

<> text needs to be removed and replaced by the necessary input! No <> symbols should remain in the code!

Performing a t-test for a Difference in Means

```
t_test(x = <NAME OF DATASET>,  
      response = <NAME OF VARIABLE>,  
      explanatory = <NAME OF VARIABLE>,  
      conf_int = TRUE,  
      conf_level = 0.90,  
      alternative = "two-sided")
```

Note: If you want a 95% confidence interval, you change `conf_level` to 0.95

Note: If you are doing a one-sided hypothesis test, you change `alternative` to either “greater” or “less”

Obtaining 1000 Bootstrap Differences in Means

```
bootstrap <- <NAME OF DATASET> %>%  
  specify(response = <NAME OF NUMERICAL VARIABLE>,  
          explanatory = <NAME OF CATEGORICAL VARIABLE>) %>%  
  generate(reps = 1000, type = "bootstrap") %>%  
  calculate(stat = "diff in means",  
            order = c("<NAME OF FIRST GROUP>", "<NAME OF FIRST GROUP>"))
```

Note: The quotation marks in the `c()` function are important! They need to be there even after you replace the values!

Note: Spelling and capitalization are important. You need to be 100% certain what the names of each group are when you specify them in the `order = step`!

Plotting the Bootstrap Distribution

```
visualize(data = bootstrap,  
          method = "simulation")
```

Note: This step *must* come after you have obtained the bootstrapped differences in means!

<> text needs to be removed and replaced by the necessary input! No <> symbols should remain in the code!

Obtaining the Sample Difference in Means

```
obs_diff_in_means <- <NAME OF DATASET> %>%  
  specify(response = <NAME OF NUMERICAL VARIABLE>,  
    explanatory = <NAME OF CATEGORICAL VARIABLE>) %>%  
  calculate(stat = "diff in means",  
    order = c("<NAME OF FIRST GROUP>", "<NAME OF FIRST GROUP>")  
  )
```

Note: This step *must* be done *before* you find your confidence interval and before finding your p-value!

Obtaining a Confidence Interval from a Bootstrap Distribution

```
get_confidence_interval(x = bootstrap,  
  level = 0.95,  
  type = "percentile",  
  point_estimate = obs_mean)
```

Note: This step *must* come after you have obtained the bootstrapped differences in means *and* the observed difference in means!

Note: If you want a 90% confidence interval, you change `level` to 0.90

Obtaining 1000 Permuted Differences in Means – Assuming the Null Hypothesis is True

```
null_dist <- <NAME OF DATASET> %>%  
  specify(response = <NAME OF NUMERICAL VARIABLE>,  
    explanatory = <NAME OF CATEGORICAL VARIABLE>) %>%  
  generate(reps = 1000, type = "permute") %>%  
  calculate(stat = "diff in means",  
    order = c("<NAME OF FIRST GROUP>", "<NAME OF FIRST GROUP>")  
  )
```

<> text needs to be removed and replaced by the necessary input! No <> symbols should remain in the code!

Plotting the Simulated Null Distribution

```
visualize(data = null_dist,  
          method = "simulation")
```

Note: This step **must** come after you have obtained the permuted differences in means!

Obtaining a p-value from a Null Distribution

```
get_pvalue(x = null_dist,  
           obs_stat = obs_diff_in_means,  
           direction = "two-sided")
```

Note: This step **must** come after you have obtained the bootstrapped differences in means **and** the observed difference in means!

Note: If you are doing a one-sided hypothesis test, you change `alternative` to either “greater” or “less”

Faceted Histograms

```
ggplot(data = <NAME OF DATASET>,  
       mapping = aes(x = <NAME OF NUMERICAL VARIABLE>)) +  
  geom_histogram(binwidth = <WIDTH OF BINS>) +  
  facet_wrap(~<NAME OF CATEGORICAL VARIABLE>) +  
  labs(x = "<TITLE FOR THE X-AXIS>")
```

Note: A histogram **must** have the variable on the x-axis!

Side-by-Side Boxplots

```
ggplot(data = <NAME OF DATASET>,  
       mapping = aes(x = <NAME OF VARIABLE>, y = <NAME OF VARIABLE>)  
       ) +  
  geom_boxplot() +  
  labs(x = "<TITLE FOR THE X-AXIS>",  
       y = "<TITLE FOR THE Y-AXIS>")
```

Note: For **horizontally stacked** boxplots, the categorical variable should be on the **x-axis**. For **vertically stacked** boxplots, the categorical variable should be on the **y-axis**.

<> text needs to be removed and replaced by the necessary input! No <> symbols should remain in the code!

Calculating Summary Statistics for One Numerical Variable and One Categorical Variable

```
favstats(<NAME OF NUMERICAL VARIABLE> ~ <NAME OF CATEGORICAL VARIABLE>,  
        data = <NAME OF DATASET>)
```

Note: The ~ **must** be included!

Obtaining an ANOVA Table

```
aov(<NAME OF NUMERICAL VARIABLE> ~ <NAME OF CATEGORICAL VARIABLE>,  
    data = <NAME OF DATASE>)
```

<> text needs to be removed and replaced by the necessary input! No <> symbols should remain in the code!

Plotting the Bootstrap Distribution

```
visualize(data = bootstrap,  
          method = "simulation")
```

Note: This is *the same* as plotting the bootstrap for one mean!

Obtaining the Sample Slope

```
obs_mean <- <NAME OF DATASET> %>%  
  specify(response = <NAME OF Y-VARIABLE>,  
          explanatory = <NAME OF X-VARIABLE>) %>%  
  calculate(stat = "mean")
```

Note: This step *must* be done *before* you find your confidence interval!

Obtaining a Confidence Interval from a Bootstrap Distribution

```
get_confidence_interval(x = bootstrap,  
                        level = 0.95,  
                        type = "percentile",  
                        point_estimate = obs_mean)
```

Note: This is *the same* as how you found a confidence interval for one mean!