PSTAT 126

Lab 9

Roupen Khanjian

Spring 2021

```
library(faraway) # Functions and Datasets for Books by Julian Faraway
library(alr4) # Data to Accompany Applied Linear Regression 4th Edition
library(tidyverse) # Easily Install and Load the 'Tidyverse'
library(patchwork) # The Composer of Plots
library(GGally) # Extension to 'ggplot2'
library(janitor) # Simple Tools for Examining and Cleaning Dirty Data
library(palmerpenguins) # Palmer Archipelago (Antarctica) Penguin Data
library(broom) # Convert Statistical Objects into Tidy Tibbles
```

Contents

Categorical Variables

- `worldcup` Dataset: Player data from the 2010 world cup.

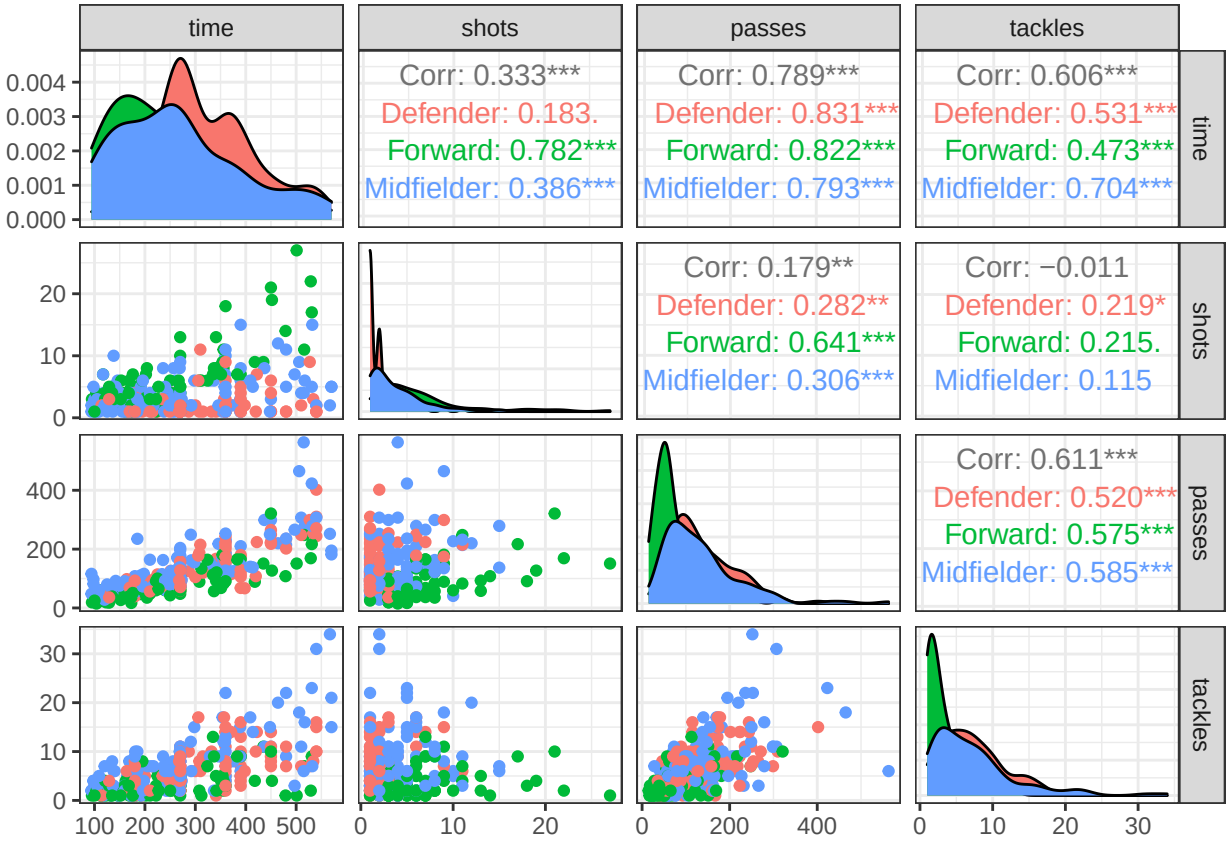
`position` = a factor with levels: (Defender, Forward, Goalkeeper, Midfielder).
`time` = Time played in minutes.
`shots` = Number of shots attempted.
`passes` = Number of passes made.
`tackles` = Number of tackles made.

```
data("worldcup") # from faraway package

glimpse(worldcup)
```

```
## Rows: 595
## Columns: 7
## $ Team     <fct> Algeria, Japan, France, France, Cameroon, Uruguay, Ghana, Gha~
## $ Position <fct> Midfielder, Midfielder, Defender, Midfielder, Forward, Forwar~
## $ Time     <int> 16, 351, 180, 270, 46, 72, 138, 33, 21, 103, 270, 55, 106, 27~
## $ Shots    <int> 0, 0, 0, 1, 2, 0, 0, 0, 5, 0, 2, 0, 2, 1, 1, 1, 5, 9, 0, 1, 0~
## $ Passes   <int> 6, 101, 91, 111, 16, 15, 51, 9, 22, 38, 120, 31, 57, 123, 172~
## $ Tackles  <int> 0, 14, 6, 5, 0, 0, 2, 0, 0, 1, 10, 2, 2, 11, 13, 7, 3, 18, 0,~
## $ Saves    <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
```

```
# Some data cleaning
worldcup1 <- worldcup %>%
```

```
  select(-c(Saves, Team)) %>%
  filter(Passes > 1 ) %>% # players that have made over 1 pass
  filter(Time > 90) %>%
  filter(Position != "Goalkeeper") %>% # remove the goalies from model
  filter(Tackles > 0) %>% # players that have at least 1 tackle
  filter(Shots > 0) %>% # players that have at least 1 shot attempted
  clean_names() # convert the columns names to lower case.

ggpairs(worldcup1,
        mapping = aes(color = position),
        columns = c("time",
                    "shots",
                    "passes",
                    "tackles")) +
  theme_bw()
```



Want to predict how many passes a player made given the other variables in our dataset. Thus, response variable will be `Passes`.

```
model <- lm(passes ~ ., data = worldcup1)
summary(model)
```

```
##
## Call:
## lm(formula = passes ~ ., data = worldcup1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -109.587  -25.154   -0.683   20.403  314.908
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -26.8017     8.8544  -3.027  0.00269 **
## positionForward    -19.7984     8.4513  -2.343  0.01981 *
## positionMidfielder  19.1624     6.6417   2.885  0.00420 **
## time                 0.4755     0.0330  14.411  < 2e-16 ***
## shots               -0.2190     0.8990  -0.244  0.80767
## tackles              1.9510     0.7126   2.738  0.00656 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 44.79 on 294 degrees of freedom
## Multiple R-squared:  0.6863, Adjusted R-squared:  0.6809
## F-statistic: 128.6 on 5 and 294 DF,  p-value: < 2.2e-16
```

What is the default or baseline group here for the Position variable?

```
d1 <- coef(model)[1] # defender
m1 <- coef(model)[1] + coef(model)[3] # midfielder
f1 <- coef(model)[1] + coef(model)[2] # forward
```

Lets say we want *Forward* to be the default or baseline group for the Position variable.

```
worldcup1$defender = ifelse(as.character(worldcup1$position) == 'Defender', 1, 0)
worldcup1$midfielder = ifelse(as.character(worldcup1$position) == 'Midfielder', 1, 0)

model1 <- lm(passes ~ defender + midfielder + time + shots + tackles
             ,data = worldcup1)
summary(model1)
```

```
##
## Call:
## lm(formula = passes ~ defender + midfielder + time + shots +
##     tackles, data = worldcup1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -109.587  -25.154   -0.683   20.403  314.908
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -46.6001     8.1089  -5.747 2.27e-08 ***
## defender     19.7984     8.4513   2.343  0.01981 *
## midfielder   38.9608     7.2095   5.404 1.35e-07 ***
## time          0.4755     0.0330  14.411  < 2e-16 ***
## shots        -0.2190     0.8990  -0.244  0.80767
## tackles       1.9510     0.7126   2.738  0.00656 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 44.79 on 294 degrees of freedom
## Multiple R-squared:  0.6863, Adjusted R-squared:  0.6809
## F-statistic: 128.6 on 5 and 294 DF,  p-value: < 2.2e-16
```

```r
# tidy and glance from broom package
model_tidy <- tidy(model)
model1_tidy <- tidy(model1)
model_tidy; model1_tidy
```

```
## # A tibble: 6 x 5
##   term               estimate std.error statistic  p.value
##   <chr>                 <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)         -26.8      8.85      -3.03  2.69e- 3
## 2 positionForward     -19.8      8.45      -2.34  1.98e- 2
## 3 positionMidfielder   19.2      6.64       2.89  4.20e- 3
## 4 time                  0.476    0.0330    14.4   5.51e-36
## 5 shots                -0.219    0.899     -0.244 8.08e- 1
## 6 tackles               1.95     0.713      2.74  6.56e- 3
```

```
## # A tibble: 6 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)  -46.6      8.11      -5.75  2.27e- 8
## 2 defender      19.8      8.45       2.34  1.98e- 2
## 3 midfielder    39.0      7.21       5.40  1.35e- 7
## 4 time           0.476    0.0330    14.4   5.51e-36
## 5 shots         -0.219    0.899     -0.244 8.08e- 1
## 6 tackles        1.95     0.713      2.74  6.56e- 3
```

```r
glance(model)[c(1:5,8:9)]; glance(model1)[c(1:5,8:9)]
```

```
## # A tibble: 1 x 7
##   r.squared adj.r.squared sigma statistic  p.value   AIC   BIC
##       <dbl>         <dbl> <dbl>     <dbl>    <dbl> <dbl> <dbl>
## 1     0.686         0.681  44.8      129. 7.70e-72 3140. 3166.
```

```
## # A tibble: 1 x 7
##   r.squared adj.r.squared sigma statistic  p.value   AIC   BIC
##       <dbl>         <dbl> <dbl>     <dbl>    <dbl> <dbl> <dbl>
## 1     0.686         0.681  44.8      129. 7.70e-72 3140. 3166.
```

```r
d2 <- coef(model1)[1] + coef(model1)[2] # defender
m2 <- coef(model1)[1] + coef(model1)[3] # midfielder
f2 <- coef(model1)[1] # forward
as.numeric(d1); as.numeric(d2) # defender
```

```
## [1] -26.80169
```

```
## [1] -26.80169
```

```r
as.numeric(m1); as.numeric(m2) # midfielder
```

```
## [1] -7.639253
```

```
## [1] -7.639253
```

```r
as.numeric(f1); as.numeric(f2) # forward
```

```
## [1] -46.60006
```

```
## [1] -46.60006
```

R chooses a default group for you.

Another example . . .

Figure 1: Dragons! How would we predict the weight of a dragon by a continuous variable (height) and a categorical variable (spotted or stripped)?

Figure 2: Interpretation of categorical predictor variable

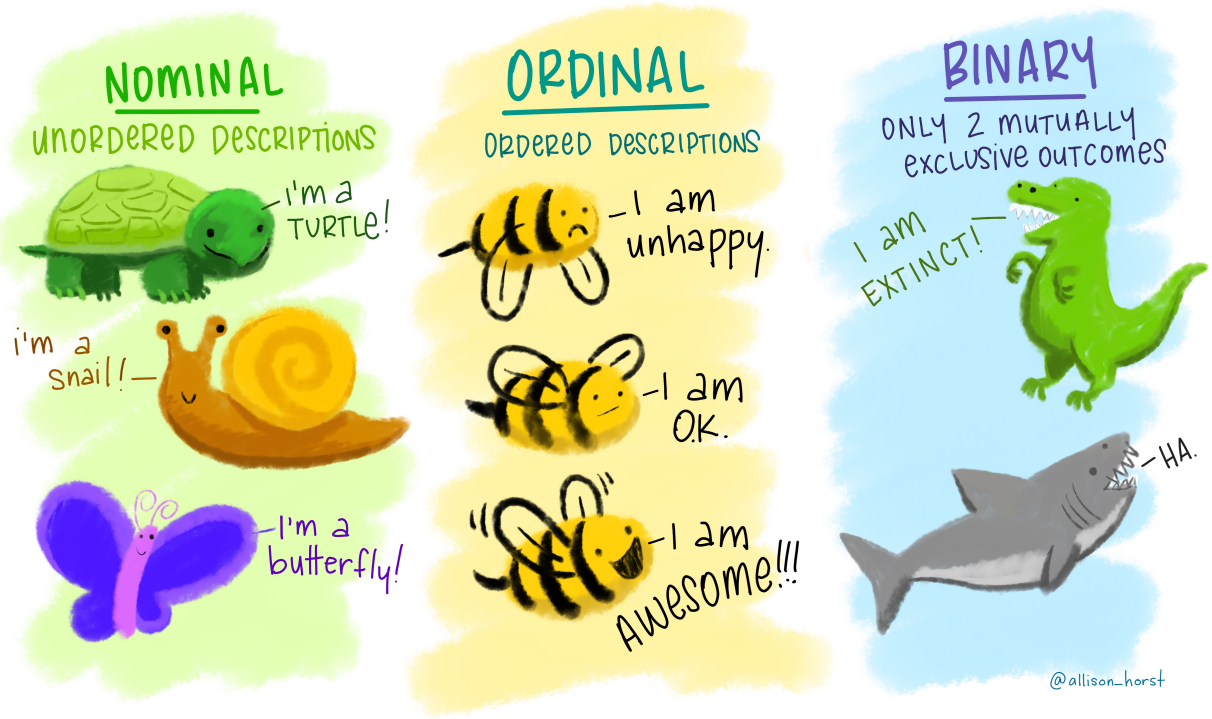


Figure 3: Review of categorical variables

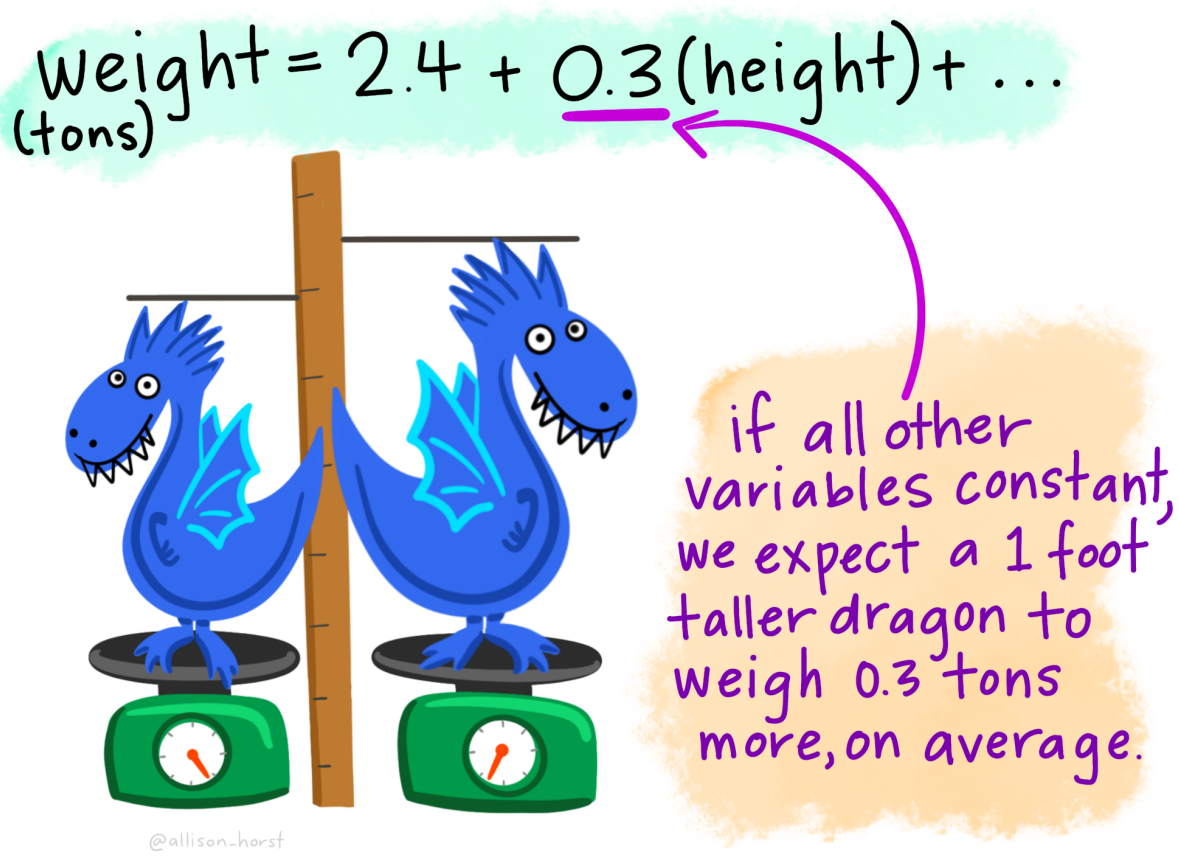


Figure 4: Interpretation of continuous predictor variable

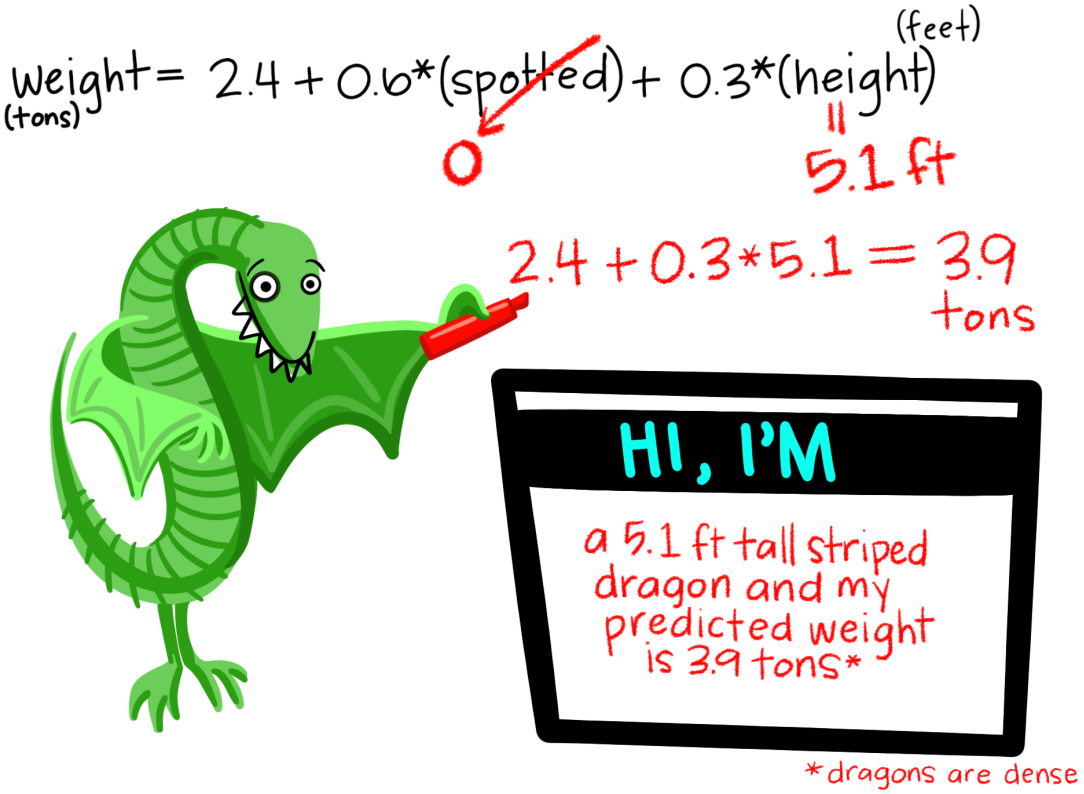


Figure 5: Making predictions based off continuous and categorical predictors

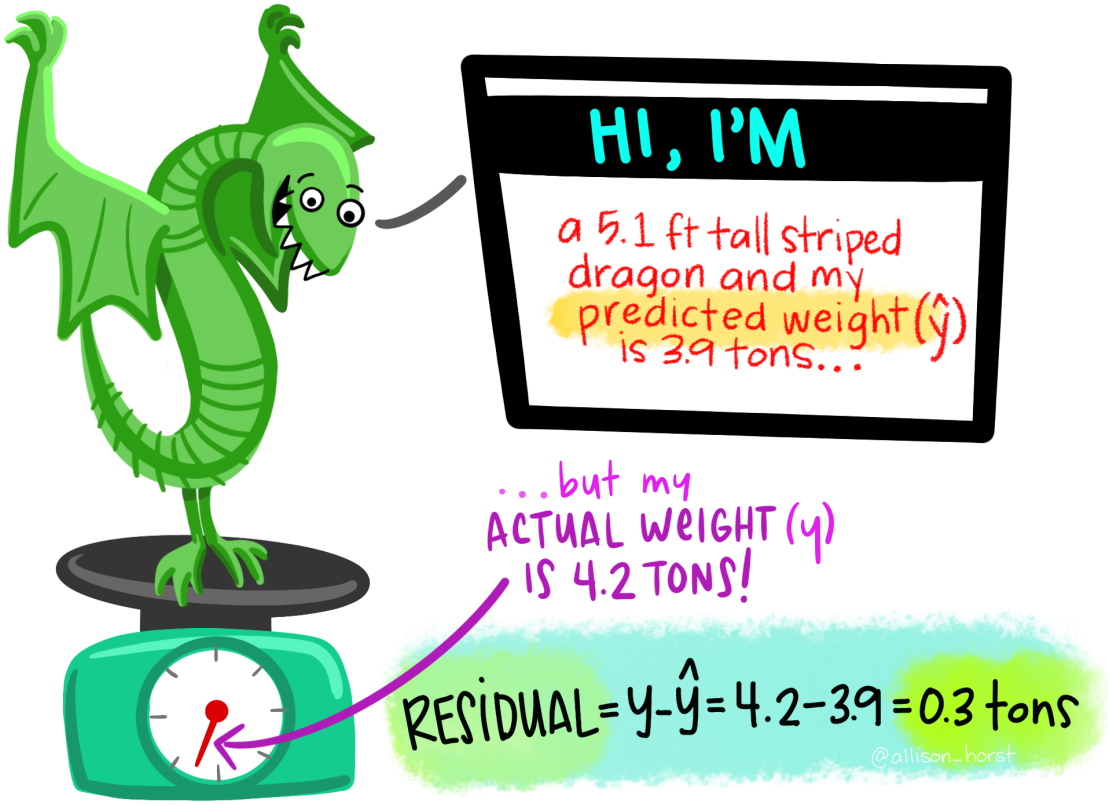


Figure 6: What about residuls in MLR?

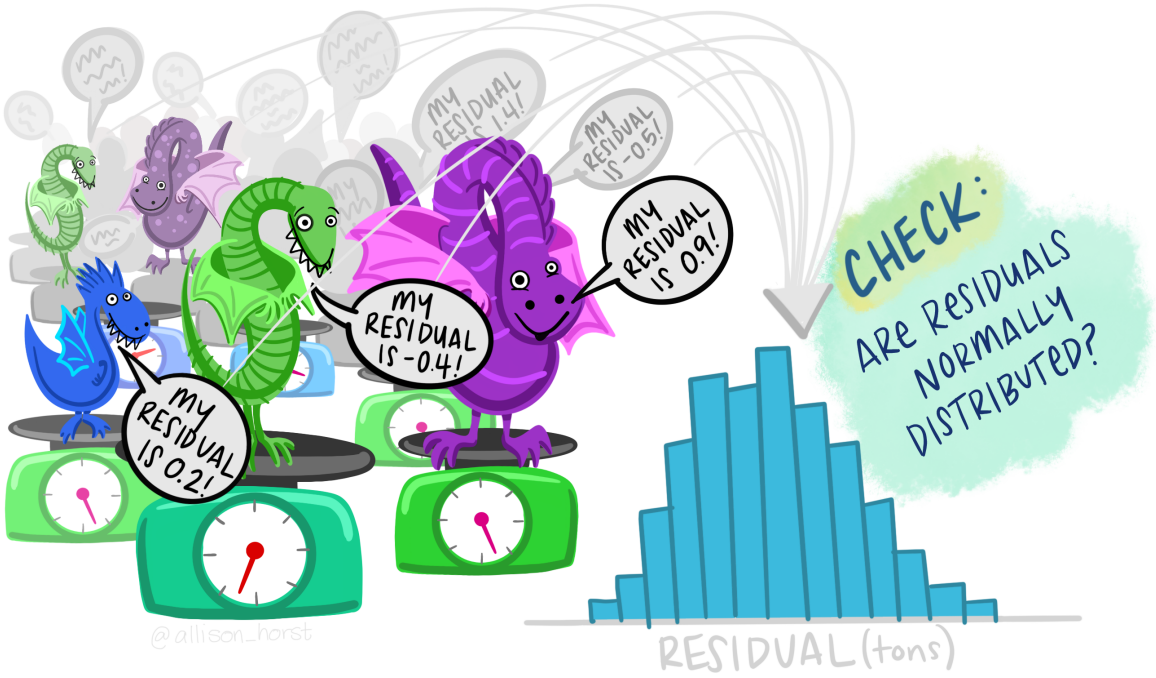


Figure 7: Make sure residulas are normally distributed!

- All above artwork by UCSB Bren professor Allison Horst.
  - https://www.twitter.com/allison_horst
  - https://github.com/allisonhorst/stats-illustrations

Interaction between quantitative and qualitative predictor variables

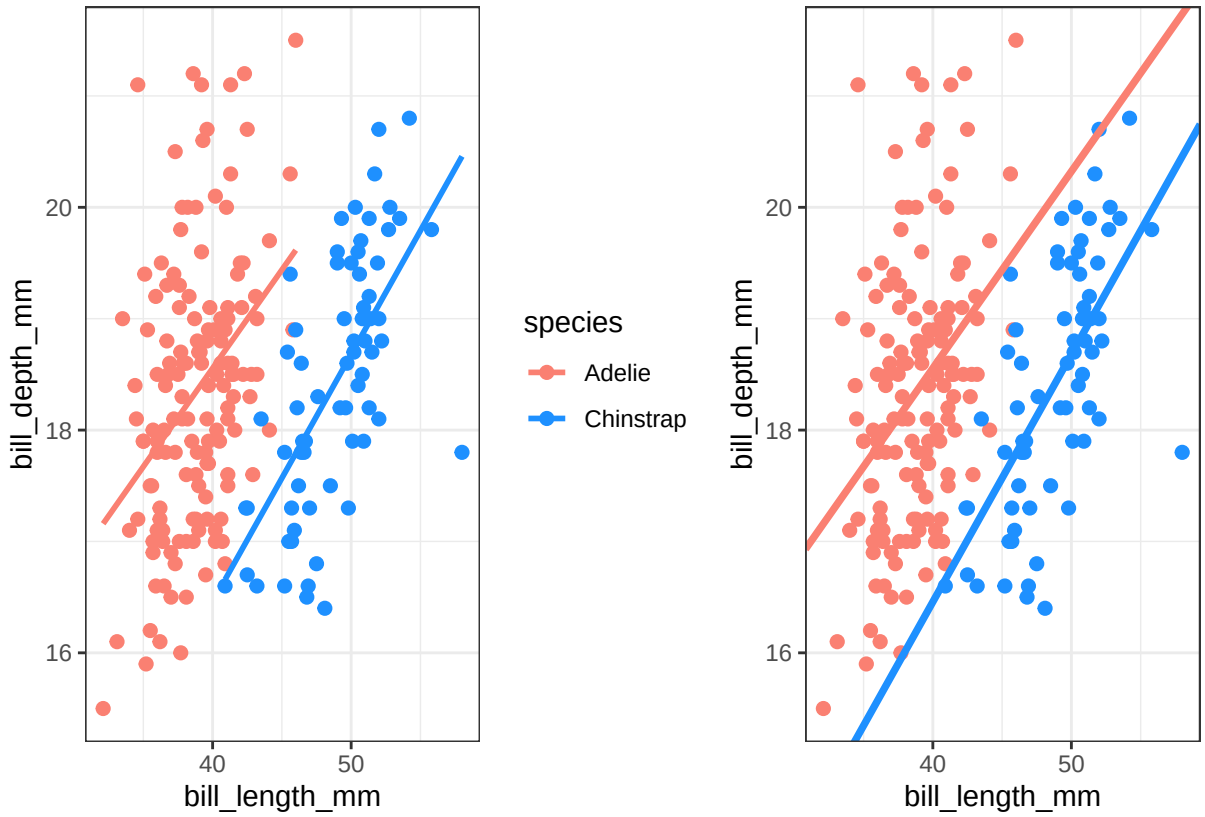
Example of a Parallel model

```
penguins_noNA_no_Gentoo <- penguins %>%
  drop_na() %>%
  filter(species != "Gentoo")

model_penguins <- lm(bill_depth_mm ~ species*bill_length_mm,
                     data = penguins_noNA_no_Gentoo)
summary(model_penguins)
```

```
##
## Call:
## lm(formula = bill_depth_mm ~ species * bill_length_mm, data = penguins_noNA_no_Gentoo)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6574 -0.7301 -0.0530  0.5743  3.4990
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      11.48771    1.27840   8.986   <2e-16 ***
## speciesChinstrap                 -3.91857    2.27857  -1.720    0.087 .
## bill_length_mm                    0.17668    0.03285   5.378    2e-07 ***
## speciesChinstrap:bill_length_mm   0.04553    0.05064   0.899    0.370
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.053 on 210 degrees of freedom
## Multiple R-squared:  0.2291, Adjusted R-squared:  0.2181
## F-statistic:  20.8 on 3 and 210 DF,  p-value: 7.713e-12
```

```
p1 <- ggplot(data = penguins_noNA_no_Gentoo,
       aes(x = bill_length_mm,
           y = bill_depth_mm,
           colour = species)) +
  geom_point(size = 2) +
  geom_smooth(method = "lm", se = F)+
  scale_color_manual(values = c("salmon", "dodgerblue")) +
  theme_bw()

p2 <- ggplot(data = penguins_noNA_no_Gentoo,
       aes(x = bill_length_mm,
           y = bill_depth_mm,
           colour = species)) +
  geom_point(size = 2) +
  geom_abline(aes(intercept = coef(model_penguins)[1],
                  slope = coef(model_penguins)[3]),
              col = "salmon", size = 1.3) +
  geom_abline(aes(intercept = coef(model_penguins)[1] +  coef(model_penguins)[2],
                  slope = coef(model_penguins)[3] +  coef(model_penguins)[4]),
              col = "dodgerblue", size = 1.3) +
  scale_color_manual(values = c("salmon", "dodgerblue")) +
  theme_bw() +
```

```
  theme(legend.position = "none")

p1 + p2
```




```
model_penguins2 <- lm(bill_depth_mm ~ species + bill_length_mm,
                      data = penguins_noNA_no_Gentoo)
summary(model_penguins2)
```

```
##
## Call:
## lm(formula = bill_depth_mm ~ species + bill_length_mm, data = penguins_noNA_no_Gentoo)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.4157 -0.7576 -0.0427  0.5929  3.5800
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      10.74376    0.97406  11.030  < 2e-16 ***
## speciesChinstrap -1.88706    0.29404  -6.418 8.97e-10 ***
## bill_length_mm    0.19585    0.02499   7.837 2.23e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.053 on 211 degrees of freedom
## Multiple R-squared:  0.2261, Adjusted R-squared:  0.2188
## F-statistic: 30.82 on 2 and 211 DF,  p-value: 1.801e-12
```

Example of a Non-parallel model

```r
data("mtcars")

mtcars1 <- mtcars %>%
  mutate(cyl = factor(cyl)) %>%
  select(c(mpg, disp, cyl))

model_1 <- lm(mpg ~ disp*cyl, data = mtcars1)
# alternative way to incorporate an interaction term into a model
# model_1 <- lm(mpg ~ disp + cyl + disp:cyl, data = mtcars1)
summary(model_1)
```

```
##
## Call:
## lm(formula = mpg ~ disp * cyl, data = mtcars1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4766 -1.8101 -0.2297  1.3523  5.0208
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  40.87196    3.02012  13.533 2.79e-13 ***
## disp         -0.13514    0.02791  -4.842 5.10e-05 ***
## cyl6        -21.78997    5.30660  -4.106 0.000354 ***
## cyl8        -18.83916    4.61166  -4.085 0.000374 ***
## disp:cyl6     0.13875    0.03635   3.817 0.000753 ***
## disp:cyl8     0.11551    0.02955   3.909 0.000592 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.372 on 26 degrees of freedom
## Multiple R-squared:  0.8701, Adjusted R-squared:  0.8452
## F-statistic: 34.84 on 5 and 26 DF,  p-value: 9.968e-11
```

```r
m1 <- ggplot(data = mtcars1, aes(x = disp, y = mpg, colour = cyl)) +
  geom_point(size = 2.5) +
  geom_smooth(method = "lm", se = F) +
  theme_bw()

m2 <- ggplot(data = mtcars1, aes(x = disp, y = mpg, colour = cyl)) +
  geom_point(size = 2.5) +
  geom_abline(aes(intercept = coef(model_1)[1],
                  slope = coef(model_1)[2]), col = "salmon", size = 1.3) +
  geom_abline(aes(intercept = coef(model_1)[1] +  coef(model_1)[3],
                  slope = coef(model_1)[2] +  coef(model_1)[5]), col = "seagreen", size = 1.3) +
  geom_abline(aes(intercept = coef(model_1)[1] +  coef(model_1)[4],
                  slope = coef(model_1)[2] +  coef(model_1)[6]), col = "dodgerblue", size = 1.3) +
  theme_bw() +
  theme(legend.position = "none")

m1 + m2
```