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

Plotting a One-Variable Bar Plot with Counts

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

Note: This bar plot has the variable names on the x-axis. If the names are squished, then you should use **`y =`** instead of **`x =`** .

Plotting a One-Variable Bar Plot with Proportions

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

Note: This bar plot has the variable names on the x-axis. If the names are squished, then you should use **`y =`** instead of **`x =`** .

Plotting a Two-Variable Bar Plot

```
ggplot(data = <NAME OF DATASET>,
       mapping = aes(x = <NAME OF VARIABLE 1>,
                     fill = <NAME OF VARIABLE 2>)) +
  geom_bar(position = "stack") +
  labs(x = "<TITLE FOR THE X-AXIS>")
```

Note: You should fill by whichever variable has **fewer** values.
Note: If you want a side-by-side bar plot you need to change `position` to "dodge". If you want a filled bar plot, you need change `position` to "fill".

Creating a Summary Table of Observations of One Variable

```
<NAME OF DATASET> %>%
   group_by(<NAME OF VARIABLE>) %>%
   summarize( n = n() )
```

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

Creating a Summary Table of Observations from Two Variables

```
<NAME OF DATASET> %>%
   group_by(<NAME OF VARIABLE 1>, <NAME OF VARIABLE 2>) %>%
   summarize( n = n() )
```

Performing a Chi-Squared Goodness-of-Fit Test (One Variable)

```
chisq_test(x = <NAME OF DATASET>,
       response = <NAME OF VARIABLE>)
```

Performing a Chi-Squared Independence / Homogeneity Test (Two Variables)

```
chisq_test(x = <NAME OF DATASET>,
       response = <NAME OF RESPONSE VARIABLE>,
       explanatory = <NAME OF EXPLANATORY VARIABLE>)
```

Obtaining the Sample X-Squared Statistic

```
obs_xsq <- <NAME OF DATASET> %>%
  specify(response = <NAME OF RESPONSE VARIABLE>,
          explanatory = <NAME OF EXPLANATORY VARIABLE>) %>%
  calculate(stat = "Chisq")
```

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

Obtaining 1000 Permuted X-Squared Statistics – Assuming the Null Hypothesis is True

```
null_dist <- <NAME OF DATASET> %>%
  specify(response = <NAME OF RESPONSE VARIABLE>,
          explanatory = <NAME OF EXPLANATORY VARIABLE>) %>%
  generate(reps = 1000, type = "permute") %>%
  calculate(stat = "Chisq")
```

< > 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_xsq,
           direction = “greater”)
```

Note: *This step* ***must*** *come after you have obtained the bootstrapped differences in means* ***and*** *the observed difference in means!*
Note: In a Chi-Squared test we **always** use a greater than alternative, since we only look in the right tail!