|  |
| --- |
| **Repairing a Non-Linear Relationship** |
| **Should you log transform the x variable? Or the y variable? Or both? Explore the plots!**  ggplot(data = hbr\_maples,  mapping = aes(x = stem\_length, y = stem\_dry\_mass)  ) +  geom\_point() +  geom\_smooth(method = "lm") +  scale\_y\_log10() +  scale\_x\_log10() +  labs(  x = "Stem Length (mm)",  y = "Log Transformed Stem Dry Mass (g)"  )  ***Note:*** In the above code scale\_y\_log10()and scale\_x\_log10()perform a log (base 10) transformation of the x **and** y-variables. If you only want to transform one variable, you can delete the other line of code! |
| **Evaluating Normality of Residuals** |
| **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 linear model that was fit, this can be either a simple **or** multiple linear regression! |
| **Evaluating Equal Variance of Residuals** |
| **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! |

|  |
| --- |
| **Repairing Independence Condition Violations** |
| **Are your observations correlated in time? Consider filtering to only include one year OR including year as an explanatory variable!**  filter(nycflights,   **year == 2015**)  flight\_delay\_lm <- lm(dep\_delay ~ airport + **year**, data = nycflights) |
| **Do you have repeated observations on the same observational unit? Consider collapsing these multiple observations into a single observation!**  **Option 1: Grab a Random Observation** evals\_small <- evals %>%  group\_by(prof\_ID) %>%  slice\_sample(n = 1)  **Option 2: Summarize the Multiple Observations** evals\_small <- evals %>%  group\_by(prof\_ID) %>%  mutate(min\_score = min(score)) %>%  distinct(prof\_ID, .keep\_all = TRUE) %>%  ungroup()  ***Note 1:*** In the above code prof\_ID is the variable that creates **groups** of observations.  ***Note 2:*** In the above code score is the **response** variable I’m interested in that I use to create summaries for each **group** of observations. |