

Functional-based reserving: From scratch to CRAN

HiTEc Summer Course – Module III | 25 – 27 August, 2026 | Athens, Greece



Matúš Maciak

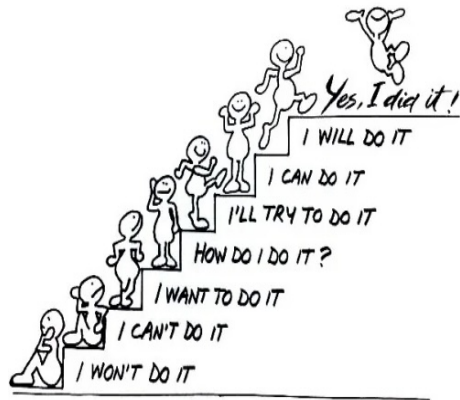
Department of Probability and Mathematical Statistics
Faculty of Mathematics and Physics
Charles University
`matus.maciak@matfyz.cuni.cz`

Motivation ...

❑ **What** should be made available on CRAN?

❑ **Why** to deploy it to CRAN?

❑ **How** to deploy it to CRAN?



Motivation ...

- ❑ **What** should be made available on CRAN?
 - ❑ Any **personal R code** that could be organized into a coherent R package...

- ❑ **Why** to deploy it to CRAN?

- ❑ **How** to deploy it to CRAN?

Motivation ...

- ❑ **What** should be made available on CRAN?
 - ❑ Any **personal R code** that could be organized into a coherent R package...
 - ❑ which may basically **contain anything** at the end (cca 25.000 CRAN packages)
- ❑ **Why** to deploy it to CRAN?
- ❑ **How** to deploy it to CRAN?

Motivation ...

- ❑ **What** should be made available on CRAN?
 - ❑ Any **personal R code** that could be organized into a coherent R package...
 - ❑ which may basically **contain anything** at the end (cca 25.000 CRAN packages)
- ❑ **Why** to deploy it to CRAN?
 - ❑ Software package + accompanying paper $\implies$ potential Q1 publication
- ❑ **How** to deploy it to CRAN?

Motivation ...

❑ What should be made available on CRAN?

- ❑ Any **personal R code** that could be organized into a coherent R package...
- ❑ which may basically **contain anything** at the end (cca 25.000 CRAN packages)

❑ Why to deploy it to CRAN?

- ❑ Software package + accompanying paper $\implies$ potential Q1 publication
- ❑ Openness to collaboration, version control, transparency, reproducibility, ...

❑ How to deploy it to CRAN?

Motivation ...

☐ What should be made available on CRAN?

- ☐ Any **personal R code** that could be organized into a coherent R package...
- ☐ which may basically **contain anything** at the end (cca 25.000 CRAN packages)

☐ Why to deploy it to CRAN?

- ☐ Software package + accompanying paper $\implies$ potential Q1 publication
- ☐ Openness to collaboration, version control, transparency, reproducibility, ...

☐ How to deploy it to CRAN?

- ☐ There are many ways to take ... **You need to find your own!**

Motivation ...

❑ What should be made available on CRAN?

- ❑ Any **personal R code** that could be organized into a coherent R package...
- ❑ which may basically **contain anything** at the end (cca 25.000 CRAN packages)

❑ Why to deploy it to CRAN?

- ❑ **Software package + accompanying paper** $\implies$ potential Q1 publication
- ❑ Openness to collaboration, version control, transparency, reproducibility, ...

❑ How to deploy it to CRAN?

- ❑ There are **many ways to take ...** **You need to find your own!**
- ❑ However, very briefly: **Build it!** $\implies$ **Validate it!** $\implies$ **Upload it!** $\implies$ **Use it!**

Motivation ...

❑ What should be made available on CRAN?

- ❑ Any **personal R code** that could be organized into a coherent R package...
- ❑ which may basically **contain anything** at the end (cca 25.000 CRAN packages)

❑ Why to deploy it to CRAN?

- ❑ Software package + accompanying paper $\implies$ potential Q1 publication
- ❑ Openness to collaboration, version control, transparency, reproducibility, ...

❑ How to deploy it to CRAN?

- ❑ There are **many ways** to take ... **You need to find your own!**
- ❑ However, very briefly: **Build it!** $\implies$ **Validate it!** $\implies$ **Upload it!** $\implies$ **Use it!**

But it is a **marathon, not a **sprint**!**

Collaboration with Michal (2001), Ivan (2010), and Rasti (2022)



Flat Top Mountain, 3.768m. (2010)



Mount Temple, 3.544m. (2013)



Gerlachovský štít, 2.656m. (2024)

Collaboration with Michal (2001), Ivan (2010), and Rasti (2022)



Flat Top Mountain, 3.768m. (2010)



Mount Temple, 3.544m. (2013)



Gerlachovský štít, 2.656m. (2024)

❑ Theoretical (mathematical) background

Maciak, M., Mizera, I., and Pešta, M. (2022). **Functional Profile Techniques for Claims Reserving.**

The ASTIN Bulletin, 52(2), 449 – 482.

DOI:10.1017/asb.2022.4

<https://www.cambridge.org/core/journals/astin-bulletin-journal-of-the-iaa>

❑ Practical (software) implementation

Maciak, M., Matúš, R., Mizera, I., and Pešta, M. (2026). **ProfileLadder: Functional-Based Reserving.**

The R Journal, 18(2), 138 – 160.

DOI:10.32614/RJ-2026-028

<https://journal.r-project.org/issues/2026-2/>

Three conceptually very different things...

1. Developing some mathematical/stochastic methodology (supplemented with some codes)
2. Organizing the existing codes into a coherent software package (e.g., for )
3. Disseminating the results and sharing the work (CRAN, JSS, The R Journal, GH Pages, ...)

Three conceptually very different things...

1. Developing some mathematical/stochastic methodology (supplemented with some codes)
2. Organizing the existing codes into a coherent software package (e.g., for )
3. Disseminating the results and sharing the work (CRAN, JSS, The R Journal, GH Pages, ...)



CRAN 0.2.2 downloads 2273 release v0.2.2

Functional claims reserving methods based on aggregated chain-ladder data, also known as a run-off triangle, implemented in three nonparametric algorithms (PARALLAX, REACT, and MACRAME) proposed in Maciak, Mizera, and Pešta (2022). [\[DOI: 10.1017/asb.2022.4\]](https://doi.org/10.1017/asb.2022.4)



FMFI, Comenius University Bratislava, June 2026



❑ **Part I** (10:30 – 12:30)

Theoretical fundations and functionality of the ProfileLadder package
(functional based—time series—forecasting, MCMC and resampling)

❑ **Part II** (14:00 – 15:30)

The key steps of the R package development process
(package skeleton, package content, packages tests, package deployment)

❑ **Part III** (16:00 – 17:30)

Dissemination, versioning, and smart automatization via GitHub
(package distribution and maintainance and some usefull and clever extensions)

Part I

Functional-based reserving with ProfileLadder

Nonparametric methods for a stochastic prediction of reserves in non-life insurance

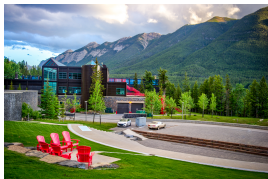
At the very beginning ...

At the very beginning: Once upon a time in Banff



Banff International Research Station (BIRS), Banff, AB, Canada (2014)

Statistical Predictions for Chain Ladder Data – 14rit2006



Statistical Predictions for Chain Ladder Data 14rit206

I. Mizera (University of Alberta, Edmonton),
M. Mašarik (Czech University of Life Sciences, Prague),
M. Pelka (Charles University, Prague)

August 24, 2014 – August 31, 2014

The main purpose of our participation was to work out our idea of a new prediction proposal, which was experimentally showed to perform better than classical methods used in the actuarial science nowadays.

The objective of the IBS's "research in nature" proposal was to meet all those of us together to continue our research developments in the area of chain ladder data prediction existing in the actuarial science and to make substantial progress towards finishing our common research in this topic, including a collaborative paper preparation.



Second impuls in 2015 ...

□ **At the Joint Statistical Meeting in Seattle (WA)**

(August 8 – 13, 2015)

Second impuls in 2015 ...

❑ At the Joint Statistical Meeting in Seattle (WA)

(August 8 – 13, 2015)



Mt. Ranier, 4.392 m. (third summit attempt)

Second impuls in 2015 ...

❑ At the Joint Statistical Meeting in Seattle (WA)

(August 8 – 13, 2015)



Mt. Ranier, 4.392 m. (third summit attempt)

❑ with some new inspiration, encouragement, and determination...



The overall goal

To propose and develop an **umbrella methodology** for general, **stochastically based**, **theoretically justified**, and **practically non-restrictive** and **easily applicable** risk assessment based on aggregated data—so-called run-off triangles

The overall goal

To propose and develop an **umbrella methodology** for general, **stochastically based**, **theoretically justified**, and **practically non-restrictive** and **easily applicable** risk assessment based on aggregated data—so-called run-off triangles

In non-life insurance in particular, it accounts for a **stochastic prediction** of the overall loss reserves that are **required by a regulator** to be kept aside to cover future claim payments... (e.g., Solvency II requires the point prediction and the 99.5% quantile of the reserve distribution)

The overall goal

To propose and develop an **umbrella methodology** for general, **stochastically based**, **theoretically justified**, and **practically non-restrictive** and **easily applicable** risk assessment based on aggregated data—so-called run-off triangles

In non-life insurance in particular, it accounts for a **stochastic prediction** of the overall loss reserves that are **required by a regulator** to be kept aside to cover future claim payments... (e.g., Solvency II requires the point prediction and the 99.5% quantile of the reserve distribution)

Related challenges involve typically very **limited data sample** with strong **time and spatial dependence**, including **nonstationarity issues**, and strong **robustness expectations** (e.g., the estimation of the 99.5 % quantile based on as few as 45 observations only)

Common approaches used by actuaries today

- ❑ **Methods based on aggregated data—the run-off triangles** (deterministic/stochastic)
 - ❑ Chain-ladder / Mack model
 - ❑ Bornhuetter–Ferguson / Cape Cod
 - ❑ Regression approaches based on GLM
 - ❑ Expert judgement / subject's intuition

Common approaches used by actuaries today

- ❑ **Methods based on aggregated data—the run-off triangles** (deterministic/stochastic)
 - ❑ Chain-ladder / Mack model
 - ❑ Bornhuetter–Ferguson / Cape Cod
 - ❑ Regression approaches based on GLM
 - ❑ Expert judgement / subject's intuition

- ❑ **Methods based on granular data—claim-by-claim reserving** (stochastic/machine learning)
 - ❑ Maximum likelihood and simulation-based methods
 - ❑ Stochastic survival models
 - ❑ Machine learning approaches (GBMs, RFs, NNs, DNNs, RL, AI... all with limited transparency!), ...

Common approaches used by actuaries today

- ❑ **Methods based on aggregated data—the run-off triangles** (deterministic/stochastic)
 - ❑ Chain-ladder / Mack model
 - ❑ Bornhuetter–Ferguson / Cape Cod
 - ❑ Regression approaches based on GLM
 - ❑ Expert judgement / subject's intuition

- ❑ **Methods based on granular data—claim-by-claim reserving** (stochastic/machine learning)
 - ❑ Maximum likelihood and simulation-based methods
 - ❑ Stochastic survival models
 - ❑ Machine learning approaches (GBMs, RFs, NNs, DNNs, RL, AI... all with limited transparency!), ...

- ❑ **Alternatives based on semi-granular data—grouped/stratified data**
 - ❑ All kinds of proposals combining strengths and limiting weaknesses of the methods above
(*statistical and machine learning, neural networks, reinforcement learning, AI tools, etc.*)

Run-off triangle: Incremental or cumulative

- ❑ **Cummulative** run-off triangle: $\{Y_{i,j}; i = 1, \dots, n; j = 1, \dots, n - i + 1\}$, with $Y_{i,0} = 0$
- ❑ **Incremental** run-off triangle: $\{X_{i,j}; i = 1, \dots, n; j = 1, \dots, n - i + 1\}$, $X_{i,j} = Y_{i,j} - Y_{i,j-1}$

Accident period i	Development period j				
	1	2	...	$n - 1$	n
1	$Y_{1,1}$	$Y_{1,2}$	...	$Y_{1,n-1}$	$Y_{1,n}$
2	$Y_{2,1}$	$Y_{2,2}$	...	$Y_{2,n-1}$	
...	...	...	...		
$\vdots$	$\vdots$	$\vdots$	$Y_{i,n+1-i}$		
$n - 1$	$Y_{n-1,1}$	$Y_{n-1,2}$			
n	$Y_{n,1}$				

- ❑ Run-off triangle with the **observed cumulative claim amounts** $Y_{i,j}$ for $i + j \leq n + 1$

Run-off triangle: Incremental or cumulative

- ❑ **Cummulative** run-off triangle: $\{Y_{i,j}; i = 1, \dots, n; j = 1, \dots, n - i + 1\}$, with $Y_{i,0} = 0$
- ❑ **Incremental** run-off triangle: $\{X_{i,j}; i = 1, \dots, n; j = 1, \dots, n - i + 1\}$, $X_{i,j} = Y_{i,j} - Y_{i,j-1}$

Accident period i	Development period j				
	1	2	...	$n - 1$	n
1	$Y_{1,1}$	$Y_{1,2}$	...	$Y_{1,n-1}$	$Y_{1,n}$
2	$Y_{2,1}$	$Y_{2,2}$	...	$Y_{2,n-1}$	?
⋮	⋮	⋮	⋱		?
			$Y_{i,n+1-i}$		?
					?
$n - 1$	$Y_{n-1,1}$	$Y_{n-1,2}$			?
n	$Y_{n,1}$				?

- ❑ Run-off triangle with the **observed cumulative claim amounts** $Y_{i,j}$ for $i + j \leq n + 1$
- ❑ The overall reserve to be estimated/predicted is $\mathcal{R} = \sum_{i=2}^n Y_{i,n} - \sum_{i=2}^n Y_{i,n-i+1}$

Alternatively, the “*patterns of loss emergence*” (Clark, 2003)

Accid. year i	Development year j									
	1	2	3	4	5	6	7	8	9	10
1	5244	9228	10823	11352	11791	12082	12120	12199	12215	12215
2	5984	9939	11725	12346	12746	12909	13034	13109	13113	13115
3	7452	12421	14171	14752	15066	15354	15637	15720	15744	15786
4	7115	11117	12488	13274	13662	13859	13872	13935	13973	13972
5	5753	8969	9917	10697	11135	11282	11255	11331	11332	11354
6	3937	6524	7989	8543	8757	8901	9013	9012	9046	9164
7	5127	8212	8976	9325	9718	9795	9833	9885	9816	9815
8	5046	8006	8984	9633	10102	10166	10261	10252	10252	10252
9	5129	8202	9185	9681	9951	10033	10133	10182	10182	10183
10	3689	6043	6789	7089	7164	7197	7253	7267	7266	7266

(a)

Alternatively, the “*patterns of loss emergence*” (Clark, 2003)

Accid. year i	Development year j									
	1	2	3	4	5	6	7	8	9	10
1	5244	9228	10823	11352	11791	12082	12120	12199	12215	12215
2	5984	9939	11725	12346	12746	12909	13034	13109	13113	13115
3	7452	12421	14171	14752	15066	15354	15637	15720	15744	15786
4	7115	11117	12488	13274	13662	13859	13872	13935	13973	13972
5	5753	8969	9917	10697	11135	11282	11255	11331	11332	11354
6	3937	6524	7989	8543	8757	8901	9013	9012	9046	9164
7	5127	8212	8976	9325	9718	9795	9833	9885	9816	9815
8	5046	8006	8984	9633	10102	10166	10261	10252	10252	10252
9	5129	8202	9185	9681	9951	10033	10133	10182	10182	10183
10	3689	6043	6789	7089	7164	7197	7253	7267	7266	7266

(c)

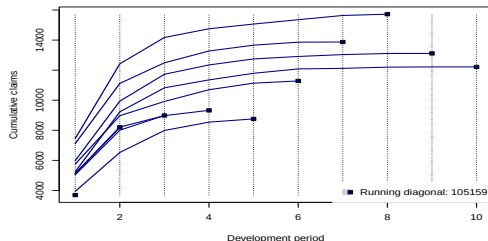
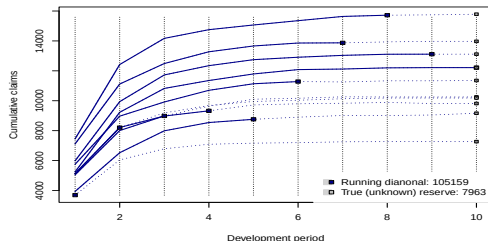
Accid. year i	Development year j									
	1	2	3	4	5	6	7	8	9	10
1	5244	9228	10823	11352	11791	12082	12120	12199	12215	12215
2	5984	9939	11725	12346	12746	12909	13034	13109	13113	.
3	7452	12421	14171	14752	15066	15354	15637	15720	.	.
4	7115	11117	12488	13274	13662	13859	13872	.	.	.
5	5753	8969	9917	10697	11135	11282	.	.	.	.
6	3937	6524	7989	8543	8757	.	.	.	.	.
7	5127	8212	8976	9325	.	.	.	.	.	.
8	5046	8006	8984	.	.	.	.	.	.	.
9	5129	8202	.	.	.	.	.	.	.	.
10	3689	.	.	.	.	.	.	.	.	.

(d)

Alternatively, the “patterns of loss emergence” (Clark, 2003)

Accid. year i	Development year j									
	1	2	3	4	5	6	7	8	9	10
1	5244	9228	10823	11352	11791	12082	12120	12199	12215	12215
2	5984	9939	11725	12346	12746	12909	13034	13109	13113	13115
3	7452	12421	14171	14752	15066	15354	15637	15720	15744	15786
4	7115	11117	12488	13274	13662	13859	13872	13935	13973	13972
5	5753	8969	9917	10697	11135	11282	11255	11331	11332	11354
6	3937	6524	7989	8543	8757	8901	9013	9012	9046	9164
7	5127	8212	8976	9325	9718	9795	9833	9885	9816	9815
8	5046	8006	8984	9633	10102	10166	10261	10252	10252	10252
9	5129	8202	9185	9681	9951	10033	10133	10182	10182	10183
10	3689	6043	6789	7089	7164	7197	7253	7267	7266	7266

Accid. year i	Development year j									
	1	2	3	4	5	6	7	8	9	10
1	5244	9228	10823	11352	11791	12082	12120	12199	12215	12215
2	5984	9939	11725	12346	12746	12909	13034	13109	13113	.
3	7452	12421	14171	14752	15066	15354	15637	15720	.	.
4	7115	11117	12488	13274	13662	13859	13872	.	.	.
5	5753	8969	9917	10697	11135	11282	.	.	.	.
6	3937	6524	7989	8543	8757	.	.	.	.	.
7	5127	8212	8976	9325	.	.	.	.	.	.
8	5046	8006	8984	.	.	.	.	.	.	.
9	5129	8202	.	.	.	.	.	.	.	.
10	3689	.	.	.	.	.	.	.	.	.



Missing functional fragments and their completion/imputation

The overall concept—the nonparametric reconstruction of missing fragments of the data—here particularly suited to the structured missingness (MNAR) in the run-off triangles, is adaptable also to other domains where curves with missing segments are observed (typically under the missing completely at random concept – MCAR)

- ❑ Censored growth curves (Delaigle and Hall, 2013, 2016)
- ❑ Fragmented functional data in medicine (Galmiche, 2016)
- ❑ Financial markets, indexes, and price curves (Liebl and Ramesede, 2019)
- ❑ Geological or seismic activity profiles (Bauer et al.2021)

Missing functional fragments and their completion/imputation

The overall concept—the nonparametric reconstruction of missing fragments of the data—here particularly suited to the structured missingness (MNAR) in the run-off triangles, is adaptable also to other domains where curves with missing segments are observed (typically under the missing completely at random concept – MCAR)

- ❑ Censored growth curves (Delaigle and Hall, 2013, 2016)
- ❑ Fragmented functional data in medicine (Galmiche, 2016)
- ❑ Financial markets, indexes, and price curves (Liebl and Ramesede, 2019)
- ❑ Geological or seismic activity profiles (Bauer et al.2021)

Application to non-life claims reserving under the MNAR assumption (MMP 2022)

There is still a growing interest in novel methods, offering [better predictive performance](#), [transparency](#), and [robustness](#)... the idea is to provide an extension and complement of other methods and packages devoted to the claims reserving problem—most importantly the ground-breaking package [ChainLadder](#) (Gesmann et al., 2025), featuring all known methods for non-life loss reserving that can be considered classical (Mack chain ladder, over-dispersed Poisson and other GLM models, Bornhuetter-Ferguson, etc.).

Triangle-based standard/benchmark methods

Goal: Use the **upper-left triangle** of the data and **predict the lower-right triangle** (i.e., the missing part of the whole square). In a cumulative triangle, the overall (unknown) reserve $\mathcal{R}$ is “represented” by the sum of the last column minus the last observed diagonal

Triangle-based standard/benchmark methods

Goal: Use the upper-left triangle of the data and predict the lower-right triangle (i.e., the missing part of the whole square). In a cumulative triangle, the overall (unknown) reserve $\mathcal{R}$ is “represented” by the sum of the last column minus the last observed diagonal

❑ **Naive: Average development ratios**

- Development ratios $f_j \geq 1$ and the assumption $E[Y_{i,j+1} | Y_{i,j}, \dots, Y_{i,1}] = f_j Y_{i,j}$
- Straightforward estimates for f_j using the available data, $\hat{f}_j = \frac{1}{n-j} \sum_{i=1}^{n-j} Y_{i,j+1} / Y_{i,j}$

Triangle-based standard/benchmark methods

Goal: Use the **upper-left triangle** of the data and **predict the lower-right triangle** (i.e., the missing part of the whole square). In a cumulative triangle, the overall (unknown) reserve $\mathcal{R}$ is “represented” by the sum of the last column minus the last observed diagonal

❑ **Naive: Average development ratios**

- Development ratios $f_j \geq 1$ and the assumption $E[Y_{i,j+1}|Y_{i,j}, \dots, Y_{i,1}] = f_j Y_{i,j}$
- Straightforward estimates for f_j using the available data, $\hat{f}_j = \frac{1}{n-j} \sum_{i=1}^{n-j} Y_{i,j+1}/Y_{i,j}$

❑ **Robust: Volume-weighted development ratios**

- Same development ratios $f_j \geq 1$ under the same assumption as above
- Slightly different (more robust) estimation, where $\hat{f}_j = \sum_{i=1}^{n-j} Y_{i,j+1} / \sum_{i=1}^{n-j} Y_{i,j}$

Triangle-based standard/benchmark methods

Goal: Use the **upper-left triangle** of the data and **predict the lower-right triangle** (i.e., the missing part of the whole square). In a cumulative triangle, the overall (unknown) reserve $\mathcal{R}$ is “represented” by the sum of the last column minus the last observed diagonal

❑ Naive: Average development ratios

- Development ratios $f_j \geq 1$ and the assumption $E[Y_{i,j+1} | Y_{i,j}, \dots, Y_{i,1}] = f_j Y_{i,j}$
- Straightforward estimates for f_j using the available data, $\hat{f}_j = \frac{1}{n-j} \sum_{i=1}^{n-j} Y_{i,j+1} / Y_{i,j}$

❑ Robust: Volume-weighted development ratios

- Same development ratios $f_j \geq 1$ under the same assumption as above
- Slightly different (more robust) estimation, where $\hat{f}_j = \sum_{i=1}^{n-j} Y_{i,j+1} / \sum_{i=1}^{n-j} Y_{i,j}$

❑ Advanced: Overdispersed Poisson or gamma models

- Parametric estimation using classical **GLM framework** for incremental data $X_{i,j} \geq 0$
- Regression parameters are row- and column-specific, i.e., $\log(E[X_{i,j}]) = \gamma + \alpha_i + \beta_j$

Algorithmic principles behind standard approaches

□ Development ratios

Accid. year i	Development year j									
	1	2	3	4	5	6	7	8	9	10
1	5244	9228	10823	11352	11791	12082	12120	12199	12215	12215
2	5984	9939	11725	12346	12746	12909	13034	13109	13113	-
3	7452	12421	14171	14752	15066	15354	15637	15720	-	-
4	7115	11117	12488	13274	13662	13859	13872	-	-	-
5	5753	8969	9917	10697	11135	11282	-	-	-	-
6	3937	6524	7989	8543	8757	-	-	-	-	-
7	5127	8212	8976	9325	-	-	-	-	-	-
8	5046	8006	8984	-	-	-	-	-	-	-
9	5129	8202	-	-	-	-	-	-	-	-
10	3689	-	-	-	-	-	-	-	-	-

Algorithmic principles behind standard approaches

□ Development ratios

Accid. year i	Development year j									
	1	2	3	4	5	6	7	8	9	10
1	5244	9228	10823	11352	11791	12082	12120	12199	12215	12215
2	5984	9939	11725	12346	12746	12909	13034	13109	13113	-
3	7452	12421	14171	14752	15066	15354	15637	15720	-	-
4	7115	11117	12488	13274	13662	13859	13872	-	-	-
5	5753	8969	9917	10697	11135	11282	-	-	-	-
6	3937	6524	7989	8543	8757	-	-	-	-	-
7	5127	8212	8976	9325	-	-	-	-	-	-
8	5046	8006	8984	-	-	-	-	-	-	-
9	5129	8202	-	-	-	-	-	-	-	-
10	3689	-	-	-	-	-	-	-	-	-

Algorithmic principles behind standard approaches

□ Development ratios

Accid. year i	Development year j									
	1	2	3	4	5	6	7	8	9	10
1	5244	9228	10823	11352	11791	12082	12120	12199	12215	12215
2	5984	9939	11725	12346	12746	12909	13034	13109	13113	-
3	7452	12421	14171	14752	15066	15354	15637	15720	-	-
4	7115	11117	12488	13274	13662	13859	13872	-	-	-
5	5753	8969	9917	10697	11135	11282	-	-	-	-
6	3937	6524	7989	8543	8757	-	-	-	-	-
7	5127	8212	8976	9325	-	-	-	-	-	-
8	5046	8006	8984	-	-	-	-	-	-	-
9	5129	8202	-	-	-	-	-	-	-	-
10	3689	-	-	-	-	-	-	-	-	-

Algorithmic principles behind standard approaches

□ Development ratios

Accid. year i	Development year j									
	1	2	3	4	5	6	7	8	9	10
1	5244	9228	10823	11352	11791	12082	12120	12199	12215	12215
2	5984	9939	11725	12346	12746	12909	13034	13109	13113	-
3	7452	12421	14171	14752	15066	15354	15637	15720	-	-
4	7115	11117	12488	13274	13662	13859	13872	-	-	-
5	5753	8969	9917	10697	11135	11282	-	-	-	-
6	3937	6524	7989	8543	8757	-	-	-	-	-
7	5127	8212	8976	9325	-	-	-	-	-	-
8	5046	8006	8984	-	-	-	-	-	-	-
9	5129	8202	-	-	-	-	-	-	-	-
10	3689	-	-	-	-	-	-	-	-	-

$$\square \hat{h}_{1,1} = \frac{9228}{5244}$$

Algorithmic principles behind standard approaches

□ Development ratios

Accid. year i	Development year j									
	1	2	3	4	5	6	7	8	9	10
1	5244	9228	10823	11352	11791	12082	12120	12199	12215	12215
2	5984	9939	11725	12346	12746	12909	13034	13109	13113	-
3	7452	12421	14171	14752	15066	15354	15637	15720	-	-
4	7115	11117	12488	13274	13662	13859	13872	-	-	-
5	5753	8969	9917	10697	11135	11282	-	-	-	-
6	3937	6524	7989	8543	8757	-	-	-	-	-
7	5127	8212	8976	9325	-	-	-	-	-	-
8	5046	8006	8984	-	-	-	-	-	-	-
9	5129	8202	-	-	-	-	-	-	-	-
10	3689	-	-	-	-	-	-	-	-	-

$$\square \hat{h}_{1,1} = \frac{9228}{5244}; \hat{f}_{2,1} = \frac{9939}{5984}$$

Algorithmic principles behind standard approaches

□ Development ratios

Accid. year i	Development year j									
	1	2	3	4	5	6	7	8	9	10
1	5244	9228	10823	11352	11791	12082	12120	12199	12215	12215
2	5984	9939	11725	12346	12746	12909	13034	13109	13113	-
3	7452	12421	14171	14752	15066	15354	15637	15720	-	-
4	7115	11117	12488	13274	13662	13859	13872	-	-	-
5	5753	8969	9917	10697	11135	11282	-	-	-	-
6	3937	6524	7989	8543	8757	-	-	-	-	-
7	5127	8212	8976	9325	-	-	-	-	-	-
8	5046	8006	8984	-	-	-	-	-	-	-
9	5129	8202	-	-	-	-	-	-	-	-
10	3689	-	-	-	-	-	-	-	-	-

$$\square \hat{f}_{1,1} = \frac{9228}{5244}; \hat{f}_{2,1} = \frac{9939}{5984}; \dots \hat{f}_{9,1} = \frac{8202}{5129}$$

Algorithmic principles behind standard approaches

□ Development ratios

Accid. year i	Development year j									
	1	2	3	4	5	6	7	8	9	10
1	5244	9228	10823	11352	11791	12082	12120	12199	12215	12215
2	5984	9939	11725	12346	12746	12909	13034	13109	13113	-
3	7452	12421	14171	14752	15066	15354	15637	15720	-	-
4	7115	11117	12488	13274	13662	13859	13872	-	-	-
5	5753	8969	9917	10697	11135	11282	-	-	-	-
6	3937	6524	7989	8543	8757	-	-	-	-	-
7	5127	8212	8976	9325	-	-	-	-	-	-
8	5046	8006	8984	-	-	-	-	-	-	-
9	5129	8202	-	-	-	-	-	-	-	-
10	3689	-	-	-	-	-	-	-	-	-

$$\square \hat{f}_{1,1} = \frac{9228}{5244}; \hat{f}_{2,1} = \frac{9939}{5984}; \dots \hat{f}_{9,1} = \frac{8202}{5129}$$

$$\square \hat{f}_1 = \frac{1}{9} \sum_{i=1}^9 \hat{f}_{i,1} \approx 1.63$$

Algorithmic principles behind standard approaches

□ Development ratios

Accid. year i	Development year j									
	1	2	3	4	5	6	7	8	9	10
1	5244	9228	10823	11352	11791	12082	12120	12199	12215	12215
2	5984	9939	11725	12346	12746	12909	13034	13109	13113	-
3	7452	12421	14171	14752	15066	15354	15637	15720	-	-
4	7115	11117	12488	13274	13662	13859	13872	-	-	-
5	5753	8969	9917	10697	11135	11282	-	-	-	-
6	3937	6524	7989	8543	8757	-	-	-	-	-
7	5127	8212	8976	9325	-	-	-	-	-	-
8	5046	8006	8984	-	-	-	-	-	-	-
9	5129	8202	-	-	-	-	-	-	-	-
10	3689	6013	-	-	-	-	-	-	-	-

$$\square \hat{f}_{1,1} = \frac{9228}{5244}; \hat{f}_{2,1} = \frac{9939}{5984}; \dots \hat{f}_{9,1} = \frac{8202}{5129}$$

$$\square \hat{f}_1 = \frac{1}{9} \sum_{i=1}^9 \hat{f}_{i,1} \approx 1.63$$

$$\square \hat{Y}_{10,2} = 3689 \times 1.63 \approx 6013$$

Algorithmic principles behind standard approaches

Development ratios

Volume-weighted ratios

Accid. year i	Development year j									
	1	2	3	4	5	6	7	8	9	10
1	5244	9228	10823	11352	11791	12082	12120	12199	12215	12215
2	5984	9939	11725	12346	12746	12909	13034	13109	13113	-
3	7452	12421	14171	14752	15066	15354	15637	15720	-	-
4	7115	11117	12488	13274	13662	13859	13872	-	-	-
5	5753	8969	9917	10697	11135	11282	-	-	-	-
6	3937	6524	7989	8543	8757	-	-	-	-	-
7	5127	8212	8976	9325	-	-	-	-	-	-
8	5046	8006	8984	-	-	-	-	-	-	-
9	5129	8202	-	-	-	-	-	-	-	-
10	3689	-	-	-	-	-	-	-	-	-

$$\hat{f}_{1,1} = \frac{9228}{5244}; \hat{f}_{2,1} = \frac{9939}{5984}; \dots \hat{f}_{9,1} = \frac{8202}{5129}$$

$$\hat{f}_1 = \frac{1}{9} \sum_{i=1}^9 \hat{f}_{i,1} \approx 1.63$$

$$\hat{Y}_{10,2} = 3689 \times 1.63 \approx 6013$$

Algorithmic principles behind standard approaches

Development ratios

Volume-weighted ratios

Accid. year i	Development year j									
	1	2	3	4	5	6	7	8	9	10
1	5244	9228	10823	11352	11791	12082	12120	12199	12215	12215
2	5984	9939	11725	12346	12746	12909	13034	13109	13113	-
3	7452	12421	14171	14752	15066	15354	15637	15720	-	-
4	7115	11117	12488	13274	13662	13859	13872	-	-	-
5	5753	8969	9917	10697	11135	11282	-	-	-	-
6	3937	6524	7989	8543	8757	-	-	-	-	-
7	5127	8212	8976	9325	-	-	-	-	-	-
8	5046	8006	8984	-	-	-	-	-	-	-
9	5129	8202	-	-	-	-	-	-	-	-
10	3689	-	-	-	-	-	-	-	-	-

$$\hat{f}_{1,1} = \frac{9228}{5244}; \hat{f}_{2,1} = \frac{9939}{5984}; \dots \hat{f}_{9,1} = \frac{8202}{5129} \quad \square \sum_{i=1}^9 Y_{i,2} = 9228 + \dots + 8202 = 82618$$

$$\hat{f}_1 = \frac{1}{9} \sum_{i=1}^9 \hat{f}_{i,1} \approx 1.63$$

$$\hat{Y}_{10,2} = 3689 \times 1.63 \approx 6013$$

Algorithmic principles behind standard approaches

Development ratios

Volume-weighted ratios

Accid. year i	Development year j									
	1	2	3	4	5	6	7	8	9	10
1	9228	9228	10823	11352	11791	12082	12120	12199	12215	12215
2	5984	9939	11725	12346	12746	12909	13034	13109	13113	-
3	7452	12421	14171	14752	15066	15354	15637	15720	-	-
4	7115	11117	12488	13274	13662	13859	13872	-	-	-
5	5753	8969	9917	10697	11135	11282	-	-	-	-
6	3937	6524	7989	8543	8757	-	-	-	-	-
7	5127	8212	8976	9325	-	-	-	-	-	-
8	5046	8006	8984	-	-	-	-	-	-	-
9	5129	8202	-	-	-	-	-	-	-	-
10	3689	-	-	-	-	-	-	-	-	-

$$\hat{f}_{1,1} = \frac{9228}{5244}; \hat{f}_{2,1} = \frac{9939}{5984}; \dots \hat{f}_{9,1} = \frac{8202}{5129} \quad \square \sum_{i=1}^9 Y_{i,2} = 9228 + \dots + 8202 = 82618$$

$$\hat{f}_1 = \frac{1}{9} \sum_{i=1}^9 \hat{f}_{i,1} \approx 1.63$$

$$\square \sum_{i=1}^9 Y_{i,1} = 5244 + \dots + 5129 = 50787$$

$$\square \hat{Y}_{10,2} = 3689 \times 1.63 \approx 6013$$

Algorithmic principles behind standard approaches

Development ratios

Volume-weighted ratios

Accid. year i	Development year j									
	1	2	3	4	5	6	7	8	9	10
1	9228	9228	10823	11352	11791	12082	12120	12199	12215	12215
2	5984	9939	11725	12346	12746	12909	13034	13109	13113	-
3	7452	12421	14171	14752	15066	15354	15637	15720	-	-
4	7115	11117	12488	13274	13662	13859	13872	-	-	-
5	5753	8969	9917	10697	11135	11282	-	-	-	-
6	3937	6524	7989	8543	8757	-	-	-	-	-
7	5127	8212	8976	9325	-	-	-	-	-	-
8	5046	8006	8984	-	-	-	-	-	-	-
9	5129	8202	-	-	-	-	-	-	-	-
10	3689	6001	-	-	-	-	-	-	-	-

$$\hat{f}_{1,1} = \frac{9228}{5244}; \hat{f}_{2,1} = \frac{9939}{5984}; \dots \hat{f}_{9,1} = \frac{8202}{5129} \quad \square \sum_{i=1}^9 Y_{i,2} = 9228 + \dots + 8202 = 82618$$

$$\hat{f}_1 = \frac{1}{9} \sum_{i=1}^9 \hat{f}_{i,1} \approx 1.63 \quad \square \sum_{i=1}^9 Y_{i,1} = 5244 + \dots + 5129 = 50787$$

$$\hat{Y}_{10,2} = 3689 \times 1.63 \approx 6013 \quad \square \hat{Y}_{10,2} = 3689 \times \frac{82618}{50787} \approx 6001$$

Algorithmic principles behind standard approaches

□ Development ratios

□ Volume-weighted ratios

□ GLM: Poisson model

Accid. year i	Development year j									
	1	2	3	4	5	6	7	8	9	10
1	5244	3984	1595	529	439	291	38	79	16	0
2	5984	3955	1786	621	400	163	125	75	4	-
3	7452	4969	1750	581	314	288	283	83	-	-
4	7115	4002	1371	786	388	197	13	-	-	-
5	5753	3216	948	780	438	147	-	-	-	-
6	3937	2587	1465	554	214	-	-	-	-	-
7	5127	3085	764	349	-	-	-	-	-	-
8	5046	2960	978	-	-	-	-	-	-	-
9	5129	3073	-	-	-	-	-	-	-	-
10	3689	-	-	-	-	-	-	-	-	-

$$\square \hat{f}_{1,1} = \frac{9228}{5244}; \hat{f}_{2,1} = \frac{9939}{5984}; \dots \hat{f}_{9,1} = \frac{8202}{5129} \quad \square \sum_{i=1}^9 Y_{i,2} = 9228 + \dots + 8202 = 82618$$

$$\square \hat{f}_1 = \frac{1}{9} \sum_{i=1}^9 \hat{f}_{i,1} \approx 1.63 \quad \square \sum_{i=1}^9 Y_{i,1} = 5244 + \dots + 5129 = 50787$$

$$\square \hat{Y}_{10,2} = 3689 \times 1.63 \approx 6013 \quad \square \hat{Y}_{10,2} = 3689 \times \frac{82618}{50787} \approx 6001$$

Algorithmic principles behind standard approaches

□ Development ratios

□ Volume-weighted ratios

□ GLM: Poisson model

Accid. year i	Development year j									
	1	2	3	4	5	6	7	8	9	10
1	5244	3984	1595	529	439	291	38	79	16	0
2	5984	3955	1786	621	400	163	125	75	4	-
3	7452	4969	1750	581	314	288	283	83	-	-
4	7115	4002	1371	786	388	197	13	-	-	-
5	5753	3216	948	780	438	147	-	-	-	-
6	3937	2587	1465	554	214	-	-	-	-	-
7	5127	3085	764	349	-	-	-	-	-	-
8	5046	2960	978	-	-	-	-	-	-	-
9	5129	3073	-	-	-	-	-	-	-	-
10	3689	-	-	-	-	-	-	-	-	-

- $\hat{f}_{1,1} = \frac{9228}{5244}$; $\hat{f}_{2,1} = \frac{9939}{5984}$; ... $\hat{f}_{9,1} = \frac{8202}{5129}$ □ $\sum_{i=1}^9 Y_{i,2} = 9228 + \dots + 8202 = 82618$ □ GLM: $\log(E[X_{i,j}]) = \alpha + \beta_i + \gamma_j$
- $\hat{f}_1 = \frac{1}{9} \sum_{i=1}^9 \hat{f}_{i,1} \approx 1.63$ □ $\sum_{i=1}^9 Y_{i,1} = 5244 + \dots + 5129 = 50787$
- $\hat{Y}_{10,2} = 3689 \times 1.63 \approx 6013$ □ $\hat{Y}_{10,2} = 3689 \times \frac{82618}{50787} \approx 6001$

Algorithmic principles behind standard approaches

□ Development ratios

□ Volume-weighted ratios

□ GLM: Poisson model

Accid. year i	Development year j									
	1	2	3	4	5	6	7	8	9	10
1	5244	3984	1595	529	439	291	38	79	16	0
2	5984	3955	1786	621	400	163	125	75	4	-
3	7452	4969	1750	581	314	288	283	83	-	-
4	7115	4002	1371	786	388	197	13	-	-	-
5	5753	3216	948	780	438	147	-	-	-	-
6	3937	2587	1465	554	214	-	-	-	-	-
7	5127	3085	764	349	-	-	-	-	-	-
8	5046	2960	978	-	-	-	-	-	-	-
9	5129	3073	-	-	-	-	-	-	-	-
10	3689	-	-	-	-	-	-	-	-	-

- $\hat{f}_{1,1} = \frac{9228}{5244}$; $\hat{f}_{2,1} = \frac{9939}{5984}$; ... $\hat{f}_{9,1} = \frac{8202}{5129}$ □ $\sum_{i=1}^9 Y_{i,2} = 9228 + \dots + 8202 = 82618$ □ GLM: $\log(E[X_{i,j}]) = \alpha + \beta_i + \gamma_j$
- $\hat{f}_1 = \frac{1}{9} \sum_{i=1}^9 \hat{f}_{i,1} \approx 1.63$ □ $\sum_{i=1}^9 Y_{i,1} = 5244 + \dots + 5129 = 50787$ □ $\hat{Y}_{10,2} = Y_{10,1} + \exp\{\hat{\alpha} + \hat{\beta}_{10} + \hat{\gamma}_2\}$
- $\hat{Y}_{10,2} = 3689 \times 1.63 \approx 6013$ □ $\hat{Y}_{10,2} = 3689 \times \frac{82618}{50787} \approx 6001$

Algorithmic principles behind standard approaches

□ Development ratios

□ Volume-weighted ratios

□ GLM: Poisson model

Accid. year i	Development year j									
	1	2	3	4	5	6	7	8	9	10
1	5244	3984	1595	529	439	291	38	79	16	0
2	5984	3955	1786	621	400	163	125	75	4	-
3	7452	4969	1750	581	314	288	283	83	-	-
4	7115	4002	1371	786	388	197	13	-	-	-
5	5753	3216	948	780	438	147	-	-	-	-
6	3937	2587	1465	554	214	-	-	-	-	-
7	5127	3085	764	349	-	-	-	-	-	-
8	5046	2960	978	-	-	-	-	-	-	-
9	5129	3073	-	-	-	-	-	-	-	-
10	3689	6001	-	-	-	-	-	-	-	-

$$\square \hat{f}_{1,1} = \frac{9228}{5244}; \hat{f}_{2,1} = \frac{9939}{5984}; \dots \hat{f}_{9,1} = \frac{8202}{5129}$$

$$\square \hat{f}_1 = \frac{1}{9} \sum_{i=1}^9 \hat{f}_{i,1} \approx 1.63$$

$$\square \hat{Y}_{10,2} = 3689 \times 1.63 \approx 6013$$

$$\square \sum_{i=1}^9 Y_{i,2} = 9228 + \dots + 8202 = 82618$$

$$\square \sum_{i=1}^9 Y_{i,1} = 5244 + \dots + 5129 = 50787$$

$$\square \hat{Y}_{10,2} = 3689 \times \frac{82618}{50787} \approx 6001$$

$$\square \text{GLM: } \log(E[X_{i,j}]) = \alpha + \beta_i + \gamma_j$$

$$\square \hat{Y}_{10,2} = Y_{10,1} + \exp\{\hat{\alpha} + \hat{\beta}_{10} + \hat{\gamma}_2\}$$

$$\square \hat{Y}_{10,2} \approx 3689 + \exp\{7.7459\} = 6001$$

Algorithmic principles behind standard approaches

Development ratios

Volume-weighted ratios

GLM: Poisson model

Accid. year i	Development year j									
	1	2	3	4	5	6	7	8	9	10
1	5244	3984	1595	529	439	291	38	79	16	0
2	5984	3955	1786	621	400	163	125	75	4	-
3	7452	4969	1750	581	314	288	283	83	-	-
4	7115	4002	1371	786	388	197	13	-	-	-
5	5753	3216	948	780	438	147	-	-	-	-
6	3937	2587	1465	554	214	-	-	-	-	-
7	5127	3085	764	349	-	-	-	-	-	-
8	5046	2960	978	-	-	-	-	-	-	-
9	5129	3073	-	-	-	-	-	-	-	-
10	3689	6001	-	-	-	-	-	-	-	-

$$\hat{f}_{1,1} = \frac{9228}{5244}; \hat{f}_{2,1} = \frac{9939}{5984}; \dots \hat{f}_{9,1} = \frac{8202}{5129}$$

$$\hat{f}_1 = \frac{1}{9} \sum_{i=1}^9 \hat{f}_{i,1} \approx 1.63$$

$$\hat{Y}_{10,2} = 3689 \times 1.63 \approx 6013$$

$$\text{Overall reserve: } \hat{\mathcal{R}} = 8687$$

$$\sum_{i=1}^9 Y_{i,2} = 9228 + \dots + 8202 = 82618$$

$$\sum_{i=1}^9 Y_{i,1} = 5244 + \dots + 5129 = 50787$$

$$\hat{Y}_{10,2} = 3689 \times \frac{82618}{50787} \approx 6001$$

$$\text{Overall reserve: } \hat{\mathcal{R}} = 8601$$

$$\text{GLM: } \log(E[X_{i,j}]) = \alpha + \beta_i + \gamma_j$$

$$\hat{Y}_{10,2} = Y_{10,1} + \exp\{\hat{\alpha} + \hat{\beta}_{10} + \hat{\gamma}_2\}$$

$$\hat{Y}_{10,2} \approx 3689 + \exp\{7.7459\} = 6001$$

$$\text{Overall reserve: } \hat{\mathcal{R}} = 8601$$

Algorithmic principles behind standard approaches

Development ratios

Volume-weighted ratios

GLM: Poisson model

Accid. year i	Development year j									
	1	2	3	4	5	6	7	8	9	10
1	5244	3984	1595	529	439	291	38	79	16	0
2	5984	3955	1786	621	400	163	125	75	4	-
3	7452	4969	1750	581	314	288	283	83	-	-
4	7115	4002	1371	786	388	197	13	-	-	-
5	5753	3216	948	780	438	147	-	-	-	-
6	3937	2587	1465	554	214	-	-	-	-	-
7	5127	3085	764	349	-	-	-	-	-	-
8	5046	2960	978	-	-	-	-	-	-	-
9	5129	3073	-	-	-	-	-	-	-	-
10	3689	6001	-	-	-	-	-	-	-	-

$$\hat{f}_{1,1} = \frac{9228}{5244}; \hat{f}_{2,1} = \frac{9939}{5984}; \dots \hat{f}_{9,1} = \frac{8202}{5129}$$

$$\hat{f}_1 = \frac{1}{9} \sum_{i=1}^9 \hat{f}_{i,1} \approx 1.63$$

$$\hat{Y}_{10,2} = 3689 \times 1.63 \approx 6013$$

$$\text{Overall reserve: } \hat{\mathcal{R}} = 8687$$

(point predictions only)

$$\sum_{i=1}^9 Y_{i,2} = 9228 + \dots + 8202 = 82618$$

$$\sum_{i=1}^9 Y_{i,1} = 5244 + \dots + 5129 = 50787$$

$$\hat{Y}_{10,2} = 3689 \times \frac{82618}{50787} \approx 6001$$

$$\text{Overall reserve: } \hat{\mathcal{R}} = 8601$$

$$\text{GLM: } \log(E[X_{i,j}]) = \alpha + \beta_i + \gamma_j$$

$$\hat{Y}_{10,2} = Y_{10,1} + \exp\{\hat{\alpha} + \hat{\beta}_{10} + \hat{\gamma}_2\}$$

$$\hat{Y}_{10,2} \approx 3689 + \exp\{7.7459\} = 6001$$

$$\text{Overall reserve: } \hat{\mathcal{R}} = 8601$$

Algorithmic principles behind standard approaches

Development ratios

Volume-weighted ratios

GLM: Poisson model

Accid. year i	Development year j									
	1	2	3	4	5	6	7	8	9	10
1	5244	3984	1595	529	439	291	38	79	16	0
2	5984	3955	1786	621	400	163	125	75	4	-
3	7452	4969	1750	581	314	288	283	83	-	-
4	7115	4002	1371	786	388	197	13	-	-	-
5	5753	3216	948	780	438	147	-	-	-	-
6	3937	2587	1465	554	214	-	-	-	-	-
7	5127	3085	764	349	-	-	-	-	-	-
8	5046	2960	978	-	-	-	-	-	-	-
9	5129	3073	-	-	-	-	-	-	-	-
10	3689	6001	-	-	-	-	-	-	-	-

$$\hat{f}_{1,1} = \frac{9228}{5244}; \hat{f}_{2,1} = \frac{9939}{5984}; \dots \hat{f}_{9,1} = \frac{8202}{5129}$$

$$\hat{f}_1 = \frac{1}{9} \sum_{i=1}^9 \hat{f}_{i,1} \approx 1.63$$

$$\hat{Y}_{10,2} = 3689 \times 1.63 \approx 6013$$

$$\text{Overall reserve: } \hat{\mathcal{R}} = 8687$$

(point predictions only)

$$\sum_{i=1}^9 Y_{i,2} = 9228 + \dots + 8202 = 82618$$

$$\sum_{i=1}^9 Y_{i,1} = 5244 + \dots + 5129 = 50787$$

$$\hat{Y}_{10,2} = 3689 \times \frac{82618}{50787} \approx 6001$$

$$\text{Overall reserve: } \hat{\mathcal{R}} = 8601$$

(point predictions only)

$$\text{GLM: } \log(E[X_{i,j}]) = \alpha + \beta_i + \gamma_j$$

$$\hat{Y}_{10,2} = Y_{10,1} + \exp\{\hat{\alpha} + \hat{\beta}_{10} + \hat{\gamma}_2\}$$

$$\hat{Y}_{10,2} \approx 3689 + \exp\{7.7459\} = 6001$$

$$\text{Overall reserve: } \hat{\mathcal{R}} = 8601$$

Algorithmic principles behind standard approaches

Development ratios

Volume-weighted ratios

GLM: Poisson model

Accid. year i	Development year j									
	1	2	3	4	5	6	7	8	9	10
1	5244	3984	1595	529	439	291	38	79	16	0
2	5984	3955	1786	621	400	163	125	75	4	-
3	7452	4969	1750	581	314	288	283	83	-	-
4	7115	4002	1371	786	388	197	13	-	-	-
5	5753	3216	948	780	438	147	-	-	-	-
6	3937	2587	1465	554	214	-	-	-	-	-
7	5127	3085	764	349	-	-	-	-	-	-
8	5046	2960	978	-	-	-	-	-	-	-
9	5129	3073	-	-	-	-	-	-	-	-
10	3689	6001	-	-	-	-	-	-	-	-

$$\hat{f}_{1,1} = \frac{9228}{5244}; \hat{f}_{2,1} = \frac{9939}{5984}; \dots \hat{f}_{9,1} = \frac{8202}{5129}$$

$$\hat{f}_1 = \frac{1}{9} \sum_{i=1}^9 \hat{f}_{i,1} \approx 1.63$$

$$\hat{Y}_{10,2} = 3689 \times 1.63 \approx 6013$$

$$\text{Overall reserve: } \hat{\mathcal{R}} = 8687$$

(point predictions only)

$$\sum_{i=1}^9 Y_{i,2} = 9228 + \dots + 8202 = 82618$$

$$\sum_{i=1}^9 Y_{i,1} = 5244 + \dots + 5129 = 50787$$

$$\hat{Y}_{10,2} = 3689 \times \frac{82618}{50787} \approx 6001$$

$$\text{Overall reserve: } \hat{\mathcal{R}} = 8601$$

(point predictions only)

$$\text{GLM: } \log(E[X_{i,j}]) = \alpha + \beta_i + \gamma_j$$

$$\hat{Y}_{10,2} = Y_{10,1} + \exp\{\hat{\alpha} + \hat{\beta}_{10} + \hat{\gamma}_2\}$$

$$\hat{Y}_{10,2} \approx 3689 + \exp\{7.7459\} = 6001$$

$$\text{Overall reserve: } \hat{\mathcal{R}} = 8601$$

(point/interval predictions)

Illustration: GLM regression based reserving in R

- ❑ The core actuarial R package [ChainLadder](#) with standard (parametric) methods
(with analogous outputs provided for all reserving methods from this package)

```
R> library(ChainLadder)
R> glmReserve(triangle) ## some run-off triangle shaped data
```

	Latest	Dev.To.Date	Ultimate	IBNR		S.E	CV
2	13113	1.0000000	13113	0	0.01035287		Inf
3	15720	0.9992372	15732	12	29.38975320	2.4491461	
4	13872	0.9934116	13964	92	73.46098788	0.7984890	
5	11282	0.9850694	11453	171	96.26787355	0.5629700	
6	8757	0.9688019	9039	282	120.58781735	0.4276164	
7	9325	0.9397360	9923	598	177.57019068	0.2969401	
8	8984	0.8905630	10088	1104	246.00693847	0.2228324	
9	8202	0.7789914	10529	2327	379.54811405	0.1631062	
10	3689	0.4788422	7704	4015	622.22022041	0.1549739	
total	92944	0.9152986	101545	8601	880.66723938	0.1023913	

Risk reserving with ChainLadder

The R package `ChainLadder` is probably the best-known R package (Gesman et al. 2025) devoted specifically to non-life insurance claims reserving. Its main purpose is to work with run-off triangles (S3 class `triangle`) and it implements both classical and stochastic (parametric) reserving methods.

- ❑ **Run-off triangle type of data:** `as.triangle()`, `cum2incr()`, and `incr2cum()`
- ❑ **Classical (deterministic) methods:** `MackChainLadder()`;
- ❑ **Stochastic (distribution-based) methods:** `glmReserve()`, `tweeddeReserve()`, `ClarkLDF()`
- ❑ **Bayesian-based reserving:** `ClarkCapeCod()`
- ❑ **Distributional prediction by resampling residuals:** `BootChanLadder()`
- ❑ **Some extensions:** `MunichChainLadder()`, `MultiChainLadder()`, `PaidIncurredChain()`

Risk reserving with ChainLadder

The R package `ChainLadder` is probably the best-known R package (Gesman et al. 2025) devoted specifically to non-life insurance claims reserving. Its main purpose is to work with run-off triangles (S3 class `triangle`) and it implements both classical and stochastic (parametric) reserving methods.

- ❑ **Run-off triangle type of data:** `as.triangle()`, `cum2incr()`, and `incr2cum()`
- ❑ **Classical (deterministic) methods:** `MackChainLadder()`;
- ❑ **Stochastic (distribution-based) methods:** `glmReserve()`, `tweedeReserve()`, `ClarkLDF()`
- ❑ **Bayesian-based reserving:** `ClarkCapeCod()`
- ❑ **Distributional prediction by resampling residuals:** `BootChanLadder()`
- ❑ **Some extensions:** `MunichChainLadder()`, `MultiChainLadder()`, `PaidIncurredChain()`

However, with no or just very poor usage of various S3 methods (`plot()`, `print()`, `summary()`, or `predict()`) (such methods may provide additional (and useful) information and better insight into the data or the method)

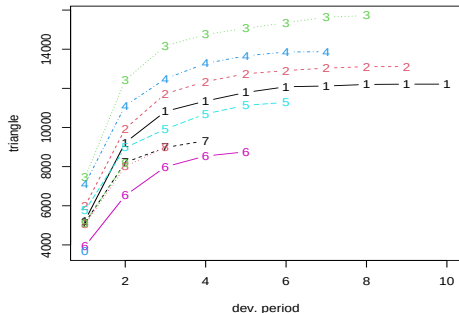
Illustration: Run-off triangle vs. prediction method (I)

Comparison of the **S3 method** `plot()`

Illustration: Run-off triangle vs. prediction method (I)

Comparison of the **S3 method** `plot()`

```
R> plot(triangle)
```



```
R> plot(glmReserve(triangle))
```

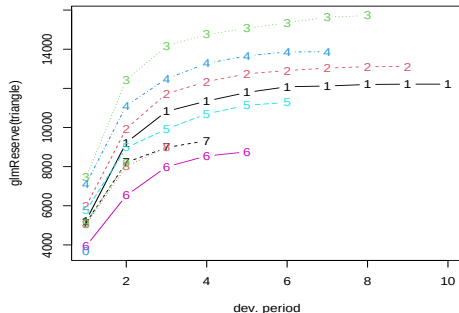


Illustration: Run-off triangle vs. prediction method (II)

Comparison of the S3 method summary()

```
R> glmReserve(triangle)
```

	Latest	Dev.To.Date	Ultimate	IBNR		S.E	CV
2	13113	1.0000000	13113	0	0.01035287		Inf
3	15720	0.9992372	15732	12	29.38975320	2.4491461	
4	13872	0.9934116	13964	92	73.46098788	0.7984890	
5	11282	0.9850694	11453	171	96.26787355	0.5629700	
6	8757	0.9688019	9039	282	120.58781735	0.4276164	
7	9325	0.9397360	9923	598	177.57019068	0.2969401	
8	8984	0.8905630	10088	1104	246.00693847	0.2228324	
9	8202	0.7789914	10529	2327	379.54811405	0.1631062	
10	3689	0.4788422	7704	4015	622.22022041	0.1549739	
total	92944	0.9152986	101545	8601	880.66723938	0.1023913	

```
R> summary(glmReserve(triangle))
```

	Latest	Dev.To.Date	Ultimate	IBNR		S.E	CV
2	13113	1.0000000	13113	0	0.01035287		Inf
3	15720	0.9992372	15732	12	29.38975320	2.4491461	
4	13872	0.9934116	13964	92	73.46098788	0.7984890	
5	11282	0.9850694	11453	171	96.26787355	0.5629700	
6	8757	0.9688019	9039	282	120.58781735	0.4276164	
7	9325	0.9397360	9923	598	177.57019068	0.2969401	
8	8984	0.8905630	10088	1104	246.00693847	0.2228324	
9	8202	0.7789914	10529	2327	379.54811405	0.1631062	
10	3689	0.4788422	7704	4015	622.22022041	0.1549739	
total	92944	0.9152986	101545	8601	880.66723938	0.1023913	

Functional-based (nonparametric) alternatives

Why?

Functional-based (nonparametric) alternatives

Why?

- ❑ Parametric benchmark techniques (development ratios, the Mack model, or ODP models) require rather strict assumptions with limited flexibility and practical applicability

Functional-based (nonparametric) alternatives

Why?

- ❑ Parametric benchmark techniques (development ratios, the Mack model, or ODP models) require rather strict assumptions with limited flexibility and practical applicability
- ❑ In practice, overall reserves for many portfolios (lines of business respectively) are only predicted in terms of some “expert judgement” with no theoretical foundations behind

Functional-based (nonparametric) alternatives

Why?

- ❑ Parametric benchmark techniques (development ratios, the Mack model, or ODP models) require rather strict assumptions with limited flexibility and practical applicability
- ❑ In practice, overall reserves for many portfolios (lines of business respectively) are only predicted in terms of some “expert judgement” with no theoretical foundations behind
- ❑ A complementary R package could implement these nonparametric methods while taking a full advantage of various S3 methods in R while providing cross-package functionality

Functional-based (nonparametric) alternatives

Why?

- ❑ Parametric benchmark techniques (development ratios, the Mack model, or ODP models) require rather strict assumptions with limited flexibility and practical applicability
- ❑ In practice, overall reserves for many portfolios (lines of business respectively) are only predicted in terms of some “expert judgement” with no theoretical foundations behind
- ❑ A complementary R package could implement these nonparametric methods while taking a full advantage of various S3 methods in R while providing cross-package functionality

How?

- ▶ The building block of all of the proposed methods are so-called functional development profiles, respectively the patterns of loss emergence (Clark, 2003)
- ▶ The overall reserves are modeled in a nonparametric way with no practical limitations for the underlying data (no need for expert judgements) (Maciak, Mizera, Pešta, 2022)
- ▶ Complex software toolbox in the R package ProfileLadder including exploratory and confirmatory statistical tools and S3 methods (Maciak, Matúš, Mizera, and Pešta, 2025)

Three competitive/complementary algorithms

Three nonparametric algorithms and also three core functions implemented in the R package [ProfileLadder](#) which complements the core actuarial R package [ChainLadder](#)

However, the implementation details of the algorithms in the R package [ProfileLadder](#) are much broader than those originally proposed in Maciak, Mizera, and Pešta (2022)

- ❑ **PARALLAX** – optimization-based (point) prediction algorithm
- ❑ **REACT** – simple recursive-based (point) prediction algorithm
- ❑ **MACRAME** – model-based Markov chain (point) prediction algorithm
- ❑ **Permutation bootstrap** – resampling method for distributional reserve predictions
(implemented for all three non-parametric algorithms above but also any parametric method from ChainLadder)

PARALLAX: Parallel approximation of missing fragments

- ▶ Observed (cummulative) triangle $Y_{i,j}$, for $i = 1, \dots, n$ and $j = 1, \dots, n + 1 - i$
- ▶ Set the observed as the predicted $\hat{Y}_{i,j} = Y_{i,j}$ for $i = 1, \dots, n$ and $j = 1, \dots, n + 1 - i$

PARALLAX: Parallel approximation of missing fragments

- ▶ Observed (cumulative) triangle $Y_{i,j}$, for $i = 1, \dots, n$ and $j = 1, \dots, n + 1 - i$
- ▶ Set the observed as the predicted $\hat{Y}_{i,j} = Y_{i,j}$ for $i = 1, \dots, n$ and $j = 1, \dots, n + 1 - i$
- ▶ For $i = 2, \dots, n$ and $j = n - i + 1, \dots, n - 1$, find the most similar development profile

$$\hat{\ell}_{i,j} = \arg \min_{\ell \in \{1, \dots, n-j\}} \left| \hat{Y}_{i,j} - Y_{\ell,j} \right| \quad (1)$$

- ▶ Predict the unobserved (future) value $Y_{i,j+1}$ such that

$$\hat{Y}_{i,j+1} = \hat{Y}_{i,j} + \left(Y_{\hat{\ell}_{i,j},j+1} - Y_{\hat{\ell}_{i,j},j} \right) \quad (2)$$

- ▶ Complete square $\{\hat{Y}_{i,j}; i = 1, \dots, n; j = 1, \dots, n\}$ and $\hat{\mathcal{R}} = \sum_{i=2}^n \hat{Y}_{i,n} - \sum_{i=2}^n \hat{Y}_{i,(n-i+1)}$

REACT: Approximation by the most recent accident year

- ▶ Observed triangle $Y_{i,j}$, for $i = 1, \dots, n$ and $j = 1, \dots, n + 1 - i$
- ▶ Set the observed as the predicted $\hat{Y}_{i,j} = Y_{i,j}$ for $i = 1, \dots, n$ and $j = 1, \dots, n + 1 - i$

REACT: Approximation by the most recent accident year

- ▶ Observed triangle $Y_{i,j}$, for $i = 1, \dots, n$ and $j = 1, \dots, n + 1 - i$
- ▶ Set the observed as the predicted $\hat{Y}_{i,j} = Y_{i,j}$ for $i = 1, \dots, n$ and $j = 1, \dots, n + 1 - i$
- ▶ Predict the unobserved (future) value $Y_{i,j+1}$ such that

$$\hat{Y}_{i,j+1} = \hat{Y}_{i,j} + (\hat{Y}_{i-1,j+1} - \hat{Y}_{i-1,j}) \quad (3)$$

- ▶ Complete square $\{\hat{Y}_{i,j}; i = 1, \dots, n; j = 1, \dots, n\}$ – i.e., the completed run-off triangle and the predicted reserve $\hat{\mathcal{R}} = \sum_{i=2}^n \hat{Y}_{i,n} - \sum_{i=2}^n \hat{Y}_{i,(n-i+1)}$

MACRAME: Markov chain fragment approximation

Step 1

(Incremental run-off triangle)

- ▶ Calculate incremental claims $X_{i,j} = Y_{i,j} - Y_{i,j-1}$, where $Y_{i,0} = 0$
- ▶ Set $\hat{X}_{i,j} = X_{i,j}$ and $\hat{Y}_{i,j} = Y_{i,j}$ for $i = 1, \dots, n$ and $j = 1, \dots, n+1-i$

Step 2

(Markov chain definition/estimation)

- ▶ Define Markov states $\mathcal{S}$ and transform $\hat{X}_{i,j}$ into $U_{i,j}$ for $i = 1, \dots, n$ and $j = 1, \dots, n+1-i$ such that $U_{i,j} = s$, if $\hat{X}_{i,j} \in [g_{k-1}, g_k)$ and $\mathcal{S} \ni s \in [g_{k-1}, g_k)$
- ▶ Estimate the transition probability matrix $\hat{\mathbb{P}} = (\hat{p}(s, s'))_{s, s' \in \mathcal{S}}$ using the set of independent Markov chains $\{U_{i,j}; j = 1, \dots, n-i+1\}_i$, for $i = 1, \dots, n$

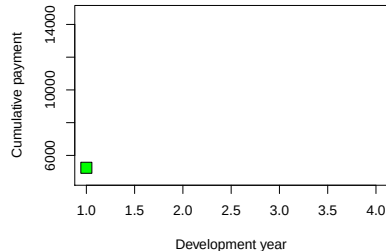
Step 3

(Increments prediction / transformation into cumulative)

- ▶ Predict the unobserved $X_{i,n+1-i+h}$ as $\hat{X}_{i,n+1-i+h} = \mathbf{c}(U_{i,n+1-i})^\top \hat{\mathbb{P}}^h \mathbf{s}$, where $\mathbf{s} = (s_1, \dots, s_t)^\top$ and $\mathbf{c}(U_{i,n+1-i})^\top = (\mathbb{1}\{U_{i,n+1-i} = s_1\}, \dots, \mathbb{1}\{U_{i,n+1-i} = s_t\})$
- ▶ Compute the predicted cumulative amount $\hat{Y}_{i,n+1-i+h} = \hat{Y}_{i,n-i+h} + \hat{X}_{i,n+1-i+h}$

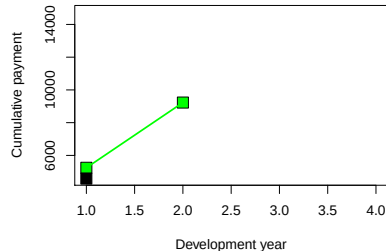
Run-off triangle vs. development profiles

Accid. year i	Development year j									
	1	2	3	4	5	6	7	8	9	10
1	5244	9228	10823	11352	11791	12082	12120	12199	12215	12215
2	4584	8739	11725	12346	12746	12909	13034	13109	13113	13115
3	7452	12421	14171	14752	15066	15354	15637	15720	15744	15786
4	7115	11117	12488	13274	13662	13859	13872	13935	13973	13972
5	5753	8969	9917	10697	11135	11282	11255	11331	11332	11354
6	3937	6524	7989	8543	8757	8901	9013	9012	9046	9164
7	5127	8212	8976	9325	9718	9795	9833	9885	9816	9815
8	5046	8006	8984	9633	10102	10166	10261	10252	10252	10252
9	5129	8202	9185	9681	9951	10033	10133	10182	10182	10183
10	3689	6043	6789	7089	7164	7197	7253	7267	7266	7266



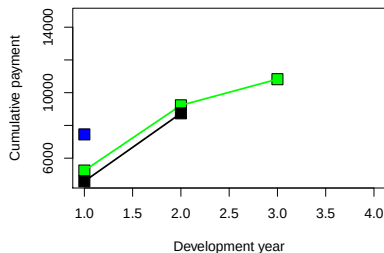
Run-off triangle vs. development profiles

Accid. year i	Development year j									
	1	2	3	4	5	6	7	8	9	10
1	5244	9228	10823	11352	11791	12082	12120	12199	12215	12215
2	4584	8739	11725	12346	12746	12909	13034	13109	13113	13115
3	7452	12421	14171	14752	15066	15354	15637	15720	15744	15786
4	7115	11117	12488	13274	13662	13859	13872	13935	13973	13972
5	5753	8969	9917	10697	11135	11282	11255	11331	11332	11354
6	3937	6524	7989	8543	8757	8901	9013	9012	9046	9164
7	5127	8212	8976	9325	9718	9795	9833	9885	9816	9815
8	5046	8006	8984	9633	10102	10166	10261	10252	10252	10252
9	5129	8202	9185	9681	9951	10033	10133	10182	10182	10183
10	3689	6043	6789	7089	7164	7197	7253	7267	7266	7266



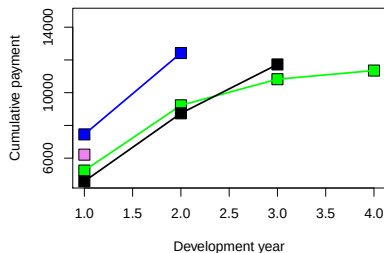
Run-off triangle vs. development profiles

Accid. year i	Development year j									
	1	2	3	4	5	6	7	8	9	10
1	5244	9228	10823	11352	11791	12082	12120	12199	12215	12215
2	4584	8739	11725	12346	12746	12909	13034	13109	13113	13115
3	7452	12421	14171	14752	15066	15354	15637	15720	15744	15786
4	7115	11117	12488	13274	13662	13859	13872	13935	13973	13972
5	5753	8969	9917	10697	11135	11282	11255	11331	11332	11354
6	3937	6524	7989	8543	8757	8901	9013	9012	9046	9164
7	5127	8212	8976	9325	9718	9795	9833	9885	9816	9815
8	5046	8006	8984	9633	10102	10166	10261	10252	10252	10252
9	5129	8202	9185	9681	9951	10033	10133	10182	10182	10183
10	3689	6043	6789	7089	7164	7197	7253	7267	7266	7266



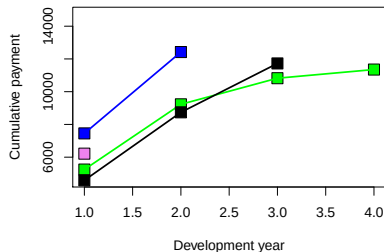
Run-off triangle vs. development profiles

Accid. year i	Development year j									
	1	2	3	4	5	6	7	8	9	10
1	5244	9228	10823	11352	11791	12082	12120	12199	12215	12215
2	4584	8739	11725	12346	12746	12909	13034	13109	13113	13115
3	7452	12421	14171	14752	15066	15354	15637	15720	15744	15786
4	7715	11117	12488	13274	13662	13859	13872	13935	13973	13972
5	5753	8969	9917	10697	11135	11282	11255	11331	11332	11354
6	3937	6524	7989	8543	8757	8901	9013	9012	9046	9164
7	5127	8212	8976	9325	9718	9795	9833	9885	9816	9815
8	5046	8006	8984	9633	10102	10166	10261	10252	10252	10252
9	5129	8202	9185	9681	9951	10033	10133	10182	10182	10183
10	3689	6043	6789	7089	7164	7197	7253	7267	7266	7266



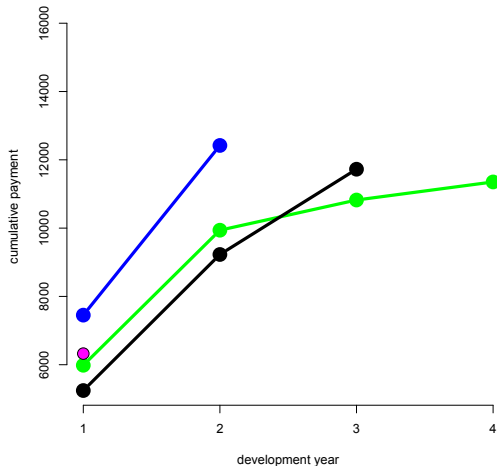
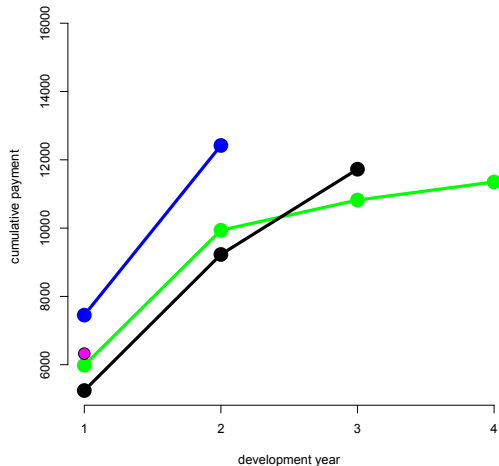
Run-off triangle vs. development profiles

Accid. year i	Development year j									
	1	2	3	4	5	6	7	8	9	10
1	5244	9228	10823	11352	11791	12082	12120	12199	12215	12215
2	4584	8739	11725	12346	12746	12909	13034	13109	13113	13115
3	7452	12421	14171	14752	15066	15354	15637	15720	15744	15786
4	7715	11117	12488	13274	13662	13859	13872	13935	13973	13972
5	5753	8969	9917	10697	11135	11282	11255	11331	11332	11354
6	3937	6524	7989	8543	8757	8901	9013	9012	9046	9164
7	5127	8212	8976	9325	9718	9795	9833	9885	9816	9815
8	5046	8006	8984	9633	10102	10166	10261	10252	10252	10252
9	5129	8202	9185	9681	9951	10033	10133	10182	10182	10183
10	3689	6043	6789	7089	7164	7197	7253	7267	7266	7266

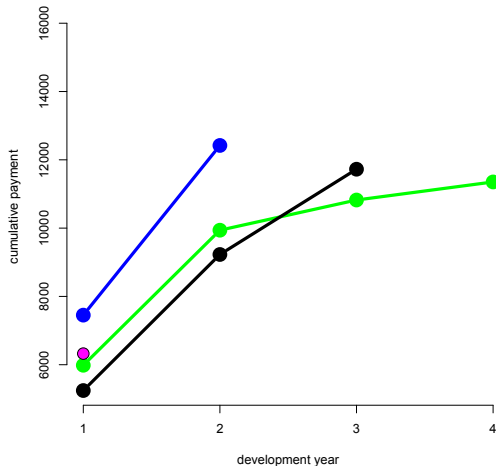
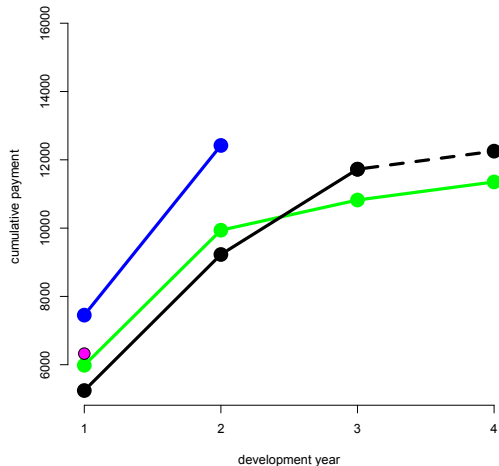


And so on ...

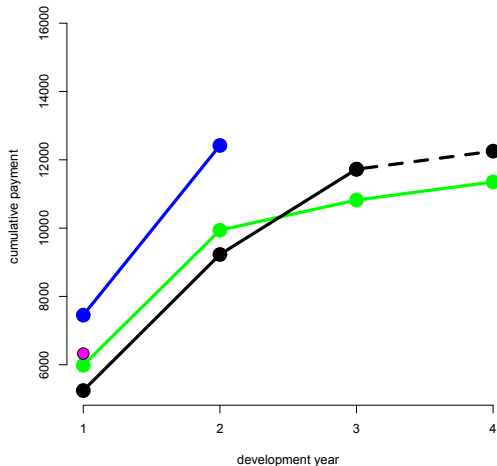
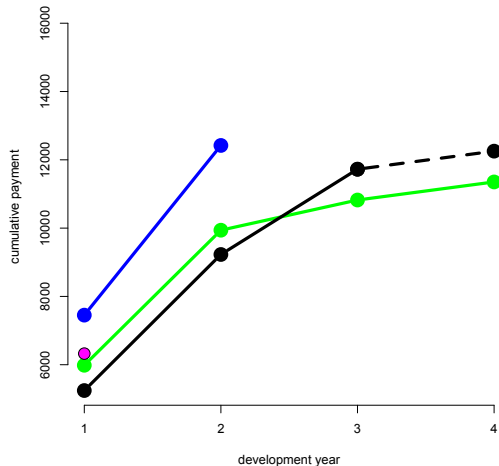
Reconstructing missing profiles: PARALLAX vs. REACT



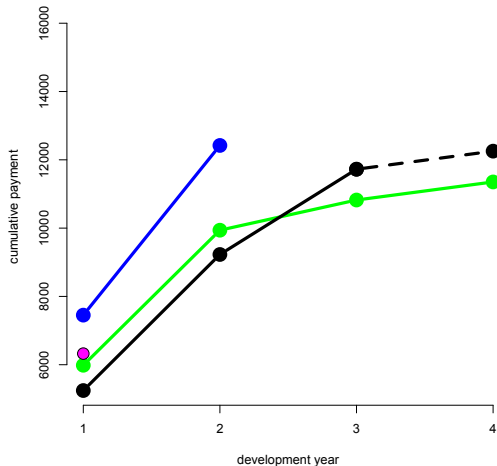
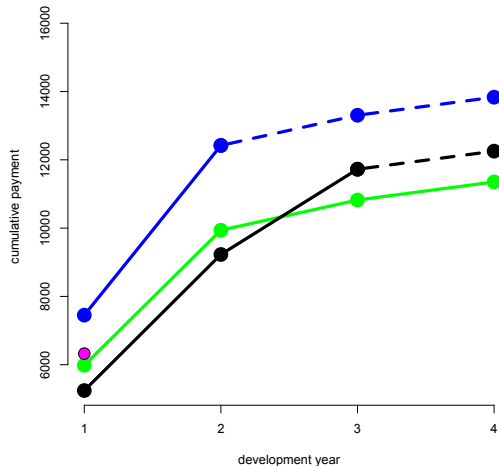
Reconstructing missing profiles: PARALLAX vs. REACT



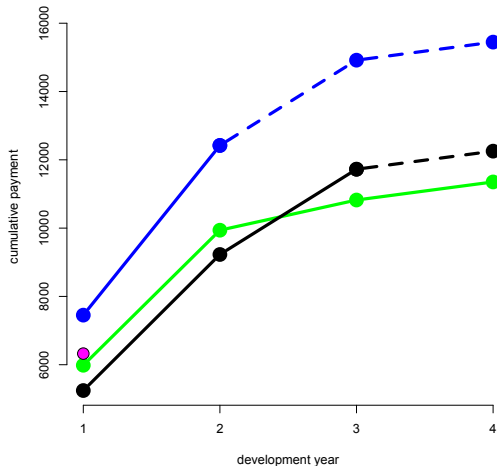
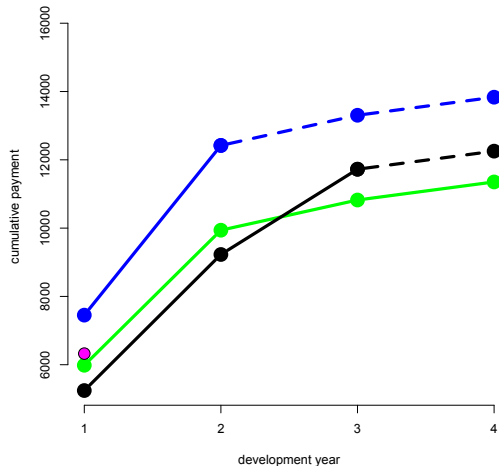
Reconstructing missing profiles: PARALLAX vs. REACT



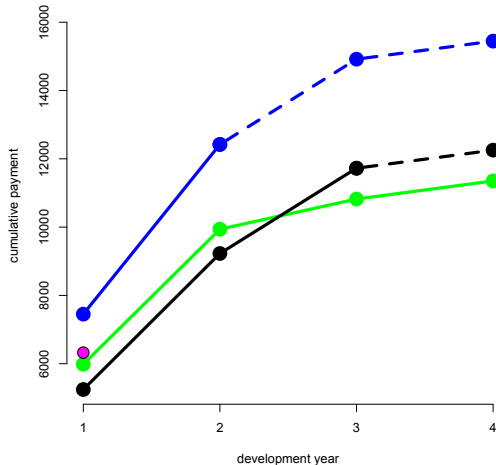
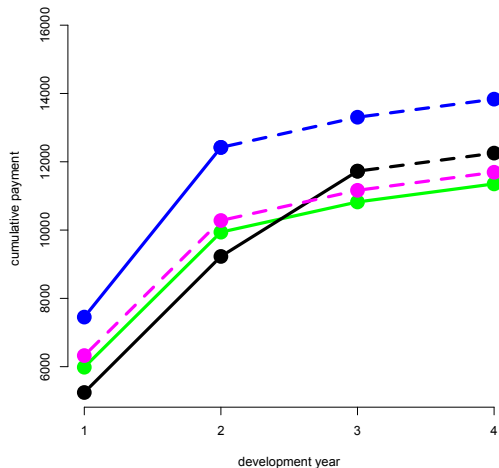
Reconstructing missing profiles: PARALLAX vs. REACT



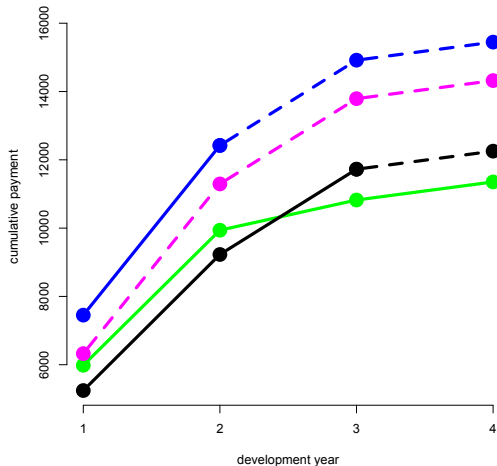
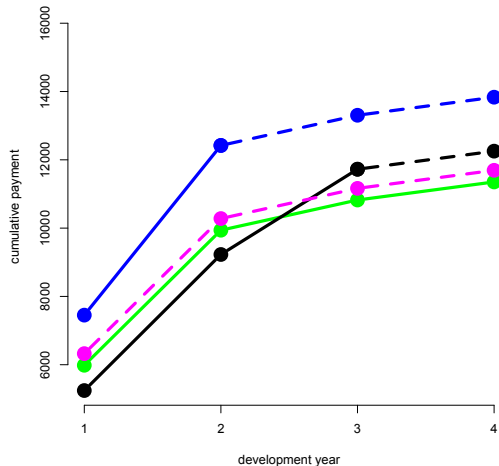
Reconstructing missing profiles: PARALLAX vs. REACT



Reconstructing missing profiles: PARALLAX vs. REACT



Reconstructing missing profiles: PARALLAX vs. REACT



Reconstructing missing profiles by MACRAME

- ❑ A relatively complex stochastic model that drives the segments reconstruction and allows for various [user-based modifications](#) and [fine-tuning adjustments](#)
- ❑ Additional software tools are provided in the R package to make appropriate user-based decisions regarding the Markov chain setup (exploratory and confirmatory analytic tools)
- ❑ **Default data-driven approach:**

Accid. year i	Development j			
	1	2	3	4
1	5244	9228	10823	11352
2	5984	9939	11725	.
3	7452	12421	.	.
4	7115	.	.	.

(a) Cumulative triangle

Reconstructing missing profiles by MACRAME

- ❑ A relatively complex stochastic model that drives the segments reconstruction and allows for various **user-based modifications** and **fine-tuning adjustments**
- ❑ Additional software tools are provided in the R package to make appropriate user-based decisions regarding the Markov chain setup (exploratory and confirmatory analytic tools)
- ❑ **Default data-driven approach:**

Accid. year i	Development j			
	1	2	3	4
1	5244	9228	10823	11352
2	5984	9939	11725	.
3	7452	12421	.	.
4	7115	.	.	.

Accid. year i	Development j			
	1	2	3	4
1	5244	3984	1595	592
2	5984	3955	1786	.
3	7452	4946	.	.
4	7115	.	.	.

(a) Cumulative triangle $\longrightarrow$ (b) Incremental triangle

Reconstructing missing profiles by MACRAME

- ❑ A relatively complex stochastic model that drives the segments reconstruction and allows for various **user-based modifications** and **fine-tuning adjustments**
- ❑ Additional software tools are provided in the R package to make appropriate user-based decisions regarding the Markov chain setup (exploratory and confirmatory analytic tools)
- ❑ **Default data-driven approach:**

Accid. year i	Development j			
	1	2	3	4
1	5244	9228	10823	11352
2	5984	9939	11725	.
3	7452	12421	.	.
4	7115	.	.	.

(a) Cumulative triangle



Accid. year i	Development j			
	1	2	3	4
1	5244	3984	1595	592
2	5984	3955	1786	.
3	7452	4946	.	.
4	7115	.	.	.

(b) Incremental triangle



Number of states:	$m = n = 4$
Ordered increments:	$\{529, 1595, 1786, 3955, 3984, 4969\}$
Break point indexes:	$\left\{ \left\lceil \frac{kn(n-1)}{2m} \right\rceil + 1; k = 1, 2, 3 \right\} \equiv \{3, 4, 6\}$
Break points $\{g_k\}_{k=0}^4$:	$\{-\infty, 1786, 3955, 4964, \infty\}$
Markov states:	medians of increments in bins $[g_{k-1}, g_k)$

(c) Increments, breaks, and MC states

Reconstructing missing profiles by MACRAME

- ❑ A relatively complex stochastic model that drives the segments reconstruction and allows for various **user-based modifications** and **fine-tuning adjustments**
- ❑ Additional software tools are provided in the R package to make appropriate user-based decisions regarding the Markov chain setup (exploratory and confirmatory analytic tools)
- ❑ **Default data-driven approach:**

Accid. year i	Development j			
	1	2	3	4
1	5244	9228	10823	11352
2	5984	9939	11725	.
3	7452	12421	.	.
4	7115	.	.	.

Accid. year i	Development j			
	1	2	3	4
1	5244	3984	1595	592
2	5984	3955	1786	.
3	7452	4946	.	.
4	7115	.	.	.

Number of states:	$m = n = 4$
Ordered increments:	$\{529, 1595, 1786, 3955, 3984, 4969\}$
Break point indexes:	$\left\{ \left\lceil \frac{kn(n-1)}{2m} \right\rceil + 1; k = 1, 2, 3 \right\} \equiv \{3, 4, 6\}$
Break points $\{g_k\}_{k=0}^4$:	$\{-\infty, 1786, 3955, 4964, \infty\}$
Markov states:	medians of increments in bins $[g_{k-1}, g_k)$

(a) Cumulative triangle $\longrightarrow$ (b) Incremental triangle $\longrightarrow$ (c) Increments, breaks, and MC states

(d) Markov chain kick off:

The resulting Markov states are: $s_1 = 1062$ (i.e., the median of the first two increments, $\{529, 1595\}$, from the first bin, $[-\infty, 1786)$); $s_2 = 1786$ (which is the only increment in the second bin being $[1786, 3955)$); $s_3 = 3970$ (again the median of two increments, $\{3955, 3984\}$, from the third bin); and, finally, $s_4 = 4969$ which is the only increment in the last bin—the interval $[g_3, \infty)$

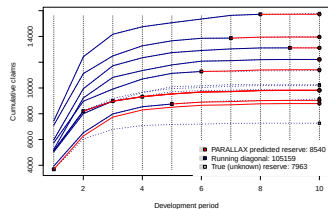
Point predictions of the overall reserve

Visualization of the predictions with the **S3 method** `plot()` from `ProfileLadder`:

Point predictions of the overall reserve

Visualization of the predictions with the **S3** method `plot()` from `ProfileLadder`:

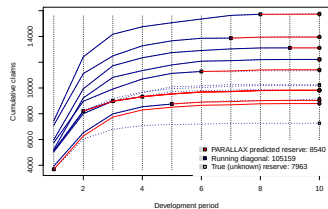
❑ PARALLAX



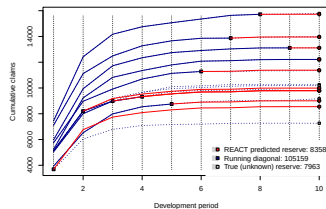
Point predictions of the overall reserve

Visualization of the predictions with the **S3** method `plot()` from `ProfileLadder`:

☐ PARALLAX



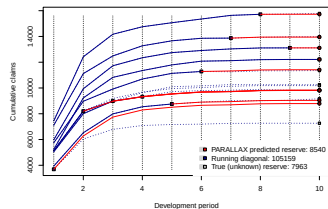
☐ REACT



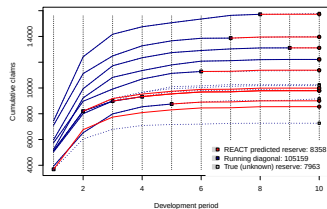
Point predictions of the overall reserve

Visualization of the predictions with the **S3** method `plot()` from `ProfileLadder`:

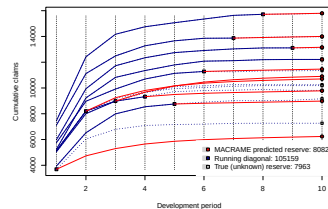
❑ PARALLAX



❑ REACT



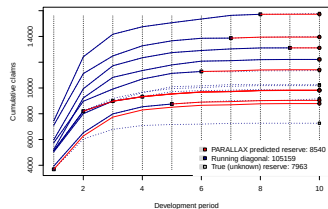
❑ MACRAME



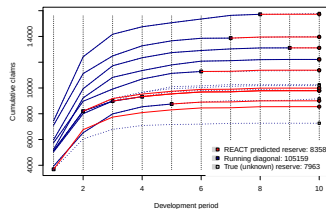
Point predictions of the overall reserve

Visualization of the predictions with the **S3** method `plot()` from `ProfileLadder`:

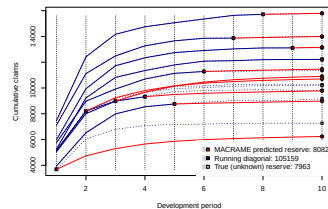
❑ PARALLAX



❑ REACT



❑ MACRAME



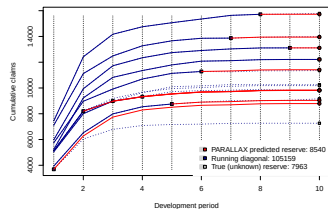
❑ The true reserve to be predicted: **7963** (in thousands of US dollars)

❑ Functional-based predictions: **PARALLAX: 8540**; **REACT: 8358**; **MACRAME: 8082**

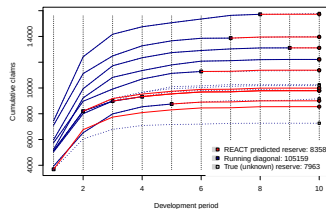
Point predictions of the overall reserve

Visualization of the predictions with the **S3 method** `plot()` from `ProfileLadder`:

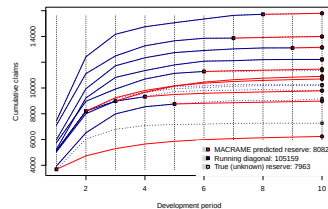
❑ PARALLAX



❑ REACT



❑ MACRAME



- ❑ The true reserve to be predicted: **7963** (in thousands of US dollars)
- ❑ Functional-based predictions: **PARALLAX: 8540**; **REACT: 8358**; **MACRAME: 8082**
- ❑ Classical methods: **ChainLadder: 8687**; **Mack model/ODP: 8601**; **Tweedie model: 8610**

From point reserve predictions to overall distribution

- ❑ Reliable (good) point reserve predictions are only the beginning for overall the risk reserving process... the goal is to get the **overall reserve distribution**
- ❑ In general, the reliability of any estimate/prediction can be assessed differently in statistics... a common concept used for reserving purposes is **consistency**
- ❑ A point prediction alone provides no information about the uncertainty surrounding the predicted reserve – in particular its variability, asymmetry, or **tail behaviour**
- ❑ Prediction methods that are proved to produce consistent reserve (point) predictions can be further generalized or equipped with some **resampling or MCMC method**
- ❑ Once a predictive distribution is available, one can evaluate quantities that are directly relevant for risk management – **tail probability, quantile/VaR, or CVaR**

Consistency of the nonparametric reserving methods

- ❑ Statistical consistency of all three nonparametric methods (for $n \rightarrow \infty$) is based on the concept of a **conditional convergence in L_2 in probability** (Belyaev, 1995)
- ❑ The consistency of the classical chain ladder method can be (also) expressed as

$$E \left[|\hat{f}_j^{(n)} - f_j|^2 \mid T_{n+1-j,j}^{(n)} \right] = \frac{\sigma_j^2}{\sum_{i=1}^{n-j} Y_{i,j}}, \quad [P] - a.s.$$

for $T_{i,j}^{(n)} \equiv \{Y_{k,l}; k \leq i; l \leq j; k+l \leq n+1\}$ and finite variances $\text{Var}Y_{i,j} = \sigma_j^2 < \infty$

Consistency of the nonparametric reserving methods

- Statistical consistency of all three nonparametric methods (for $n \rightarrow \infty$) is based on the concept of a **conditional convergence in L_2 in probability** (Belyaev, 1995)
- The consistency of the classical chain ladder method can be (also) expressed as

$$E \left[|\hat{f}_j^{(n)} - f_j|^2 \mid T_{n+1-j,j}^{(n)} \right] = \frac{\sigma_j^2}{\sum_{i=1}^{n-j} Y_{i,j}}, \quad [P] - a.s.$$

for $T_{i,j}^{(n)} \equiv \{Y_{k,l}; k \leq i; l \leq j; k+l \leq n+1\}$ and finite variances $Var Y_{i,j} = \sigma_j^2 < \infty$

- Analogous assumptions + some technical conditions are also imposed for the nonparametric methods (PARALLAX, REACT, and MACRAME)
 - the closest historical value $Y_{\hat{\ell}_{n+\kappa-j,j}}$ from $\{Y_{1,j}, \dots, Y_{n-j,j}\}$ should be sufficiently close to $EY_{n+\kappa-j,j}$
 - within the same development period j the previous value $Y_{n-j,j}$ and the current value $Y_{n-j+1,j}$ are getting closer in probability as $n \rightarrow \infty$
 - homogeneous underlying Markov chain and $\frac{1}{n} \sum_{j=1}^{n-1} \frac{1}{n-j} \sum_{i=1}^{n-j} P[U_{i,j} = s]$ is bounded away from zero

Consistency of PARALLAX

- PARALLAX estimator of the development ratios f_j

$$\widehat{f}_{i,j}^{(n)} \xrightarrow[n \rightarrow \infty]{L_2(T_{i,j}^{(n)})} f_j \quad \text{in probability } P$$

- The conditional mean square error of the PARALLAX estimator of f_j is

$$\mathbb{E} \left[\left\{ \widehat{f}_{i,j}^{(n)} - f_j \right\}^2 \middle| T_{i,j}^{(n)} \right] = \frac{\sigma_j^2 Y_{\widehat{\ell}_{i,j},j}}{Y_{n-j+1,j}^2} + (f_j - 1)^2 \left(\frac{Y_{\widehat{\ell}_{i,j},j}}{Y_{n-j+1,j}} - 1 \right)^2 \quad [P] - a.s.,$$

- Thus, the conditional mean square error is roughly $\frac{\sigma_j^2}{Y_{n-j+1}}$ which is typically greater than the conditional mean square error of the chain ladder development factors' estimators, since $Y_{n-j+1} < \sum_{i=1}^{n-j} Y_{i,j}$, $[P] - a.s.$ (but there is some row-wise additional flexibility)

Consistency of REACT

- REACT estimator of the development ratios f_j

$$\widehat{f}_j^{(n)} \xrightarrow[n \rightarrow \infty]{L_2(T_{n-j+1,j}^{(n)})} f_j \quad \text{in probability } P$$

- The conditional mean square error of the REACT estimator of f_j is

$$\mathbb{E} \left[\left\{ \widehat{f}_j^{(n)} - f_j \right\}^2 \middle| T_{n-j+1,j}^{(n)} \right] = \frac{\sigma_j^2 Y_{n-j,j}}{Y_{n-j+1,j}^2} + (f_j - 1)^2 \left(\frac{Y_{n-j,j}}{Y_{n-j+1,j}} - 1 \right)^2 \quad [P] - a.s.,$$

Consistency of REACT

- REACT estimator of the development ratios f_j

$$\hat{f}_j^{(n)} \xrightarrow[n \rightarrow \infty]{L_2(T_{n-j+1,j}^{(n)})} f_j \quad \text{in probability } P$$

- The conditional mean square error of the REACT estimator of f_j is

$$\mathbb{E} \left[\left\{ \hat{f}_j^{(n)} - f_j \right\}^2 \middle| T_{n-j+1,j}^{(n)} \right] = \frac{\sigma_j^2 Y_{n-j,j}}{Y_{n-j+1,j}^2} + (f_j - 1)^2 \left(\frac{Y_{n-j,j}}{Y_{n-j+1,j}} - 1 \right)^2 \quad [P] - a.s.,$$

- The independence of the development factor estimates $\hat{f}_j$ on the triangle rows $i \in \{1, \dots, n\}$ comes from the fact that $\hat{f}_j = 1 + \frac{Y_{n-j,j+1} - Y_{n-j,j}}{Y_{n-j+1,j}}$

Consistency of MACRAME

- Consistent estimation of the transition probabilities (transitions among increments $X_{i,j}$)

$$\hat{p}(s_1, s_2) = p(s_1, s_2) + O_P(n^{-1/2}), \quad n \rightarrow \infty,$$

- Unbiased prediction of the future states in the underlying Markov chain

$$\mathbb{E} \left[\hat{\mathbb{E}}[U_{i,j+1}] \right] = \mathbb{E} \left[\sum_{s_1 \in \mathcal{S}} \hat{p}(s_1) \left(\sum_{s_2 \in \mathcal{S}} \hat{p}(s_1, s_2) s_2 \right) \right] = \mathbb{E} [U_{i,j+1}]$$

where $\hat{X}_{i,j+1} = \hat{E}[U_{i,j+1} | U_{i,j} = s_1] = \sum_{s_2 \in \mathcal{S}} \hat{p}(s_1, s_2) s_2$ is the prediction for $X_{i,j+1}$ and $\hat{p}(s_1) = \frac{1}{n-1} \sum_{j=1}^{n-1} \frac{1}{n-j} \sum_{i=1}^{n-j} \mathbb{I}_{\{U_{i,j}=s_1\}}$ and $\hat{p}(s_1, s_2) = \frac{1}{n-1} \sum_{j=1}^{n-1} \frac{1}{n-j} \sum_{i=1}^{n-j} \mathbb{I}_{\{U_{i,j}=s_1; U_{i,j+1}=s_2\}} / \hat{p}(s_1)$

Overall distributional prediction

European Parliament and Council (2009). *Directive 2009/138/EC of the European Parliament and of the Council of 25 November 2009 on the taking-up and pursuit of the business of Insurance and Reinsurance (Solvency II).* Official Journal of the European Union, 52(L 335):1–5, 2009

Overall distributional prediction

European Parliament and Council (2009). *Directive 2009/138/EC of the European Parliament and of the Council of 25 November 2009 on the taking-up and pursuit of the business of Insurance and Reinsurance (Solvency II).*

Official Journal of the European Union, 52(L 335):1–5, 2009

- ❑ **Imposing second order conditions or some specific distributional assumptions**
 - ❑ Mack model
 - ❑ Tweedie model
 - ❑ GLM-based reserving

Overall distributional prediction

European Parliament and Council (2009). *Directive 2009/138/EC of the European Parliament and of the Council of 25 November 2009 on the taking-up and pursuit of the business of Insurance and Reinsurance (Solvency II).*

Official Journal of the European Union, 52(L 335):1–5, 2009

- ❑ **Imposing second order conditions or some specific distributional assumptions**
 - ❑ Mack model
 - ❑ Tweedie model
 - ❑ GLM-based reserving
- ❑ **Implementing resampling (MCMC or bootstrap) methods**
 - ❑ incremental residual bootstrap (with standard or back-fitted residuals)
 - ❑ functional permutation bootstrap (proposed for nonparametric reserving)

Overall distributional prediction

European Parliament and Council (2009). *Directive 2009/138/EC of the European Parliament and of the Council of 25 November 2009 on the taking-up and pursuit of the business of Insurance and Reinsurance (Solvency II).* Official Journal of the European Union, 52(L 335):1–5, 2009

- ❑ **Imposing second order conditions or some specific distributional assumptions**
 - ❑ Mack model
 - ❑ Tweedie model
 - ❑ GLM-based reserving
- ❑ **Implementing resampling (MCMC or bootstrap) methods**
 - ❑ incremental residual bootstrap (with standard or back-fitted residuals)
 - ❑ functional permutation bootstrap (proposed for nonparametric reserving)
- ❑ **Conformal inference techniques**
 - ❑ adaptations beyond exchangeability (Shafer and Vovk, 2007; HHMP 2026?)

Overall distributional prediction

European Parliament and Council (2009). *Directive 2009/138/EC of the European Parliament and of the Council of 25 November 2009 on the taking-up and pursuit of the business of Insurance and Reinsurance (Solvency II).*

Official Journal of the European Union, 52(L 335):1–5, 2009

- ❑ **Imposing second order conditions or some specific distributional assumptions**
 - ❑ Mack model
 - ❑ Tweedie model
 - ❑ GLM-based reserving
- ❑ **Implementing resampling (MCMC or bootstrap) methods**
 - ❑ incremental residual bootstrap (with standard or back-fitted residuals)
 - ❑ **functional permutation bootstrap** (proposed for nonparametric reserving)
- ❑ **Conformal inference techniques**
 - ❑ adaptations beyond exchangeability (Shafer and Vovk, 2007; HHMP 2026?)

Resampling predicted development profiles

- ❑ **Step 1** The **predicted lower triangle** $\{\hat{Y}_{i,j} : i = 2, \dots, n; j = n + 2 - i, \dots, n\}$, where the upper triangle predicted elements are kept as the original ones, i.e., $\hat{Y}_{i,j} = Y_{i,j}$ for $i + j \leq n + 1$
- ❑ **Step 2** The full predicted square $\{\hat{Y}_{i,j}\}_{i=1,j=1}^{n,n}$ is **standardized** such that each row value is divided by the first positive value within the row, i.e., $\tilde{Y}_{i,j} := \hat{Y}_{i,j} / \hat{Y}_{i,p_i}$, $p_i = \min \{j \in \{1, \dots, n\} : \hat{Y}_{i,j} > 0\}$
- ❑ **Step 3** The square $\{\tilde{Y}_{i,j}\}_{i=1,j=1}^{n,n}$ is resampled in a **row-wise manner without replacement**—for every permutation $\pi^{(b)} : (1, \dots, n) \mapsto (\pi^{(b)}(1), \dots, \pi^{(b)}(n))$, where $b = 1, \dots, B$ such that $\pi^{(b)} \neq \pi^{(s)}$ if $b \neq s$, a new permuted square $\{\tilde{Y}_{\pi^{(b)}(i),j}\}_{i=1,j=1}^{n,n}$ is obtained
- ❑ **Step 4** The prediction algorithms is applied on the **cut upper triangles** $\{\tilde{Y}_{\pi^{(b)}(i),j} : \pi^{(b)}(i) + j \leq n + 1\}$, obtaining new standardized cumulative amounts $\{\check{Y}_{i,j}^{(b)}\}_{i,j}$ for $i = 2, \dots, n$ and $j = n + 2 - i, \dots, n$
- ❑ **Step 5** Standardized $\check{Y}_{i,j}^{(b)}$'s are **back-standardized** yielding the bootstrapped predicted values

$$\left\{ \check{Y}_{i,j}^{(b)} := \check{Y}_{i,j}^{(b)} \hat{Y}_{i,p_i} \right\}_{i,j} \quad \text{for } i = 2, \dots, n \quad \text{and} \quad j = n + 2 - i, \dots, n.$$

Permutation bootstrap by the ProfileLadder package

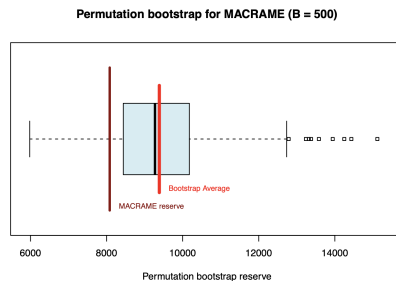
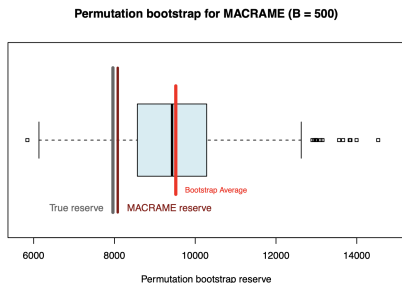
- ❑ Solvency II requires at least 10.000 bootstrap replications ($8! \approx 40.000$; $10! \approx 3.6mil.$)
- ❑ Graphical illustration with the **S3 method** `plot()` from `ProfileLadder`
- ❑ Complete (retrospective) portfolio vs.the actual run-off triangle

Accid.	Development year j									
year i	1	2	3	4	5	6	7	8	9	10
1	5244	9228	10823	11352	11791	12082	12120	12199	12215	12215
2	5984	9939	11725	12346	12746	12909	13034	13109	13113	13115
3	7452	12421	14171	14752	15066	15354	15637	15720	15744	15786
4	7115	11117	12488	13274	13662	13859	13872	13935	13973	13972
5	5753	8969	9917	10697	11135	11282	11255	11331	11332	11354
6	3937	6524	7989	8543	8757	8901	9013	9012	9046	9164
7	5127	8212	8976	9325	9718	9795	9833	9885	9816	9815
8	5046	8006	8984	9633	10102	10166	10261	10252	10252	10252
9	5129	8202	9185	9681	9951	10033	10133	10182	10182	10183
10	3689	6043	6789	7089	7164	7197	7253	7267	7266	7266

Accid.	Development year j									
year i	1	2	3	4	5	6	7	8	9	10
1	5244	9228	10823	11352	11791	12082	12120	12199	12215	12215
2	5984	9939	11725	12346	12746	12909	13034	13109	13113	.
3	7452	12421	14171	14752	15066	15354	15637	15720	.	.
4	7115	11117	12488	13274	13662	13859	13872	.	.	.
5	5753	8969	9917	10697	11135	11282	.	.	.	.
6	3937	6524	7989	8543	8757	.	.	.	.	.
7	5127	8212	8976	9325	.	.	.	.	.	.
8	5046	8006	8984	.	.	.	.	.	.	.
9	5129	8202	.	.	.	.	.	.	.	.
10	3689	.	.	.	.	.	.	.	.	.

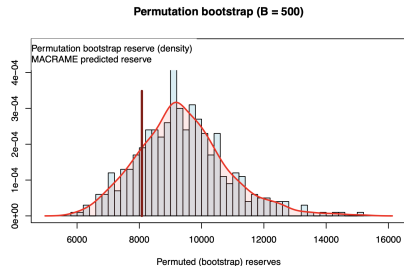
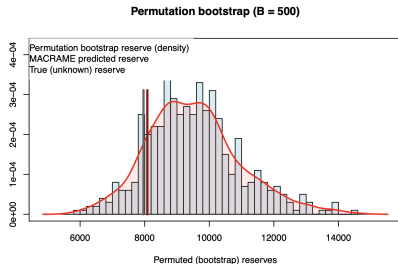
Permutation bootstrap by the ProfileLadder package

- ❑ Solvency II requires at least 10.000 bootstrap replications ($8! \approx 40.000$; $10! \approx 3.6\text{mil.}$)
- ❑ Graphical illustration with the **S3 method `plot()`** from **ProfileLadder**
- ❑ Boxplots of (permuted) bootstrap reserves $\{\hat{\mathcal{R}}^{(b)}\}_{b=1}^{1000}$



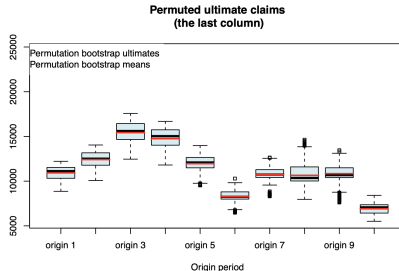
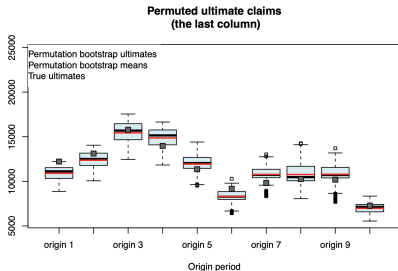
Permutation bootstrap by the ProfileLadder package

- ❑ Solvency II requires at least 10.000 bootstrap replications ($8! \approx 40.000$; $10! \approx 3.6\text{mil.}$)
- ❑ Graphical illustration with the **S3 method `plot()`** from **ProfileLadder**
- ❑ The bootstrap reserve distribution (histogram and the kernel density estimate)



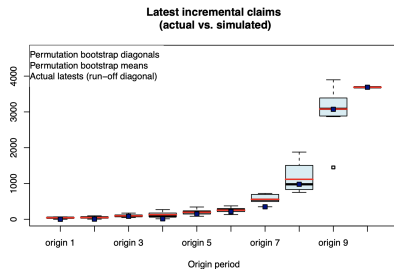
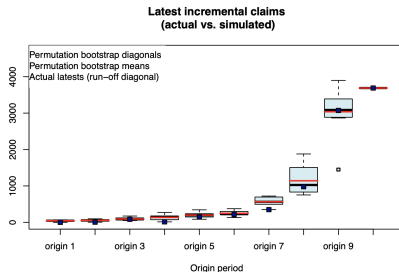
Permutation bootstrap by the ProfileLadder package

- ❑ Solvency II requires at least 10.000 bootstrap replications ($8! \approx 40.000$; $10! \approx 3.6\text{mil.}$)
- ❑ Graphical illustration with the **S3 method plot()** from **ProfileLadder**
- ❑ Simulated ultimate claims (row specific reserve predictions)



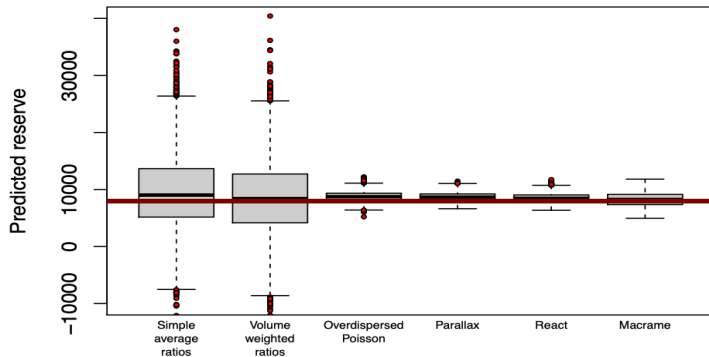
Permutation bootstrap by the ProfileLadder package

- ❑ Solvency II requires at least 10.000 bootstrap replications ($8! \approx 40.000$; $10! \approx 3.6\text{mil.}$)
- ❑ Graphical illustration with the **S3 method plot()** from **ProfileLadder**
- ❑ Simulated run-off diagonals – values $Y_{i,n-i+1}$, for $i = 1, \dots, n$



Distributional prediction: Classical vs. nonparametric methods

- ❑ **Cameron Mutual** insurance portfolio (10.000 bootstrap resamples/permutations)



Empirical “bullet-proof” comparisons on real run-off triangles

- ❑ The database of 518 real and completed run-off triangles (provided for testing purposes) from the [National Association of Insurance Commissioners \(NAIC\)](#) (Meyers and Shi, 2011)
- ❑ All triangles are also available in the [R package raw](#) however, the data are given in the so-called long-type format (data frames) not as typical run-off triangles

Empirical “bullet-proof” comparisons on real run-off triangles

- ❑ The database of 518 real and completed run-off triangles (provided for testing purposes) from the [National Association of Insurance Commissioners \(NAIC\)](#) (Meyers and Shi, 2011)
- ❑ All triangles are also available in the [R package raw](#) however, the data are given in the so-called long-type format (data frames) not as typical run-off triangles

- (i) **130 run-off triangles that were ODP compliant** (with non-negative increments only, but profiles being entirely zero not allowed)
- (ii) **299 not ODP compliant triangles** (for instance, triangles with negative or too many increments or some zero profiles)
- (iii) **89 rather atypical triangles**, that are not uncommon in the actuarial practice but the actual reserve is often determined by an expert judgement

Out-of-sample bootstrap performance measures

Reserve% gives an absolute relative difference of the predicted reserve and the true reserve defined for each triangle as $100 \times \left| \frac{\text{predicted reserve}}{\text{true reserve}} - 1 \right|$ and averaged over all triangles in the given scenario (smaller values are better);

BootCoV% expresses a coefficient of variation for the bootstrapped reserve distribution relative to the bootstrap mean $100 \times \frac{\text{Std.Dev}(\text{bootstrapped reserves})}{\text{Avg}(\text{bootstrapped reserves})}$ averaged, again, over all triangles in the given scenario (smaller values are better);

BootVaR.995 denotes the 99.5% quantile of the bootstrap distribution relative to the bootstrapped mean $\frac{\text{Quantile}_{0.995}(\text{bootstrapped reserves})}{\text{Avg}(\text{bootstrapped reserves})}$ and averaged over all triangles in the given scenario (smaller values are better);

BootQnt.950 provides a percentage proportion of the triangles in the given scenario for which the true reserve is dominated by the 95% quantile of the bootstrapped distribution (values closest to 95% are preferred).

Claims reserves evaluation – Type I

Method	Reserve%	BootCoV%	BootVaR _{.995}	BootQnt _{.950}
<i>Average</i>	58.79 (186.00)	79.46 (144.67)	3.67 (3.57)	100.00%
<i>Weighted</i>	47.13 (130.91)	53.60 (61.46)	2.63 (1.81)	98.46%
<i>ODP Model</i>	47.10 (130.89)	16.98 (10.16)	1.54 (0.39)	86.92%
<i>PARALLAX</i>	57.85 (125.45)	22.34 (16.13)	1.59 (0.46)	96.92%
<i>REACT</i>	43.19 (78.28)	24.08 (18.03)	1.64 (0.51)	97.69%
<i>MACRAME</i>	45.32 (76.43)	23.93 (12.65)	1.73 (0.42)	95.38%

Table: Overall empirical performance of six claims reserving techniques when applied to the group (i), 130 **ODP compliant** run-off triangles from Meyers and Shi (2011). The corresponding standard deviations are given in parentheses; two best results are indicated by bold typeface.

Claims reserves evaluation – Type II

Method	Reserve%		BootCoV%		BootVaR _{.995}		BootQnt _{.950}
<i>Average</i>	215.95	(1128.77)	4045.61	(4.0e+04)	43.14	(461.41)	99.67%
<i>Weighted</i>	541.33	(6135.24)	−3e+03	(2.3e+04)	−7.43	(132.37)	97.99%
<i>Chainladder</i>	541.33	(6135.24)	29.78	(212.59)	1.97	(7.58)	83.28%
<i>PARALLAX</i>	68.83	(132.40)	9.53	(628.55)	1.70	(11.04)	92.98%
<i>REACT</i>	97.85	(334.97)	66.60	(182.67)	2.92	(4.99)	94.31%
<i>MACRAME</i>	68.38	(93.76)	51.26	(36.96)	2.75	(1.59)	91.97%

Table: Empirical performance of six claims reserving techniques applied to the group (ii), 299 “rather typical” but ODP non-compliant run-off triangles from Meyers and Shi (2011). The corresponding standard deviations are given in parentheses; two best results are indicated by bold typeface.

Claims reserves evaluation – Type III

Method	Reserve%		BootCoV%		BootVaR _{.995}		BootQnt _{.950}
<i>Average</i>	255.88	(654.91)	−2e+03	(2.1e+04)	−47.74	(505.00)	91.01%
<i>Weighted</i>	181.32	(526.35)	4.6e+04	(4.3e+05)	167.99	(1492.61)	91.01%
<i>Chainladder</i>	181.32	(526.35)	177.17	(472.73)	6.02	(11.65)	79.78%
<i>PARALLAX</i>	142.08	(567.07)	69.77	(75.02)	3.09	(4.63)	77.53%
<i>REACT</i>	111.03	(256.82)	240.60	(1294.93)	7.94	(35.34)	76.40%
<i>MACRAME</i>	111.02	(141.21)	256.41	(1175.31)	10.25	(52.48)	69.66%

Table: Empirical performance of six claims reserving techniques applied to the group (iii), 89 “atypical” run-off triangles from Meyers and Shi (2011). The corresponding standard deviations are given in parentheses; two best results are indicated by bold typeface.

Bootstrap computational efficiency

Prediction method	Point prediction	Permutation bootstrap	Residual bootstrap
PARALLAX	0.03s. (0.006)	15.01s. (0.341)	X
REACT	0.01s. (0.001)	9.04s. (0.116)	X
MACRAME	0.04s. (0.012)	48.75s. (11.51)	X
Chainladder	0.03s. (0.008)	31.70s. (0.276)	8.21s. (0.093)
Mack model	0.04s. (0.002)	42.10s. (0.650)	57.10s. (0.520)
ODP model	3.86s. (0.043)	63.37m. (0.148)	∞ m. (NA*)
Tweedie formula	0.24s. (0.026)	4.19m. (0.307)	3.69m. (0.252)

Three large (60×60) run-off triangles from the Guarantee Fund of the Czech Insurers' Bureau are used to average computational times with the number of bootstrap resamples being always set to $B = 1000$ using the MacOS Architecture with 12 Apple M2 Pro chips (8 for performance and for 4 efficiency) with 32GB memory

Implementation improvements wrt. ChainLadder

- ❑ **The core actuarial package ChainLadder**
 - S3 class `triangle` (resp. `as.triangle()` function) which provided only very limited information about the underlying data
 - limited usage of various S3 methods available (by default) in R

Implementation improvements wrt. ChainLadder

❑ The core actuarial package ChainLadder

- S3 class `triangle` (resp. `as.triangle()` function) which provided only very limited information about the underlying data
- limited usage of various S3 methods available (by default) in R

❑ The nonparametric methods implemented in ProfileLadder

- extended S3 class `profileLadder` (resp. `as.profileLadder()` function) with additional information about the underlying run-off triangle
- straightforward usage of S3 methods `plot()`, `summary()`, `print()`, or `predict()`
- additional summary/graphical tools for run-off triangle exploratory
- wide-scale compatibility with the parametric methods from ChainLadder

Compatibility with the ChainLadder package

S3 class triangle from ChainLadder vs. **S3 class ProfileLadder** from ProfileLadder

```
R> options(profileLadder.fancy = TRUE)
R> as.profileLadder(observed(CameronMutual))
R> as.profileLadder(CameronMutual)
```

Run-off triangle										Run-off triangle (complete/fully observed)									
5244	9228	10823	11352	11791	12082	12120	12199	12215	12215	5244	9228	10823	11352	11791	12082	12120	12199	12215	12215
5984	9939	11725	12346	12746	12909	13034	13109	13113	.	5984	9939	11725	12346	12746	12909	13034	13109	13113	13115
7452	12421	14171	14752	15066	15354	15637	15720	.	.	7452	12421	14171	14752	15066	15354	15637	15720	15744	15786
7115	11117	12488	13274	13662	13859	13872	.	.	.	7115	11117	12488	13274	13662	13859	13872	13935	13973	13972
5753	8969	9917	10697	11135	11282	.	.	.	.	5753	8969	9917	10697	11135	11282	11255	11331	11332	11354
3937	6524	7989	8543	8757	.	.	.	.	.	3937	6524	7989	8543	8757	8901	9013	9012	9046	9164
5127	8212	8976	9325	.	.	.	.	.	.	5127	8212	8976	9325	9718	9795	9833	9885	9816	9815
5046	8006	8984	.	.	.	.	.	.	.	5046	8006	8984	9633	10102	10166	10261	10252	10252	10252
5129	8202	.	.	.	.	.	.	.	.	5129	8202	9185	9681	9951	10033	10133	10182	10182	10183
3689	.	.	.	.	.	.	.	.	.	3689	6043	6789	7089	7164	7197	7253	7267	7266	7266

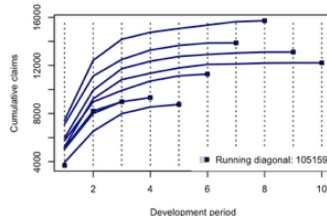
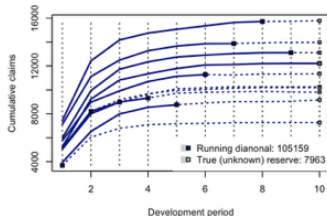
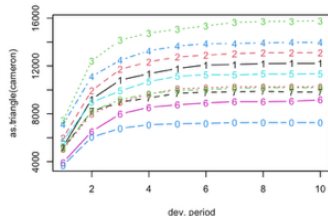
Compatibility with the ChainLadder package

S3 class triangle from ChainLadder vs. **S3 class ProfileLadder** from ProfileLadder

```
par(mfrow = c(1,3))
### S3 class 'triangle' from the ChainLadder pkg
plot(as.triangle(cameron))

### S3 class 'profileLadder' from the ProfileLadder pkg
plot(as.profileLadder(cameron))

### S3 class 'profileLadder' (observed run-off triangle only)
plot(as.profileLadder(observed(cameron)))
```



Implementation consistency with the ChainLadder package

```
R> summary(glmReserve(triangle))
```

	Latest	Dev.To.Date	Ultimate	IBNR	S.E	CV
2	13113	1.0000000	13113	0	0.01035287	Inf
3	15720	0.9992372	15732	12	29.38975320	2.4491461
4	13872	0.9934116	13964	92	73.46098788	0.7984890
5	11282	0.9850694	11453	171	96.26787355	0.5629700
6	8757	0.9688019	9039	282	120.58781735	0.4276164
7	9325	0.9397360	9923	598	177.57019068	0.2969401
8	8984	0.8905630	10088	1104	246.00693847	0.2228324
9	8202	0.7789914	10529	2327	379.54811405	0.1631062
10	3689	0.4788422	7704	4015	622.22022041	0.1549739
total	92944	0.9152986	101545	8601	880.66723938	0.1023913

```
R> summary(parallelReserve(triangle))
```

PARALLAX reserve prediction (by origins)

	First	Latest	Dev.To.Date	Ultimate	IBNR
2	5984	13113	1.0000000	13113	0
3	7452	15720	0.9997456	15724	4
4	7115	13872	0.9943373	13951	79
5	5753	11282	0.9883487	11415	133
6	3937	8757	0.9690163	9037	280
7	5127	9325	0.9496894	9819	494
8	5046	8984	0.9142159	9827	843
9	5129	8202	0.8361709	9809	1607
10	3689	3689	0.4197292	8789	5100
total	49232	92944	0.9158488	101484	8540

Overall reserve summary

Est.Reserve	Est.Ultimate	Paid Amount	True Reserve	Reserve%
8540.00	113699.00	105159.00	7963.00	7.25

- ❑ The default output is fully consistent with the standard outputs provided by the classical parametric methods implemented in the [ChainLadder package](#)
- ❑ In addition, using the extended run-off triangle class [profileLadder](#) the outputs contain more detailed information relevant for actuaries

	Latest	Dev.To.Date	Ultimate	IBNR	S.E	CV
2	13113	1.0000000	13113	0	0.00000	NaN
3	15720	0.9992372	15732	12	35.18573	2.9321441
4	13872	0.9934116	13964	92	72.68189	0.7900206
5	11282	0.9850694	11453	171	96.58273	0.5648113
6	8757	0.9688019	9039	282	118.48121	0.4201461
7	9325	0.9397360	9923	598	171.92040	0.2874923
8	8984	0.8905630	10088	1104	248.31790	0.2249256
9	8202	0.7788914	10529	2327	354.75196	0.1524504
10	3689	0.4788422	7704	4015	660.11910	0.1644132
total	92944	0.9152986	101545	8601	896.34280	0.1042138

PARALLAX based reserve prediction (with B = 500 permutations)

Overall reserve distribution

The PARALLAX predicted reserve represents the 29.34% quantile of the permuted bootstrap reserve distribution. Bootstrap simulated reserves beyond 2sigma rule: 11 (out of 500)

- ▲

Functional-based claims reserving – overview and next steps

Three competitive unsupervised loss reserving techniques based on **non-parametric and distribution free approaches** applied to **aggregated data** structured in **run-off triangles**...

- (i) they are **simple, straightforward, and easily applicable**;
- (ii) they require **neither distributional nor parametric assumptions** and apply to all kinds of run-off triangles, including those with negative increments or zero development periods;
- (iii) various stochastic assumptions can be postulated in order to derive **desirable statistical properties** serving as the methods' **formal justification**;
- (iv) straightforward to obtain the **overall reserve distribution** via resampling—the proposed **permutation bootstrap** add-on in particular;
- (v) the proposed methods are **intuitive** and **simple, robust** against outliers, computationally very **effective**, and promising for the actuarial practice;

Functional-based claims reserving – overview and next steps

Three competitive unsupervised loss reserving techniques based on **non-parametric and distribution free approaches** applied to **aggregated data** structured in **run-off triangles**...

- (i) they are **simple, straightforward, and easily applicable**;
- (ii) they require **neither distributional nor parametric assumptions** and apply to all kinds of run-off triangles, including those with negative increments or zero development periods;
- (iii) various stochastic assumptions can be postulated in order to derive **desirable statistical properties** serving as the methods' **formal justification**;
- (iv) straightforward to obtain the **overall reserve distribution** via resampling—the proposed **permutation bootstrap** add-on in particular;
- (v) the proposed methods are **intuitive** and **simple, robust** against outliers, computationally very **effective**, and promising for the actuarial practice;

↪ **let's create an R package and deploy it on CRAN (with a Q1 paper as a promotion)**

Part II

The R package development and deployment

Package skeleton, package documentation, functionality checks, package deployment

Why CRAN?

- ❑ **CRAN** can turn your **R codes** into a standard component of the **R ecosystem**
(visibility ↔ accessibility ↔ quality control ↔ compatibility ↔ credibility ↔ R community)

Why CRAN?

- ❑ **CRAN** can turn your **R codes** into a standard component of the **R ecosystem**
(visibility ↔ accessibility ↔ quality control ↔ compatibility ↔ credibility ↔ R community)
 - being available on CRAN signals that the R package satisfies high technical standards and meets all software organizational requirements
 - CRAN performs regular and extensive automated checks of the package structure, documentation, examples, dependencies, and code
 - CRAN provides binaries and all previous versions are archived, which supports transparency, reproducibility, and maintenance
 - CRAN automatically checks the packages on different operating systems and R versions, helping developers detect portability problems

Why CRAN?

- ❑ **CRAN** can turn your **R codes** into a standard component of the **R ecosystem**
(visibility ↔ accessibility ↔ quality control ↔ compatibility ↔ credibility ↔ R community)
 - being available on CRAN signals that the R package satisfies high technical standards and meets all software organizational requirements
 - CRAN performs regular and extensive automated checks of the package structure, documentation, examples, dependencies, and code
 - CRAN provides binaries and all previous versions are archived, which supports transparency, reproducibility, and maintenance
 - CRAN automatically checks the packages on different operating systems and R versions, helping developers detect portability problems
- ❑ Thus, it may be **challenging at the beginning** but it can make the R package development efficient and **much easier at the end** ...

Why is CRAN generally preferred?

CRAN is usually the preferred final package destination although it probably can not be universally described as “the best option available”. However...

Why is CRAN generally preferred?

CRAN is usually the preferred final package destination although it probably can not be universally described as “the best option available”. However...

❑ From the user perspective:

- allows for trivial package installation, no additional installer, no GitHub, ...
- high level of trust (many technical and organizational requirements)
- ongoing compatibility checks, developments and dependencies
- **But CRAN does not provide scientific peer review check!**

Why is CRAN generally preferred?

CRAN is usually the preferred final package destination although it probably can not be universally described as “the best option available”. However...

❑ From the user perspective:

- allows for trivial package installation, no additional installer, no GitHub, ...
- high level of trust (many technical and organizational requirements)
- ongoing compatibility checks, developments and dependencies
- **But CRAN does not provide scientific peer review check!**

❑ From the developer perspective:

- continuous automated quality control checks (Linux, Macintosh, Windows)
- dependency integration and simple user adoption process (multiple R versions)
- good visibility, high credibility, automatic binaries, transparency
- **But CRAN could be the most challenging option to handle!**

Why is CRAN generally preferred?

CRAN is usually the preferred final package destination although it probably can not be universally described as “the best option available”. However...

❑ From the user perspective:

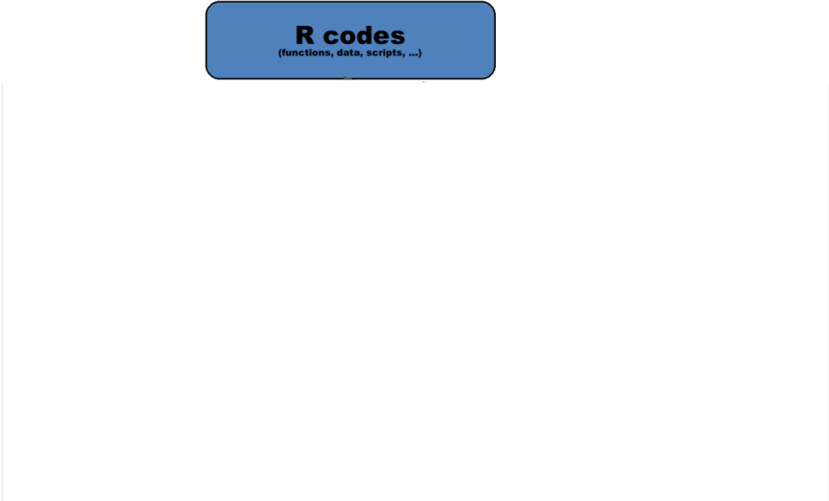
- allows for trivial package installation, no additional installer, no GitHub, ...
- high level of trust (many technical and organizational requirements)
- ongoing compatibility checks, developments and dependencies
- **But CRAN does not provide scientific peer review check!**

❑ From the developer perspective:

- continuous automated quality control checks (Linux, Macintosh, Windows)
- dependency integration and simple user adoption process (multiple R versions)
- good visibility, high credibility, automatic binaries, transparency
- **But CRAN could be the most challenging option to handle!**

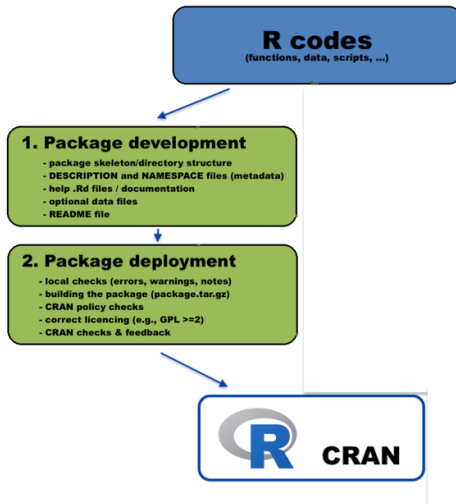
CRAN alternatives: GitHub, Gitlab, Bioconductor, R-universe, private repositories (e.g., MSD)

In principle, two possible paths to take ...

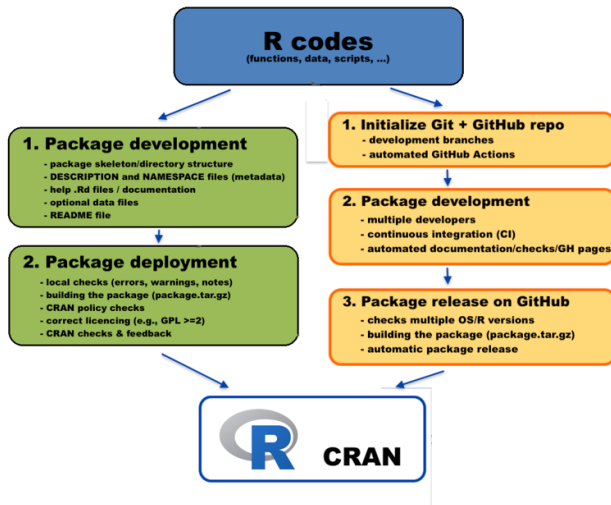


R codes
(functions, data, scripts, ...)

In principle, two possible paths to take ...



In principle, two possible paths to take ...



Mille viae ducunt homines per saecula Romam (Alain de Lille, 1175)



Mille viae ducunt homines per saecula Romam (Alain de Lille, 1175)



Similarly, there are many ways that lead to a package development and its final deployment to CRAN ...

- ☐ **Naive**: Reediting some existing package (copy + paste)
- ☐ **Ad-hoc**: Manual scripting and packaging
- ☐ **Tool-assisted**, relatively efficient (local) package development
- ☐ **Quality-focused**: testing, documentation, and checks
- ☐ **Automated workflows**, versioning with reproducibility

Mille viae ducunt homines per saecula Romam (Alain de Lille, 1175)



Similarly, there are many ways that lead to a package development and its final deployment to CRAN ...

- ☐ **Naive:** Reediting some existing package (copy + paste)
- ☐ **Ad-hoc:** Manual scripting and packaging
- ☐ **Tool-assisted**, relatively efficient (local) package development
- ☐ **Quality-focused:** testing, documentation, and checks
- ☐ **Automated workflows**, versioning with reproducibility
- ☐ **Bonus:** Continuous Integration (CI) via GitHub Actions (GA)

Mille viae ducunt homines per saecula Romam (Alain de Lille, 1175)



Similarly, there are many ways that lead to a package development and its final deployment to CRAN ...

- ☐ **Naive:** Reediting some existing package (copy + paste)
- ☐ **Ad-hoc:** Manual scripting and packaging
- ☐ **Tool-assisted, relatively efficient (local) package development**
- ☐ **Quality-focused:** testing, documentation, and checks
- ☐ **Automated workflows**, versioning with reproducibility
- ☐ **Bonus:** Continuous Integration (CI) via GitHub Actions (GA)

Tool-assisted (local) package development

Practical software tools – **R packages** – designed for an individual user-based package development and management (so-called R package-development toolkit)

- ❑ `utils`, `tools` – base-R utilities for installing, updating and managing packages
- ❑ `devtools` – toolkit for the main package-development workflow: build, check, test, document, install
- ❑ `roxygen2` – generating of the package documentation (`.Rd` files) and `NAMESPACE`
- ❑ `testthat` – package with a framework for writing and automatically running unit tests
- ❑ `pkgdown` – creates an R package website from its documentation, vignettes and README
- ❑ `usethis` – automatization of some steps: creates files, tests, licenses, GitHub configuration, CI, etc.
- ❑ `remotes` – installs packages from GitHub, GitLab, URLs, local sources, etc., rather than only CRAN

Tool-assisted (local) package development

Practical software tools – **R packages** – designed for an individual user-based package development and management (so-called R package-development toolkit)

- ❑ `utils`, `tools` – base-R utilities for installing, updating and managing packages
- ❑ `devtools` – toolkit for the main package-development workflow: build, check, test, document, install
- ❑ `roxygen2` – generating of the package documentation (`.Rd` files) and `NAMESPACE`
- ❑ `testthat` – package with a framework for writing and automatically running unit tests
- ❑ `pkgdown` – creates an R package website from its documentation, vignettes and README
- ❑ `usethis` – automatization of some steps: creates files, tests, licenses, GitHub configuration, CI, etc.
- ❑ `remotes` – installs packages from GitHub, GitLab, URLs, local sources, etc., rather than only CRAN

↪ although, not all of these packages are strictly needed for an own package development!

R package-development toolkit – some background

- ❑ `utils`, `tools` – part of **base R**; the traditional infrastructure for package installation, building, checking, and administration (although, not that much user-friendly)
- ❑ `testthat` – introduced already in **2009**; brought systematic **unit testing** and regression testing into the R package-development workflow
- ❑ `devtools` – introduced in **2011** as a convenient interface to the complete package development cycle: `load`, `document`, `test`, `check`, `build`, `install`
- ❑ Over time, the original devtools functionality was **separated into specialized packages** (e.g., `roxygen2` for documentation; `remotes` for installation; `usethis` for the overall project setup; ...)

R package-development toolkit – some background

- ❑ `utils`, `tools` – part of **base R**; the traditional infrastructure for package installation, building, checking, and administration (although, not that much user-friendly)
- ❑ `testthat` – introduced already in **2009**; brought systematic **unit testing** and regression testing into the R package-development workflow
- ❑ `devtools` – introduced in **2011** as a convenient interface to the complete package development cycle: load, document, test, check, build, install
- ❑ Over time, the original devtools functionality was **separated into specialized packages** (e.g., `roxygen2` for documentation; `remotes` for installation; `usethis` for the overall project setup; ...)
- ❑ Thus, modern `devtools` acts largely as a **high-level interface** to an ecosystem of smaller, specialized development tools.

Base R → devtools → modular package-development ecosystem

Typical package development workflow

❑ Two main R packages: devtools and roxygen2

```
R> utils::package.skeleton( )          ## package core
R> devtools::create()
R> usethis::create_package()

R> roxygen2::roxygenize()              ## documentation/help files
R> devtools::check()                   ## functionality check
R> devtools::build()                   ## package builder
R> devtools::check(as_cran = TRUE)     ## CRAN feasibility check
```

Typical package development workflow

❑ Two main R packages: devtools and roxygen2

```
R> utils::package.skeleton( )           ## package core
R> devtools::create()
R> usethis::create_package()

R> roxygen2::roxygenize()                ## documentation/help files
R> devtools::check()                     ## functionality check
R> devtools::build()                     ## package builder
R> devtools::check(as_cran = TRUE)       ## CRAN feasibility check
```

❑ Package ready for CRAN...

→ submission on <https://cran.r-project.org/submit.html>

Typical package development workflow

❑ Two main R packages: devtools and roxygen2

```
R> utils::package.skeleton( )           ## package core
R> devtools::create()
R> usethis::create_package()

R> roxygen2::roxygenize()               ## documentation/help files
R> devtools::check()                   ## functionality check
R> devtools::build()                   ## package builder
R> devtools::check(as_cran = TRUE)     ## CRAN feasibility check
```

❑ Package ready for CRAN...

→ submission on <https://cran.r-project.org/submit.html>

With another level of tests to undertake... automatic/manual CRAN checks after the package submission

System terminal vs. the R console

(cheatsheet)

own R functions and documentation

1. R functions in R directory
2. R functions documentation
3. data into in data directory

minimal package structure

4. DESCRIPTION file
5. R -e 'roxygen2::roxygenise()' (package directory)

check and build

6. add README.md file
7. R CMD build \$(PKGSRCE) (package directory)
8. R CMD check \$(PKGSRCE) (tar.gz file)
9. R CMD check --as-cran

optionally -- upload to CRAN

10. Use <https://cran.r-project.org/submit.html>

CRAN automatic/manual checks

11. CRAN checks and manual inspection by CRAN team

package on CRAN

12. source tar.gz and binaries

build GitHub pages (from the package root directory in R)

13. pkgdown::build_favicons()
14. pkgdown::build_site() ## repeat after logo change

create unit tests and run tests (package root directory in R)

15. add tests into tests/testthat
16. devtools::test()

push to GitHub (from the package directory in console)

17. git status ## kontrola ci doslo k zmene
18. git add , ## zoznam zmien sa prida do skryteho git suboru
19. git commit -m "message"
20. git push

automatization on GitHub via yaml configuration

21. R-CMD-check.yaml, _pkgdown.yaml test-coverage.yaml, GH workflow

logo & vignette

22. package vignette
23. logo / hexSticker package for R

24. rjtools checks

25. ready for submission in R

Basic development tool: devtools

- ❑ developer-friendly interface for basic tools like R CMD build, R CMD check, and others... (but incorporates also other tools from other development packages)
 - ❑ started as a minimalist but efficient package to facilitate local package development ... however, it accumulated more and more functionality over the years and became too complex
 - ❑ thus, the package was split into a few smaller but more tightly focused packages ... (roxygen2, testthat, pkgbuild, usethis, remote, ...)
-



A central toolkit for R package development—helps to make the whole development process easier by providing specific functions that simplify common tasks while following important standards of a good development workflow.



Package skeleton

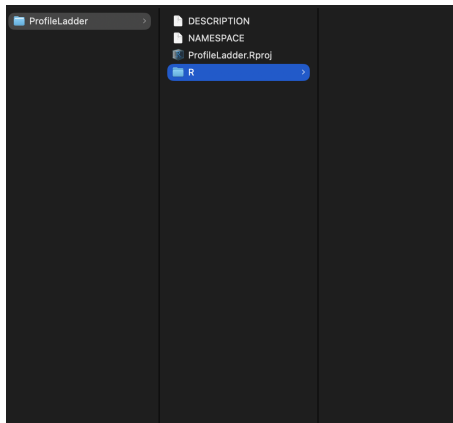
(devtools)

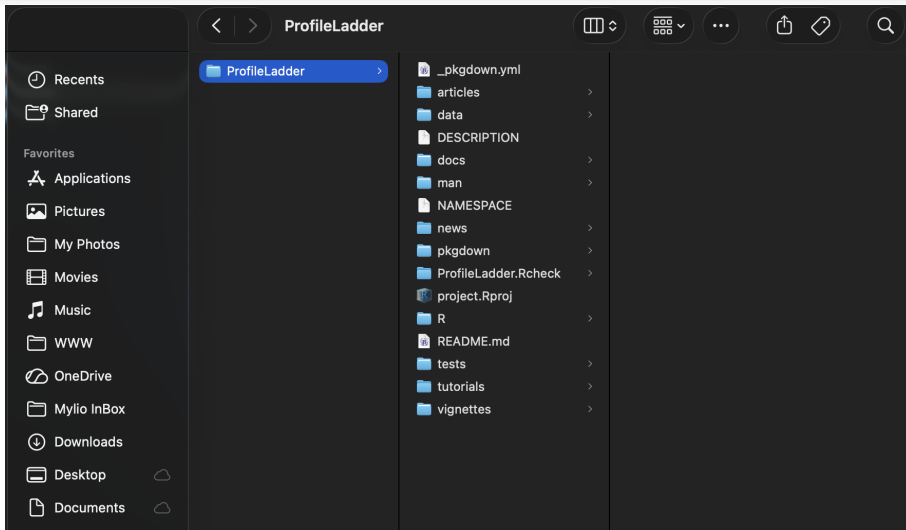
```
> setwd("/Users/maciak/Software/")
>
> devtools::create("ProfileLadder")
✓ Creating ProfileLadder/.
✓ Setting active project to "/Users/maciak/Software/ProfileLadder".
✓ Creating R/.
✓ Writing DESCRIPTION.
Package: ProfileLadder
Title: What the Package Does (One Line, Title Case)
Version: 0.0.0.9000
Authors@R (parsed):
  * First Last <first.last@example.com> [aut, cre]
Description: What the package does (one paragraph).
License: `use_mit_license()`, `use_gpl3_license()` or friends to
  pick a license
Encoding: UTF-8
Roxygen: list(markdown = TRUE)
RoxygenNote: 7.3.3
✓ Writing NAMESPACE.
✓ Writing ProfileLadder.Rproj.
✓ Adding "^ProfileLadder\\.Rproj$" to .Rbuildignore.
✓ Adding ".Rproj.user" to .gitignore.
✓ Adding "\\\\.Rproj\\.user$" to .Rbuildignore.
✓ Setting active project to "<no active project>".
> |
```

Package skeleton

(devtools)

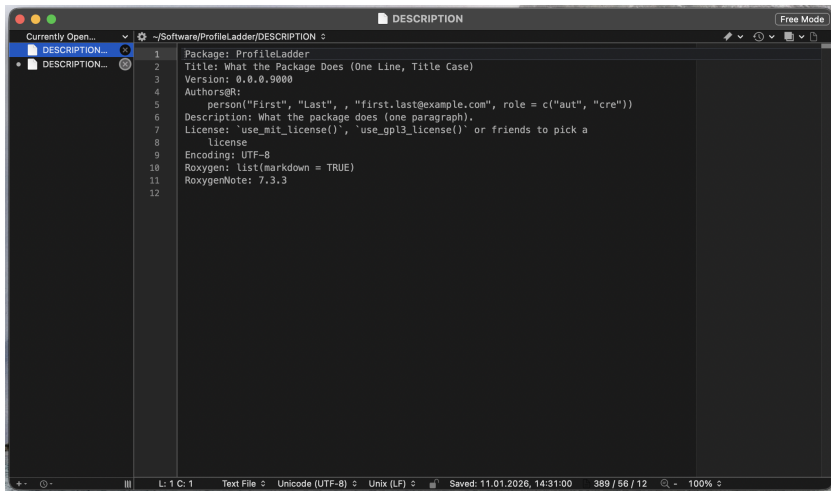
```
> setwd("/Users/maciak/Software/")
>
> devtools::create("ProfileLadder")
✓ Creating ProfileLadder/.
✓ Setting active project to "/Users/maciak/Software/ProfileLadder".
✓ Creating R/.
✓ Writing DESCRIPTION.
Package: ProfileLadder
Title: What the Package Does (One Line, Title Case)
Version: 0.0.0.9000
Authors@R (parsed):
  * First Last <first.last@example.com> [aut, cre]
Description: What the package does (one paragraph).
License: `use_mit_license()`, `use_gpl3_license()` or friends to
  pick a license
Encoding: UTF-8
Roxygen: list(markdown = TRUE)
RoxygenNote: 7.3.3
✓ Writing NAMESPACE.
✓ Writing ProfileLadder.Rproj.
✓ Adding "^ProfileLadder\\.Rproj$" to .Rbuildignore.
✓ Adding ".Rproj.user" to .gitignore.
✓ Adding "\\ProfileLadder\\.Rproj\\.user$" to .Rbuildignore.
✓ Setting active project to "<no active project>".
> |
```





DESCRIPTION file

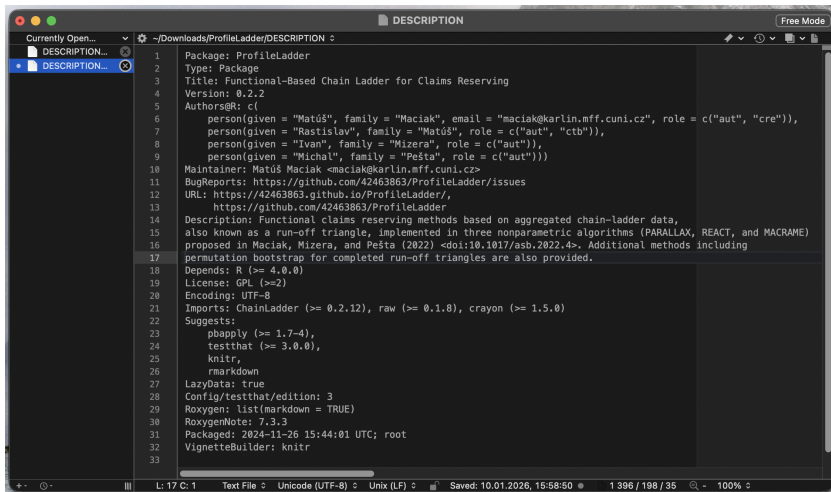
Contains the package metadata: package name, version, title, authors, description, license, dependencies, required R version, URLs, etc. It tells R and CRAN what the package is and what it depends on.



```
1 Package: ProfileLadder
2 Title: What the Package Does (One Line, Title Case)
3 Version: 0.0.0.9000
4 Authors@R:
5   person("First", "Last", , "first.last@example.com", role = c("aut", "cre"))
6 Description: What the package does (one paragraph).
7 License: `use_mit_license()`, `use_gpl3_license()` or friends to pick a
8   license
9 Encoding: UTF-8
10 Roxygen: list(markdown = TRUE)
11 RoxygenNote: 7.3.3
12
```

DESCRIPTION file

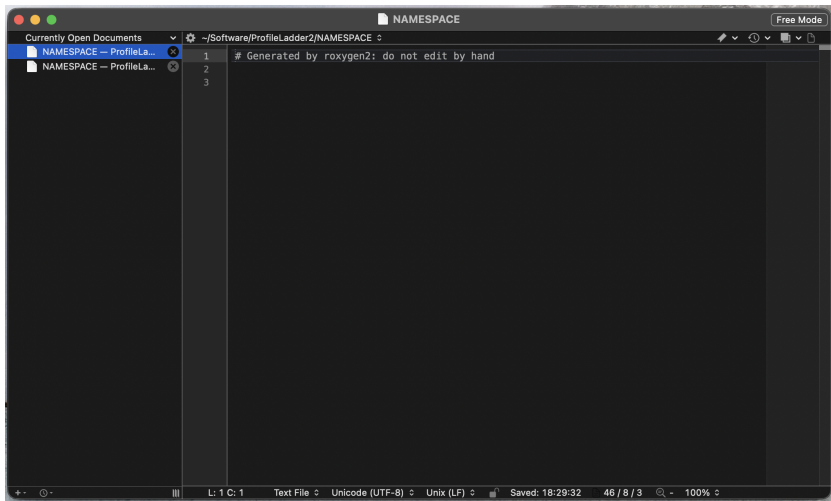
Contains the package metadata: package name, version, title, authors, description, license, dependencies, required R version, URLs, etc. It tells R and CRAN what the package is and what it depends on.



```
1 Package: ProfileLadder
2 Type: Package
3 Title: Functional-Based Chain Ladder for Claims Reserving
4 Version: 0.2.2
5 Authors@R: c(
6   person(given = "Matúš", family = "Maciak", email = "maciak@karlin.mff.cuni.cz", role = c("aut", "cre")),
7   person(given = "Rastislav", family = "Matúš", role = c("aut", "ctb")),
8   person(given = "Ivan", family = "Mizera", role = c("aut")),
9   person(given = "Michal", family = "Pešta", role = c("aut")))
10 Maintainer: Matúš Maciak <maciak@karlin.mff.cuni.cz>
11 BugReports: https://github.com/42463863/ProfileLadder/issues
12 URL: https://42463863.github.io/ProfileLadder/,
13     https://github.com/42463863/ProfileLadder
14 Description: Functional claims reserving methods based on aggregated chain-ladder data,
15 also known as a run-off triangle, implemented in three nonparametric algorithms (PARALLAX, REACT, and MACRAME)
16 proposed in Maciak, Mizera, and Pešta (2022) <doi:10.1017/asb.2022.4>. Additional methods including
17 permutation bootstrap for completed run-off triangles are also provided.
18 Depends: R (>= 4.0.0)
19 License: GPL (>=2)
20 Encoding: UTF-8
21 Imports: ChainLadder (>= 0.2.12), raw (>= 0.1.8), crayon (>= 1.5.0)
22 Suggests:
23   pbapply (>= 1.7-4),
24   testthat (>= 3.0.0),
25   knitr,
26   rmarkdown
27 LazyData: true
28 Config/testthat/edition: 3
29 Roxygen: list(markdown = TRUE)
30 RoxygenNote: 7.3.3
31 Packaged: 2024-11-26 15:44:01 UTC; root
32 VignetteBuilder: knitr
33
```

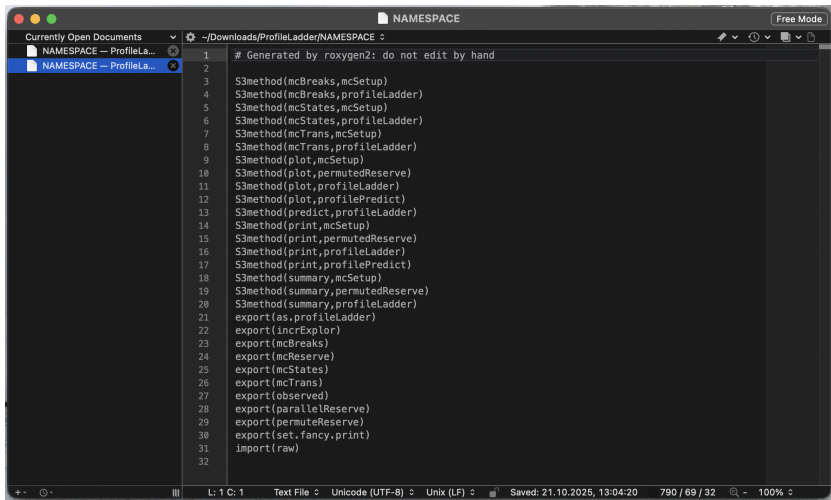
NAMESPACE file

Controls what the package exports to users and what it imports from other packages. In other words, it manages the package's public interface and avoids name conflicts.



NAMESPACE file

Controls what the package exports to users and what it imports from other packages. In other words, it manages the package's public interface and avoids name conflicts.



The screenshot shows an RStudio editor window titled 'NAMESPACE'. The file path is '~/Downloads/ProfileLadder/NAMESPACE'. The editor displays the following R code:

```
1 # Generated by roxygen2: do not edit by hand
2
3 S3method(mcbreaks,mcSetup)
4 S3method(mcbreaks,profileLadder)
5 S3method(mcStates,mcSetup)
6 S3method(mcStates,profileLadder)
7 S3method(mcTrans,mcSetup)
8 S3method(mcTrans,profileLadder)
9 S3method(plot,mcSetup)
10 S3method(plot,permutedReserve)
11 S3method(plot,profileLadder)
12 S3method(plot,profilePredict)
13 S3method(predict,profileLadder)
14 S3method(print,mcSetup)
15 S3method(print,permutedReserve)
16 S3method(print,profileLadder)
17 S3method(print,profilePredict)
18 S3method(summary,mcSetup)
19 S3method(summary,permutedReserve)
20 S3method(summary,profileLadder)
21 export(as.profileLadder)
22 export(incrExplor)
23 export(mcbreaks)
24 export(mcReserve)
25 export(mcStates)
26 export(mcTrans)
27 export(observed)
28 export(parallelReserve)
29 export(permuteReserve)
30 export(set.fancy.print)
31 import(raw)
32
```

The status bar at the bottom indicates the file is a Text File in Unicode (UTF-8) encoding, using Unix (LF) line endings. It also shows the file was saved on 21.10.2025 at 13:04:20, with 790 / 69 / 32 lines and a zoom level of 100%.

Core functions of the R package ProfileLadder

❑ **Point predictions** – three functional-based reserve prediction algorithms

❑ **PARALLAX** – **Parallel** approximation by missing fragments

```
R> parallelReserve(data)      ## parallelReserve.R | 230 lines / 80 documentation
```

❑ **REACT** – Approximation by the most **recent** **accident** segment

```
R> parallelReserve(data, method = "react")
```

❑ **MACRAME** – **Markov** chain **fragment** approximation

```
R> mcReserve(data)           ## mcReserve.R | 405 lines / 110 documentation
```

Core functions of the R package ProfileLadder

❑ **Point predictions** – three functional-based reserve prediction algorithms

❑ **PARALLAX** – **Parallel** approximation by missing fragments

```
R> parallelReserve(data)      ## parallelReserve.R | 230 lines / 80 documentation
```

❑ **REACT** – Approximation by the most **recent** **accident** segment

```
R> parallelReserve(data, method = "react")
```

❑ **MACRAME** – **Markov** chain **fragment** approximation

```
R> mcReserve(data)           ## mcReserve.R | 405 lines / 110 documentation
```

❑ **Distributional prediction** – performed in terms of permutation bootstrap resampling

❑ **Permutation bootstrap** – applicable to all nonparametric and also classical parametric methods

```
R> permuteReserve(...)       ## permuteReserve.R | 320 lines / 120 documentation
```

Core functions of the R package ProfileLadder

❑ **Point predictions** – three functional-based reserve prediction algorithms

❑ **PARALLAX** – **Parallel** approximation by missing fragments

```
R> parallelReserve(data)      ## parallelReserve.R | 230 lines / 80 documentation
```

❑ **REACT** – Approximation by the most **recent** **accident** segment

```
R> parallelReserve(data, method = "react")
```

❑ **MACRAME** – **Markov** **chain** **fragment** approximation

```
R> mcReserve(data)           ## mcReserve.R | 405 lines / 110 documentation
```

❑ **Distributional prediction** – performed in terms of permutation bootstrap resampling

❑ **Permutation bootstrap** – applicable to all nonparametric and also classical parametric methods

```
R> permuteReserve(...)       ## permuteReserve.R | 320 lines / 120 documentation
```

❑ **Additional utility/design functions** – for an easy integration with other packages

```
R> as.profileLadder(); observed(); set.fancy.print(); zzz(); ...
```

S3 methods in the R package ProfileLadder

S3 methods provide standard, natural R interface without requiring a complicated class system... thus, the package easily becomes familiar to R users, different model classes can use the same generic functions, package code becomes easier to organize, and new classes can be added without changing the generic itself.

❑ S3 methods in R

Create an object $\Rightarrow$ assign it a class $\Rightarrow$ use or define a generic function $\Rightarrow$ use easily

4x `print()` method, 4x `plot()` method, 3x `summary()` method, and one `predict()` method

❑ Accessor functions

AFunctions specifically designed to ensure easy-to-use functionality of the package

`mcBreaks()`, `mcStates()`, and `mcTrans()`

❑ Default setup functions

Additional functions to set up the R environment when launched

`set.fancy.print()` and `zzz.R`

And remember, it is the era of data... **big data** actually

- ❑ **Reproducibility data for testing purposes and comparisons**
 - real data from Mayers and Shi (2011) used in the theoretical paper (10×10)
(although, provided in a different structure in the raw package)

And remember, it is the era of data... **big data** actually

- ❑ **Reproducibility data for testing purposes and comparisons**
 - real data from Mayers and Shi (2011) used in the theoretical paper (10×10)
(although, provided in a different structure in the raw package)
- ❑ **New/novel data for robustness demonstration**
 - real run-off triangles not made available in any other public sources yet (17×17)
(provided by Generali-Česká Pojišťovna, a.s.)

And remember, it is the era of data... **big data** actually

- ❑ **Reproducibility data for testing purposes and comparisons**
 - real data from Mayers and Shi (2011) used in the theoretical paper (10×10)
(although, provided in a different structure in the raw package)
- ❑ **New/novel data for robustness demonstration**
 - real run-off triangles not made available in any other public sources yet (17×17)
(provided by Generali-Česká Pojišťovna, a.s.)
- ❑ **Large data sets for effectivity illustrations**
 - large triangles—compared to other commonly available actuarial data (60×60)
(provided by the Czech Insurers' Bureau)

And remember, it is the era of data... **big data** actually

- ❑ **Reproducibility data for testing purposes and comparisons**
 - real data from Mayers and Shi (2011) used in the theoretical paper (10×10)
(although, provided in a different structure in the raw package)
- ❑ **New/novel data for robustness demonstration**
 - real run-off triangles not made available in any other public sources yet (17×17)
(provided by Generali-Česká Pojišťovna, a.s.)
- ❑ **Large data sets for effectivity illustrations**
 - large triangles—compared to other commonly available actuarial data (60×60)
(provided by the Czech Insurers' Bureau)
- ❑ **Data beyond actuarial context...**
 - panel data collected in cohorts as an example beyond actuarial reserving
(Covid 19 data extracted from public data available from ÚZIS ČR)

And remember, it is the era of data... **big data** actually

- ❑ **Reproducibility data for testing purposes and comparisons**
 - real data from Mayers and Shi (2011) used in the theoretical paper (10×10)
(although, provided in a different structure in the raw package)
- ❑ **New/novel data for robustness demonstration**
 - real run-off triangles not made available in any other public sources yet (17×17)
(provided by Generali-Česká Pojišťovna, a.s.)
- ❑ **Large data sets for effectivity illustrations**
 - large triangles—compared to other commonly available actuarial data (60×60)
(provided by the Czech Insurers' Bureau)
- ❑ **Data beyond actuarial context...**
 - panel data collected in cohorts as an example beyond actuarial reserving
(Covid 19 data extracted from public data available from ÚZIS ČR)

All data – .RData files in data/ – needs analogous documentation (cca 30 – 80 lines per each dataset) as the one used for the R functions – required for each data file included in the package

Classical chain-ladder methods vs. functional-based reserving

- ❑ **Leading actuarial R package ChainLadder** `install.packages("ChainLadder")`
 - ❑ all common and typically used reserving methods – utilizing **parametric statistical techniques** and the aggregated **data structured in the run-off triangles** (e.g., `chainladder()`, `glmReserve()`, `tweedieReserve()`, ...)
- ❑ **Newly deployed R package ProfileLadder** `install.packages("ProfileLadder")`
 - ❑ non-restrictive, functional-based, non-parametric approaches **proposed and theoretically proved** in Maciak, Mizera, and Pešta (2022) (e.g., `profileLadder()`, `parallelReserve()`, `mcReserve()`, ...)

Package documentation

(roxygen2)

- ❑ domain-specific documentation syntax (system) for R that uses structured, special-purpose comments directly embedded in the underlying R code (.R files)
- ❑ provides automatic NAMESPACE generation (although not complete), consistent help (.Rd) files, data description, and package metadata (e.g., the DESCRIPTION file)



The main purpose of roxygen2 is to let users/developers to describe their functions directly in the R scripts and let roxygen2 to process the source code and comments to automatically generate .Rd files in man/, NAMESPACE, and modify the DESCRIPTION file.



Automatization with roxygen2

Instead of writing and editing the `.Rd` files and `NAMESPACE` directly (manually), the R code is directly commended with `#'` comments in the R source and the documentation is generated automatically with roxygen2

```
R> roxygen2::roxygenise()  ## directly runs the roxygen documentation generation
R> devtools::document()   ## devtools wrapper for the same documentation process
```

Automation with roxygen2

Instead of writing and editing the `.Rd` files and `NAMESPACE` directly (manually), the R code is directly commended with `#'` comments in the R source and the documentation is generated automatically with roxygen2

```
R> roxygen2::roxygenise()    ## directly runs the roxygen documentation generation
R> devtools::document()     ## devtools wrapper for the same documentation process
```

This ensures the following:

- ❑ creates the `man/` directory and generates the `.Rd` help files in there
- ❑ appropriately updates the `NAMESPACE` file (creates it if not made yet)
- ❑ documents functions, arguments, return values and examples
- ❑ documents datasets, classes, and S3 methods

Package validity checks

(R CMD check)

```

maciak@mbp ~ % Downloads -- ssh -- 88x67
maciak@mbp$79 ProfileLadder % cd ..
maciak@mbp$79 Download N R CMD check ProfileLadder.0.2.2.tar.gz
* using 3rd directory: /Users/maciek/Downloads/ProfileLadder.Rcheck
* using R version 4.5.1 (2025-06-13)
* using platform: x86_64-apple-darwin20
* R was compiled by
  Apple clang version 14.0.0 (clang-1400.0.29.202)
  GNU Fortran (GCC) 14.2.0
* running under: macOS Tahoe 26.0.1
* using session charset: UTF-8
* checking for file 'ProfileLadder/DESCRIPTION' ... OK
* checking extension type ... Package
* this is package 'ProfileLadder' version '0.2.2'
* package encoding: UTF-8
* checking package namespace information ... OK
* checking package dependencies ... OK
* checking if this is a source package ... OK
* checking if there is a namespace ... OK
* checking for executable files ... OK
* checking for hidden files and directories ... OK
* checking for portable file names ... OK
* checking for sufficient/correct file permissions ... OK
* checking whether package 'ProfileLadder' can be installed ... OK
* checking installed package size ... OK
* checking package directory ... OK
* checking 'build' directory ... OK
* checking DESCRIPTION meta-information ... OK
* checking top-level files ... OK
* checking for left-over files ... OK
* checking index information ... OK
* checking package subdirectories ... OK
* checking code files for non-ASCII characters ... OK
* checking R files for syntax errors ... OK
* checking whether the package can be loaded ... OK
* checking whether the package can be loaded with stated dependencies ... OK
* checking whether the package can be unloaded cleanly ... OK
* checking whether the namespace can be loaded with stated dependencies ... OK
* checking whether the namespace can be unloaded cleanly ... OK
* checking whether startup messages can be suppressed ... OK
* checking dependencies in R code ... OK
* checking S3 generic/method consistency ... OK
* checking replacement functions ... OK
* checking foreign function calls ... OK
* checking R code for possible problems ... OK
* checking Rd files ... OK
* checking Rd metadata ... OK
* checking Rd cross-references ... OK
* checking for missing documentation entries ... OK
* checking for code/documentation mismatches ... OK
* checking Rd usage sections ... OK
* checking Rd contents ... OK
* checking for unstated dependencies in examples ... OK
* checking contents of 'data' directory ... OK
* checking data for ASCII and non-ASCII characters ... OK
* checking lazyData ... OK
* checking data for ASCII and uncompressed saves ... OK
* checking installed files from 'inst/doc' ... OK
* checking files in 'vignettes' ... OK
* checking examples ... OK
* checking for unstated dependencies in vignettes ... OK
* checking package vignettes ... OK
* checking re-building of vignette outputs ... OK
* checking PDF version of manual ... OK
* DONE
Status: OK

```

Once the main package core is created the tarball
(.tar.gz) can be generated by R CMD build

Package validity checks

(R CMD check)

```

maciak@nbkaps79: Downloads — zsh — 88x67
maciak@nbkaps79: Downloads % cd ..
maciak@nbkaps79: Downloads % R CMD check ProfileLadder_0.2.2.tar.gz
* using 'log' directory: '/Users/maciek/Downloads/ProfileLadder.Rcheck/'
* using R version 4.5.1 (2025-06-13)
* using platform: x86_64-apple-darwin20
* R was compiled by
  Apple clang version 14.0.0 (clang-1400.0.29.202)
  GNU Fortran (GCC) 14.2.0
* running under: macOS Tahoe 26.0.1
* using session charset: UTF-8
* checking for file 'ProfileLadder/DESCRIPTION' ... OK
* checking extension type ... Package
* this is package 'ProfileLadder' version '0.2.2'
* package encoding: UTF-8
* checking package namespace information ... OK
* checking package dependencies ... OK
* checking if this is a source package ... OK
* checking if there is a namespace ... OK
* checking for executable files ... OK
* checking for hidden files and directories ... OK
* checking for portable file names ... OK
* checking for sufficient/correct file permissions ... OK
* checking whether package 'ProfileLadder' can be installed ... OK
* checking installed package size ... OK
* checking package directory ... OK
* checking 'build' directory ... OK
* checking DESCRIPTION meta-information ... OK
* checking top-level files ... OK
* checking for left-over files ... OK
* checking index information ... OK
* checking package subdirectories ... OK
* checking code files for non-ASCII characters ... OK
* checking R files for syntax errors ... OK
* checking whether the package can be loaded ... OK
* checking whether the package can be loaded with stated dependencies ... OK
* checking whether the package can be unloaded cleanly ... OK
* checking whether the namespace can be loaded with stated dependencies ... OK
* checking whether the namespace can be unloaded cleanly ... OK
* checking whether startup messages can be suppressed ... OK
* checking dependencies in R code ... OK
* checking S3 generic/method consistency ... OK
* checking replacement functions ... OK
* checking foreign function calls ... OK
* checking R code for possible problems ... OK
* checking Rd files ... OK
* checking Rd metadata ... OK
* checking Rd cross-references ... OK
* checking for missing documentation entries ... OK
* checking for code/documentation mismatches ... OK
* checking Rd usage sections ... OK
* checking Rd contents ... OK
* checking for unstated dependencies in examples ... OK
* checking contents of 'data' directory ... OK
* checking data for ASCII and unprocessed saves ... OK
* checking data for ASCII and unprocessed saves ... OK
* checking lazyData ... OK
* checking data for ASCII and unprocessed saves ... OK
* checking files in 'vignettes' ... OK
* checking examples ... OK
* checking for unstated dependencies in vignettes ... OK
* checking package vignettes ... OK
* checking re-building of vignette outputs ... OK
* checking PDF version of manual ... OK
* DONE
Status: OK

```

Once the main package core is created the tarball
(.tar.gz) can be generated by R CMD build

Game begins: package quality control by R CMD check

- ❑ package structure, files, directories, ...
- ❑ DESCRIPTION file (required fields like Package, Version, Title, Description, License, ...)
- ❑ NAMESPACE file (exported/imported functions, public/private functions, possible duplicates)
- ❑ documentation checks – valid syntax in .Rd files, all parameters should be documented, references
- ❑ no console spam, no marketing, no test fails on any OS
- ❑ installation checks, examples below 5s.

CRAN feasibility checks

(R CMD check -as-cran)

```

maciak@mbkpm79 ProfileLadder ~ -zsh -- 89x68
maciak@mbkpm79 ProfileLadder ~ R CMD check --as-cran ../ProfileLadder_0.2.2.tar.gz
* using log directory '/Users/maciek/Downloads/ProfileLadder/ProfileLadder.Rcheck'
* using R version 4.2.1 (2022-08-31)
* using platform: x86_64-apple-darwin20
* R was compiled by
  Apple clang version 14.0.0 (clang-1400.0.29.202)
  GNU Fortran (GCC) 14.2.0
* running under: macOS Tahoe 22.0.1
* using session charset: UTF-8
* using option '--as-cran'
* checking for file 'ProfileLadder/DESCRIPTION' ... OK
* checking extension type ... Package
* this is package 'ProfileLadder' version '0.2.2'
* package encoding: UTF-8
* checking CRAN incoming feasibility ... [Bs/39s] OK
* checking package namespace information ... OK
* checking package dependencies ... OK
* checking if this is a source package ... OK
* checking if there is a namespace file ... OK
* checking for executable files ... OK
* checking for hidden files and directories ... OK
* checking for portable file names ... OK
* checking for sufficient/correct file permissions ... OK
* checking whether package 'ProfileLadder' can be installed ... OK
* checking installed package size ... OK
* checking package directory ... OK
* checking for future file timestamps ... OK
* checking 'build' directory ... OK
* checking DESCRIPTION meta-information ... OK
* checking top-level files ... OK
* checking for left-over files ... OK
* checking index information ... OK
* checking package subdirectories ... OK
* checking code files for non-ASCII characters ... OK
* checking R files for syntax errors ... OK
* checking whether the package can be loaded ... OK
* checking whether the package can be loaded with stated dependencies ... OK
* checking whether the package can be unloaded cleanly ... OK
* checking whether the namespace can be loaded with stated dependencies ... OK
* checking whether the namespace can be unloaded cleanly ... OK
* checking whether startup messages can be suppressed ... OK
* checking use of S3 registration ... OK
* checking dependencies in R code ... OK
* checking S3 generic/method consistency ... OK
* checking replacement functions ... OK
* checking foreign function calls ... OK
* checking R code for possible problems ... OK
* checking Rd files ... OK
* checking Rd metadata ... OK
* checking Rd line widths ... OK
* checking Rd cross-references ... OK
* checking for missing documentation entries ... OK
* checking for code/documentation mismatches ... OK
* checking Rd \usage sections ... OK
* checking Rd contents ... OK
* checking for unstated dependencies in examples ... OK
* checking contents of 'data' directory ... OK
* checking data for non-ASCII characters ... OK
* checking lazydata ... OK
* checking data for ASCII and uncompressed saves ... OK
* checking installed files from 'inst/doc' ... OK
* checking files in 'vignettes' ... OK
* checking examples ... [14s/15s] OK
* checking for unstated dependencies in vignettes ... OK
* checking package vignettes ... OK
* checking re-building of vignette outputs ... [13s/14s] OK
* checking PDF version of manual ... OK

```

Question 1: Is the software package a **valid R package**?Question 2: Does the package satisfy the **CRAN policy**?

CRAN feasibility checks

(R CMD check -as-cran)

```

maciak@mbkpm79 ProfileLadder ~ -zsh -- 89x68
❯ R CMD check --as-cran ../ProfileLadder_0.2.2.tar.gz
* using log directory '/Users/maciak/Downloads/ProfileLadder/ProfileLadder.Rcheck'
* using R version 4.2.1 (2023-05-13)
* using platform: x86_64-apple-darwin20
* R was compiled by
  Apple clang version 14.0.0 (clang-1400.0.29.202)
  GNU Fortran (GCC) 14.2.0
* running under: macOS Tahoe 23.0.1
* using session charset: UTF-8
* using option '--as-cran'
* checking for file 'ProfileLadder/DESCRIPTION' ... OK
* checking extension type ... Package
* this is package 'ProfileLadder' version '0.2.2'
* package encoding: UTF-8
* checking CRAN incoming feasibility ... [8s/39s] OK
* checking package namespace information ... OK
* checking package dependencies ... OK
* checking if this is a source package ... OK
* checking if there is a namespace ... OK
* checking for executable files ... OK
* checking for hidden files and directories ... OK
* checking for portable file names ... OK
* checking for sufficient/correct file permissions ... OK
* checking whether package 'ProfileLadder' can be installed ... OK
* checking installed package size ... OK
* checking package directory ... OK
* checking for future file timestamps ... OK
* checking 'build' directory ... OK
* checking DESCRIPTION meta-information ... OK
* checking top-level files ... OK
* checking for left-over files ... OK
* checking index information ... OK
* checking package subdirectories ... OK
* checking code files for non-ASCII characters ... OK
* checking R files for syntax errors ... OK
* checking whether the package can be loaded ... OK
* checking whether the package can be loaded with stated dependencies ... OK
* checking whether the package can be unloaded cleanly ... OK
* checking whether the namespace can be loaded with stated dependencies ... OK
* checking whether the namespace can be unloaded cleanly ... OK
* checking whether startup messages can be suppressed ... OK
* checking use of S3 registration ... OK
* checking dependencies in R code ... OK
* checking S3 generic/method consistency ... OK
* checking replacement functions ... OK
* checking foreign function calls ... OK
* checking R code for possible problems ... OK
* checking Rd files ... OK
* checking Rd metadata ... OK
* checking Rd line widths ... OK
* checking Rd cross-references ... OK
* checking for missing documentation entries ... OK
* checking for code/documentation mismatches ... OK
* checking Rd \usage sections ... OK
* checking Rd contents ... OK
* checking for unstated dependencies in examples ... OK
* checking contents of 'data' directory ... OK
* checking data for non-ASCII characters ... OK
* checking data for ASCII and uncompressed saves ... OK
* checking installed files from 'inst/doc' ... OK
* checking files in 'vignettes' ... OK
* checking examples ... [14s/15s] OK
* checking for unstated dependencies in vignettes ... OK
* checking package vignettes ... OK
* checking re-building of vignette outputs ... [13s/14s] OK
* checking PDF version of manual ... OK

```

Question 1: Is the software package a **valid R package**?Question 2: Does the package satisfy the **CRAN policy**?

CRAN feasibility check with R CMD check -as-cran

- ❑ **NOTEs** and **WARNINGS** are typically considered to be **serious issues** in CRAN Repository Policy
- ❑ clean CRAN-like environment – no user options, no profiles, no environment variables, no pre-installed files
- ❑ documentation quantlity (proper title/sentence styles, capitalization, spelling)
- ❑ multiple OS platform checks (Linux, macOS, Windows)
- ❑ additional checks that may not be reproduced locally

CRAN package checks and review

Before the package is finally accepted by CRAN, it undergoes several (automatic and manual) levels of **technical checking and quality control**

- ❑ **Developer/local check** – before submission, the package should pass a (local) CRAN-like check:

```
R CMD check --as-cran package.tar.gz
```

```
devtools::check()
```
- ❑ Check results are classified as **ERROR**, **WARNING** and **NOTE** – errors and warnings must normally be removed; significant notes must be eliminated or properly explained/justified
- ❑ **Cross-platform checks** – Win-builder, R-hub or GitHub Actions can be used before submission to test different R versions, Linux/Windows/Macintosh operating systems, or different compilers
- ❑ **Automatic CRAN checks** – after submission CRAN runs `R CMD check` on its own infrastructure; package updates may also undergo **reverse-dependency checks**.
- ❑ **Manual CRAN review** – especially for new packages, CRAN maintainers additionally inspect policy compliance, metadata, licensing, documentation and unresolved check issues.

user-based local check → CRAN automatic check → human review → CRAN

ProfileLadder package on CRAN

(Version 0.2.2)



Maciak, M., Mizera, I., and Pešta, M. (2022). **Functional Profile Techniques for Claims Reserving.** *ASTIN Bulletin*, 52(2), 449-482. DOI:10.1017/asb.2022.4

Maciak, M., Matúš, R., Mizera, I., and Pešta, M. (2026). **ProfileLadder: Functional-Based Reserving.** *The R Journal*, 18(2), 138-160. DOI:10.32614/RJ-2026-028

ProfileLadder package on CRAN

(Version 0.2.2)



Maciak, M., Mizera, I., and Pešta, M. (2022). **Functional Profile Techniques for Claims Reserving.** *ASTIN Bulletin*, 52(2), 449-482. DOI:10.1017/asb.2022.4

Maciak, M., Matúš, R., Mizera, I., and Pešta, M. (2026). **ProfileLadder: Functional-Based Reserving.** *The R Journal*, 18(2), 138-160. DOI:10.32614/RJ-2026-028

Part III

Dissemination of the work and package distribution

Accompanying manuscript, automatization and versioning via GitHub repository

What's next?

Publishing the **R package on CRAN** makes it potentially available to the whole R community... But, **how to promote the package** and **how to make it utilized in practice?**

What's next?

Publishing the **R package on CRAN** makes it potentially available to the whole R community... But, **how to promote the package** and **how to make it utilized in practice?**

The package on CRAN is only the beginning...

- ❑ focus also on the **package maintenance**, **regular updates**, and **proper versioning**
- ❑ pay attention to **bug reports and reported errors** (emails or bug reporting page)
- ❑ try to **promote the package among practitioners** (actuaries in this particular case)
- ❑ accompany the paper with a **vignette/package website/manuscript...**
- ❑ etc.

Dissemination: Different audiences need different channels

A successful methodological contribution should ideally reach **several different communities** ...

❑ Scientific community

- journal article, conference presentation, invited seminars, research communication
- methodological contribution, theoretical properties, empirical comparisons, practical utilization

❑ R software community

- CRAN, GitHub, package website, vignette
- reproducible examples, documentation, source code

❑ Insurance practitioners

- insurance companies, actuarial meetings and workshops, practical trainings
- apply it for real problems with real datasets, practical interpretation, computational workflow

Dissemination: Different audiences need different channels

A successful methodological contribution should ideally reach **several different communities** ...

❑ Scientific community

- journal article, conference presentation, invited seminars, research communication
- methodological contribution, theoretical properties, empirical comparisons, practical utilization

❑ R software community

- CRAN, GitHub, package website, vignette
- reproducible examples, documentation, source code

❑ Insurance practitioners

- insurance companies, actuarial meetings and workshops, practical trainings
- apply it for real problems with real datasets, practical interpretation, computational workflow

good idea $\Rightarrow$ methodology $\Rightarrow$ software $\Rightarrow$ communication $\Rightarrow$ adoption

"The value of an idea lies in the using of it." Thomas A. Edison

From a rough idea to a real actuarial practice ...

Proposing a new method, deriving its theoretical background, proving its superiority, making the corresponding R package available on CRAN, and communicating the results among the scientific community does **not automatically imply practical adoption** of the method...

- ❑ Demonstrate the method on **real/actual insurance data** rather than some artificial examples
- ❑ Present the results in a form familiar to actuarial practitioners: **reserve estimates, prediction intervals, risk measures, diagnostic plots, residuals, ...**
- ❑ Compare the new approach with **established benchmarks** well known to actuaries and well-established in practice (ChainLadder, Mack, Bootstrap Chain Ladder, etc.)
- ❑ Organize **presentations, workshops, or training sessions** directly for insurance companies and actuarial associations
- ❑ **Collect feedback** from practitioners: important pros and cons/ computational speed, interpretability, required outputs, missing functionality, data compatibility, etc.

From a rough idea to a real actuarial practice ...

Proposing a new method, deriving its theoretical background, proving its superiority, making the corresponding R package available on CRAN, and communicating the results among the scientific community does **not automatically imply practical adoption** of the method...

- ❑ Demonstrate the method on **real/actual insurance data** rather than some artificial examples
- ❑ Present the results in a form familiar to actuarial practitioners: **reserve estimates, prediction intervals, risk measures, diagnostic plots, residuals, ...**
- ❑ Compare the new approach with **established benchmarks** well known to actuaries and well-established in practice (ChainLadder, Mack, Bootstrap Chain Ladder, etc.)
- ❑ Organize **presentations, workshops, or training sessions** directly for insurance companies and actuarial associations
- ❑ **Collect feedback** from practitioners: important pros and cons/ computational speed, interpretability, required outputs, missing functionality, data compatibility, etc.

A practically useful method must solve a practically relevant problem but the overall adoption still may take years (mainly due to various regulatory restrictions)

Extensive collaboration with actuaries

❑ Generali – Česká pojišťovna, a.s.

- one of the largest insurance companies in the Czech Republic, part of the international Generali Group, whose parent company is Assicurazioni Generali S.p.A. in Italy
- provided three exclusive datasets for the R package ProfileLadder (CZ.casco, CZ.liability, and CZ.property)

Extensive collaboration with actuaries

❑ Generali – Česká pojišťovna, a.s.

- one of the largest insurance companies in the Czech Republic, part of the international Generali Group, whose parent company is Assicurazioni Generali S.p.A. in Italy
- [provided three exclusive datasets](#) for the R package ProfileLadder (CZ.casco, CZ.liability, and CZ.property)

❑ Czech Insurers' Bureau

- professional organization of insurers that are authorized to provide motor third-party liability insurance (MTPL) in the Czech Republic
- [provided a large dataset](#) from the MTPL insurance in the Czech Republic (the dataset GFCIB with 60 origins and 60 developments)

Extensive collaboration with actuaries

❑ Generali – Česká pojišťovna, a.s.

- one of the largest insurance companies in the Czech Republic, part of the international Generali Group, whose parent company is Assicurazioni Generali S.p.A. in Italy
- provided three exclusive datasets for the R package ProfileLadder (CZ.casco, CZ.liability, and CZ.property)

❑ Czech Insurers' Bureau

- professional organization of insurers that are authorized to provide motor third-party liability insurance (MTPL) in the Czech Republic
- provided a large dataset from the MTPL insurance in the Czech Republic (the dataset GFCIB with 60 origins and 60 developments)

But there is (usually) no free lunch!

From the scientist's perspective: JSS or The R Journal

From the scientist's perspective: JSS or The R Journal

Journal of Statistical Software

The R Journal

From the scientist's perspective: JSS or The R Journal

Journal of Statistical Software

The R Journal

- ❑ Theoretical/methodological paper
- ❑ R package available on CRAN
- ❑ JSS working paper with different dataset than the one used in the theoretical paper
- ❑ That's all! $\implies$ **Good-to-go!**

From the scientist's perspective: JSS or The R Journal

Journal of Statistical Software

- ❑ Theoretical/methodological paper
- ❑ R package available on CRAN
- ❑ JSS working paper with different dataset than the one used in the theoretical paper
- ❑ That's all! $\implies$ **Good-to-go!**

The R Journal

- ❑ **GitHub** integration/versioning
- ❑ **Package website** (e.g., GH pages)
- ❑ **Vignette** – ideally with different dataset than the one used in the RJ paper
- ❑ **rjtools** submission check
- ❑ **Finally, good-to-go now!**

Aims and scopes of both leading (software) journals

Journal of Statistical Software | JSS

- ❑ Founded in 1996 by Jan de Leeuw (Department of Statistics, UCLA)
- ❑ Volume 114 at the beginning of 2026
- ❑ JIF: 8.1 Q1 | AIS: 5/166 Q1
- ❑ **Editors-in-Chief**
B.Grün, R.Killick, T.Hothorn,
N. Vialaneix, and A. Zeileis
- ❑ **Publication time:** rather long (> 1 year)

The R Journal | RJ

- ❑ Established in 2009 by the R Foundation for Statistical Computing
- ❑ 4 issues per year, cca 65 issues
- ❑ JIF: 2.3 Q2 | AIS: 33/166 Q1
- ❑ **Editor-in-Chief**
Emi Tanaka, Rob Hyndman
- ❑ **Publication time:** 4 – 6 months

Aims and scopes of both leading (software) journals

Journal of Statistical Software | JSS

- ❑ Founded in 1996 by Jan de Leeuw (Department of Statistics, UCLA)
- ❑ Volume 114 at the beginning of 2026
- ❑ JIF: 8.1 Q1 | AIS: 5/166 Q1
- ❑ **Editors-in-Chief**
B.Grün, R.Killick, T.Hothorn,
N. Vialaneix, and A. Zeileis
- ❑ **Publication time:** rather long (> 1 year)
- ❑ **Final decision:** two days

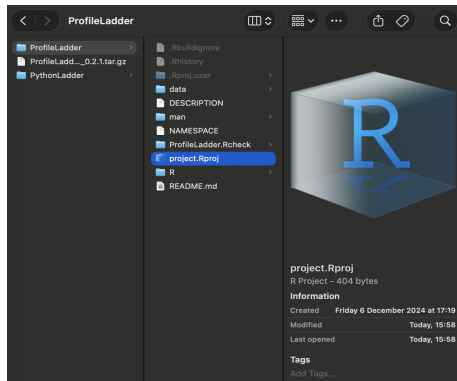
The R Journal | RJ

- ❑ Established in 2009 by the R Foundation for Statistical Computing
- ❑ 4 issues per year, cca 65 issues
- ❑ JIF: 2.3 Q2 | AIS: 33/166 Q1
- ❑ **Editor-in-Chief**
Emi Tanaka, Rob Hyndman
- ❑ **Publication time:** 4 – 6 months
- ❑ **Final decision:** 6 weeks / 10 weeks

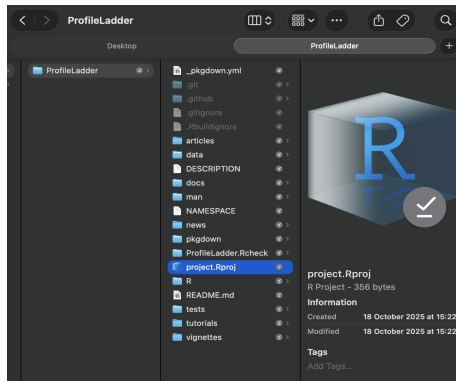
ProfileLadder: Package deployment vs. paper submission

ProfileLadder package structure for

- CRAN deployment



- The R Journal submission



Publications submitted to JSS/RJ

Both journals dedicated to [statistical software](#) and [computational methods](#) with theoretical methodology, proper source codes, and replication materials suitable for R or Python, MATLAB, C++, etc.

❑ Journal of Statistical Software

- ❑ journal dedicated to [substantial statistical software](#) and [new computational methods](#)
- ❑ software in [R](#), [Python](#), [MATLAB](#), [C++](#), and [others](#) — as long as reproducibility is shown
- ❑ [R packages are welcome](#), code must be high quality, well documented, vignettes recommended

Publications submitted to JSS/RJ

Both journals dedicated to [statistical software](#) and [computational methods](#) with theoretical methodology, proper source codes, and replication materials suitable for R or Python, MATLAB, C++, etc.

❑ Journal of Statistical Software

- ❑ journal dedicated to [substantial statistical software](#) and [new computational methods](#)
- ❑ software in [R, Python, MATLAB, C++, and others](#) — as long as reproducibility is shown
- ❑ [R packages are welcome](#), code must be high quality, well documented, vignettes recommended

❑ The R Journal

- ❑ official journal of the R Foundation, [primarily oriented toward the R community](#)
- ❑ [articles about R](#), software packages, teaching resources, benchmark comparisons, and applications
- ❑ [enforces strict guidelines](#) for package papers focusing on [design](#), [performance](#), and [implementation](#)

Publications submitted to JSS/RJ

Both journals dedicated to **statistical software** and **computational methods** with theoretical methodology, proper source codes, and replication materials suitable for R or Python, MATLAB, C++, etc.

❑ Journal of Statistical Software

- ❑ journal dedicated to **substantial statistical software** and **new computational methods**
- ❑ software in **R, Python, MATLAB, C++, and others** — as long as reproducibility is shown
- ❑ **R packages are welcome**, code must be high quality, well documented, vignettes recommended

Suitable for major theoretical/methodological contributions (a published paper) with a well-engineered software, standard code requirements, examples beyond the original paper, and rigorous reproducibility – all meant for broad statistical audience and high academic impact

❑ The R Journal

- ❑ official journal of the R Foundation, **primarily oriented toward the R community**
- ❑ **articles about R**, software packages, teaching resources, benchmark comparisons, and applications
- ❑ **enforces strict guidelines** for package papers focusing on **design**, **performance**, and **implementation**

Publications submitted to JSS/RJ

Both journals dedicated to **statistical software and computational methods** with theoretical methodology, proper source codes, and replication materials suitable for R or Python, MATLAB, C++, etc.

❑ Journal of Statistical Software

- ❑ journal dedicated to **substantial statistical software** and **new computational methods**
- ❑ software in **R, Python, MATLAB, C++, and others** — as long as reproducibility is shown
- ❑ **R packages are welcome**, code must be high quality, well documented, vignettes recommended

Suitable for major theoretical/methodological contributions (a published paper) with a well-engineered software, standard code requirements, examples beyond the original paper, and rigorous reproducibility – all meant for broad statistical audience and high academic impact

❑ The R Journal

- ❑ official journal of the R Foundation, **primarily oriented toward the R community**
- ❑ **articles about R**, software packages, teaching resources, benchmark comparisons, and applications
- ❑ **enforces strict guidelines** for package papers focusing on **design, performance, and implementation**

Suitable for new R software packages or codes, with examples, implementation details, and insights for users with a community-oriented impact with a focus on package design, structure, effectivity, and functionality

Our own experience...

Journal of Statistical Software

Our own experience...

Journal of Statistical Software

- ❑ **Desk rejection (after a few days) due to ...**
 - ❑ typical justification “the paper does not fully meet the journal’s scope and standards”
 - ❑ the submission resembles more some supplementary material to the previous theoretical article in ASTIN Bulletin than a general software implementation for broader audience and interest

Our own experience...

Journal of Statistical Software

❑ Desk rejection (after a few days) due to ...

- ❑ typical justification “the paper does not fully meet the journal’s scope and standards”
- ❑ the submission resembles more some supplementary material to the previous theoretical article in ASTIN Bulletin than a general software implementation for broader audience and interest

❑ Some constructive remarks

- ❑ print/summary/plot/predict methods should be implemented for each S3 class (!)
- ❑ exposition of objects’ internal structure is not a recommended programming practice—instead, dedicated accessor methods should be implemented to assess specific entries of the objects
- ❑ additional methods should be implemented for model based objects (predictions, fitted values, etc.)
- ❑ it is a good practice, that complex objects (outputs) always have a dedicated class with the corresponding PS3/S4 methods

Our own experience...

Journal of Statistical Software

❑ Desk rejection (after a few days) due to ...

- ❑ typical justification “the paper does not fully meet the journal’s scope and standards”
- ❑ the submission resembles more some supplementary material to the previous theoretical article in ASTIN Bulletin than a general software implementation for broader audience and interest

❑ Some constructive remarks

- ❑ print/summary/plot/predict methods should be implemented for each S3 class (!)
- ❑ exposition of objects’ internal structure is not a recommended programming practice—instead, dedicated accessor methods should be implemented to assess specific entries of the objects
- ❑ additional methods should be implemented for model based objects (predictions, fitted values, etc.)
- ❑ it is a good practice, that complex objects (outputs) always have a dedicated class with the corresponding PS3/S4 methods

Detailed package/paper instructions: <https://www.jstatsoft.org/authors>

S3 methods vs. S4 methods in R

Both **object-oriented programming systems** provide **class-specific methods** for generic functions:

`generic(object)` $\implies$ method determined by the object's class

S3 methods vs. S4 methods in R

Both **object-oriented programming systems** provide **class-specific methods** for generic functions:

`generic(object)` $\implies$ method determined by the object's class

S3 methods

Informal, lightweight, and easy to use OOP system

Class is stored as an object attribute and methods follow the convention

```
class(x) <- "myModel"  
summary.myModel <- function(x, ...)
```

Advantages

- ✓ very simple and flexible to implement
- ✓ natural for statistical model objects

Disadvantages

- ✗ no formal definition of object structure
- ✗ errors may appear only at run time

S4 methods

More formal / strictly defined OOP system

Classes, slots, inheritance and methods are explicitly defined

```
setClass("myModel", ...)  
setMethod("summary", "myModel", ...)
```

Advantages

- ✓ explicit structure and type validation
- ✓ suitable for large, complex object systems

Disadvantages

- ✗ more verbose and more difficult to learn
- ✗ greater development/maintenance overhead

Back to our own experience...

The R Journal

- ❑ Major revision (after six weeks) due to ...
 - ❑ the submission makes important contributions to the R community, particularly for actuarial sciences, insurance practitioners, and other researchers

Back to our own experience...

The R Journal

- ❑ Major revision (after six weeks) due to ...
 - ❑ the submission makes important contributions to the R community, particularly for actuarial sciences, insurance practitioners, and other researchers
 - ❑ the nonparametric approach is sensible and addresses real limitations of existing parametric methods

Back to our own experience...

The R Journal

- ❑ **Major revision (after six weeks) due to ...**
 - ❑ the submission makes important contributions to the R community, particularly for actuarial sciences, insurance practitioners, and other researchers
 - ❑ the nonparametric approach is sensible and addresses real limitations of existing parametric methods
 - ❑ the integration with the established ChainLadder package is very thoughtful, allowing the permutation bootstrap to be applied to both classical and novel methods

Back to our own experience...

The R Journal

- ❑ Major revision (after six weeks) due to ...
 - ❑ the submission makes important contributions to the R community, particularly for actuarial sciences, insurance practitioners, and other researchers
 - ❑ the nonparametric approach is sensible and addresses real limitations of existing parametric methods
 - ❑ the integration with the established ChainLadder package is very thoughtful, allowing the permutation bootstrap to be applied to both classical and novel methods
 - ❑ the technology appears current, and the methods are well theoretically grounded in recent research (Maciak, Mizera, and Pešta, 2022) published in a leading actuarial journal

Back to our own experience...

The R Journal

❑ Major revision (after six weeks) due to ...

- ❑ the submission makes important contributions to the R community, particularly for actuarial sciences, insurance practitioners, and other researchers
- ❑ the nonparametric approach is sensible and addresses real limitations of existing parametric methods
- ❑ the integration with the established ChainLadder package is very thoughtful, allowing the permutation bootstrap to be applied to both classical and novel methods
- ❑ the technology appears current, and the methods are well theoretically grounded in recent research (Maciak, Mizera, and Pešta, 2022) published in a leading actuarial journal
- ❑ the ProfileLadder package extends and complements the widely used ChainLadder package for loss reserving; both the paper and the package are very clear and intuitive

Back to our own experience...

The R Journal

❑ Major revision (after six weeks) due to ...

- ❑ the submission makes important contributions to the R community, particularly for actuarial sciences, insurance practitioners, and other researchers
- ❑ the nonparametric approach is sensible and addresses real limitations of existing parametric methods
- ❑ the integration with the established ChainLadder package is very thoughtful, allowing the permutation bootstrap to be applied to both classical and novel methods
- ❑ the technology appears current, and the methods are well theoretically grounded in recent research (Maciak, Mizera, and Pešta, 2022) published in a leading actuarial journal
- ❑ the ProfileLadder package extends and complements the widely used ChainLadder package for loss reserving; both the paper and the package are very clear and intuitive
- ❑ the package workflow is clear and intuitive, with logical function names, good use of S3 methods, and elegant integration with ChainLadder through observed() and as.profileLadder() functions

Back to our own experience...

The R Journal

❑ Major revision (after six weeks) due to ...

- ❑ the submission makes important contributions to the R community, particularly for actuarial sciences, insurance practitioners, and other researchers
- ❑ the nonparametric approach is sensible and addresses real limitations of existing parametric methods
- ❑ the integration with the established ChainLadder package is very thoughtful, allowing the permutation bootstrap to be applied to both classical and novel methods
- ❑ the technology appears current, and the methods are well theoretically grounded in recent research (Maciak, Mizera, and Pešta, 2022) published in a leading actuarial journal
- ❑ the ProfileLadder package extends and complements the widely used ChainLadder package for loss reserving; both the paper and the package are very clear and intuitive
- ❑ the package workflow is clear and intuitive, with logical function names, good use of S3 methods, and elegant integration with ChainLadder through observed() and as.profileLadder() functions
- ❑ documentation appears comprehensive with helpful examples, thoughtful parameter defaults, flexible user control through parameters states, breaks, and method, good accessor functions, and incrExplor() function for exploratory analysis is a valuable addition

Back to our own experience...

The R Journal

❑ Major revision (after six weeks) due to ...

- ❑ the submission makes important contributions to the R community, particularly for actuarial sciences, insurance practitioners, and other researchers
- ❑ the nonparametric approach is sensible and addresses real limitations of existing parametric methods
- ❑ the integration with the established ChainLadder package is very thoughtful, allowing the permutation bootstrap to be applied to both classical and novel methods
- ❑ the technology appears current, and the methods are well theoretically grounded in recent research (Maciak, Mizera, and Pešta, 2022) published in a leading actuarial journal
- ❑ the ProfileLadder package extends and complements the widely used ChainLadder package for loss reserving; both the paper and the package are very clear and intuitive
- ❑ the package workflow is clear and intuitive, with logical function names, good use of S3 methods, and elegant integration with ChainLadder through observed() and as.profileLadder() functions
- ❑ documentation appears comprehensive with helpful examples, thoughtful parameter defaults, flexible user control through parameters states, breaks, and method, good accessor functions, and incrExplor() function for exploratory analysis is a valuable addition

Detailed package/paper instructions: <https://journal.r-project.org/submissions.html>

Additional requirements of The R Journal (Rob Hyndman)

❑ Remarks on the package structure/design

- ❑ there is **no version control for your package**. I recommend using [GitHub](#)
- ❑ there is **no package website** – this can be handled straightforwardly using [GH Pages](#)
- ❑ the documentation and help files in the package are ok, but on their own provide limited guidance to users—the **package vignette is requested** to be added
- ❑ in addition, **complex unit tests should be added** to the package

Additional requirements of The R Journal (Rob Hyndman)

❑ Remarks on the package structure/design

- ❑ there is **no version control for your package**. I recommend using [GitHub](#)
- ❑ there is **no package website** – this can be handled straightforwardly using [GH Pages](#)
- ❑ the documentation and help files in the package are ok, but on their own provide limited guidance to users—the **package vignette is requested** to be added
- ❑ in addition, **complex unit tests should be added** to the package

❑ Remarks on package functionality

- ❑ the algorithms seem effective but **consider adding progress bars** for computationally intensive **operations** (especially permutation bootstrap)
- ❑ for large triangles, **storing all B bootstrap samples might be memory-intensive**—consider options **for summary statistics only** that will be stored instead
- ❑ in order to broaden the perspective and potential user base of the package, it would be useful if **at least one detailed example outside the loss reserving context** is provided

It is good to consider the following ...

- ❑ GitHub repository and version control
- ❑ GitHub Pages / Package website
- ❑ Package vignette
- ❑ The R Journal feasibility checks

It is good to consider the following ...

- ❑ **GitHub repository and version control**

- ✓ transparent development history, collaboration, issue tracking, reproducibility, easy access to development versions (with [GitHub Actions](#) and [Continuous Integration](#) as a bonus)

- ✗ requires regular commits, meaningful versioning and repository maintenance

- ❑ **GitHub Pages / Package website**

- ❑ **Package vignette**

- ❑ **The R Journal feasibility checks**

It is good to consider the following ...

❑ **GitHub repository and version control**

✓ transparent development history, collaboration, issue tracking, reproducibility, easy access to development versions (with [GitHub Actions](#) and [Continuous Integration](#) as a bonus)

✗ requires regular commits, meaningful versioning and repository maintenance

❑ **GitHub Pages / Package website**

✓ attractive online documentation, searchable function reference, examples and vignettes; easy dissemination outside CRAN

✗ another component that must be generated, updated and kept synchronized with the package

❑ **Package vignette**

❑ **The R Journal feasibility checks**

It is good to consider the following ...

❑ GitHub repository and version control

✓ transparent development history, collaboration, issue tracking, reproducibility, easy access to development versions (with [GitHub Actions](#) and [Continuous Integration](#) as a bonus)

✗ requires regular commits, meaningful versioning and repository maintenance

❑ GitHub Pages / Package website

✓ attractive online documentation, searchable function reference, examples and vignettes; easy dissemination outside CRAN

✗ another component that must be generated, updated and kept synchronized with the package

❑ Package vignette

✓ explains the complete workflow on a realistic example: *problem* → *data* → *model* → *output* → *interpretation*

✗ computationally expensive examples may slow down package building/checking and must remain reproducible

❑ The R Journal feasibility checks

It is good to consider the following ...

❑ GitHub repository and version control

✓ transparent development history, collaboration, issue tracking, reproducibility, easy access to development versions (with [GitHub Actions](#) and [Continuous Integration](#) as a bonus)

✗ requires regular commits, meaningful versioning and repository maintenance

❑ GitHub Pages / Package website

✓ attractive online documentation, searchable function reference, examples and vignettes; easy dissemination outside CRAN

✗ another component that must be generated, updated and kept synchronized with the package

❑ Package vignette

✓ explains the complete workflow on a realistic example: *problem* → *data* → *model* → *output* → *interpretation*

✗ computationally expensive examples may slow down package building/checking and must remain reproducible

❑ The R Journal feasibility checks

✓ automatic pre-submission checks of manuscript structure, filenames, package availability, spelling, sections, and other RJ requirements

✗ passing the automated checks guarantees only formal compliance—not [acceptance or scientific quality](#)

GitHub for R package development

GitHub provides much more than just an online copy of the source code – it can serve as the **central platform for development, maintenance, and dissemination** of an R package.

GitHub for R package development

GitHub provides much more than just an online copy of the source code – it can serve as the **central platform for development, maintenance, and dissemination** of an R package.

Development & maintenance

- ❑ **Version control** – complete history of changes, commits, branches, and releases
- ❑ **Collaboration** – several developers can work on the same package using branches
- ❑ **Issues** – structured reporting of bugs, feature requests, and user feedback
- ❑ **GitHub Actions / CI** – automatic testing, R CMD check, package building, etc.

GitHub for R package development

GitHub provides much more than just an online copy of the source code – it can serve as the **central platform for development, maintenance, and dissemination** of an R package.

Development & maintenance

- ❑ **Version control** – complete history of changes, commits, branches, and releases
- ❑ **Collaboration** – several developers can work on the same package using branches
- ❑ **Issues** – structured reporting of bugs, feature requests, and user feedback
- ❑ **GitHub Actions / CI** – automatic testing, R CMD check, package building, etc.

Distribution & dissemination

- ❑ **Direct installation:**
`remotes::install_github("...")`
- ❑ **GitHub Releases** – named/versioned snapshots of stable software releases
- ❑ **GitHub Pages** – hosting of the package website generated, e.g., by pkgdown
- ❑ public source code improves **transparency and reproducibility**

GitHub for R package development

GitHub provides much more than just an online copy of the source code – it can serve as the **central platform for development, maintenance, and dissemination** of an R package.

Development & maintenance

- ❑ **Version control** – complete history of changes, commits, branches, and releases
- ❑ **Collaboration** – several developers can work on the same package using branches
- ❑ **Issues** – structured reporting of bugs, feature requests, and user feedback
- ❑ **GitHub Actions / CI** – automatic testing, R CMD check, package building, etc.

Distribution & dissemination

- ❑ **Direct installation:**
`remotes::install_github("...")`
- ❑ **GitHub Releases** – named/versioned snapshots of stable software releases
- ❑ **GitHub Pages** – hosting of the package website generated, e.g., by pkgdown
- ❑ public source code improves **transparency and reproducibility**

local development → **GitHub** → **testing / collaboration** → **CRAN release**

Continuous Integration (CI) and GitHub Actions (GA)

Continuous Integration (CI) ensures that any changes to the package are automatically tested and checked in a clean environment whenever the source code is updated/modified...

Code changes $\Rightarrow$ GitHub push $\Rightarrow$ fresh environment $\Rightarrow$ build/test/check $\Rightarrow$ report

- ❑ **GitHub Actions** – GitHub infrastructure for implementing automated CI workflows stored in `.github/workflows/*.yaml`.
- ❑ Typical R package workflows can automatically run:
 - ✓ R CMD check and testthat tests
 - ✓ checks on Linux, Windows and macOS / different R versions
 - ✓ code coverage, package website generation, deployment, etc.
- ❑ **Advantages:** reproducibility, early detection of errors, cross-platform checking, automatic validation of every update, easy package management
- ❑ **Disadvantages:** initial configuration is required; workflows may fail because of external dependencies or platform-specific issues (also may not be free of charge)

Continuous Integration (CI) and GitHub Actions (GA)

Continuous Integration (CI) ensures that any changes to the package are **automatically tested and checked** in a clean environment whenever the source code is updated/modified...

Code changes $\Rightarrow$ GitHub push $\Rightarrow$ fresh environment $\Rightarrow$ build/test/check $\Rightarrow$ report

- ❑ **GitHub Actions** – GitHub infrastructure for implementing automated CI workflows stored in `.github/workflows/*.yaml`.
- ❑ Typical R package workflows can automatically run:
 - ✓ R CMD check and testthat tests
 - ✓ checks on Linux, Windows and macOS / different R versions
 - ✓ code coverage, package website generation, deployment, etc.
- ❑ **Advantages:** reproducibility, early detection of errors, cross-platform checking, automatic validation of every update, easy package management
- ❑ **Disadvantages:** initial configuration is required; workflows may fail because of external dependencies or platform-specific issues (also may not be free of charge)

“It works on my computer” is not enough!

GitHub pages

(pkgdown)

```

ProfileLadder - R - 89x69

> pkgdown::build_site()
- Installing package ProfileLadder into temporary library -

- Building pkgdown site for package ProfileLadder -
Reading from: /Users/maciek/ProfileLadder
Writing to: /Users/maciek/ProfileLadder/docs
- Sitemap
  ✓ URLs ok.
  ✓ Favicons ok.
  ✓ Open graph metadata ok.
  ✓ Articles metadata ok.
  ✓ Reference metadata ok.
- Initialising site
Copying pkgdown/favicon/apple-touch-icon.png to apple-touch-icon.png
Copying pkgdown/favicon/favicon-96x96.png to favicon-96x96.png
Copying pkgdown/favicon/favicon.ico to favicon.ico
Copying pkgdown/favicon/favicon.svg to favicon.svg
Copying pkgdown/favicon/web-app-manifest-192x192.png to
web-app-manifest-192x192.png
Copying pkgdown/favicon/web-app-manifest-512x512.png to
web-app-manifest-512x512.png
- Building home
Reading README.md
Writing 404.html
- Building function reference -
Reading man/C2.esec.Rd
Reading man/C2.liability.Rd
Reading man/C2.property.Rd
Reading man/CenterMutual.Rd
Reading man/GFCR.Rd
Reading man/HideWestMutual.Rd
Reading man/NeveadGeneral.Rd
Reading man/ps.profileLadder.Rd
Reading man/covid19C2.Rd
Reading man/LincExplorer.Rd
Reading man/mcReake.Rd
Reading man/mcReserve.Rd
Reading man/mcStatus.Rd
Reading man/mcTrans.Rd
Reading man/observed.Rd
Reading man/parallelReserve.Rd
Reading man/permutReserve.Rd
Writing reference/permutReserve.html
Reading man/plot.mcSetup.Rd
Reading man/plot.permutReserve.Rd
Writing reference/plot.permutReserve.html
Reading man/plot.profileLadder.Rd
Reading man/plot.profilePredict.Rd
Reading man/predict.profileLadder.Rd
Reading man/print.mcSetup.Rd
Reading man/print.permutReserve.Rd
Writing reference/print.permutReserve.html
Reading man/print.profileLadder.Rd
Reading man/print.profilePredict.Rd
Reading man/set.fancy.print.Rd
Reading man/summary.mcSetup.Rd
Reading man/summary.permutReserve.Rd
Writing reference/summary.permutReserve.html
Reading man/summary.profileLadder.Rd
Reading man/summary.subscribe.Rd
- Building articles
Reading vignettes/ev.vignette.Rd
g7232f6e4.md line 709 column 1, closing implicitly.n 1 unclosed at filebase
- Building sitemap -
- Building search index -
- Checking for problems -
- Finished building pkgdown site for package ProfileLadder -
- Finished building pkgdown site for package ProfileLadder -
    
```

- ❑ automatically turns the R package into a website (using the information from the DESCRIPTION file, README, and .Rd documentation files)
- ❑ provides online reference pages (i.e., html manuals) for all functions, datasets, and classes methods



Generates a full working website from the R package – uses the package documentation, vignettes, README, and metadata to share and promote information about the package online.



ProfileLadder



CRAN 0.2.2 downloads 2273 release v0.2.2

Overview

The R package **ProfileLadder** provides nonparametric, functional-based methods for claims reserving based on aggregated chain-ladder data also known as the run-off triangles. The package implements three estimation/prediction algorithms (PARALLAX, REACT, and MACRAME) and the permutation bootstrap add-on proposed in Maciak, Mizera, and Pešta (2022).

The package offers a flexible and computationally effective framework for point-wise and distributional reserve predictions and includes pertinent visualization and diagnostic tools through S3 methods. It also provides accessor functions and real-world datasets to support exploratory analysis across insurance, operational risks, and other domains where triangular data structures arise, making modern, transparent, and extensible alternatives to classical approaches accessible in insurance industry and academic research.

Installation

The R package **ProfileLadder** can be downloaded from CRAN and installed in R by using

```
install.packages("ProfileLadder")
```

Core package functions

- `parallelReserve` implements the PARALLAX and REACT algorithms
- `mcReserve` implements the MACRAME algorithm
- `permuteReserve` provides a distributional reserve prediction (permutation bootstrap)
- `predict.profileLadder` 1-step-ahead prediction (prediction of the next running diagonal)

Links

[Browse source code](#)

[Report a bug](#)

License

GPL (>=2)

Citation

[Citing ProfileLadder](#)

Developers

Matúš Maciak

Author, maintainer

Rastislav Matúš

Author, contributor

Ivan Mizera

Author

Michal Pešta

Author

Vignette: How the package should be used for real work?

- ❑ explains the problem the package is meant to solve
- ❑ gives details about the statistical/methodological idea and typical workflow
initial problem ⇒ data ⇒ model ⇒ output ⇒ interpretation
- ❑ typically it should involve more complex data beyond simple illustration purposes

Vignette: How the package should be used for real work?

- ❑ explains the problem the package is meant to solve
- ❑ gives details about the statistical/methodological idea and typical workflow
initial problem ⇒ data ⇒ model ⇒ output ⇒ interpretation
- ❑ typically it should involve more complex data beyond simple illustration purposes

Vignette preparation

```
R> usethis::use_vignette("vignette")
```

which will create the vignette source .Rmd file and modifies the DESCRIPTION file correspondingly (in particular, the lines `VignetteBuilder: knitr`, and `Suggests: knitr/rmarkdown`)

ProfileLadder package vignette: https://cran.r-project.org/web/packages/ProfileLadder/vignettes/my_vignette.html

Functional-Based Reserving with the R Package ProfileLadder



Matúš Maciak, Rastislav Matúš, Ivan Mizera, and Michal Pešta

Source: [vignettes/my_vignette.Rmd](#)

The R package `ProfileLadder` provides nonparametric, functional-based methods for claims reserving based on aggregated chain-ladder data also known as the *run-off triangles*. The package implements three estimation/prediction algorithms (PARALLAX, REACT, and MACRAME) and the permutation bootstrap add-on proposed in Maciak, Mizera, and Pešta (2022).

The package offers a flexible and computationally effective framework for point-wise and distributional reserve predictions and includes pertinent visualization and diagnostic tools through standard S3 methods—such as `plot()`, `predict()`, `print()`, or `summary()`. It also provides accessor functions and real-world datasets to support exploratory analysis across insurance, operational risks, and other domains where triangular data structures arise, making modern, transparent, and extensible alternatives to classical approaches accessible in insurance industry and academic research.

The following lines provide an illustrative sample analysis and reserve prediction for two specific run-off triangles using all three nonparametric prediction algorithms and other functions introduced within the `ProfileLadder` package. The functionality of the core functions of the package are and some important features are presented and discussed. Further theoretical details can be found in

Maciak, M., Mizera, I., and Pešta, M. (2022). **Functional Profile Techniques for Claims Reserving**. *ASTIN Bulletin*, 52(2), 449 – 482. DOI: [10.1017/asb.2022.4](#)

Package manual and GitHub pages

- PDF/HTML manual: <https://CRAN.R-project.org/package=ProfileLadder>
- GitHub Pages: <https://42463863.github.io/ProfileLadder/index.html>

The package `ProfileLadder` can be installed directly from CRAN using a standard command

```
install.packages("ProfileLadder")
```

The package `ProfileLadder` also allows for a convenient and elegant integration with the leading actuarial R package `ChainLadder` commonly utilized for risk assessment and actuarial practice.

On this page

Package manual and GitHub pages

Package structure

Notation and example datasets

1. Data visualization with ProfileLadder

2. PARALLAX

3. REACT

4. MACRAME

5. Permutation bootstrap

6. Generic S3 method `predict()`

Conclusion

The R journal feasibility checks

(rjtools)

The R package that helps to create .Rmd, .html, and .pdf files of the manuscript. In addition, there are also some automated checks implemented to assess if the submission follows the required R Journal format

- ☐ `initial_check_article()`
- ☐ `check_filenames()`
- ☐ `check_unnecessary_files()`
- ☐ `check_cover_letter()`
- ☐ `check_title()`
- ☐ `check_section()`
- ☐ `check_spelling()`
- ☐ `check_proposed_pkg()`
- ☐ `check_packages_available()`



Publication is not the end of the whole process...

A published paper and a CRAN release represent only **one stage in the software life cycle**.

- ❑ **Maintain the package** – bug fixes, compatibility with new R versions, dependency updates, CRAN checks.
- ❑ **Maintain communication channels** – GitHub issues, e-mail, package website, NEWS/changelog.
- ❑ **Listen to users** – reported bugs and feature requests often reveal important practical limitations.
- ❑ **Extend the package** – new methods, additional datasets, faster algorithms, additional S3 methods and diagnostics.
- ❑ **Keep disseminating** – new applications, conference talks, workshops, teaching materials, collaborations.

Publication is not the end of the whole process...

A published paper and a CRAN release represent only **one stage in the software life cycle**.

- ❑ **Maintain the package** – bug fixes, compatibility with new R versions, dependency updates, CRAN checks.
- ❑ **Maintain communication channels** – GitHub issues, e-mail, package website, NEWS/changelog.
- ❑ **Listen to users** – reported bugs and feature requests often reveal important practical limitations.
- ❑ **Extend the package** – new methods, additional datasets, faster algorithms, additional S3 methods and diagnostics.
- ❑ **Keep disseminating** – new applications, conference talks, workshops, teaching materials, collaborations.

release → users → feedback → improvement → new release

Want an R package logo? Also doable! (hexSticker)

- ☐ Well, like it or not...
 - ☐ ... but nobody will make you to have one!
-

Want an R package logo? Also doable! (hexSticker)

- ❑ Well, like it or not...
- ❑ ... but nobody will make you to have one!



Not particularly that nice, but it is working...
Or you can way do way better by yourself...

<https://swag.posit.co>



A business plan for future?

(or the next R package?)

A business plan for future?

(or the next R package?)



ProfileLadder package for



ProfileLadder package for



Thank you very much! Any questions?

ProfileLadder package for



Thank you very much! Any questions?

For any comments, questions, or suggestions use matus.maciak@matfyz.cuni.cz