Wherever you see **red characters**, these need to be replaced by your information. This includes the < > symbols!

Performing a t-test for One Mean

```
t_test(x = <NAME OF DATASET>,
       response = <NAME OF VARIABLE>,
       conf_int = TRUE,
       conf_level = 0.90,
       alternative = "two-sided",
       mu = <VALUE OF NULL HYPOTHESIS FOR MU>)
```

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 Means

```
bootstrap <- <NAME OF DATASET> %>%
  specify(response = <NAME OF VARIABLE>) %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "mean")
```

Plotting the Bootstrap Distribution

```
visualize(data = bootstrap,
          method = “simulation”)
```

Note: *This step* ***must*** *come after you have obtained the bootstrap means!*

Obtaining the Sample Mean

```
obs_mean <- <NAME OF DATASET> %>%
  specify(response = <NAME OF 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 step* ***must*** *come after you have obtained the bootstrap means* ***and*** *the observed mean!*
Note: If you want a 90% confidence interval, you change `level` to 0.90

Wherever you see **red characters**, these need to be replaced by your information. This includes the < > symbols!

Scatterplot

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

Scatterplot with Regression Line

```
ggplot(data = <NAME OF DATASET>,
       mapping = aes(x = <NAME OF X-VARIABLE>,
                     y = <NAME OF Y-VARIABLE>)
       ) +
  geom_point() +
  geom_smooth(method = “lm”) +
  labs(x = "<TITLE FOR THE X-AXIS>",
       y = “<TITLE FOR THE Y-AXIS>”)
```

Fitting a Linear Regression

```
model <- lm(<NAME OF Y-VARIABLE> ~ <NAME OF X-VARIABLE>,
                       data = <NAME OF DATASET>)
```

Note: The ~ is necessary! It has to be there!

Obtaining Coefficient Table

```
get_regression_table(model,
                     conf.level = 0.95)
```

Note: You need to have fit the linear regression **before** this step!
Note: If you want a 90% confidence interval, you change `conf.level` to 0.90

Obtaining 1000 Bootstrap Slopes

```
bootstrap <- <NAME OF DATASET> %>%
  specify(response = <NAME OF Y-VARIABLE>,
          explanatory = <NAME OF X-VARIABLE>) %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "slope")
```

Wherever you see **red characters**, these need to be replaced by your information. This includes the < > symbols!

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!*