# rsample

rsample contains a set of functions that can create different types of resamples and corresponding classes for their analysis. The goal is to have a modular set of methods that can be used across different R packages for:

* traditional resampling techniques for estimating the sampling distribution of a statistic and
* estimating model performance using a holdout set

The scope of rsample is to provide the basic building blocks for creating and analyzing resamples of a data set but does not include code for modeling or calculating statistics. The “Working with Resample Sets” vignette gives demonstrations of how rsample tools can be used.

To install it, use:

[install.packages](https://www.rdocumentation.org/packages/utils/topics/install.packages)("rsample")

## For the devel version:

[require](https://www.rdocumentation.org/packages/base/topics/library)(devtools)

[install\_github](https://www.rdocumentation.org/packages/devtools/topics/reexports)("tidymodels/rsample")

Note that resampled data sets created by rsample are directly accessible in a resampling object but do not contain much overhead in memory. Since the original data is not modified, R does not make an automatic copy.

For example, creating 50 bootstraps of a data set does not create an object that is 50-fold larger in memory:

> [library](https://www.rdocumentation.org/packages/base/topics/library)(rsample)

> [library](https://www.rdocumentation.org/packages/base/topics/library)(mlbench)

> [library](https://www.rdocumentation.org/packages/base/topics/library)(pryr)

>

> [data](https://www.rdocumentation.org/packages/utils/topics/data)(LetterRecognition)

>

> object\_size(LetterRecognition)

2.64 MB

>

> [set.seed](https://www.rdocumentation.org/packages/base/topics/Random)(35222)

> boots <- [bootstraps](https://tidymodels.github.io/rsample/reference/bootstraps.html)(LetterRecognition, times = 50)

>

> object\_size(boots)

6.69 MB

>

> # Object size per resample

> object\_size(boots)/[nrow](https://www.rdocumentation.org/packages/base/topics/nrow)(boots)

134 kB

>

> # Fold increase is <<< 50

> [as.numeric](https://www.rdocumentation.org/packages/base/topics/numeric)(object\_size(boots)/object\_size(LetterRecognition))

[1] 2.528695

The memory usage for 50 boostrap samples is less than 3-fold more than the original data set.

# Working with rsets

## Introduction

rsample can be used to create objects containing resamples of the original data. This page contains examples of how those objects can be used for data analysis.

For illustration, the attrition data is used. From the help file:

These data are from the IBM Watson Analytics Lab. The website describes the data with “Uncover the factors that lead to employee attrition and explore important questions such as ‘show me a breakdown of distance from home by job role and attrition’ or ‘compare average monthly income by education and attrition’. This is a fictional data set created by IBM data scientists.” There are 1470 rows.

The data can be accessed using

[library](https://www.rdocumentation.org/packages/base/topics/library)(rsample)

[data](https://www.rdocumentation.org/packages/utils/topics/data)("attrition")

[names](https://www.rdocumentation.org/packages/base/topics/names)(attrition)

#> [1] "Age" "Attrition" "BusinessTravel"

#> [4] "DailyRate" "Department" "DistanceFromHome"

#> [7] "Education" "EducationField" "EnvironmentSatisfaction"

#> [10] "Gender" "HourlyRate" "JobInvolvement"

#> [13] "JobLevel" "JobRole" "JobSatisfaction"

#> [16] "MaritalStatus" "MonthlyIncome" "MonthlyRate"

#> [19] "NumCompaniesWorked" "OverTime" "PercentSalaryHike"

#> [22] "PerformanceRating" "RelationshipSatisfaction" "StockOptionLevel"

#> [25] "TotalWorkingYears" "TrainingTimesLastYear" "WorkLifeBalance"

#> [28] "YearsAtCompany" "YearsInCurrentRole" "YearsSinceLastPromotion"

#> [31] "YearsWithCurrManager"

[table](https://www.rdocumentation.org/packages/base/topics/table)(attrition$Attrition)

#>

#> No Yes

#> 1233 237

## Model Assessment

Let’s fit a logistic regression model to the data with model terms for the job satisfaction, gender, and monthly income.

If we were fitting the model to the entire data set, we might model attrition using

[glm](https://www.rdocumentation.org/packages/stats/topics/glm)(Attrition ~ JobSatisfaction + Gender + MonthlyIncome, data = attrition, family = binomial)

For convenience, we’ll create a formula object that will be used later:

mod\_form <- [as.formula](https://www.rdocumentation.org/packages/stats/topics/formula)(Attrition ~ JobSatisfaction + Gender + MonthlyIncome)

To evaluate this model, we will use 10 repeats of 10-fold cross-validation and use the 100 holdout samples to evaluate the overall accuracy of the model.

First, let’s make the splits of the data:

[library](https://www.rdocumentation.org/packages/base/topics/library)(rsample)

[set.seed](https://www.rdocumentation.org/packages/base/topics/Random)(4622)

rs\_obj <- [vfold\_cv](https://tidymodels.github.io/rsample/reference/vfold_cv.html)(attrition, v = 10, repeats = 10)

rs\_obj

#> # 10-fold cross-validation repeated 10 times

#> # A tibble: 100 x 3

#> splits id id2

#> <list> <chr> <chr>

#> 1 <split [1.3K/147]> Repeat01 Fold01

#> 2 <split [1.3K/147]> Repeat01 Fold02

#> 3 <split [1.3K/147]> Repeat01 Fold03

#> 4 <split [1.3K/147]> Repeat01 Fold04

#> 5 <split [1.3K/147]> Repeat01 Fold05

#> 6 <split [1.3K/147]> Repeat01 Fold06

#> 7 <split [1.3K/147]> Repeat01 Fold07

#> 8 <split [1.3K/147]> Repeat01 Fold08

#> 9 <split [1.3K/147]> Repeat01 Fold09

#> 10 <split [1.3K/147]> Repeat01 Fold10

#> # ... with 90 more rows

Now let’s write a function that will, for each resample:

1. obtain the analysis data set (i.e. the 90% used for modeling)
2. fit a logistic regression model
3. predict the assessment data (the other 10% not used for the model) using the broom package
4. determine if each sample was predicted correctly.

Here is our function:

## splits will be the `rsplit` object with the 90/10 partition

holdout\_results <- function(splits, ...) {

# Fit the model to the 90%

mod <- [glm](https://www.rdocumentation.org/packages/stats/topics/glm)(..., data = [analysis](https://tidymodels.github.io/rsample/reference/as.data.frame.rsplit.html)(splits), family = binomial)

# Save the 10%

holdout <- [assessment](https://tidymodels.github.io/rsample/reference/as.data.frame.rsplit.html)(splits)

# `augment` will save the predictions with the holdout data set

res <- broom::[augment](https://www.rdocumentation.org/packages/broom/topics/augment)(mod, newdata = holdout)

# Class predictions on the assessment set from class probs

lvls <- [levels](https://www.rdocumentation.org/packages/base/topics/levels)(holdout$Attrition)

predictions <- [factor](https://www.rdocumentation.org/packages/base/topics/factor)([ifelse](https://www.rdocumentation.org/packages/base/topics/ifelse)(res$.fitted > 0, lvls[2], lvls[1]),

levels = lvls)

# Calculate whether the prediction was correct

res$correct <- predictions == holdout$Attrition

# Return the assessment data set with the additional columns

res

}

For example:

example <- holdout\_results(rs\_obj$splits[[1]], mod\_form)

[dim](https://www.rdocumentation.org/packages/base/topics/dim)(example)

#> [1] 147 35

[dim](https://www.rdocumentation.org/packages/base/topics/dim)([assessment](https://tidymodels.github.io/rsample/reference/as.data.frame.rsplit.html)(rs\_obj$splits[[1]]))

#> [1] 147 31

## newly added columns:

example[1:10, [setdiff](https://www.rdocumentation.org/packages/base/topics/sets)([names](https://www.rdocumentation.org/packages/base/topics/names)(example), [names](https://www.rdocumentation.org/packages/base/topics/names)(attrition))]

#> # A tibble: 10 x 4

#> .rownames .fitted .se.fit correct

#> <chr> <dbl> <dbl> <lgl>

#> 1 7 -1.28 0.182 TRUE

#> 2 14 -1.16 0.187 TRUE

#> 3 32 -3.25 0.353 TRUE

#> 4 54 -0.934 0.193 TRUE

#> 5 72 -1.97 0.193 TRUE

#> 6 74 -1.67 0.188 TRUE

#> 7 76 -2.10 0.171 TRUE

#> 8 138 -1.51 0.165 TRUE

#> 9 142 -1.54 0.149 FALSE

#> 10 147 -1.58 0.150 FALSE

For this model, the .fitted value is the linear predictor in log-odds units.

To compute this data set for each of the 100 resamples, we’ll use the map function from the purrr package:

[library](https://www.rdocumentation.org/packages/base/topics/library)(purrr)

rs\_obj$results <- [map](https://www.rdocumentation.org/packages/purrr/topics/map)(rs\_obj$splits,

holdout\_results,

mod\_form)

rs\_obj

#> # 10-fold cross-validation repeated 10 times

#> # A tibble: 100 x 4

#> splits id id2 results

#> <list> <chr> <chr> <list>

#> 1 <split [1.3K/147]> Repeat01 Fold01 <tibble [147 × 35]>

#> 2 <split [1.3K/147]> Repeat01 Fold02 <tibble [147 × 35]>

#> 3 <split [1.3K/147]> Repeat01 Fold03 <tibble [147 × 35]>

#> 4 <split [1.3K/147]> Repeat01 Fold04 <tibble [147 × 35]>

#> 5 <split [1.3K/147]> Repeat01 Fold05 <tibble [147 × 35]>

#> 6 <split [1.3K/147]> Repeat01 Fold06 <tibble [147 × 35]>

#> 7 <split [1.3K/147]> Repeat01 Fold07 <tibble [147 × 35]>

#> 8 <split [1.3K/147]> Repeat01 Fold08 <tibble [147 × 35]>

#> 9 <split [1.3K/147]> Repeat01 Fold09 <tibble [147 × 35]>

#> 10 <split [1.3K/147]> Repeat01 Fold10 <tibble [147 × 35]>

#> # ... with 90 more rows

Now we can compute the accuracy values for all of the assessment data sets:

rs\_obj$accuracy <- [map\_dbl](https://www.rdocumentation.org/packages/purrr/topics/map)(rs\_obj$results, function(x) [mean](https://www.rdocumentation.org/packages/base/topics/mean)(x$correct))

[summary](https://www.rdocumentation.org/packages/base/topics/summary)(rs\_obj$accuracy)

#> Min. 1st Qu. Median Mean 3rd Qu. Max.

#> 0.748 0.816 0.837 0.839 0.864 0.898

Keep in mind that the baseline accuracy to beat is the rate of non-attrition, which is 0.839. Not a great model so far.

## Using the Bootstrap to Make Comparisons

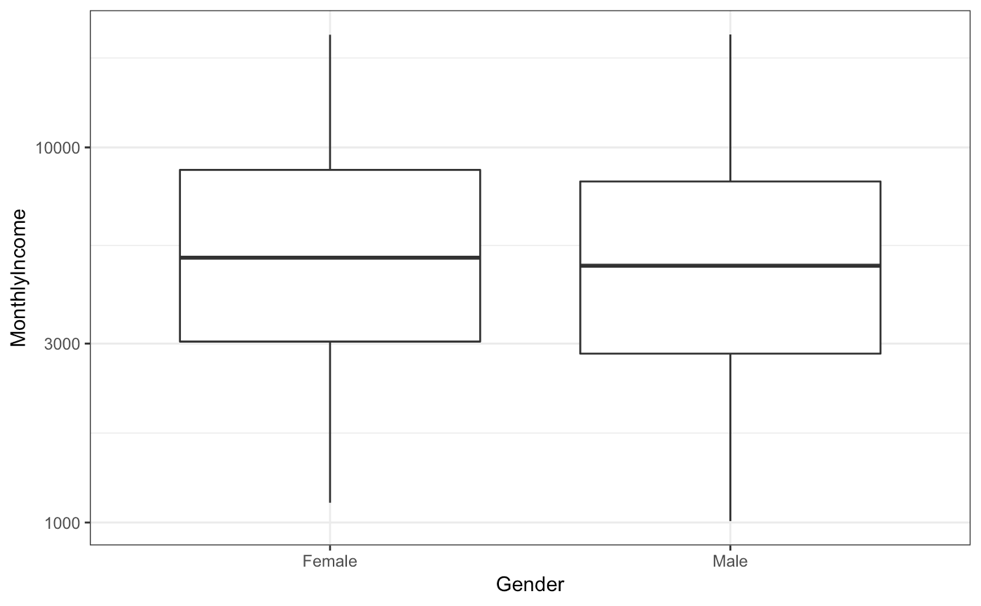
Traditionally, the bootstrap has been primarily used to empirically determine the sampling distribution of a test statistic. Given a set of samples with replacement, a statistic can be calculated on each analysis set and the results can be used to make inferences (such as confidence intervals).

For example, are there differences in the median monthly income between genders?

ggplot(attrition, aes(x = Gender, y = MonthlyIncome)) +

geom\_boxplot() +

scale\_y\_log10()



If we wanted to compare the genders, we could conduct a t-test or rank-based test. Instead, let’s use the bootstrap to see if there is a difference in the median incomes for the two groups. We need a simple function to compute this statistic on the resample:

median\_diff <- function(splits) {

x <- [analysis](https://tidymodels.github.io/rsample/reference/as.data.frame.rsplit.html)(splits)

[median](https://www.rdocumentation.org/packages/stats/topics/median)(x$MonthlyIncome[x$Gender == "Female"]) -

[median](https://www.rdocumentation.org/packages/stats/topics/median)(x$MonthlyIncome[x$Gender == "Male"])

}

Now we would create a large number of bootstrap samples (say 2000+). For illustration, we’ll only do 500 in this document.

[set.seed](https://www.rdocumentation.org/packages/base/topics/Random)(353)

bt\_resamples <- [bootstraps](https://tidymodels.github.io/rsample/reference/bootstraps.html)(attrition, times = 500)

This function is then computed across each resample:

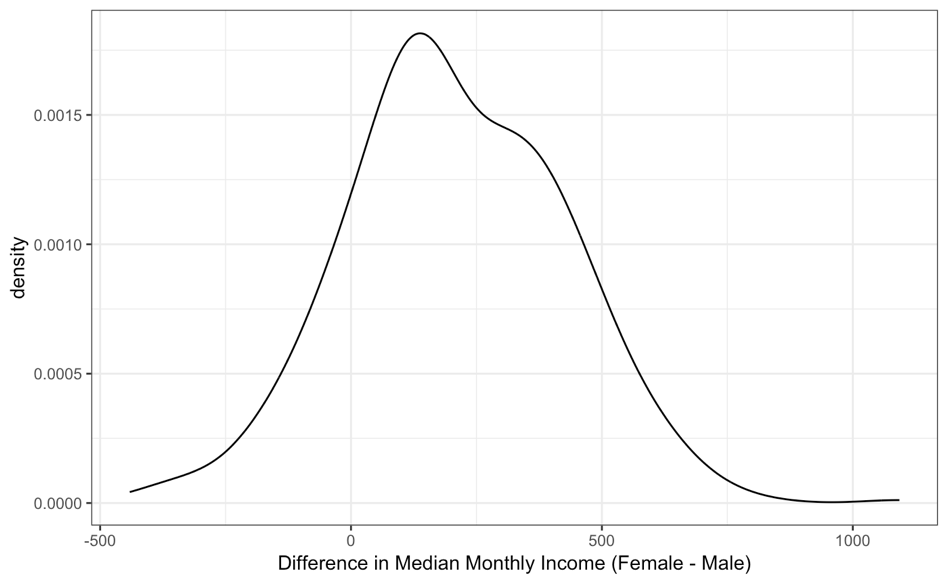
bt\_resamples$wage\_diff <- [map\_dbl](https://www.rdocumentation.org/packages/purrr/topics/map)(bt\_resamples$splits, median\_diff)

The bootstrap distribution of this statistic has a slightly bimodal and skewed distribution:

ggplot(bt\_resamples, aes(x = wage\_diff)) +

geom\_line(stat = "density", adjust = 1.25) +

xlab("Difference in Median Monthly Income (Female - Male)")



The variation is considerable in this statistic. One method of computing a confidence interval is to take the percentiles of the bootstrap distribution. A 95% confidence interval for the difference in the means would be:

[quantile](https://www.rdocumentation.org/packages/stats/topics/quantile)(bt\_resamples$wage\_diff,

probs = [c](https://www.rdocumentation.org/packages/base/topics/c)(0.025, 0.500, 0.975))

#> 2.5% 50% 97.5%

#> -207 190 618

On average, there is no evidence for a difference in the genders.

## Bootstrap Estimates of Model Coefficients

Unless there is already a column in the resample object that contains the fitted model, a function can be used to fit the model and save all of the model coefficients. The [broom package](https://cran.r-project.org/package=broom) package has a tidy function that will save the coefficients in a data frame. Instead of returning a data frame with a row for each model term, we will save a data frame with a single row and columns for each model term. As before, [purrr::map](https://www.rdocumentation.org/packages/purrr/topics/map) can be used to estimate and save these values for each split.

glm\_coefs <- function(splits, ...) {

## use `analysis` or `as.data.frame` to get the analysis data

mod <- [glm](https://www.rdocumentation.org/packages/stats/topics/glm)(..., data = [analysis](https://tidymodels.github.io/rsample/reference/as.data.frame.rsplit.html)(splits), family = binomial)

[as.data.frame](https://www.rdocumentation.org/packages/base/topics/as.data.frame)([t](https://www.rdocumentation.org/packages/base/topics/t)([coef](https://www.rdocumentation.org/packages/stats/topics/coef)(mod)))

}

bt\_resamples$betas <- [map](https://www.rdocumentation.org/packages/purrr/topics/map)(.x = bt\_resamples$splits,

.f = glm\_coefs,

mod\_form)

bt\_resamples

#> # Bootstrap sampling

#> # A tibble: 500 x 4

#> splits id wage\_diff betas

#> <list> <chr> <dbl> <list>

#> 1 <split [1.5K/545]> Bootstrap001 80 <data.frame [1 × 6]>

#> 2 <split [1.5K/540]> Bootstrap002 68.5 <data.frame [1 × 6]>

#> 3 <split [1.5K/535]> Bootstrap003 20 <data.frame [1 × 6]>

#> 4 <split [1.5K/534]> Bootstrap004 177 <data.frame [1 × 6]>

#> 5 <split [1.5K/547]> Bootstrap005 18 <data.frame [1 × 6]>

#> 6 <split [1.5K/522]> Bootstrap006 594. <data.frame [1 × 6]>

#> 7 <split [1.5K/540]> Bootstrap007 137 <data.frame [1 × 6]>

#> 8 <split [1.5K/526]> Bootstrap008 55 <data.frame [1 × 6]>

#> 9 <split [1.5K/531]> Bootstrap009 32.5 <data.frame [1 × 6]>

#> 10 <split [1.5K/546]> Bootstrap010 451 <data.frame [1 × 6]>

#> # ... with 490 more rows

bt\_resamples$betas[[1]]

#> (Intercept) JobSatisfaction.L JobSatisfaction.Q JobSatisfaction.C GenderMale

#> 1 -0.828 -0.297 0.0979 -0.259 0.175

#> MonthlyIncome

#> 1 -0.000135

## Keeping Tidy

As previously mentioned, the [broom package](https://cran.r-project.org/package=broom) contains a class called tidy that created representations of objects that can be easily used for analysis, plotting, etc. rsample contains tidy methods for rset and rsplit objects. For example:

first\_resample <- bt\_resamples$splits[[1]]

[class](https://www.rdocumentation.org/packages/base/topics/class)(first\_resample)

#> [1] "rsplit" "boot\_split"

[tidy](https://www.rdocumentation.org/packages/generics/topics/tidy)(first\_resample)

#> # A tibble: 1,470 x 2

#> Row Data

#> <int> <chr>

#> 1 1 Analysis

#> 2 4 Analysis

#> 3 5 Analysis

#> 4 9 Analysis

#> 5 14 Analysis

#> 6 15 Analysis

#> 7 16 Analysis

#> 8 22 Analysis

#> 9 23 Analysis

#> 10 24 Analysis

#> # ... with 1,460 more rows

and

[class](https://www.rdocumentation.org/packages/base/topics/class)(bt\_resamples)

#> [1] "bootstraps" "rset" "tbl\_df" "tbl" "data.frame"

[tidy](https://www.rdocumentation.org/packages/generics/topics/tidy)(bt\_resamples)

#> # A tibble: 735,000 x 3

#> Row Data Resample

#> <int> <chr> <chr>

#> 1 1 Analysis Bootstrap001

#> 2 1 Analysis Bootstrap003

#> 3 1 Analysis Bootstrap007

#> 4 1 Analysis Bootstrap008

#> 5 1 Analysis Bootstrap010

#> 6 1 Analysis Bootstrap011

#> 7 1 Analysis Bootstrap012

#> 8 1 Analysis Bootstrap013

#> 9 1 Analysis Bootstrap014

#> 10 1 Analysis Bootstrap015

#> # ... with 734,990 more rows