Modeling Data

Dominguez Center for Data Science Workshop

2025-03-21

Questions Round-Up

How do we add a totals row at the bottom of a table?

See the "Wrangling Extra" Project on Posit Cloud for examples.

Any other questions?

Plan for today

- Linear regression models in `R`. Let's look at lots of different examples.
- Remember that you can either fill in the "modeling.qmd" file or follow along with the "modeling_key.qmd" file.
  - Both documents have code related to topics we covered in previous weeks already filled in.

Disclaimer

Modeling is a HUGE topic. We are going to focus more on the how to build models in `R` and less on deciding which model to use and what variables to include.

I recommend thinking about your modeling goals first and then using your goals to determine the type of model.

Modeling Goals:

Descriptive: Want to estimate quantities related to the population.

Ex: How many trees are in Alaska?

Predictive: Want to predict the value of a variable.

Ex: Can I use remotely sensed data to predict forest types in Alaska?

Causal: Want to determine if changes in a variable cause changes in another variable.

Ex: Are insects causing the increased mortality rates for pinyon-juniper woodlands?

Linear regression models are good for descriptive and causal goals.

Load Packages

```
# For data wrangling and plotting
library(tidyverse)

# For diagnostic plots
library(gglm)

# To add labels to ggplots
library(ggrepel)
```

Simple Linear Regression

Consider this model when:

- Response variable (y): quantitative
- Explanatory variable (x): quantitative (simple = 1 explanatory variable)

And the relationship between x and y is reasonably approximated by a line:

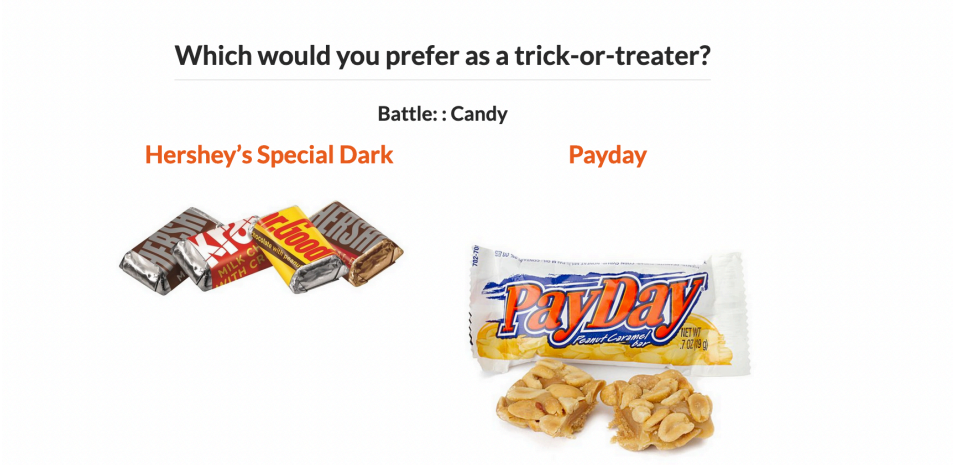
Form of the Model:

$$y = \beta_o + \beta_1 x + \epsilon$$

Data Example:

> "The social contract of Halloween is simple: Provide adequate treats to costumed masses, or be prepared for late-night tricks from those dissatisfied with your offer. To help you avoid that type of vengeance, and to help you make good decisions at the supermarket this weekend, we wanted to figure out what Halloween candy people most prefer. So we devised an experiment: Pit dozens of fun-sized candy varietals against one another, and let the wisdom of the crowd decide which one was best." – Walt Hickey

> "While we don't know who exactly voted, we do know this: 8,371 different IP addresses voted on about 269,000 randomly generated matchups.2 So, not a scientific survey or anything, but a good sample of what candy people like."



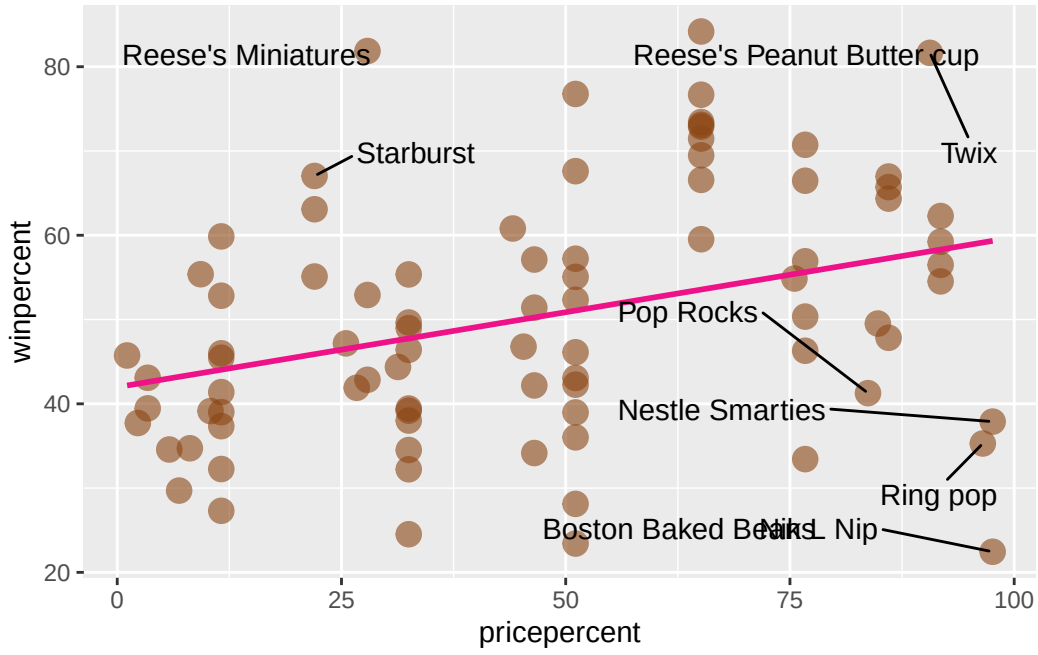

```
candy <- read_csv("https://raw.githubusercontent.com/fivethirtyeight/data/master/candy-power
  mutate(pricepercent = pricepercent*100)


glimpse(candy)
```

```
Rows: 85
Columns: 13
$ competitorname   <chr> "100 Grand", "3 Musketeers", "One dime", "One quarter~
$ chocolate        <dbl> 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,~
$ fruity           <dbl> 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1,~
$ caramel          <dbl> 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,~
$ peanutyalmondy   <dbl> 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
$ nougat           <dbl> 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,~
```

```
$ crispedricewafer <dbl> 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
$ hard             <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1,~
$ bar              <dbl> 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,~
$ pluribus         <dbl> 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1,~
$ sugarpercent     <dbl> 0.732, 0.604, 0.011, 0.011, 0.906, 0.465, 0.604, 0.31~
$ pricepercent     <dbl> 86.0, 51.1, 11.6, 51.1, 51.1, 76.7, 76.7, 51.1, 32.5,~
$ winpercent       <dbl> 66.97173, 67.60294, 32.26109, 46.11650, 52.34146, 50.~
```

Does a linear seem reasonable?

```
# Always graph the data!
ggplot(data = candy,
       mapping = aes(x = pricepercent,
                     y = winpercent)) +
  geom_point(alpha = 0.6, size = 4,
             color = "chocolate4") +
  geom_smooth(method = "lm", se = FALSE,
              color = "deeppink2") +
  geom_text_repel(aes(label = competitorname), size = 4,
                  force = 2,
                  box.padding = 1)
```



```r
# Can also compute the correlation coefficient
candy %>%
  summarize(cor = cor(pricepercent, winpercent))
```

```
# A tibble: 1 x 1
    cor
  <dbl>
1 0.345
```

Linear Model in R

```r
mod <- lm(winpercent ~ pricepercent, data = candy)
summary(mod)
```

```
Call:
lm(formula = winpercent ~ pricepercent, data = candy)

Residuals:
    Min      1Q  Median      3Q     Max
-36.889  -8.573  -0.544   8.784  34.926

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  41.97874    2.90809  14.435  < 2e-16 ***
pricepercent  0.17783    0.05305   3.352  0.00121 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 13.89 on 83 degrees of freedom
Multiple R-squared:  0.1192,    Adjusted R-squared:  0.1086
F-statistic: 11.24 on 1 and 83 DF,  p-value: 0.001209
```

Checking Model Assumptions

Assumptions:

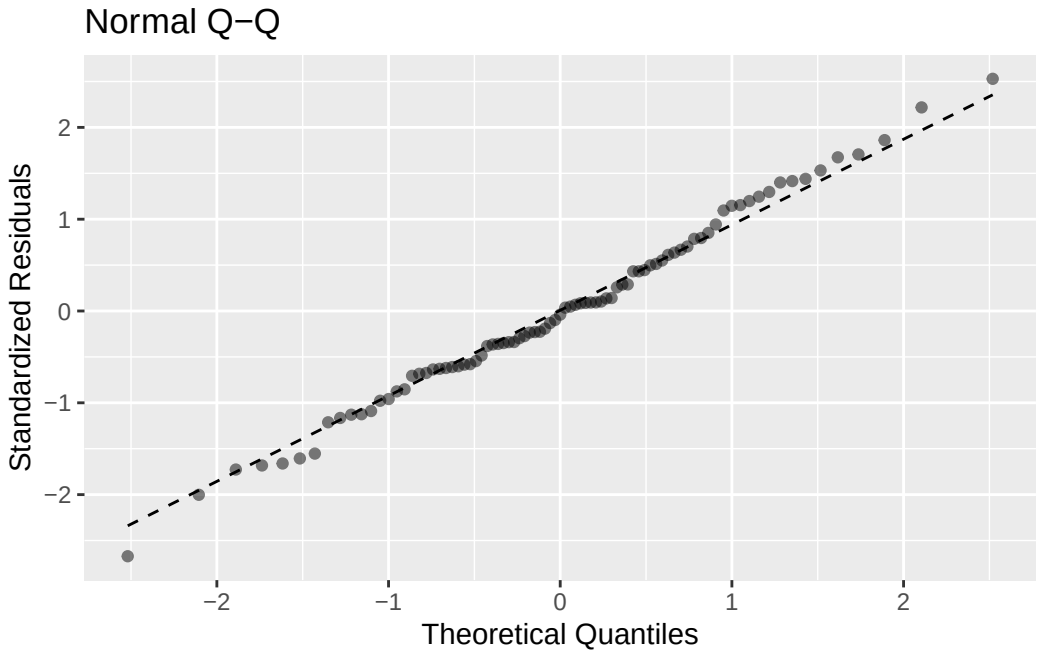
- Model errors are normally.
- Observations are independent.

- Model form is appropriate.
- The variability in the errors is constant.

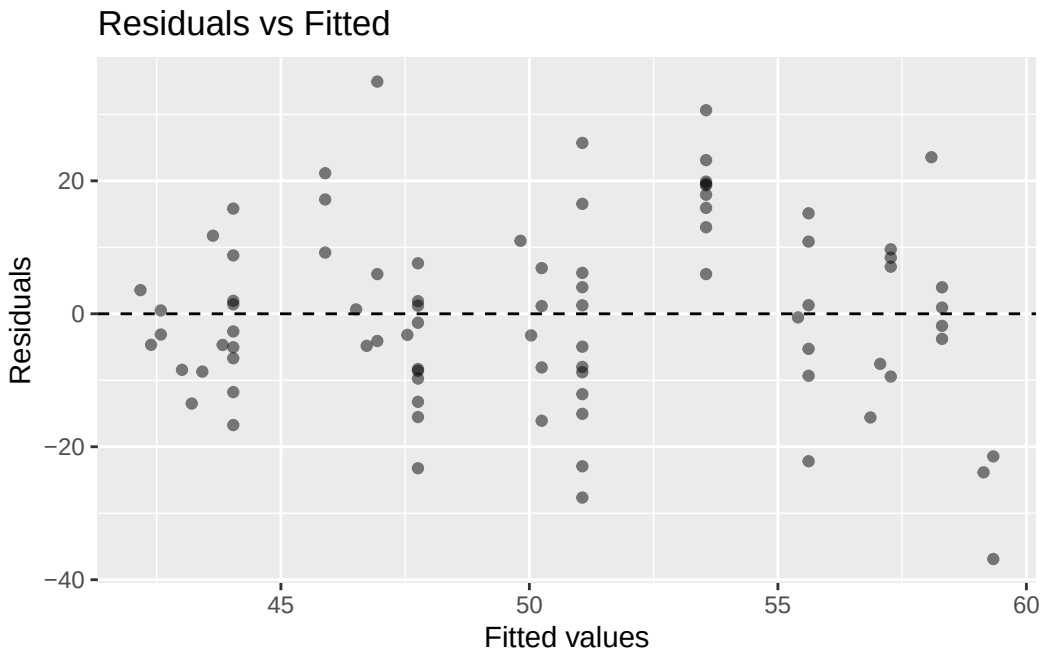
Key term: residual $= e = y - \hat{y}$

Useful graphs for checking assumptions are the QQ Plot and the Residual Plot.

```
# QQ Plot - Normality of errors
ggplot(data = mod) +
  stat_normal_qq()
```




```
# Residual Plot
ggplot(data = mod) +
  stat_fitted_resid()
```



Prediction

Careful: R will let you predict outside the range of your explanatory variable!

```
new_cases <- data.frame(pricepercent = c(25, 85, 150))
predict(mod, newdata = new_cases, interval = "prediction", level = 0.95)
```

```
       fit      lwr      upr
1 46.42443 18.53574 74.31312
2 57.09409 29.01180 85.17638
3 68.65289 38.80661 98.49917
```

```
predict(mod, newdata = new_cases, interval = "confidence", level = 0.95)
```

```
       fit      lwr      upr
1 46.42443 42.64089 50.20797
2 57.09409 52.07901 62.10917
3 68.65289 57.36854 79.93724
```

Multiple Linear Regression

We are going to look at different data examples that need different forms of the multiple linear regression model. We are not going to check model assumptions because the code and interpretation of the output is the same as in the simple linear regression case.

General model form:

$$y = \beta_o + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \epsilon$$

Equal Slopes Model

Data Example

Meadowfoam is a plant that grows in the Pacific Northwest and is harvested for its seed oil. In a randomized experiment, researchers at Oregon State University looked at how two light-related factors influenced the number of flowers per meadowfoam plant, the primary measure of productivity for this plant. The two light measures were light intensity (in mmol/ m^2 /sec) and the timing of onset of the light (early or late in terms of photo periodic floral induction).

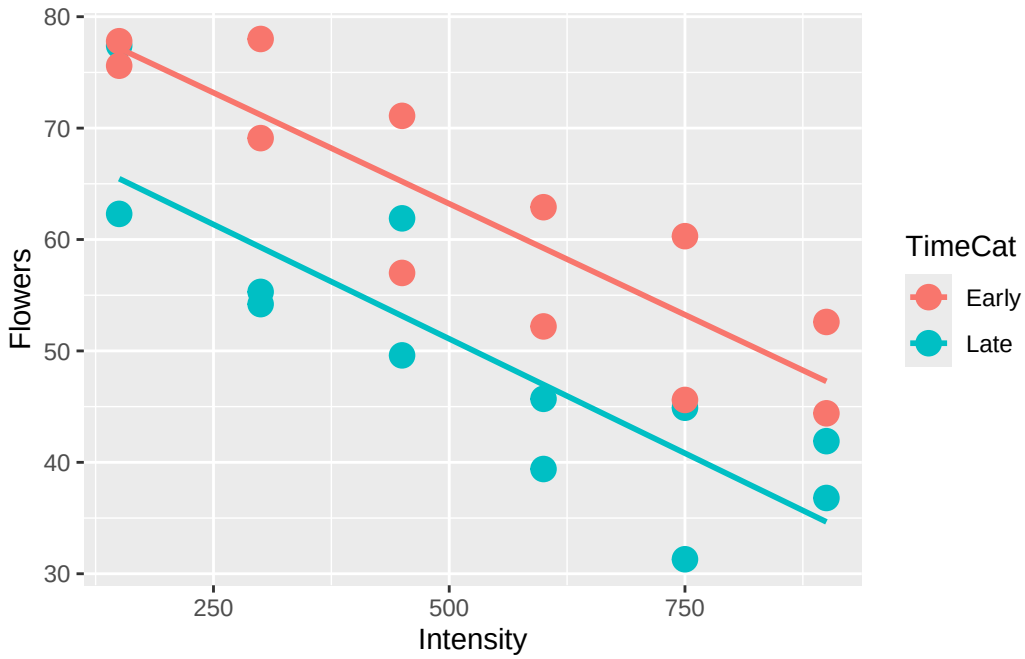
Let's first load and wrangle the data.

```
library(tidyverse)
library(Sleuth3)
data(case0901)

# Recode the timing variable
case0901 <- case0901 %>%
  mutate(TimeCat = case_match(Time,
                              1 ~ "Late",
                              2 ~ "Early"
    ))

count(case0901, Time, TimeCat)
```

```
  Time TimeCat  n
1    1    Late 12
2    2   Early 12
```

Next let's visualize the data.

```
ggplot(data = case0901,
       mapping = aes(x = Intensity,
                     y = Flowers,
                     color = TimeCat)) +
  geom_point(size = 4) +
  geom_smooth(method = lm, se = FALSE)
```



And, then we can fit the model and use it for prediction:

```
# Fit the model
modFlowers <- lm(Flowers ~ Intensity + TimeCat, data = case0901)
summary(modFlowers)
```

```
Call:
lm(formula = Flowers ~ Intensity + TimeCat, data = case0901)

Residuals:
   Min     1Q Median     3Q    Max
-9.652 -4.139 -1.558  5.632 12.165

Coefficients:
```

```
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  83.464167   3.273772  25.495  < 2e-16 ***
Intensity    -0.040471   0.005132  -7.886 1.04e-07 ***
TimeCatLate -12.158333   2.629557  -4.624 0.000146 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.441 on 21 degrees of freedom
Multiple R-squared:  0.7992,    Adjusted R-squared:   0.78
F-statistic: 41.78 on 2 and 21 DF,  p-value: 4.786e-08
```

```r
# Predict new values
flowersNew <- data.frame(Intensity = c(700, 700), TimeCat = c("Early", "Late"))
flowersNew
```

```
  Intensity TimeCat
1       700   Early
2       700    Late
```

```r
predict(modFlowers, newdata = flowersNew, interval = "prediction", level = 0.95)
```

```
       fit      lwr      upr
1 55.13417 41.06770 69.20063
2 42.97583 28.90937 57.04230
```

Question: Is the assumption of **equal** slopes reasonsable here?

Different Slopes Model

Data Example

For this example, we will use data collected by the website pollster.com, which aggregated 102 presidential polls from August 29th, 2008 through the end of September. We want to determine the best model to explain the variable `Margin`, measured by the difference in preference between Barack Obama and John McCain. Our potential predictors are `Days` (the number of days after the Democratic Convention) and `Charlie` (indicator variable on whether poll was conducted before or after the first ABC interview of Sarah Palin with Charlie Gibson).

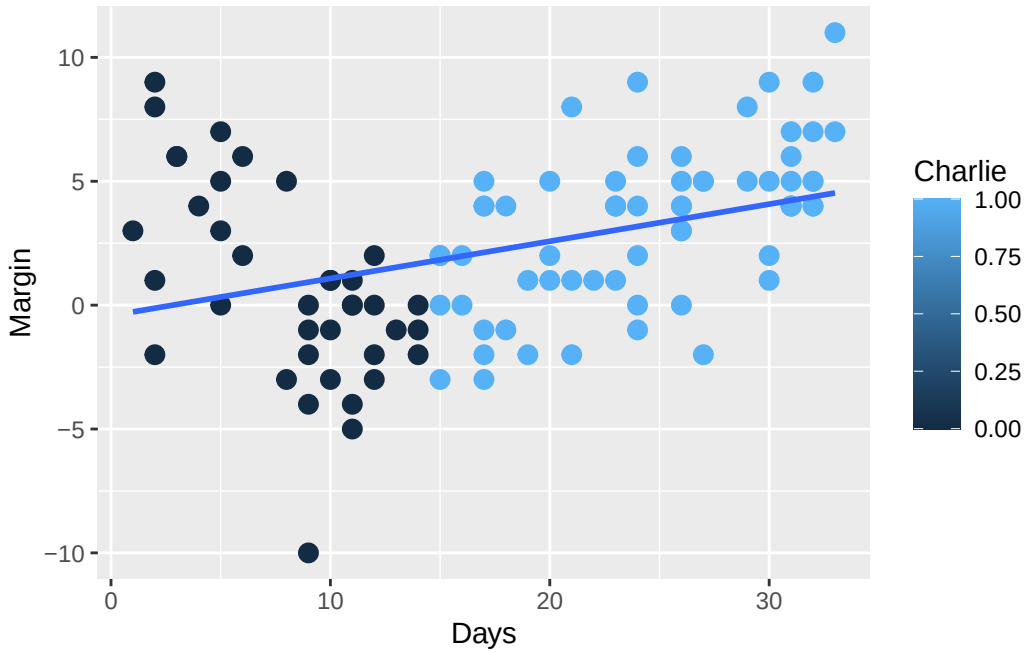
Let's load the data.

```
library(Stat2Data)
data("Pollster08")
glimpse(Pollster08)
```

```
Rows: 102
Columns: 11
$ PollTaker <fct> Rasmussen, Zogby, Diageo/Hotline, CBS, CNN, Rasmussen, ARG, ~
$ PollDates <fct> 8/28-30/08, 8/29-30/08, 8/29-31/08, 8/29-31/08, 8/29-31/08, ~
$ MidDate   <fct> 8/29, 8/30, 8/30, 8/30, 8/30, 8/31, 8/31, 9/1, 9/2, 9/2, 9/2~
$ Days      <int> 1, 2, 2, 2, 2, 3, 3, 4, 5, 5, 5, 5, 6, 6, 8, 8, 9, 9, 9, 9, ~
$ n         <int> 3000, 2020, 805, 781, 927, 3000, 1200, 1728, 2771, 1000, 734~
$ Pop       <fct> LV, LV, RV, RV, RV, LV, LV, RV, RV, A, RV, LV, LV, RV, RV, R~
$ McCain    <int> 46, 47, 39, 40, 48, 45, 43, 36, 42, 39, 42, 44, 46, 40, 48, ~
$ Obama     <int> 49, 45, 48, 48, 49, 51, 49, 40, 49, 42, 42, 49, 48, 46, 45, ~
$ Margin    <int> 3, -2, 9, 8, 1, 6, 6, 4, 7, 3, 0, 5, 2, 6, -3, 5, -4, -1, -2~
$ Charlie   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
$ Meltdown  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
```

And, then visualize the data.

```
ggplot(Pollster08,
       aes(x = Days,
           y = Margin,
           color = Charlie)) +
  geom_point(size = 3) +
  geom_smooth(method = lm, se = FALSE)
```



Questions:

- What is wrong with how one of the variables is mapped in the graph?
- Is the assumption of **equal slopes** reasonable here?

For the different slopes model, we need to include an interaction term.

```
modPoll <- lm(Margin ~ factor(Charlie)*Days, data = Pollster08)
summary(modPoll)
```

```
Call:
lm(formula = Margin ~ factor(Charlie) * Days, data = Pollster08)

Residuals:
     Min       1Q   Median       3Q      Max
-10.1803  -1.7702   0.1641   1.7862   5.8089

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)            5.5656     1.0885   5.113 1.57e-06 ***
factor(Charlie)1     -10.1117     1.9251  -5.253 8.74e-07 ***
Days                  -0.5984     0.1206  -4.960 2.96e-06 ***
```

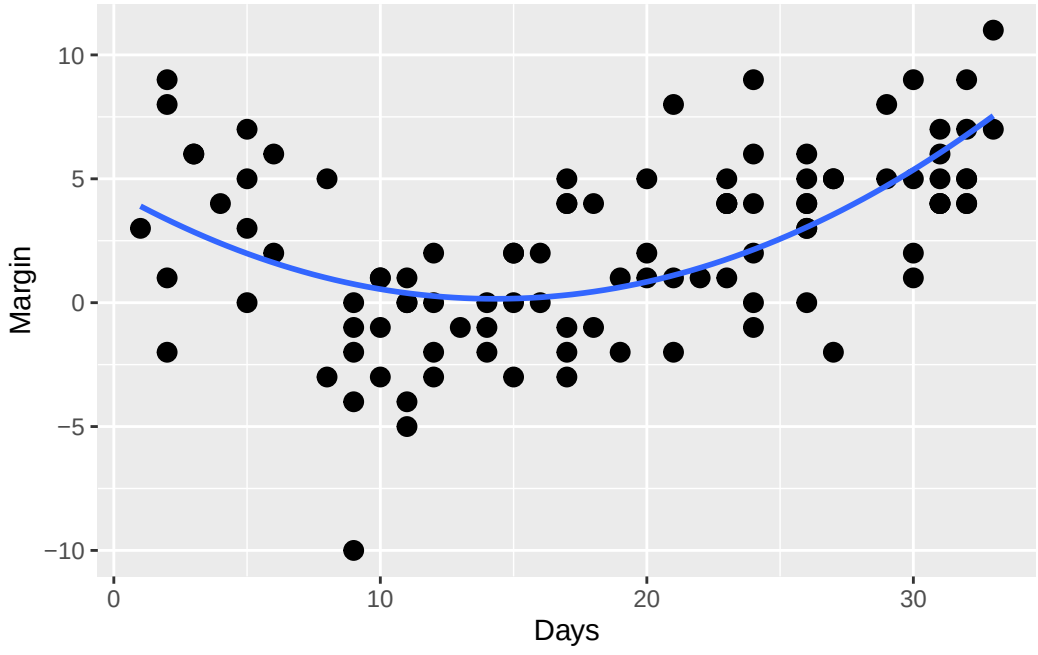
```
factor(Charlie)1:Days   0.9207     0.1364   6.752 1.04e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.868 on 98 degrees of freedom
Multiple R-squared:  0.417, Adjusted R-squared:  0.3992
F-statistic: 23.37 on 3 and 98 DF,  p-value: 1.712e-11
```

Polynomial Terms

Instead of using the categorical `Charlie` variable, we could also try fitting a parabola. This also fits under the umbrella of multiple linear regression.

We can add a parabola to our graph.

```
ggplot(Pollster08,
       aes(x = Days,
           y = Margin)) +
  geom_point(size = 3) +
  geom_smooth(method = lm, se = FALSE,
              formula = y ~ poly(x, degree = 2))
```



And, we can add higher order terms to the model.

```r
modPollPoly <- lm(Margin ~ poly(Days, degree = 2, raw = TRUE), data = Pollster08)
summary(modPollPoly)
```

```
Call:
lm(formula = Margin ~ poly(Days, degree = 2, raw = TRUE), data = Pollster08)

Residuals:
     Min       1Q   Median       3Q      Max
-10.7496  -2.0461  -0.1227   1.9297   6.8969

Coefficients:
                                     Estimate Std. Error t value Pr(>|t|)
(Intercept)                          4.477958   1.095676   4.087 8.89e-05 ***
poly(Days, degree = 2, raw = TRUE)1 -0.604426   0.138598  -4.361 3.18e-05 ***
poly(Days, degree = 2, raw = TRUE)2  0.021129   0.003776   5.595 1.97e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.014 on 99 degrees of freedom
Multiple R-squared:  0.3495,    Adjusted R-squared:  0.3363
F-statistic: 26.59 on 2 and 99 DF,  p-value: 5.711e-10
```

Homework

No optional homework this week. You will receive some practice problems after our next modeling session.