|  |
| --- |
| **Creating a Summary Table of Observations of One Variable**  <NAME OF DATASET> %>%  count(<NAME OF CATEGORICAL VARIABLE>) |
| **Creating a Summary Table of Observations from Two Variables**  <NAME OF DATASET> %>%  count(<NAME OF CATEGORICAL VARIABLE 1>,  <NAME OF CATEGORICAL VARIABLE 2>) |
| **Creating a Contingency Table of Observations from Two Variables**  <NAME OF DATASET> %>%  count(<NAME OF CATEGORICAL VARIABLE 1>,  <NAME OF CATEGORICAL VARIABLE 2>) %>%  pivot\_wider(names\_from = <NAME OF EXPLANATORY VARIABLE>,  values\_from = n) %>%  janitor::adorn\_totals(where = c(“row”, “col”))  ***Note:*** Your explanatory variable should be in the rows and your response variable should be in the columns. So, the variable you insert into names\_from should be the response variable you are interested in. |

|  |
| --- |
| **Simulation-based one-way ANOVA for Categorical Variable with three or more levels** |
| **Obtaining the Sample (Observed) F-Statistic**  obs\_F <- <NAME OF DATASET> %>%  specify(response = <NAME OF VARIABLE>,  explanatory = <NAME OF VARIABLE>) %>%  calculate(stat = "F") |
| **Obtaining 1000 Permuted F-Statistics**  null\_distribution <- <NAME OF DATASET> %>%  specify(response = <NAME OF NUMERICAL VARIABLE>,  explanatory = <NAME OF CATEGORICAL VARIABLE>) %>%  hypothesize(null = “independence”) %>%  generate(reps = 1000, type = "permute") %>%  calculate(stat = "F") |

|  |
| --- |
| **Plotting the Null Distribution**  visualize(null\_distribution)  ***Note:*** You can add axis labels to this plot! All you need to do is connect the visualize() step to labs() using a **+** sign. |
| **Shading the p-value**  visualize(null\_distribution)+  shade\_p\_value(obs\_stat = obs\_F,  direction = “greater”)  ***Note:*** You can add axis labels to this plot! All you need to do is connect the visualize() step to labs() using a **+** sign. |
| **Obtaining a p-value**  get\_p\_value(null\_distribution,  obs\_stat = obs\_F,  direction = “greater”) |
| **Theory-based one-way ANOVA for Categorical Variable with three or more levels** |
| **Conducting a Theory-based One-Way ANOVA**  aov(<NAME OF RESPONSE VARIABLE> ~ <NAME OF EXPLANATORY VARIABLE>) %>%  broom::tidy() |

|  |
| --- |
| **Simulation-based one-way ANOVA for Categorical Variable with two levels** |
| **Obtaining the Sample (Observed) Difference in Means**  obs\_diff <- <NAME OF DATASET> %>%  specify(response = <NAME OF VARIABLE>,  explanatory = <NAME OF VARIABLE>) %>%  calculate(stat = "diff in means") |

|  |
| --- |
| **Obtaining 1000 Permuted Differences in Means**  null\_distribution <- <NAME OF DATASET> %>%  specify(response = <NAME OF NUMERICAL VARIABLE>,  explanatory = <NAME OF CATEGORICAL VARIABLE>) %>%  hypothesize(null = “independence”) %>%  generate(reps = 1000, type = "permute") %>%  calculate(stat = "diff in means") |
| **Plotting the Null Distribution**  visualize(null\_distribution)  ***Note:*** You can add axis labels to this plot! All you need to do is connect the visualize() step to labs() using a **+** sign. |
| **Shading the p-value**  visualize(null\_distribution)+  shade\_p\_value(obs\_stat = obs\_diff,  direction = “two-sided”)  ***Note:*** You can add axis labels to this plot! All you need to do is connect the visualize() step to labs() using a **+** sign. |
| **Obtaining a p-value**  get\_p\_value(null\_distribution,  obs\_stat = obs\_diff,  direction = “two-sided”) |
| **Theory-based one-way ANOVA for Categorical Variable with two levels** |
| **Conducting a Theory-based One-Way ANOVA**  aov(<NAME OF RESPONSE VARIABLE> ~ <NAME OF EXPLANATORY VARIABLE>) %>%  broom::tidy() |

**(more on next page!)**

|  |
| --- |
| **Evaluating Conditions of Residuals for ANOVA** |
| **Distribution of residuals (to evaluate normality condition)**  broom::augment(my\_model) %>%  ggplot(mapping = aes(x = .resid)) +  geom\_histogram() +  labs(x = "Residual")  ***Note:*** In the above code my\_model represents the ANOVA model that was fit, this can be either a one-way **or** two-way ANOVA! |
| **Residuals versus fitted values (to evaluate equal variance condition)**  broom::augment(my\_model) %>%  ggplot(mapping = aes(y = .resid, x = `<NAME OF EXPLANATORY VARIABLE>`)) +  geom\_point() +  geom\_hline(yintercept = 0, color = "red", linewidth = 3) +  labs(x = "<NAME OF EXPLANATORY VARIABLE>",  y = "Residuals")  ***Note:*** In the above code my\_model represents the linear model that was fit, this can be either a simple **or** multiple linear regression! |