Bootstrapping

AU STAT-427/627

Emil Hvitfeldt

2021-2-23

Bootstrapping

Last week we looked at a couple of different Cross-Validation methods

- Leave-One-Out Cross-Validation (LOOCV)
- K-fold Cross-Validation

Bootstrapping

This week we will look at Bootstrapping

This is a technique that uses resampling with replacement to estimate the uncertainty with a given estimator or statistical learning method

It is a powerful and general statistical tool, and can be used with most estimators/methods

Bootstrapping VS Cross-Validation

- Cross-Validation: provide estimates of the test error.
- Bootstrap: provides the standard error of the estimates.

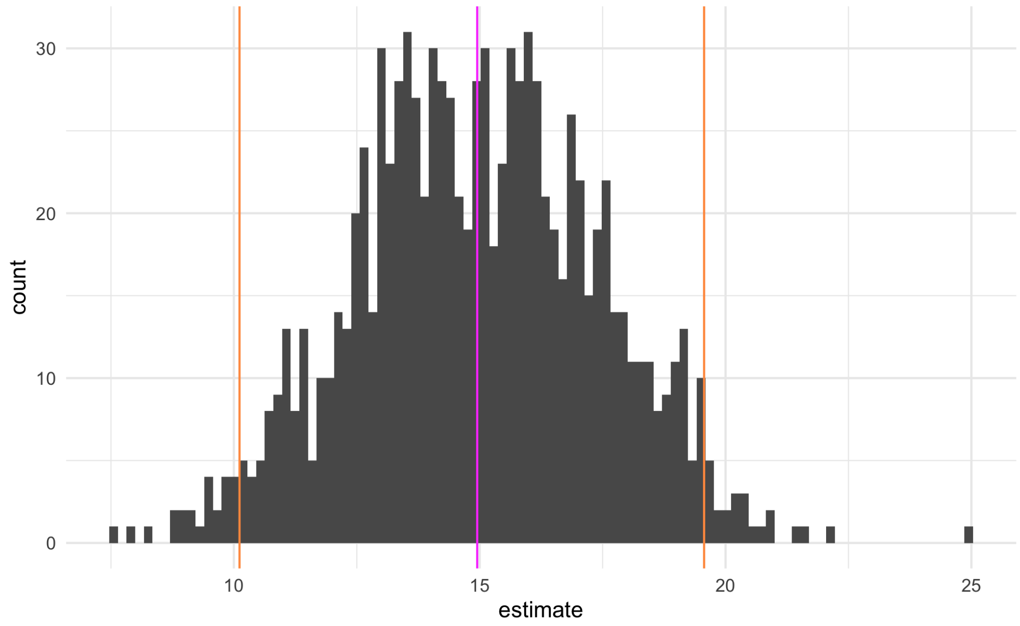
Motivation

Suppose We have an estimate we want to find out how variable it is.

We could collect data n times and calculate the estimates.

We then have a distribution of and can see the how well it is doing

1000 realizations
pink line is the mean
orange lines 95% percent quantiles



Motivation

The Problem

We are not always able to conduct multiple data collections at will

Sometimes for resource issues or time-sensitive data

We need the different samples to come from the same underlying distribution

Motivation

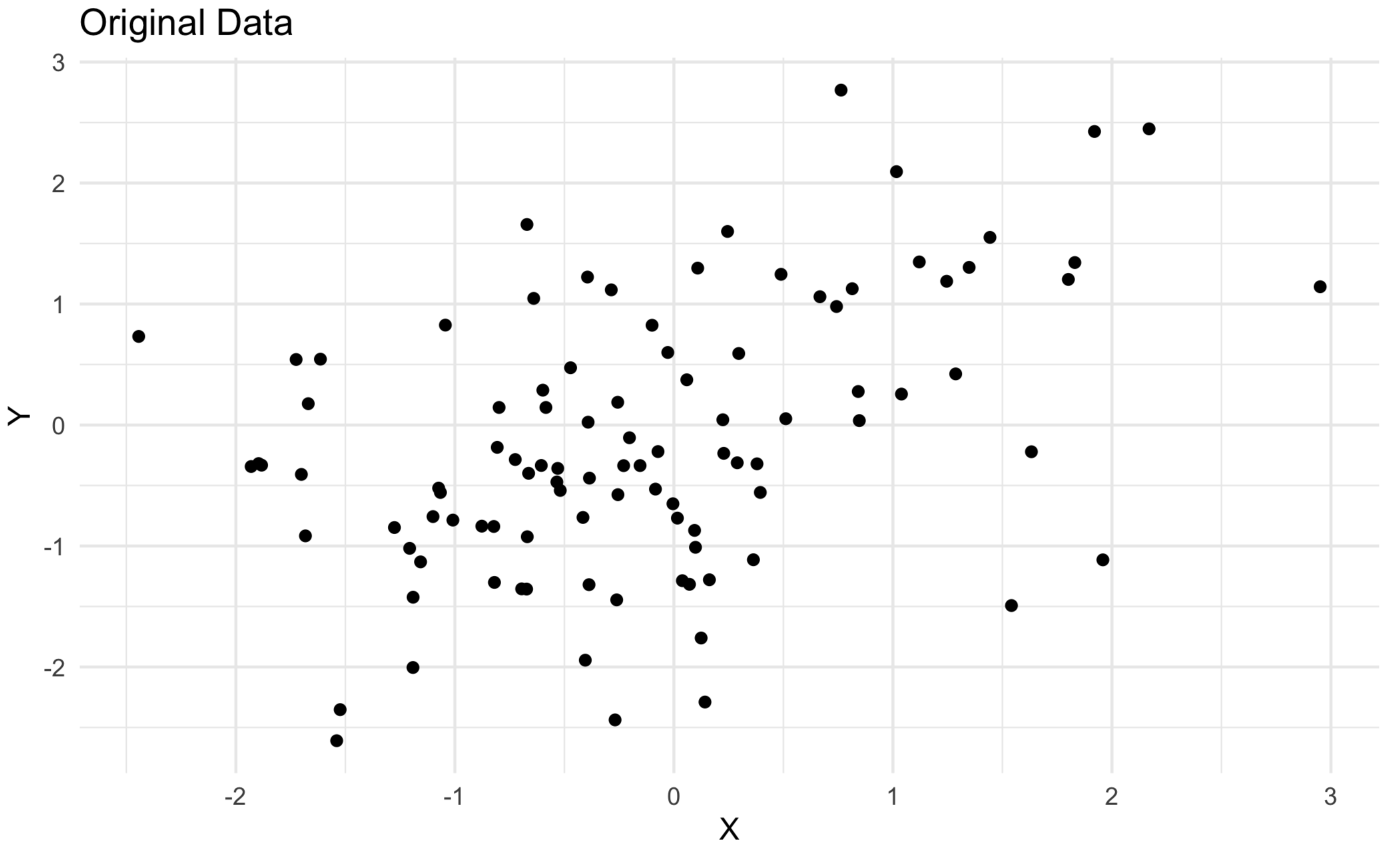
The Solution

We take our one data set and resample the rows with replacement. This allows us to get new data sets that approximate the original data set

If the original data set is close to the underlying true distribution then the resampled data sets are also approximations of the true underlying distribution

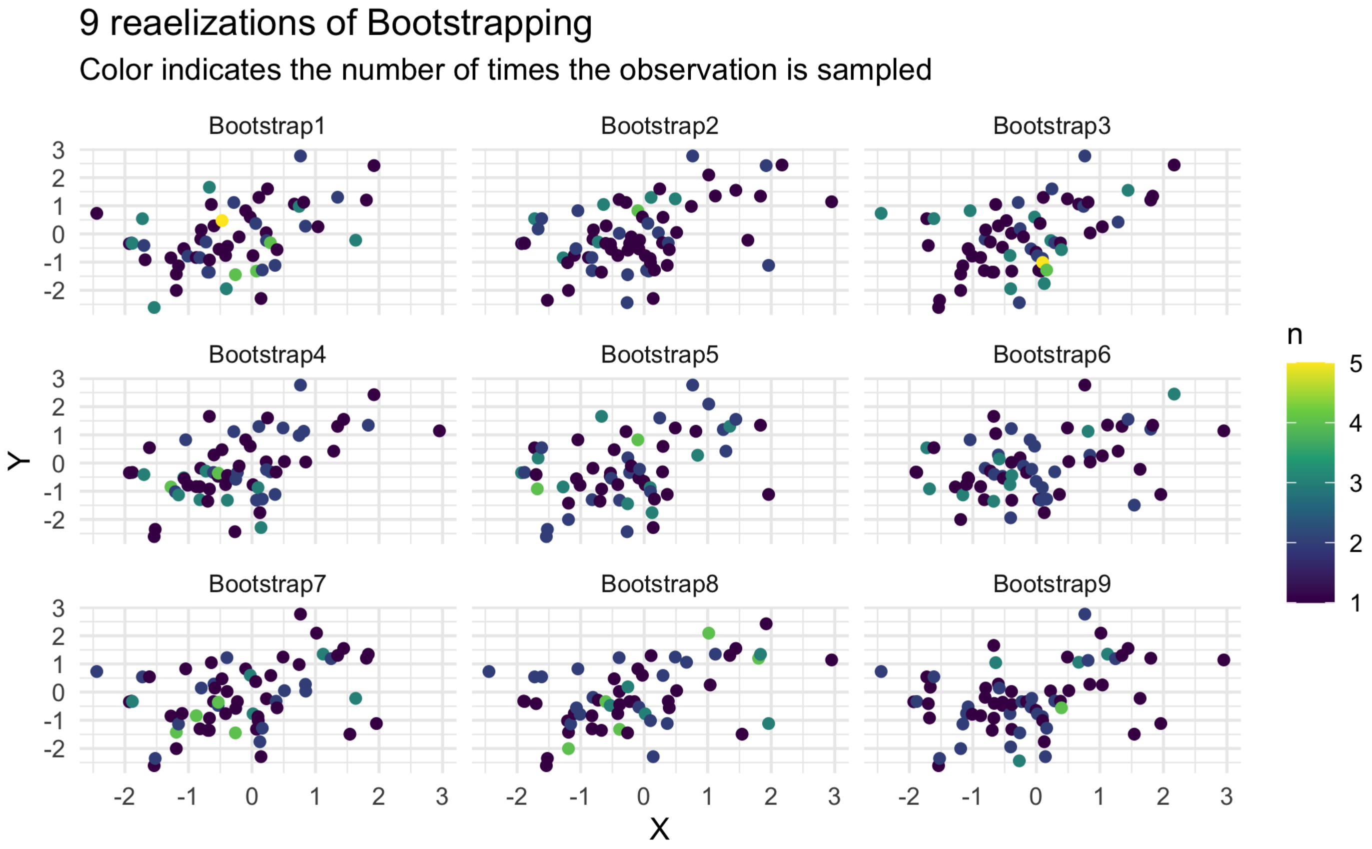
Example

From "An Introduction to Statistical Learning"



Example

Visualizing multiple bootstrappings



Example

We want to minimize

$$\alpha = \frac{\sigma_Y^2 - \sigma_{XY}}{\sigma_X^2 + \sigma_Y^2 - 2\sigma_{XY}}$$

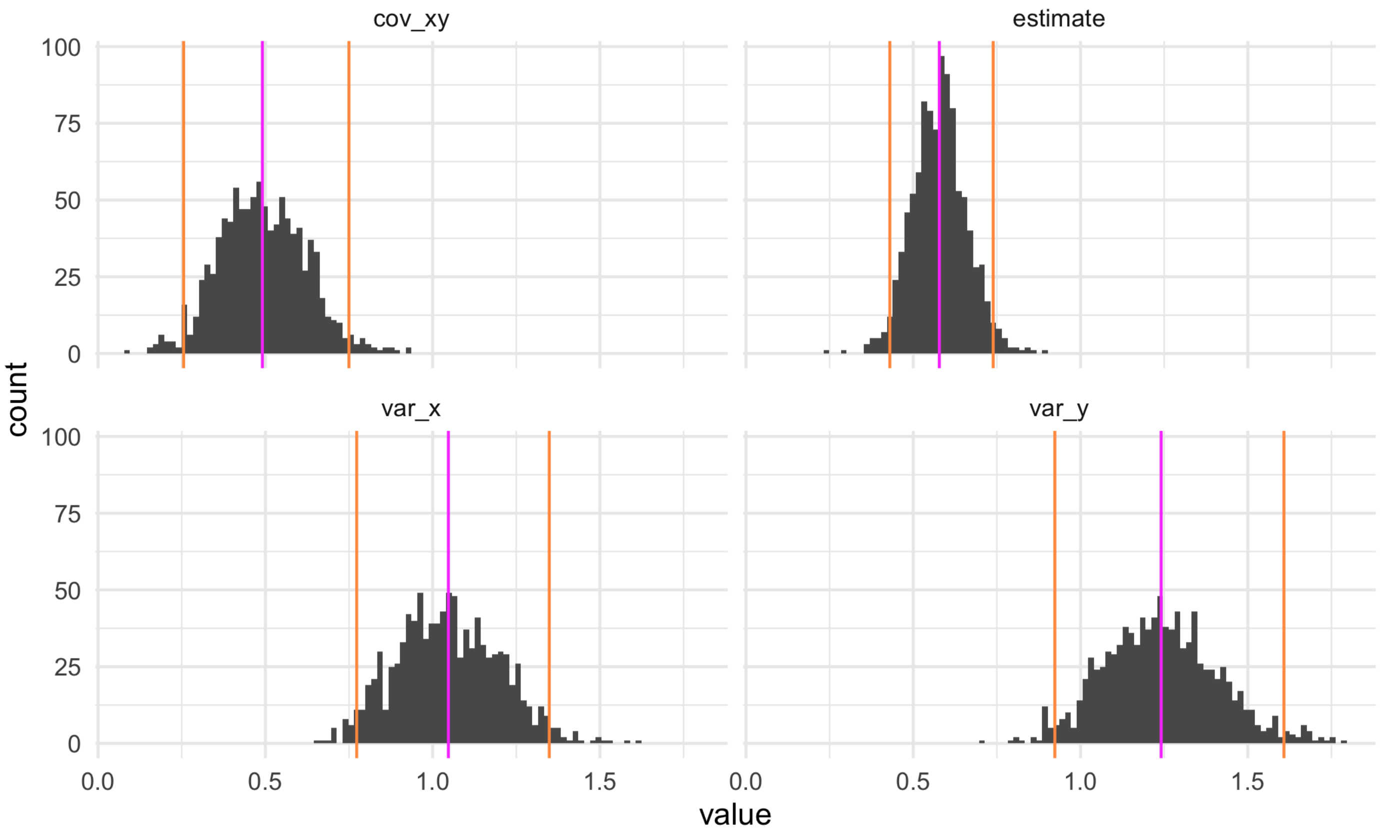
Where $\sigma_X^2 = \mathrm{Var}(X)$, $\sigma_Y^2 = \mathrm{Var}(Y)$, and $\sigma_{XY} = \mathrm{Cov}(X, Y)$

Bootstrapping results

```
## # A tibble: 1,000 x 5
##    id             var_x var_y cov_xy estimate
##  * <chr>          <dbl> <dbl>  <dbl>    <dbl>
##  1 Bootstrap0001 1.04   1.33  0.583    0.618
##  2 Bootstrap0002 0.958  1.21  0.416    0.596
##  3 Bootstrap0003 0.950  1.44  0.479    0.671
##  4 Bootstrap0004 0.909  1.27  0.326    0.617
##  5 Bootstrap0005 1.05   1.24  0.413    0.563
##  6 Bootstrap0006 0.747  1.52  0.386    0.759
##  7 Bootstrap0007 0.899  1.33  0.488    0.673
##  8 Bootstrap0008 0.897  1.43  0.515    0.705
##  9 Bootstrap0009 1.21   1.29  0.531    0.527
## 10 Bootstrap0010 0.879  1.06  0.381    0.576
## # … with 990 more rows
```

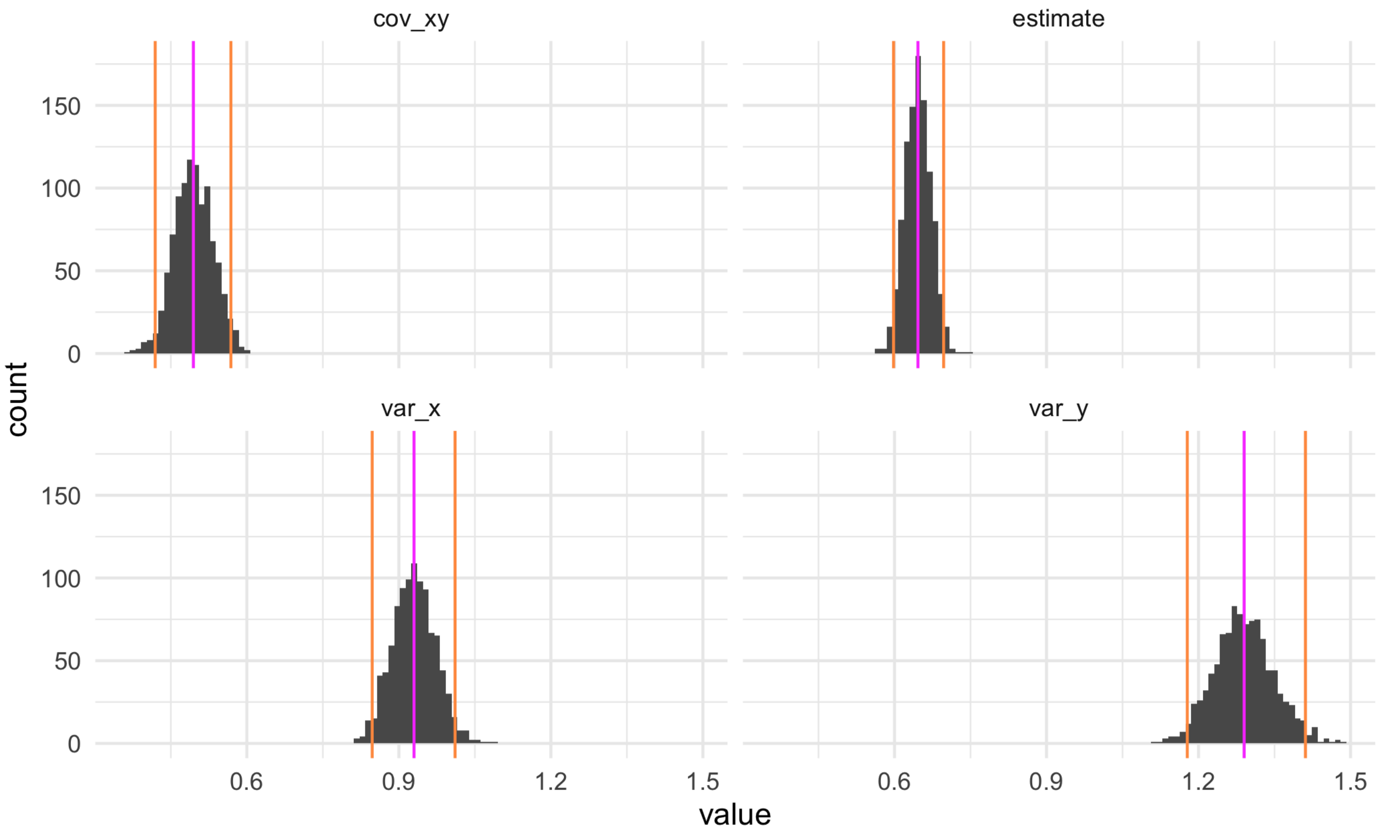
Bootstrapping results

With $n = 100$ in original data set



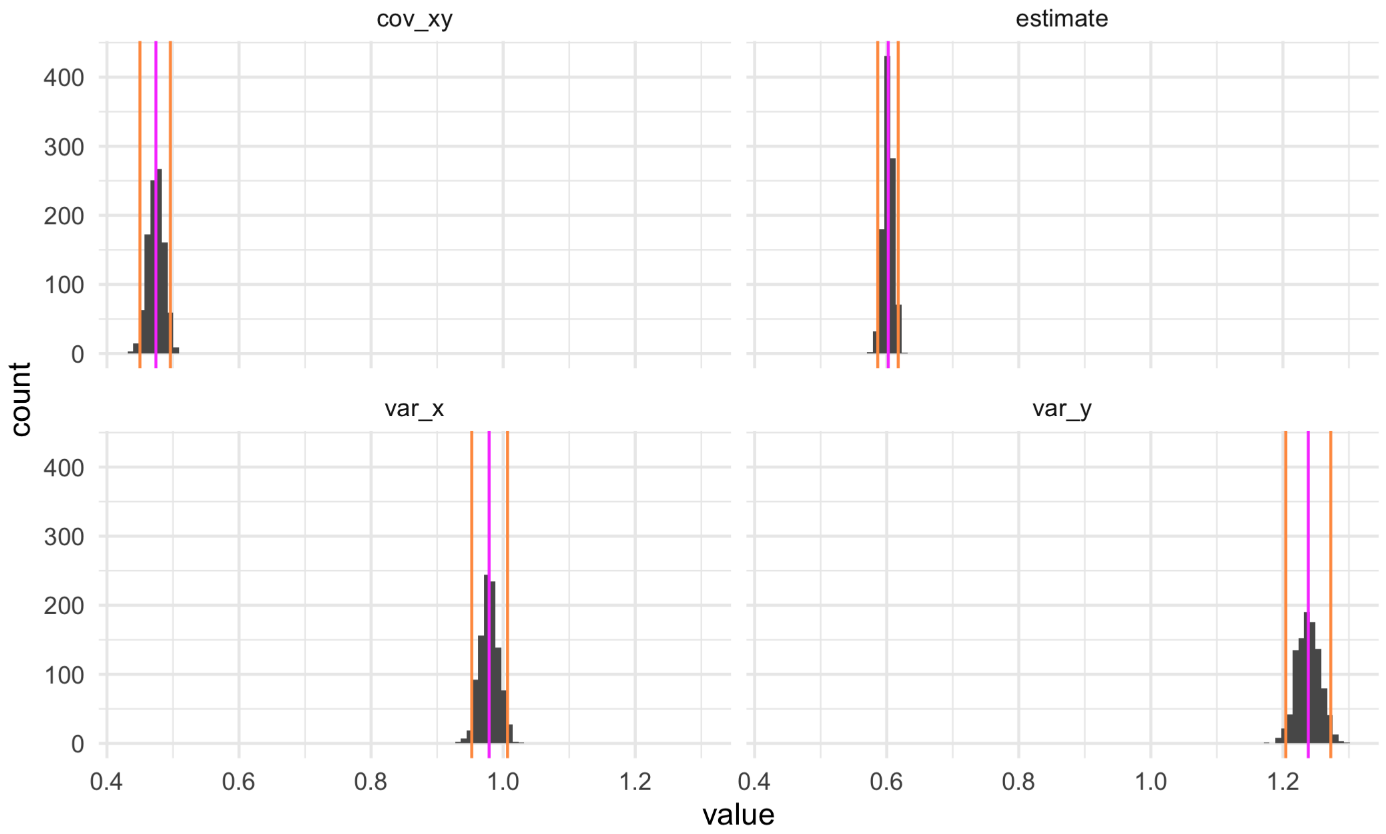
Bootstrapping results

With $n = 1000$ in original data set



Bootstrapping results

With $n = 10000$ in original data set



What size of bootstrappings are we looking for?

We are using bootstrapping sizes to be the same size of to get a comparatively estimate of the variation

Rsample

We are back with `rsample` and the `mtcars` data set

```r
library(rsample)

mtcars
```

```
##                      mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4           21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag       21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710          22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive      21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout   18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## Valiant             18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
## Duster 360          14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
## Merc 240D           24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## Merc 230            22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
## Merc 280            19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
## Merc 280C           17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
## Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
## Merc 450SL          17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
## Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
```

Rsample

We can use the `bootstraps()` function on a data.frame to create a `bootstraps` object

```
mtcars_boots <- bootstraps(mtcars, times = 10
mtcars_boots
```

```
## # Bootstrap sampling
## # A tibble: 100 x 2
##    splits          id
##    <list>          <chr>
##  1 <split [32/12]> Bootstrap001
##  2 <split [32/11]> Bootstrap002
##  3 <split [32/12]> Bootstrap003
##  4 <split [32/9]>  Bootstrap004
##  5 <split [32/10]> Bootstrap005
##  6 <split [32/11]> Bootstrap006
##  7 <split [32/12]> Bootstrap007
##  8 <split [32/11]> Bootstrap008
##  9 <split [32/11]> Bootstrap009
## 10 <split [32/11]> Bootstrap010
## # … with 90 more rows
```

Rsample

An under the hood, we have 100 analysis/assesment splits similar to `initial_split()` and `vfold_cv()`

```
mtcars_boots <- bootstraps(mtcars, times = 10
mtcars_boots$splits
```

```
## [[1]]
## <Analysis/Assess/Total>
## <32/12/32>
##
## [[2]]
## <Analysis/Assess/Total>
## <32/12/32>
##
## [[3]]
## <Analysis/Assess/Total>
## <32/9/32>
##
## [[4]]
## <Analysis/Assess/Total>
## <32/14/32>
##
## [[5]]
```

Using resamples in action

We start by creating a linear regression specification and create a `workflow` object with `workflows()`

```
library(parsnip)
linear_spec <- linear_reg() %>%
  set_mode("regression") %>%
  set_engine("lm")

library(workflows)

linear_wf <- workflow() %>%
  add_model(linear_spec) %>%
  add_formula(mpg ~ disp + hp + wt)
```

Tune

We can use `fit_resamples()` to fit the workflow we created within each bootstrap

```
library(tune)

linear_fold_fits <- fit_resamples(
    linear_wf,
    resamples = mtcars_boots
)
```

Tune

The results of this resampling comes as a data.frame

```
linear_fold_fits
```

```
## # Resampling results
## # Bootstrap sampling
## # A tibble: 100 x 4
##    splits          id           .metrics         .notes
##    <list>          <chr>        <list>           <list>
##  1 <split [32/12]> Bootstrap001 <tibble [2 × 4]> <tibble [0 × 1]>
##  2 <split [32/12]> Bootstrap002 <tibble [2 × 4]> <tibble [0 × 1]>
##  3 <split [32/9]>  Bootstrap003 <tibble [2 × 4]> <tibble [0 × 1]>
##  4 <split [32/14]> Bootstrap004 <tibble [2 × 4]> <tibble [0 × 1]>
##  5 <split [32/16]> Bootstrap005 <tibble [2 × 4]> <tibble [0 × 1]>
##  6 <split [32/13]> Bootstrap006 <tibble [2 × 4]> <tibble [0 × 1]>
##  7 <split [32/15]> Bootstrap007 <tibble [2 × 4]> <tibble [0 × 1]>
##  8 <split [32/12]> Bootstrap008 <tibble [2 × 4]> <tibble [0 × 1]>
##  9 <split [32/14]> Bootstrap009 <tibble [2 × 4]> <tibble [0 × 1]>
## 10 <split [32/11]> Bootstrap010 <tibble [2 × 4]> <tibble [0 × 1]>
## # … with 90 more rows
```

Tune

`collect_metrics()` can be used to extract the CV estimate

```
library(tune)

collect_metrics(linear_fold_fits)
```

```
## # A tibble: 2 x 6
##   .metric .estimator  mean     n std_err .config
##   <chr>   <chr>      <dbl> <int>   <dbl> <chr>
## 1 rmse    standard   2.95    100 0.0633  Preprocessor1_Model1
## 2 rsq     standard   0.828   100 0.00670 Preprocessor1_Model1
```

Tune

Setting `summarize = FALSE` in `collect_metrics()` Allows us the see the individual performance metrics for each fold

```
collect_metrics(linear_fold_fits, summarize = FALSE)
```

```
## # A tibble: 200 x 5
##    id           .metric .estimator .estimate .config
##    <chr>        <chr>   <chr>          <dbl> <chr>
##  1 Bootstrap001 rmse    standard       2.78  Preprocessor1_Model1
##  2 Bootstrap001 rsq     standard       0.938 Preprocessor1_Model1
##  3 Bootstrap002 rmse    standard       3.53  Preprocessor1_Model1
##  4 Bootstrap002 rsq     standard       0.752 Preprocessor1_Model1
##  5 Bootstrap003 rmse    standard       2.49  Preprocessor1_Model1
##  6 Bootstrap003 rsq     standard       0.802 Preprocessor1_Model1
##  7 Bootstrap004 rmse    standard       2.52  Preprocessor1_Model1
##  8 Bootstrap004 rsq     standard       0.811 Preprocessor1_Model1
##  9 Bootstrap005 rmse    standard       2.98  Preprocessor1_Model1
## 10 Bootstrap005 rsq     standard       0.826 Preprocessor1_Model1
## # … with 190 more rows
```