pokemon dataset

sarah, mars, juniper

```
library(ggplot2)
library(dplyr)
```

```
Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```
library(mltools)
library(data.table)
```

```
Attaching package: 'data.table'

The following objects are masked from 'package:dplyr':

    between, first, last
```

```
library(car)
```

```
Loading required package: carData
```

```
Attaching package: 'car'

The following object is masked from 'package:dplyr':

    recode
```

The professor recommended doing this so the empty string " " will be converted to a NA value so we can then use the is.na() function.

```
pokemon <- read.csv("/Users/sarah/Desktop/SDS291/FinalProject/pokemon.csv", na.strings = c
```

Making the dual_type column that says if the Pokemon is a dual type based on if there is a second type.

```
pokemon <- mutate(pokemon, dual_type = !is.na(type2))
```

Making sure that there are only TRUE and FALSE values in the dual_type column.

```
unique(pokemon$'dual_type')
```

```
[1]  TRUE FALSE
```

Making sure that these variables are the right data type.

```
pokemon$'capture_rate' = as.numeric(pokemon$'capture_rate')
```

```
Warning: NAs introduced by coercion
```

```
pokemon$'type1' = as.factor(pokemon$'type1')
pokemon$'type2' = as.factor(pokemon$'type2')
```

Dataset

We are planning to use the Complete Pokemon Dataset that has information on different Pokemon up to Gen 7. The link where we got the dataset is included below. Dataset Link

Research Question

How do different Pokemon's base stats influence capture rate?

```
pokemon$'capture_rate' = as.numeric(pokemon$'capture_rate')
pokemon$'type1' = as.factor(pokemon$'type1')
pokemon$'type2' = as.factor(pokemon$'type2')
```

Different Possible Stats

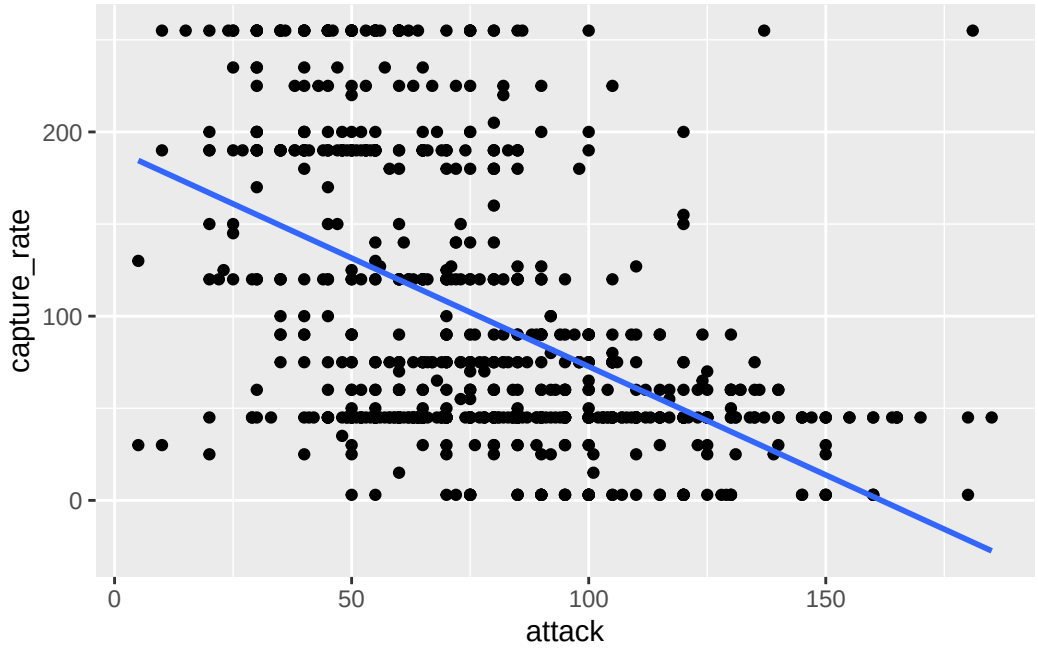
- attack
- base_happiness
- base_egg_steps
- base_total
- defense
- hp
- sp_attack
- sp_defense
- speed

The base stats I'll focus on are attack, hp, defense, and speed. I'm not sure what are important base stats for Pokemon but I'm guessing.

```
ggplot(data = pokemon, mapping = aes(x = attack, y = capture_rate)) +
  geom_point() +
  geom_smooth(method = lm, se = FALSE, formula = y~x)
```

```
Warning: Removed 1 rows containing non-finite values (stat_smooth).
```

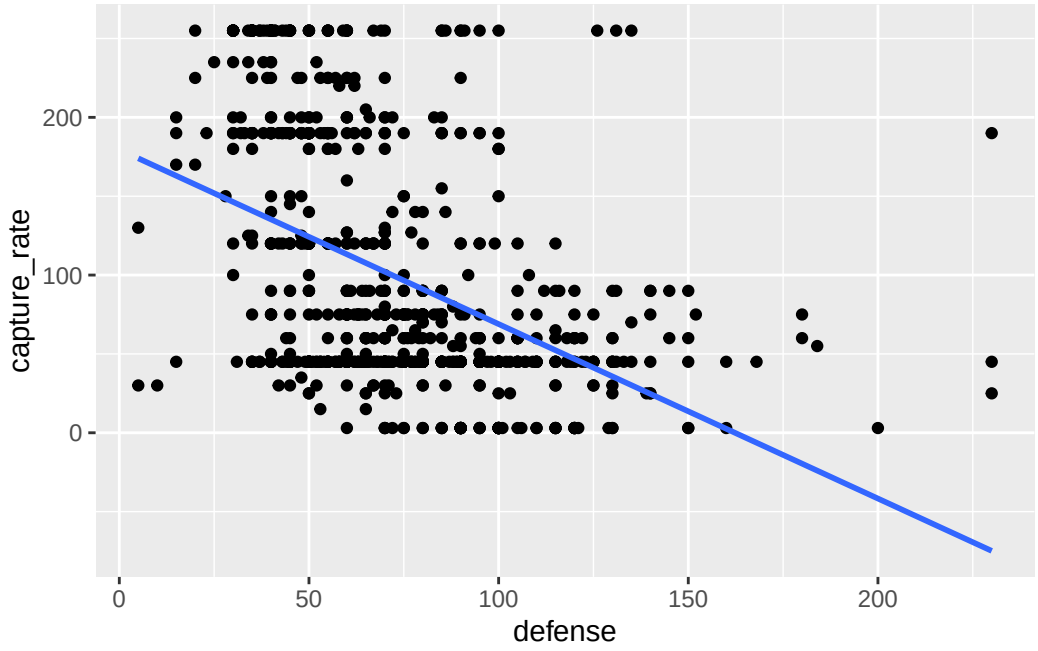
```
Warning: Removed 1 rows containing missing values (geom_point).
```




```
ggplot(data = pokemon, mapping = aes(x = defense, y = capture_rate)) +
  geom_point() +
  geom_smooth(method = lm, se = FALSE, formula = y~x)
```

```
Warning: Removed 1 rows containing non-finite values (stat_smooth).
```

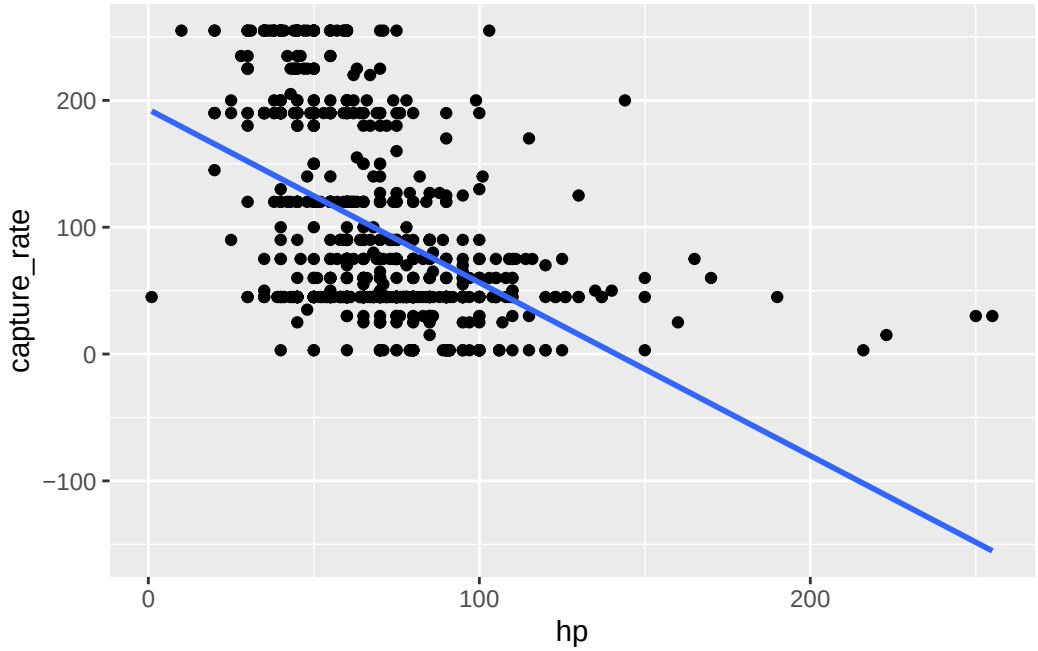
```
Warning: Removed 1 rows containing missing values (geom_point).
```




```
ggplot(data = pokemon, mapping = aes(x = hp, y = capture_rate)) +
  geom_point() +
  geom_smooth(method = lm, se = FALSE, formula = y~x)
```

```
Warning: Removed 1 rows containing non-finite values (stat_smooth).
```

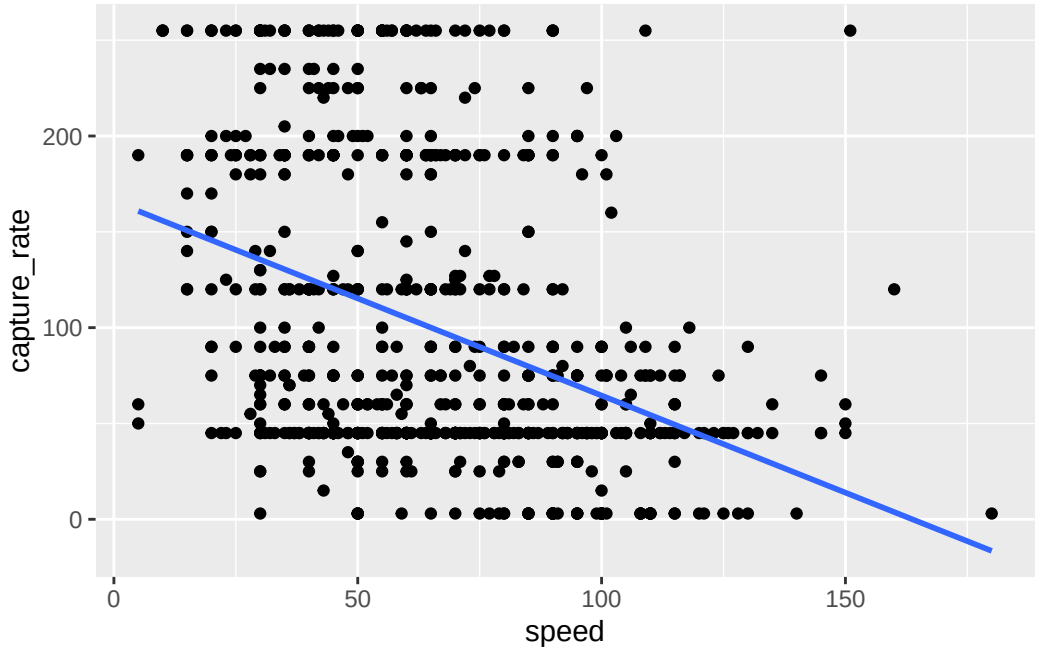
```
Warning: Removed 1 rows containing missing values (geom_point).
```




```
ggplot(data = pokemon, mapping = aes(x = speed, y = capture_rate)) +
  geom_point() +
  geom_smooth(method = lm, se = FALSE, formula = y~x)
```

```
Warning: Removed 1 rows containing non-finite values (stat_smooth).
```

```
Warning: Removed 1 rows containing missing values (geom_point).
```



Making the models

Additive Model

```
additive_capture_model2 <- lm(capture_rate ~ attack + defense + hp + speed, data = pokemon
```

```
summary(additive_capture_model2)
```

```
Call:
lm(formula = capture_rate ~ attack + defense + hp + speed, data = pokemon)

Residuals:
     Min       1Q   Median       3Q      Max
-150.832  -36.856   -5.036   36.345  255.371

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 288.48251    7.67680  37.578  < 2e-16 ***
attack       -0.29466    0.08042  -3.664 0.000265 ***
```

```
defense      -0.77384    0.07492 -10.329  < 2e-16 ***
hp           -0.86774    0.08251 -10.517  < 2e-16 ***
speed        -0.76101    0.07525 -10.113  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 56.4 on 795 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.4558,    Adjusted R-squared:  0.453
F-statistic: 166.4 on 4 and 795 DF,  p-value: < 2.2e-16
```

Adjusted R-squared: 0.453

Interactive Model

```r
interact_capture_model2 <- lm(capture_rate ~ attack * defense * hp * speed, data = pokemon
```

```r
summary(interact_capture_model2)
```

```
Call:
lm(formula = capture_rate ~ attack * defense * hp * speed, data = pokemon)

Residuals:
     Min       1Q   Median       3Q      Max
-167.956  -31.949   -1.533   28.992  219.782

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)
(Intercept)             2.615e+02  5.582e+01   4.686 3.29e-06 ***
attack                 -7.279e-02  9.924e-01  -0.073  0.94154
defense                 1.677e-01  5.320e-01   0.315  0.75270
hp                     -7.348e-02  8.772e-01  -0.084  0.93326
speed                   1.101e+00  9.999e-01   1.101  0.27102
attack:defense         -1.027e-02  1.017e-02  -1.010  0.31278
attack:hp              -1.235e-02  1.323e-02  -0.933  0.35101
defense:hp             -1.291e-02  1.015e-02  -1.273  0.20355
attack:speed           -2.156e-02  1.591e-02  -1.355  0.17572
defense:speed          -3.126e-02  1.115e-02  -2.803  0.00519 **
hp:speed               -2.085e-02  1.636e-02  -1.274  0.20303
```

```
attack:defense:hp        1.798e-04  1.415e-04   1.270  0.20436
attack:defense:speed     3.228e-04  1.606e-04   2.010  0.04482 *
attack:hp:speed          2.942e-04  2.235e-04   1.316  0.18844
defense:hp:speed         2.655e-04  1.861e-04   1.426  0.15422
attack:defense:hp:speed -3.217e-06  2.283e-06  -1.409  0.15925
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 55.09 on 784 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.488, Adjusted R-squared:  0.4782
F-statistic: 49.81 on 15 and 784 DF,  p-value: < 2.2e-16
```

Adjusted R-squared: 0.4782

Nested F-test

```
anova(additive_capture_model2, interact_capture_model2)
```

```
Analysis of Variance Table

Model 1: capture_rate ~ attack + defense + hp + speed
Model 2: capture_rate ~ attack * defense * hp * speed
  Res.Df     RSS Df Sum of Sq      F    Pr(>F)
1    795 2528696
2    784 2379127 11    149569 4.4807 1.385e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-value is below 0.05 implying that the change from additive to interactive is necessary, but I feel like the adjusted r-square doesn't justify the change.

Opinion: Would be interesting to look at but haven't tested removing some explanatory variables to see how it affects the model.

Model we plan to use

We plan to use an interaction model to predict the capture rate of pokemon (outcome variable). We plan to use the base stats of the Pokemon as our explanatory variables. We are currently

looking into attack, hp, defense, and speed, but we are thinking of adding base_total as a possible explanatory variable and also seeing if we can add our own variable that says if the Pokemon is a dual type, but are having trouble with creating the variable at the moment.

Current Plan

- Look at how using a different number and combination of variables affect the model
- what stats we plan to look at: dual_type, attack, defense, hp, speed, base_total, base_happiness, sp_attack, and sp_defense. Plan to look at a different combination of these things, but should also look at the VIF of these variables as well to make sure there isn't too much multicollinearity.

VIF and Multicollinearity

```
vif(additive_capture_model2)
```

```
  attack  defense       hp    speed 
1.681200 1.336157 1.209118 1.184981 
```

All values are below 10 so multicollinearity wont be an issue here.

```
vif(interact_capture_model2)
```

```
there are higher-order terms (interactions) in this model
consider setting type = 'predictor'; see ?vif
```

```
                attack                defense                     hp 
             268.34144               70.61604              143.24012 
                 speed          attack:defense              attack:hp 
             219.30342              520.29476              644.42856 
           defense:hp            attack:speed           defense:speed 
             275.89995              996.83905              279.76076 
            hp:speed       attack:defense:hp    attack:defense:speed 
             572.21339              950.03798             1054.74108 
       attack:hp:speed        defense:hp:speed attack:defense:hp:speed 
            1766.61410              753.94466             1918.23200 
```

So all these values are above 10 so they should be centered if we chose to use them.

Adding dual types to the current models

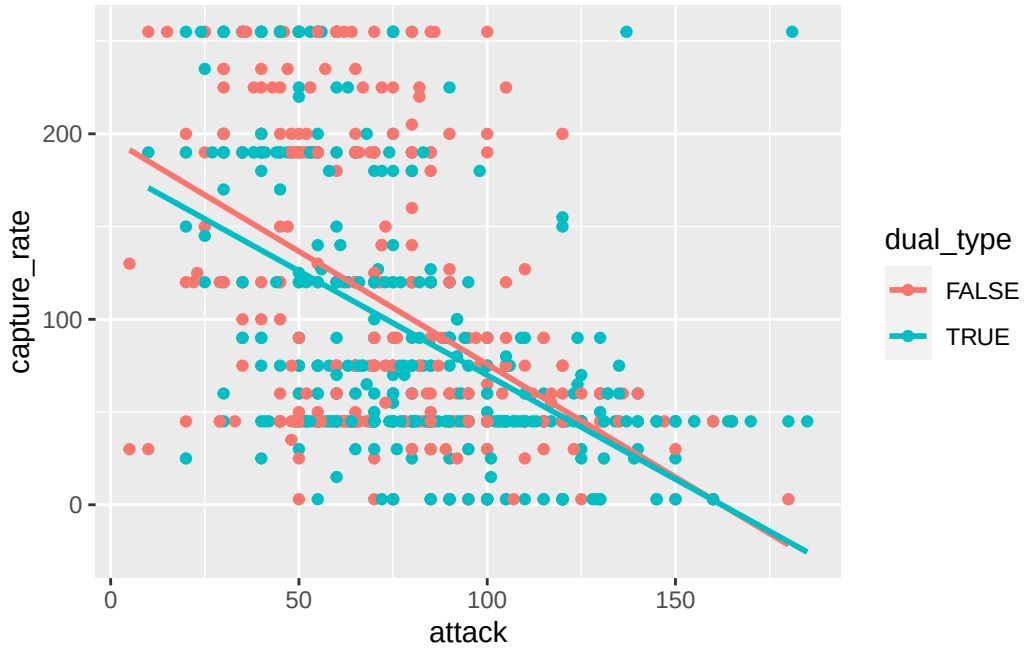
So I'll be using attack, speed, hp, defense,and dual_type as explanatory variables to predict capture rate. I'll make the mode, and tey to add to the visualizations but haven't tried using new base stat explanatory variables since I'm not sure if using adjusted r-squared or a nested f-test would be the best way to compare so once we know that we can drop, add,or try new variables a bit better.

Visualizations

```
ggplot(data = pokemon, mapping = aes(x = attack, y = capture_rate, color=dual_type)) +
  geom_point() +
  geom_smooth(method = lm, se = FALSE, formula = y~x)
```

```
Warning: Removed 1 rows containing non-finite values (stat_smooth).
```

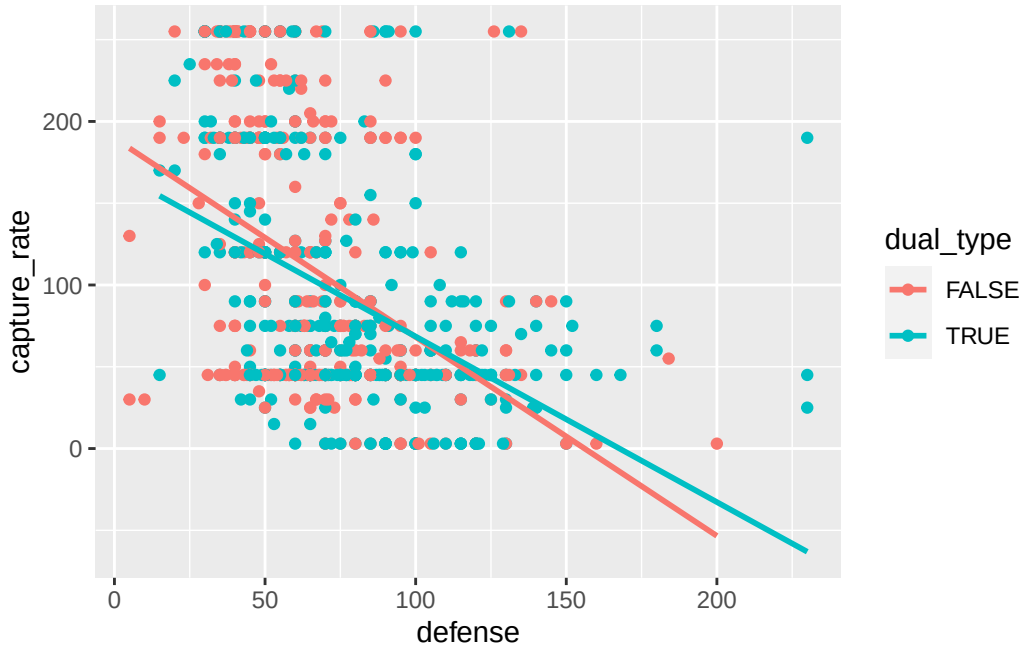
```
Warning: Removed 1 rows containing missing values (geom_point).
```



```
ggplot(data = pokemon, mapping = aes(x = defense, y = capture_rate, color=dual_type)) +
  geom_point() +
  geom_smooth(method = lm, se = FALSE, formula = y~x)
```

```
Warning: Removed 1 rows containing non-finite values (stat_smooth).
```

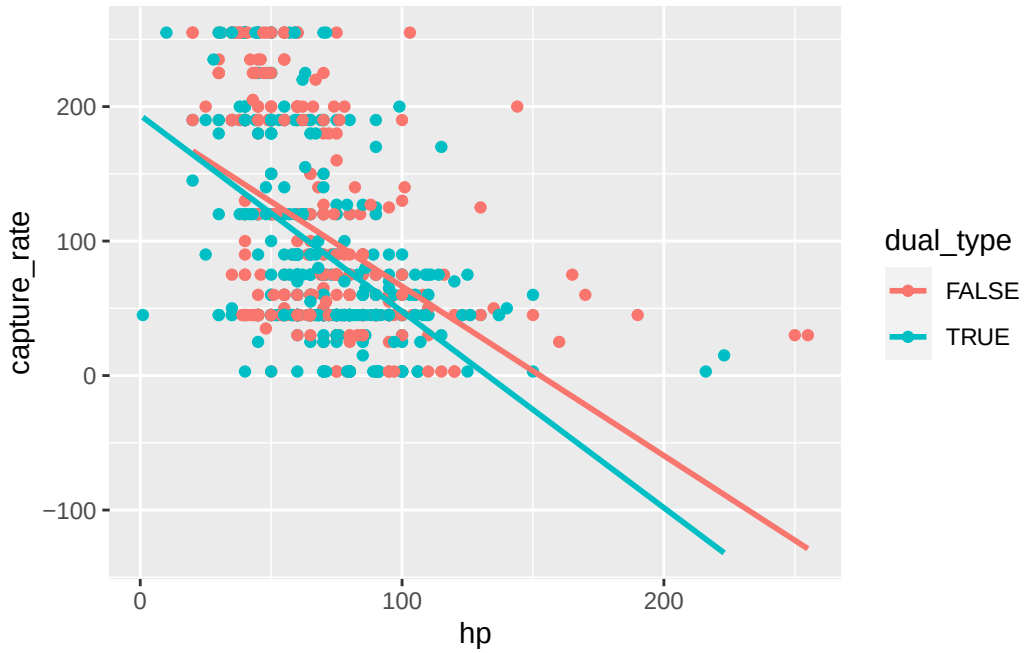
```
Warning: Removed 1 rows containing missing values (geom_point).
```




```
ggplot(data = pokemon, mapping = aes(x = hp, y = capture_rate, color=dual_type)) +
  geom_point() +
  geom_smooth(method = lm, se = FALSE, formula = y~x)
```

```
Warning: Removed 1 rows containing non-finite values (stat_smooth).
```

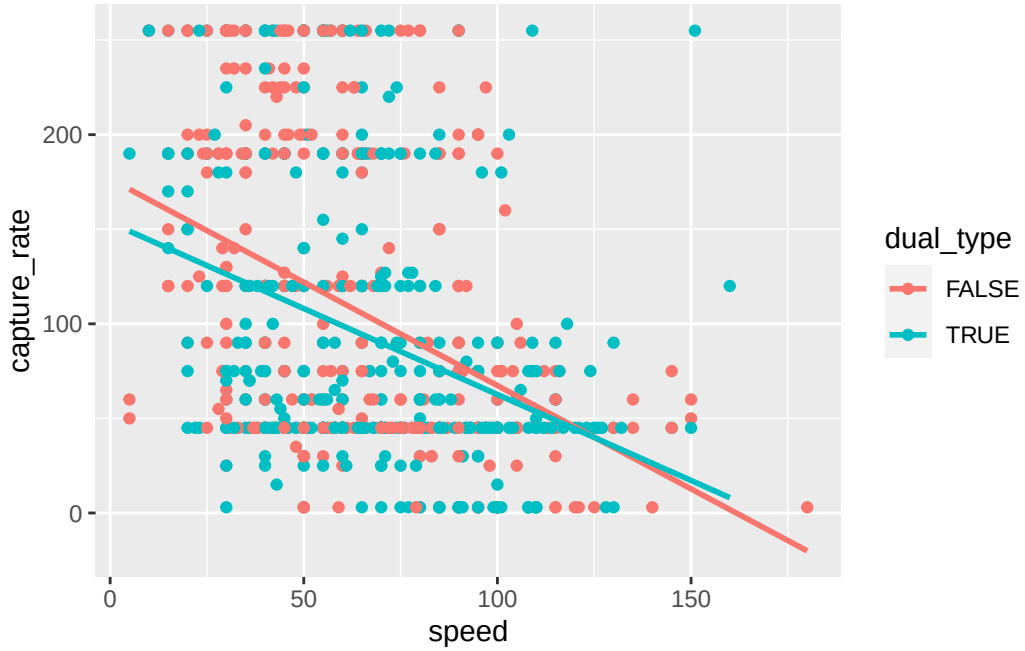
```
Warning: Removed 1 rows containing missing values (geom_point).
```




```
ggplot(data = pokemon, mapping = aes(x = speed, y = capture_rate, color=dual_type)) +
  geom_point() +
  geom_smooth(method = lm, se = FALSE, formula = y~x)
```

```
Warning: Removed 1 rows containing non-finite values (stat_smooth).
```

```
Warning: Removed 1 rows containing missing values (geom_point).
```



Making the models

Additive Model

```
add_capture_model3 <- lm(capture_rate ~ attack + defense + hp + speed + dual_type, data =
```

```
summary(add_capture_model3)
```

```
Call:
lm(formula = capture_rate ~ attack + defense + hp + speed + dual_type,
    data = pokemon)

Residuals:
     Min       1Q   Median       3Q      Max
-150.647  -36.921   -4.964   36.208  255.453

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   288.55739    7.74097  37.277  < 2e-16 ***
```

```
attack         -0.29463    0.08047  -3.661 0.000267 ***
defense        -0.77304    0.07567 -10.217  < 2e-16 ***
hp             -0.86777    0.08256 -10.511  < 2e-16 ***
speed          -0.76053    0.07554 -10.068  < 2e-16 ***
dual_typeTRUE  -0.31776    4.06062  -0.078 0.937646
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 56.43 on 794 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.4558,    Adjusted R-squared:  0.4523
F-statistic:   133 on 5 and 794 DF,  p-value: < 2.2e-16
```

Adjusted R-squared: 0.4523

Interactive Model

```r
interact_capture_model3 <- lm(capture_rate ~ attack * defense * hp * speed * dual_type, da
```

```r
summary(interact_capture_model3)
```

```
Call:
lm(formula = capture_rate ~ attack * defense * hp * speed * dual_type,
    data = pokemon)

Residuals:
    Min      1Q  Median      3Q     Max
-180.69  -31.66   -2.61   30.32  216.10

Coefficients:
                                    Estimate Std. Error t value Pr(>|t|)
(Intercept)                        1.605e+02  9.847e+01   1.630   0.1035
attack                             2.804e+00  1.601e+00   1.752   0.0802 .
defense                            9.493e-01  1.494e+00   0.635   0.5254
hp                                -7.295e-02  1.408e+00  -0.052   0.9587
speed                              1.957e+00  1.692e+00   1.157   0.2477
dual_typeTRUE                      1.818e+02  1.382e+02   1.316   0.1886
attack:defense                    -3.548e-02  2.065e-02  -1.718   0.0862 .
attack:hp                         -2.844e-02  1.927e-02  -1.476   0.1404
```

```
defense:hp                                -1.052e-02  2.184e-02  -0.482   0.6303
attack:speed                              -6.283e-02  2.559e-02  -2.455   0.0143 *
defense:speed                             -2.841e-02  2.636e-02  -1.078   0.2815
hp:speed                                  -1.533e-02  2.578e-02  -0.595   0.5522
attack:dual_typeTRUE                      -4.673e+00  2.263e+00  -2.065   0.0393 *
defense:dual_typeTRUE                     -1.145e+00  1.645e+00  -0.696   0.4865
hp:dual_typeTRUE                          -2.954e-01  2.103e+00  -0.140   0.8883
speed:dual_typeTRUE                       -1.211e+00  2.364e+00  -0.512   0.6088
attack:defense:hp                          3.114e-04  2.575e-04   1.209   0.2269
attack:defense:speed                       6.150e-04  3.243e-04   1.896   0.0583 .
attack:hp:speed                            5.601e-04  3.237e-04   1.730   0.0840 .
defense:hp:speed                           1.040e-04  3.763e-04   0.276   0.7823
attack:defense:dual_typeTRUE               3.641e-02  2.533e-02   1.437   0.1511
attack:hp:dual_typeTRUE                    2.736e-02  3.026e-02   0.904   0.3663
defense:hp:dual_typeTRUE                  -3.543e-03  2.594e-02  -0.137   0.8914
attack:speed:dual_typeTRUE                 6.003e-02  3.591e-02   1.671   0.0951 .
defense:speed:dual_typeTRUE               -9.959e-03  3.045e-02  -0.327   0.7437
hp:speed:dual_typeTRUE                    -1.139e-02  3.673e-02  -0.310   0.7565
attack:defense:hp:speed                   -4.839e-06  4.045e-06  -1.196   0.2320
attack:defense:hp:dual_typeTRUE           -1.865e-04  3.375e-04  -0.552   0.5808
attack:defense:speed:dual_typeTRUE        -3.515e-04  3.985e-04  -0.882   0.3781
attack:hp:speed:dual_typeTRUE             -3.609e-04  4.901e-04  -0.736   0.4617
defense:hp:speed:dual_typeTRUE             3.258e-04  4.559e-04   0.715   0.4750
attack:defense:hp:speed:dual_typeTRUE      1.376e-06  5.313e-06   0.259   0.7957
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 54.88 on 768 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.5022,    Adjusted R-squared:  0.4821
F-statistic: 24.99 on 31 and 768 DF,  p-value: < 2.2e-16
```

Adjusted R-squared: 0.4821

Nested F-test

```
anova(add_capture_model3, interact_capture_model3)
```

```
Analysis of Variance Table

Model 1: capture_rate ~ attack + defense + hp + speed + dual_type
```

```
Model 2: capture_rate ~ attack * defense * hp * speed * dual_type
  Res.Df     RSS Df Sum of Sq     F    Pr(>F)
1    794 2528677
2    768 2313168 26    215508 2.752 7.923e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-value when comparing the additive and interactive model for attack, speed, hp, defense, and dual_type is below 0.05 so it seems like there is some significance for going from the additive model to an interactive model. The difference in adjusted r-squared is 0.0298 which doesn't seem that significant.

Also when looking at the adjusted r-squared of the additive model when adding dual_type, when adding dual_type it is lower meaning it fits the model slightly less, so the extra complexity isn't needed since it doesn't help or model.

When looking at the adjusted r-squared between the interaction models shows that the dual_type model to the interaction model that shows that adding it improved the adjusted r-squared by 0.0039 which is really small so I argue it doesn't add to these 4 base stats based on adjusted r-squared.

```r
additive_capture_model4 <- lm(capture_rate ~ attack + defense + hp + speed + base_total, d
```

```r
vif(additive_capture_model4)
```

```
   attack    defense         hp      speed base_total
 2.198357   2.924339   2.293341   2.586556   8.718589
```