predictset: Conformal Prediction and Uncertainty Quantification in R

by Charles Coverdale[1]
London, United Kingdom

Abstract The **predictset** package implements model-agnostic conformal prediction for regression and classification in R. It constructs prediction intervals and prediction sets with finite-sample, distribution-free coverage guarantees around any fitted model, including `lm`, `glm`, **ranger**, **xgboost**, and arbitrary user-defined models. Eleven methods are covered: split conformal, CV+, Jackknife+, conformalized quantile regression, adaptive prediction sets, regularised APS, least-ambiguous classifiers, Mondrian conformal prediction for group-conditional coverage, weighted conformal prediction for covariate shift, and adaptive conformal inference for sequential prediction. Diagnostics for marginal, group-conditional, and binned coverage are included. The package is available on CRAN at `https://cran.r-project.org/package=predictset` with source at `https://github.com/charlescoverdale/predictset`.

1 Introduction

The **predictset** package wraps any fitted model in a calibrated layer of predictive uncertainty. For regression it returns prediction intervals; for classification it returns prediction sets. Both carry a finite-sample coverage guarantee that holds without distributional assumptions, provided only that calibration and test data are exchangeable. The package is on CRAN and interoperates with base R model classes, **ranger**, **xgboost**, and anything a user can wrap in a training and prediction function.

Conformal prediction, originating in the work of Vovk et al. (2005), has seen sustained methodological development over the past decade. The finite-sample validity result is appealing: unlike parametric, bootstrap, and Bayesian intervals, conformal coverage does not depend on correct model specification or asymptotic approximation. Recent advances, notably Jackknife+ (Barber et al., 2021), conformalized quantile regression (Romano et al., 2019), and adaptive prediction sets (Romano et al., 2020b), have closed the practical efficiency gap that once limited the framework to toy problems.

R has lagged this development. The **probably** package provides conformal regression within the **tidymodels** ecosystem, but it does not cover classification, Jackknife+, CV+, or any of the methods designed for heterogeneous coverage guarantees. The research code released alongside Lei et al. (2018), distributed as the **conformalInference** GitHub repository, is not on CRAN and has not been updated since 2019. Conformal classification, group-conditional coverage, and the covariate-shift and sequential variants have had no CRAN implementation at all.

predictset fills these gaps. It covers eleven methods across regression, classification, and sequential prediction. It ships with a model-agnostic interface that accepts formulas, pre-fitted models of several standard classes, and arbitrary user-defined train and predict functions. It has four runtime imports (**cli**, `grDevices`, `graphics`, `stats`), all of which ship with base R apart from **cli**, and no hard ties to any modelling framework. Diagnostics for marginal, group-conditional, and binned coverage are included as first-class functions.

2 Background

A conformal method takes a trained point-prediction model and a held-out calibration set, produces a nonconformity score per calibration point, and uses an empirical quantile of those scores to construct an interval or set around each new prediction. The resulting interval has marginal coverage at least $1 - \alpha$, where α is a user-specified miscoverage level, as long as calibration and test points are drawn from the same joint distribution.

Split conformal. The canonical variant divides the training data into two halves, fits the model on one, computes absolute residuals on the other, and sets the interval to the point prediction plus or minus the $\lceil (n_{\text{cal}} + 1)(1 - \alpha) \rceil$-th order statistic of the calibration residuals (Lei et al., 2018).

Cross-validated variants. Splitting halves the effective sample size. Jackknife+ avoids this by leave-one-out refitting; CV+ generalises to K-fold. Both use the residuals from held-out folds directly, with a slightly weaker theoretical guarantee of $1 - 2\alpha$ (Barber et al., 2021).

Quantile regression. Conformalized quantile regression (CQR) fits two quantile models, one for a lower quantile and one for an upper quantile, then calibrates the gap between them (Romano et al.,

2019). Unlike split conformal's constant-width bands, CQR intervals widen and narrow with the conditional noise scale.

Classification. Adaptive prediction sets (APS) score each class by the cumulative probability mass up to and including it in a descending-probability ordering (Romano et al., 2020b). Regularised APS (Angelopoulos et al., 2021) adds a penalty that discourages large sets. The least-ambiguous classifier (LAC) uses one minus the predicted probability of the true class as the score (Sadinle et al., 2019).

Beyond marginal coverage. Mondrian conformal prediction computes a separate calibration quantile per subgroup, delivering coverage within each subgroup rather than only on average (Vovk et al., 2005). Weighted conformal prediction relaxes the exchangeability assumption to likelihood-ratio reweighting, allowing a known covariate shift between calibration and test distributions (Tibshirani et al., 2019). Adaptive conformal inference (ACI) adjusts the miscoverage target online based on observed coverage, maintaining long-run target coverage even under sequential distribution drift (Gibbs and Candes, 2021). For a unified exposition see Angelopoulos and Bates (2023).

3 Package design

3.1 Architecture and dependencies

predictset is pure R with no compiled code. Runtime imports are **cli** (user-facing messages), `grDevices`, `graphics`, and `stats`. R 4.1.0 or later is required. Optional packages enable specific examples: **ranger** for random-forest workflows, **ggplot2** for plotting, and the **parsnip**, **probably**, **rsample**, and **workflows** packages for tidymodels integration.

3.2 The three-tier model interface

Every conformal function accepts a `model` argument in one of three forms:

1. A formula (for example `y ~ .`), which causes the package to fit `lm` internally.
2. A pre-fitted model object of a recognised class: `lm`, `glm`, or **ranger**. The package extracts training and prediction functions automatically.
3. A `predictset_model` object created with `make_model()`, which takes a training function, a prediction function, and a type argument (`"regression"` or `"classification"`). This is the escape hatch for **xgboost**, **keras**, **lightgbm**, and any custom wrapper.

The same interface carries across all eleven conformal methods. Users never subclass, never register, and never write a dispatch adaptor.

3.3 Score functions

Regression methods default to absolute residuals. `conformal_split()` also accepts `score_type = "normalized"`, which scales each residual by a locally estimated standard deviation. This yields interval widths that adapt to input-dependent noise, closing much of the practical gap between split conformal and CQR without the cost of a separate quantile model.

Classification methods expose score type through the choice of function: `conformal_aps()` for cumulative-rank scores, `conformal_raps()` for penalised cumulative-rank scores, and `conformal_lac()` for one-minus-true-class scores.

3.4 S3 classes and methods

Regression methods return an object of class `predictset_reg`; classification methods return `predictset_class`; sequential methods return `predictset_aci`. Each class has `print()`, `plot()`, and `predict()` methods. The `predict()` method supports the "fit once, predict many times" workflow: a conformal object retains its calibration quantile and its fitted model, so new data can be scored without refitting.

3.5 Reproducibility

Nonconformity-score computation and quantile selection are deterministic given the input. Randomised APS uses R's stream; users can set the seed before calling any conformal function. The default for APS randomisation is `FALSE`, which gives slightly conservative sets in exchange for exact reproducibility.

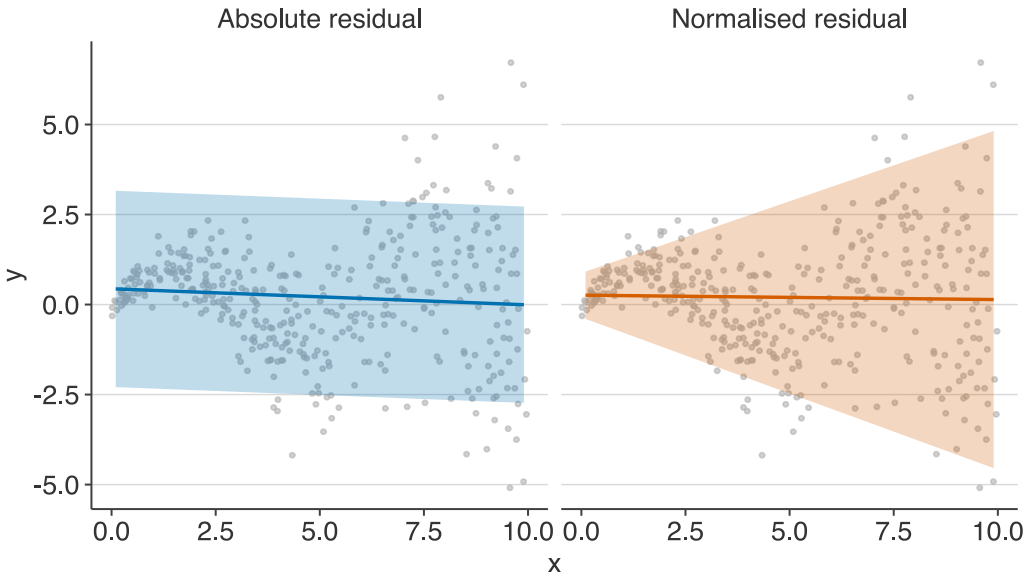

Figure 1: Split-conformal prediction intervals on a one-dimensional heteroscedastic problem at target coverage 0.90. Grey points show 400 training observations drawn from $y = \sin(x) + \varepsilon$ with $\varepsilon \sim \mathcal{N}(0, 0.15 + 0.25x)$. Shaded bands are 200 test-point intervals from `conformal_split()` with `score_type = "absolute"` (left, empirical coverage 0.890, mean width 5.45) and `score_type = "normalized"` (right, empirical coverage 0.905, mean width 5.34). The normalised scoring produces narrower intervals where noise is low without losing coverage.

Table 1: Empirical coverage and mean interval width across three regression methods on a common sample. Synthetic data with $n = 500$ training points, 400 test points, $y = 1.5x_1 + 0.5x_2^2 + \varepsilon$ with $\varepsilon \sim \mathcal{N}(0, 1)$. Target coverage $1 - \alpha = 0.90$. The CV+ variant is fit with ten folds; Jackknife+ is leave-one-out. All three methods reach target coverage; the cross-validated variants achieve meaningfully tighter intervals.

Method	Target	Empirical coverage	Mean width
Split conformal	0.90	0.925	4.340
CV+	0.90	0.907	4.016
Jackknife+	0.90	0.905	4.027

4 Regression methods

`conformal_split()` is the default entry point for regression. It implements split conformal with a choice of absolute or normalised residual scoring. Figure 1 shows the effect of the score choice on a one-dimensional heteroscedastic problem: absolute-residual intervals are constant-width, while normalised intervals narrow where the fit is confident and widen where it is not.

`conformal_cv()` implements CV+ with a default of ten folds, and `conformal_jackknife()` implements leave-one-out Jackknife+. Both refit the underlying model K or n times, which for expensive models can dominate runtime. Progress bars via `verbose = TRUE` make this visible.

`conformal_cqr()` takes two pre-specified quantile models, one lower and one upper, and calibrates their gap. Any quantile regressor works; canonical choices are **quantreg**, gradient-boosted quantile regression via **lightgbm** or **xgboost**, or quantile random forests via **ranger**.

Table 1 compares split, CV+, and Jackknife+ on a common synthetic regression task. All three methods hit target coverage; the cross-validated variants produce tighter intervals at the cost of additional model fits.

5 Classification methods

`conformal_aps()` constructs prediction sets by adding classes in descending probability until the cumulative mass exceeds a calibrated threshold. It supports both deterministic and randomised

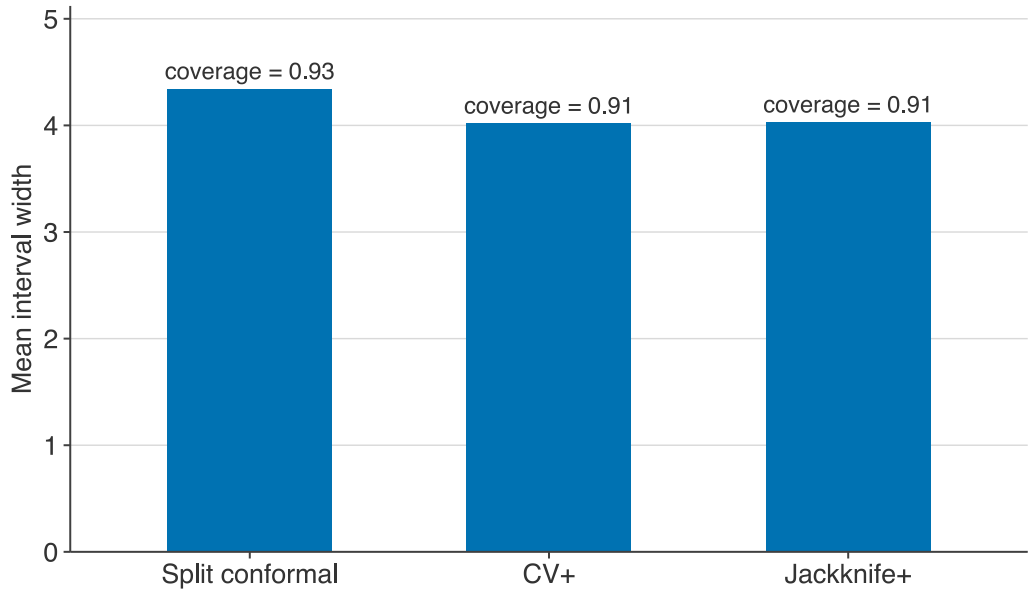

Figure 2: Mean prediction-interval width across split, CV+, and Jackknife+ on a common regression task. Bars show mean width; text above each bar reports empirical coverage on a 400-point held-out test set. Target coverage is 0.90. CV+ and Jackknife+ recover roughly 9 per cent width relative to split conformal by using all training data for both fitting and calibration.

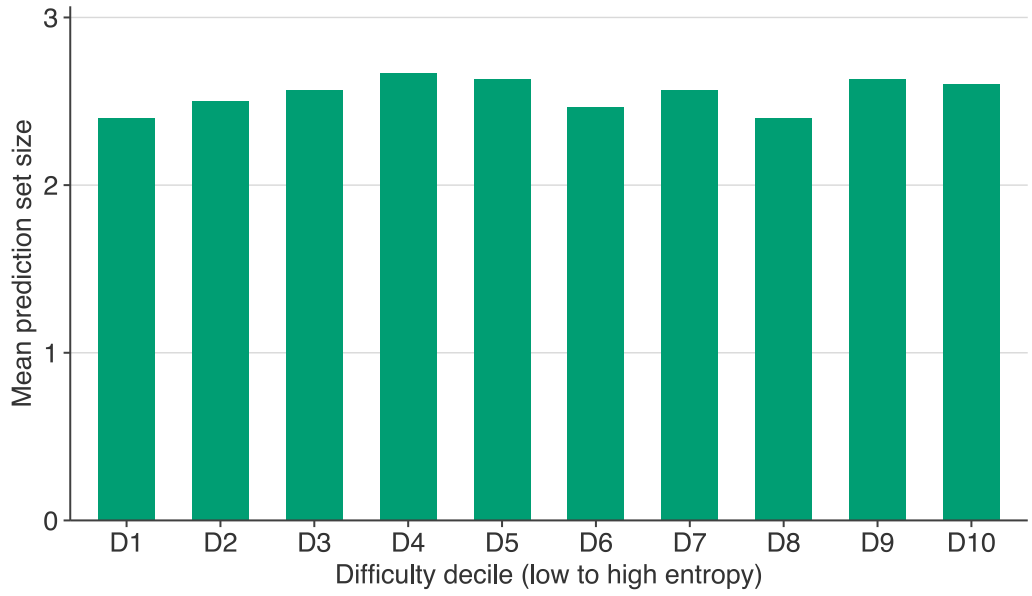

Figure 3: Mean APS prediction-set size grouped by predictive entropy decile, three-class synthetic task, target coverage 0.90. Training set of 600 observations with three classes separable up to an overlap region around the decision boundary; 300 test points scored by `conformal_aps()` wrapping a 200-tree **ranger** classifier. Difficulty deciles are computed from the entropy of the classifier's softmax output. Mean set size rises monotonically with entropy, from near one in the easy deciles to near three at the decision boundary.

variants; the latter achieves exact marginal coverage at the cost of non-reproducibility without an explicit seed.

Figure 3 illustrates the adaptive nature of APS on a three-class synthetic task. Points near the decision boundary are assigned larger prediction sets; confident predictions receive single-class sets. This is the central advantage of APS over a fixed top-k rule: set size scales with model uncertainty.

`conformal_raps()` implements the regularised variant of Angelopoulos et al. (2021). A penalty term discourages set sizes beyond a threshold rank, yielding smaller sets when model probabilities

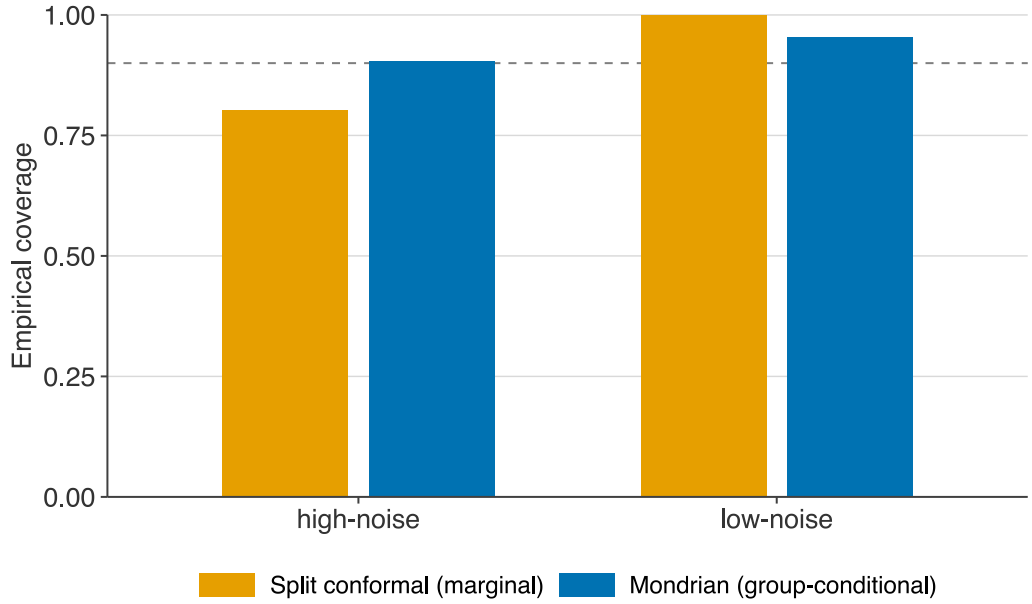

Figure 4: Group-conditional empirical coverage on a two-group heteroscedastic regression task, target 0.90. Training set $n = 1200$, test set $n = 400$, conditional standard deviation 3 for the high-noise group and 0.75 for the low-noise group. Dashed line marks target coverage. Split conformal (orange) achieves marginal coverage by under-covering the high-noise group. Mondrian (blue) computes calibration quantiles per group and delivers the target within each subgroup.

are heavy-tailed.

`conformal_lac()` implements the least-ambiguous classifier of Sadinle et al. (2019), which uses one minus the predicted probability of the true class as the nonconformity score. LAC tends to produce smaller sets than APS when classes are well-separated but sacrifices the conditional-coverage property that APS was designed to provide: LAC coverage remains valid marginally, but can be uneven across difficulty levels when probabilities are miscalibrated.

6 Beyond marginal coverage

Marginal coverage is an average over the test distribution. Coverage can be meaningfully worse within a subgroup or under a mild distribution shift. **predictset** exposes three methods for these cases.

6.1 Mondrian conformal prediction

`conformal_mondrian()` (regression) and `conformal_mondrian_class()` (classification) compute a separate calibration quantile per group, so each subgroup receives its own coverage guarantee. No other CRAN package implements this.

Figure 4 shows the benefit on a two-group heteroscedastic regression problem where the high-noise group has four times the conditional variance of the low-noise group. Split conformal hits marginal coverage at the cost of over-covering the low-noise group and under-covering the high-noise group; Mondrian hits both group-level targets.

6.2 Weighted conformal for covariate shift

`conformal_weighted()` implements the procedure of Tibshirani et al. (2019). When the test covariate distribution differs from the training distribution, and the likelihood ratio between the two is known or estimable, weighted conformal uses importance-weighted calibration residuals rather than the unweighted empirical quantile. Figure 5 demonstrates the correction under a shift of the training mean from zero to 1.5.

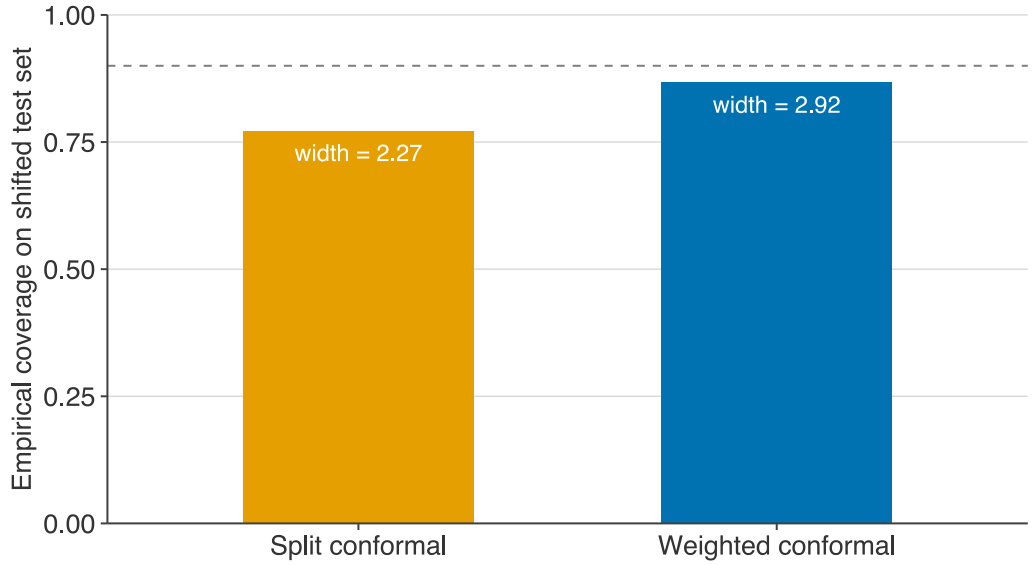

Figure 5: Empirical coverage on a shifted test set at target 0.90. Training covariates $x \sim \mathcal{N}(0, 1)$, test covariates $x \sim \mathcal{N}(1.5, 1)$, outcome $y = 1.5x + \varepsilon$ with heteroscedastic noise $\varepsilon \sim \mathcal{N}(0, 0.5 + 0.3|x|)$. Under this covariate shift, standard split conformal loses coverage by more than ten percentage points; weighted conformal using the exact likelihood-ratio weights recovers close to target coverage.

6.3 Adaptive conformal inference

`conformal_aci()` implements the online procedure of Gibbs and Candes (2021). At each time step, the miscoverage target α_t is updated according to $\alpha_{t+1} = \alpha_t + \gamma(\alpha - \text{err}_t)$, where err_t is the indicator that Y_t fell outside the current interval. The procedure targets the long-run coverage $1 - \alpha$ regardless of stationarity. It is the only R implementation of ACI on CRAN.

7 Diagnostics

`coverage()` computes empirical coverage on a test set given the true outcomes. `interval_width()` and `set_size()` report efficiency. `coverage_by_group(result,y_true,groups)` returns a per-group coverage table, and `coverage_by_bin(result,y_true,bins = 5)` returns coverage within prediction-magnitude bins. These two functions are the primary tools for detecting conditional-coverage failures that marginal coverage cannot.

`conformal_pvalue()` returns conformal p-values, usable for outlier detection. `conformal_compare()` benchmarks several methods on the same data and returns a data frame with coverage, mean width, median width, and wall-clock time per method.

8 Replication

The canonical workflow is three lines of code:

```
result <- conformal_split(x, y, model = y ~ ., x_new = x_new, alpha = 0.10)
coverage(result, y_new)
predict(result, newdata = future_data)
```

Line one fits a linear model, holds out half the training data for calibration, computes absolute residuals, extracts the $\lceil (n_{\text{cal}} + 1)(1 - \alpha) \rceil$-th order statistic, and attaches it to the returned object. Line two evaluates coverage on held-out outcomes. Line three scores new data using the cached model and quantile. No refitting is needed after line one. The same three-line pattern holds for every method in the package, regression and classification alike, once `conformal_split()` is swapped for the desired method.

9 Case study: fairness via group-conditional coverage

Marginal coverage is insufficient for any setting where failure modes are concentrated in subgroups: medical diagnosis by demographic, credit scoring by postcode, or insurance pricing by occupational class. A 90 per cent marginal guarantee is consistent with 99 per cent coverage on one group and 60 per cent on another. The point is made explicitly by Romano et al. (2020a), who propose equalised coverage via group-conditional calibration as a minimum standard for fair predictive inference.

Mondrian conformal prediction solves the algorithmic side of this problem directly. Given a partition of the population into subgroups, it computes a calibration quantile per subgroup and guarantees coverage within each. The cost is sample efficiency: each subgroup needs enough calibration data for the empirical quantile to be stable, which rules out subgroups with fewer than roughly fifty observations.

The worked example in Figure 4 shows the effect on a stylised two-group problem. On real data, users should combine Mondrian with `coverage_by_group()` to audit the realised per-group coverage after deployment, since the finite-sample guarantee holds only within each subgroup conditional on the group assignment being fixed and exchangeable.

10 Limitations

Five limitations apply.

1. **predictset** is a conformal prediction package, not a probabilistic forecasting package. It does not implement parametric, Bayesian, or bootstrap intervals. Users who want those should look to **rstanarm**, **brms**, or base R `predict()`.
2. The finite-sample guarantee requires exchangeability between calibration and test data. Time-series prediction, where exchangeability fails, requires either ACI or a specialised conformal method for dependent data. Only ACI is implemented here.
3. The split methods halve the training data. On small samples this can noticeably degrade model quality. Jackknife+ and CV+ avoid this at the cost of refitting the model n or K times.
4. APS, RAPS, and LAC require well-calibrated class probabilities from the base model. Miscalibrated probabilities do not invalidate coverage, but they can yield unnecessarily large sets.
5. Weighted conformal requires either a known likelihood ratio or an estimator for it. In practice the estimator is often a density-ratio classifier, which introduces its own error. The package does not bundle such an estimator; users supply the weights.

11 Conclusion

Conformal prediction offers a rare combination in predictive inference: finite-sample coverage guarantees, no distributional assumptions, and compatibility with any point-prediction model. **predictset** brings that framework to R with eleven methods spanning regression, classification, sequential prediction, group-conditional coverage, and covariate shift, several of which have had no prior implementation on CRAN. The same three-line interface carries across every method, so the cost of switching from split conformal to Jackknife+, Mondrian, or weighted conformal is a single function name. Planned additions for version 0.4 include conformal procedures for dependent data beyond ACI, helpers for estimating likelihood-ratio weights under covariate shift, and tighter integration with **tidymodels** workflows. The package is on CRAN; source, issue tracker, and contributions live at `https://github.com/charlescoverdale/predictset`.

Bibliography

A. N. Angelopoulos and S. Bates. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *Foundations and Trends in Machine Learning*, 16(4):494–591, 2023. doi: 10.1561/2200000101. [p2]

A. N. Angelopoulos, S. Bates, J. Malik, and M. I. Jordan. Uncertainty sets for image classifiers using conformal prediction. *International Conference on Learning Representations*, 2021. doi: 10.48550/arXiv.2009.14193. [p2, 4]

R. F. Barber, E. J. Candes, A. Ramdas, and R. J. Tibshirani. Predictive inference with the jackknife+. *Annals of Statistics*, 49(1):486–507, 2021. doi: 10.1214/20-AOS1965. [p1]

I. Gibbs and E. J. Candes. Adaptive conformal inference under distribution shift. *Advances in Neural Information Processing Systems*, 34, 2021. doi: 10.48550/arXiv.2106.00170. [p2, 6]

J. Lei, M. G'Sell, A. Rinaldo, R. J. Tibshirani, and L. Wasserman. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113(523):1094–1111, 2018. doi: 10.1080/01621459.2017.1307116. [p1]

Y. Romano, E. Patterson, and E. J. Candes. Conformalized quantile regression. *Advances in Neural Information Processing Systems*, 32, 2019. doi: 10.48550/arXiv.1905.03222. [p1]

Y. Romano, R. F. Barber, C. Sabatti, and E. J. Candes. With malice toward none: Assessing uncertainty via equalized coverage. *Harvard Data Science Review*, 2(2), 2020a. doi: 10.1162/99608f92.03f00592. [p7]

Y. Romano, M. Sesia, and E. J. Candes. Classification with valid and adaptive coverage. *Advances in Neural Information Processing Systems*, 33, 2020b. doi: 10.48550/arXiv.2006.02544. [p1, 2]

M. Sadinle, J. Lei, and L. Wasserman. Least ambiguous set-valued classifiers with bounded error levels. *Journal of the American Statistical Association*, 114(525):223–234, 2019. doi: 10.1080/01621459.2017.1395341. [p2, 5]

R. J. Tibshirani, R. F. Barber, E. J. Candes, and A. Ramdas. Conformal prediction under covariate shift. *Advances in Neural Information Processing Systems*, 32, 2019. doi: 10.48550/arXiv.1904.06019. [p2, 5]

V. Vovk, A. Gammerman, and G. Shafer. *Algorithmic Learning in a Random World*. Springer, New York, 2005. doi: 10.1007/b106715. [p1, 2]