# Linear Regression

Aleksandr Fisher

4/17/2020

# Introduction

- How can we use one variable to predict another?
- Big technical tool: linear regression

# Predicting Happiness

- Can we use a country's income to predict it's citizens' level of happiness?

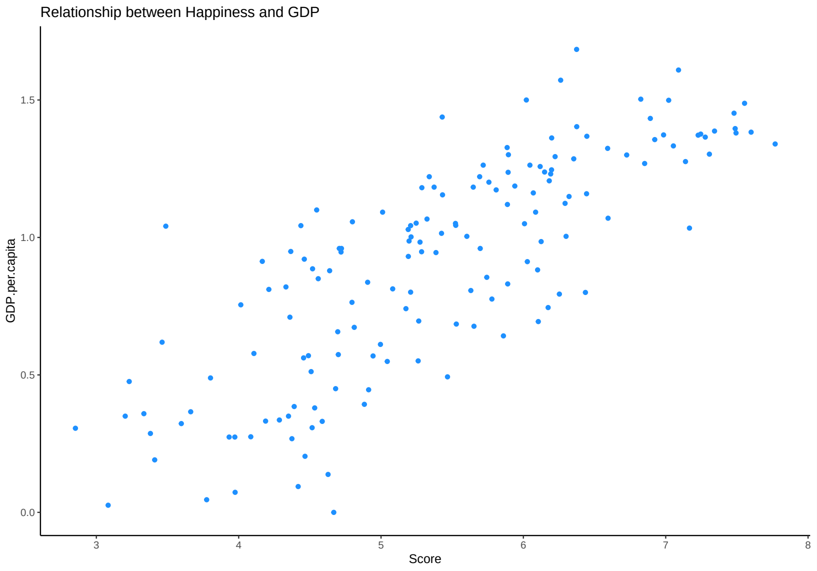
```r
# Read Happiness Data
happ2019 = read.csv("C:/Users/afisher/Documents/R Code/Resources/Data/Happiness/2019.csv")

# Structure of dataset
str(happ2019)
```

```
## 'data.frame':    156 obs. of  9 variables:
##  $ Overall.rank                : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Country.or.region           : Factor w/ 156 levels "Afghanistan",..: 44 37 106 58 99 134 133 100 24
##  $ Score                       : num  7.77 7.6 7.55 7.49 7.49 ...
##  $ GDP.per.capita              : num  1.34 1.38 1.49 1.38 1.4 ...
##  $ Social.support              : num  1.59 1.57 1.58 1.62 1.52 ...
##  $ Healthy.life.expectancy     : num  0.986 0.996 1.028 1.026 0.999 ...
##  $ Freedom.to.make.life.choices: num  0.596 0.592 0.603 0.591 0.557 0.572 0.574 0.585 0.584 0.532 ...
##  $ Generosity                  : num  0.153 0.252 0.271 0.354 0.322 0.263 0.267 0.33 0.285 0.244 ...
##  $ Perceptions.of.corruption   : num  0.393 0.41 0.341 0.118 0.298 0.343 0.373 0.38 0.308 0.226 ...
```

## Predicting using bivariate relationship

- Goal: What's our best guess about Y if we know what X is?
    - what's our best guess about a country's happiness if I know its income level?
- Terminology:
    - **Dependent/outcome variable:** the variable we want to predict (happiness).
    - **Independent/explanatory variable:** the variable we're using to predict (GDP per capita).

# Plotting the data



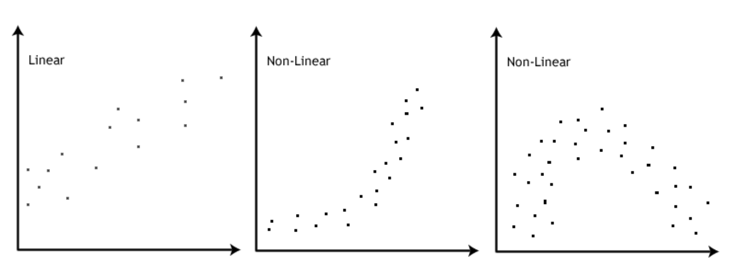
# Correlation and scatter-plots:

Recall the definition of correlation:

$$r = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2(y_i - \overline{y})^2}}$$

- positive correlation $\sim$ upward slope
- negative correlation $\sim$ downward slope
- high correlation $\sim$ tighter, closer to a line
- correlation cannot capture nonlinear relationship.

# Must be linear!



# Linear Regression Model

- Prediction: for any value of X, what's the best guess about Y?
- Simplest possible way to relate two variables: a line.

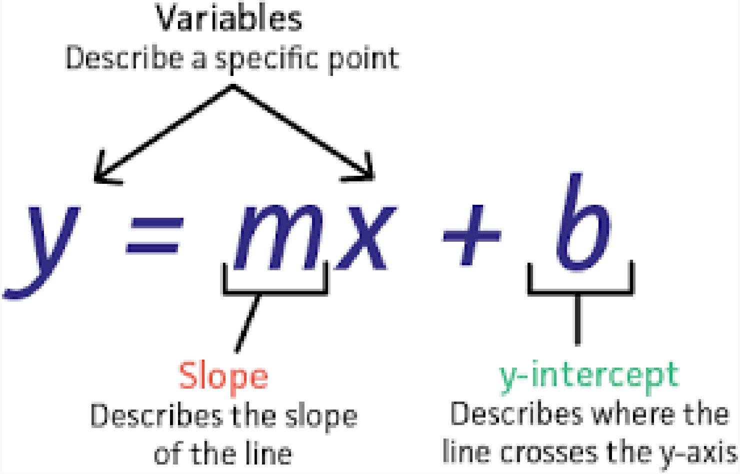
$$y = mx + b$$

- Where:
  - y = how far up
  - x = how far along
  - m = Slope or Gradient (how steep the line is)
  - b = the Y Intercept (where the line crosses the Y axis)

# Linear Regression Model

- Problem: for any line we draw, not all the data is on the line.
    - Some weights will be above the line, some below.
    - Need a way to account for chance variation away from the line

## A line


Variables
Describe a specific point
y = mx + b
Slope
Describes the slope of the line
y-intercept
Describes where the line crosses the y-axis

# Linear Regression Model

- Model for the line of best fit:

Population regression line:

$$Y_i = \underbrace{\beta_0}_{\text{intercept}} + \underbrace{\beta_1}_{\text{slope}} X_i + \underbrace{\epsilon_i}_{\text{error term}}$$

- **Coefficients/parameters($\alpha, \beta$):** true unknown intercept/slope of the line of best fit.
- **Chance error ($\epsilon$):** accounts for the fact that the line doesn't perfectly fit the data.
  - Each observation allowed to be off the regression line.
  - Chance errors are 0 on average.

# Interpreting the regression line

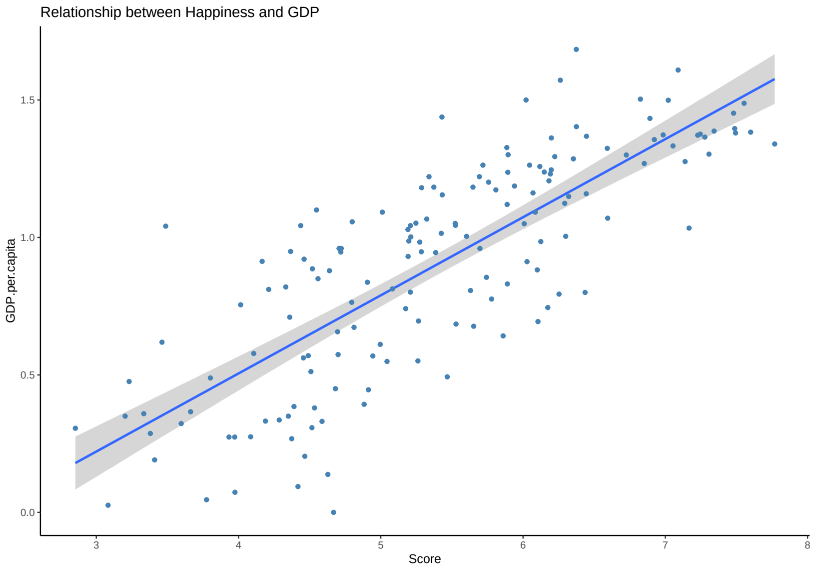
$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$$

- **Intercept** $\alpha$: average of Y when X is 0
  - Average happiness when I GDP is 0.
- **Slope** $\beta$: average change in Y when X increase by one unit.
  - Average increase in happiness when gdp increases by 1 unit (what unit is your variable in?)
- But we don't know $\alpha$ or $\beta$ is. How do we estimate it?

# Estimated Coefficients

- Parameters: $\alpha, \beta$
  - Unknown features of the **data-generating process**
  - Chance error makes these impossible to observe directly
- Estimates $\hat{\alpha}, \hat{\beta}$
  - An **estimate** is a function of the data that is our best guess about some parameter
- **Regression line:** $\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 X_i + \hat{\epsilon}_i$
- Average value of Y when X is equal to x
- Represents the best guess or **predicted value** of the outcome at x

# Plotting our data



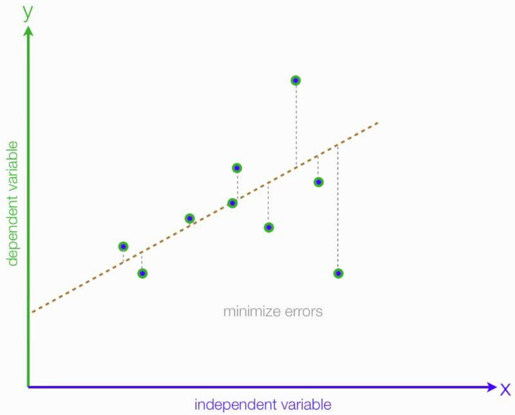
## Least Squares

- How do we figure out the best line to draw?
    - **Fitted/predicted value** for each observation: $\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 X_i + \hat{\epsilon}_i$
    - **Residual/prediction error:** $\hat{\epsilon}_i = Y_i - \hat{Y}_i$
- Get these estiamtes by the **least squares method**
- Minimize the **sum of the squared resisduals (SSR)**:

$$\sum_{j=1}^{n}(y_j - \beta_0 - \beta_1 x_j)^2.$$

- This find the line that minimizes the magnitude of the prediction errors

## Minimize the errors



# Linear Regression in R

- `R` will calculate least squares line for a data set using `lm()`.
- Jargon: "fit the model"
- Syntax: `lm(y ~ x, data = mydata)`
- `y` is the name of the dependent variance, `x` is the name of the independent variable and `mydata` is the data.frame where they live

# Linear Regression in R

```r
fit = lm(Score ~ GDP.per.capita, data=happ2019)
fit
```

```
##
## Call:
## lm(formula = Score ~ GDP.per.capita, data = happ2019)
##
## Coefficients:
##    (Intercept)  GDP.per.capita
##          3.399           2.218
```

- What does this mean?

# Coefficients and fitted values

- Use `coef()` to extract estimated coefficients:

```
coef(fit)
```

```
##    (Intercept) GDP.per.capita
##       3.399345       2.218148
```

- R can show you eachof the fitted values as well:

```
head(fitted(fit))
```

```
##        1        2        3        4        5        6
## 6.371663 6.467044 6.699949 6.460389 6.495880 6.620096
```

## Properties of least squares

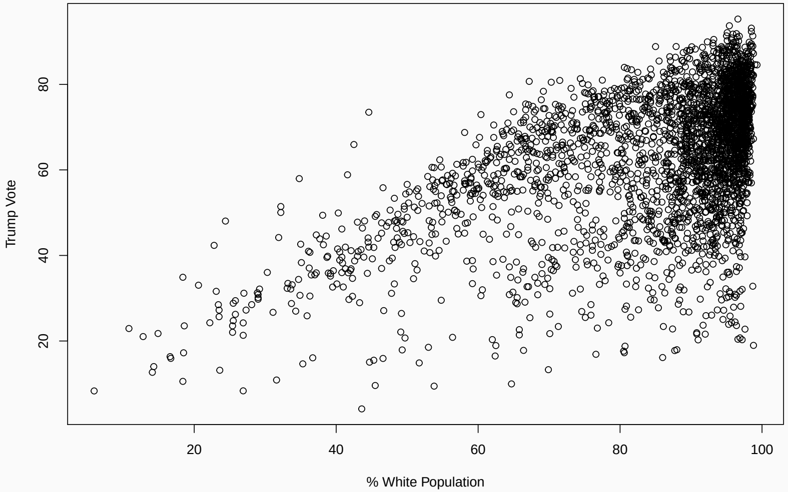
- Least squares line always goes throught $(\bar{X}, \bar{Y})$
- Estimated slop is relatd to correlation

$$\hat{\beta} = \text{(correlation of X AND Y)} \times \frac{\text{SD of Y}}{\text{SD of X}}$$

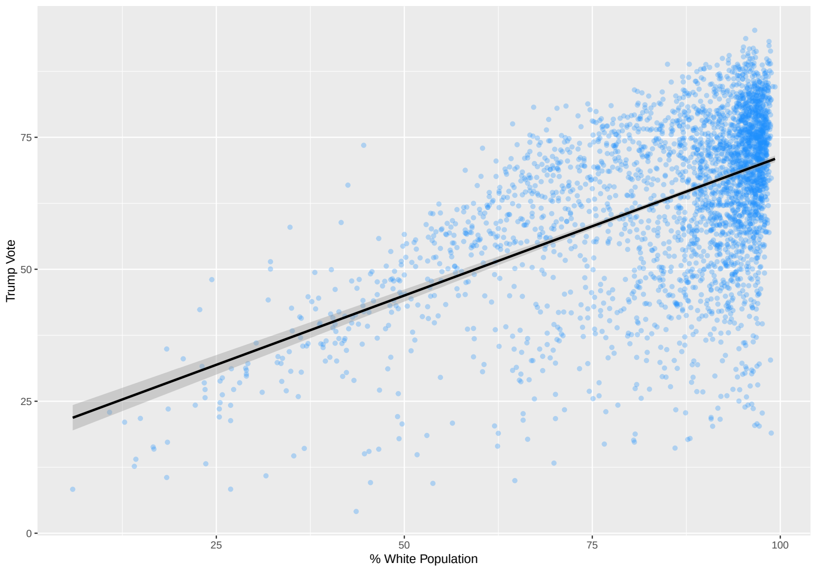
- mean of residuals is always 0

## Looking at the 2016 Election

# White Population and Trump Vote (Base R)



# White Population and Trump Vote (ggplot)



# Let's run our first regression!

```r
## Linear Regression
m1 = lm(Trump ~ White, data=votes)
m1
```

```
##
## Call:
## lm(formula = Trump ~ White, data = votes)
##
## Coefficients:
## (Intercept)        White
##      18.779        0.525
```

```r
plot(votes$White, votes$Trump, xlab = "% White Population",
     ylab = "Trump Vote")
abline(m1, col='red')
```

# Making Predictions

- What is the predicted Trump vote for a county thats 30% white

```r
coef(m1)
```

```
## (Intercept)       White
##  18.7788513   0.5250146
```

```r
a.hat <- coef(m1)[1] ## estimated intercept
b.hat <- coef(m1)[2] ## estimated slope

pred30 = a.hat + b.hat * 0.3
pred30
```

```
## (Intercept)
##    18.93636
```

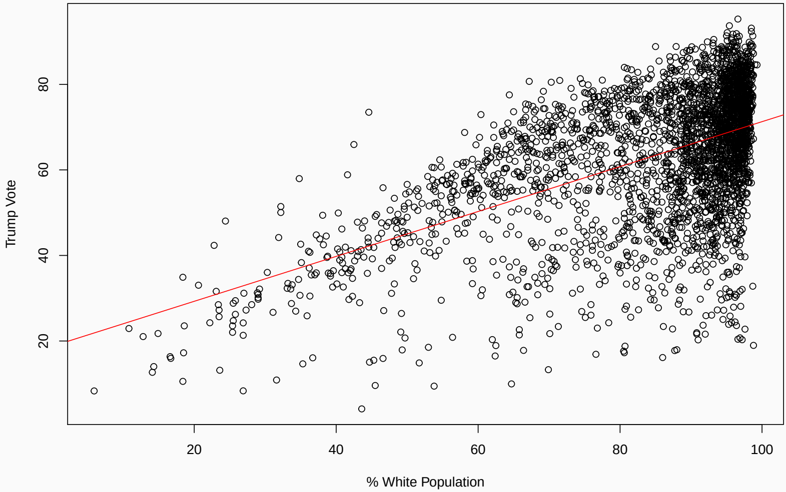
# Making Predictions

- What is the predicted Trump vote for a county thats 80% white

```
pred80 = a.hat + b.hat * 0.8
pred80
```

```
## (Intercept)
##    19.19886
```

# Plotting our predictions



# Breaking it down by State

- How does the relationship between racial composition of a county and vote for Trump change from state to state?

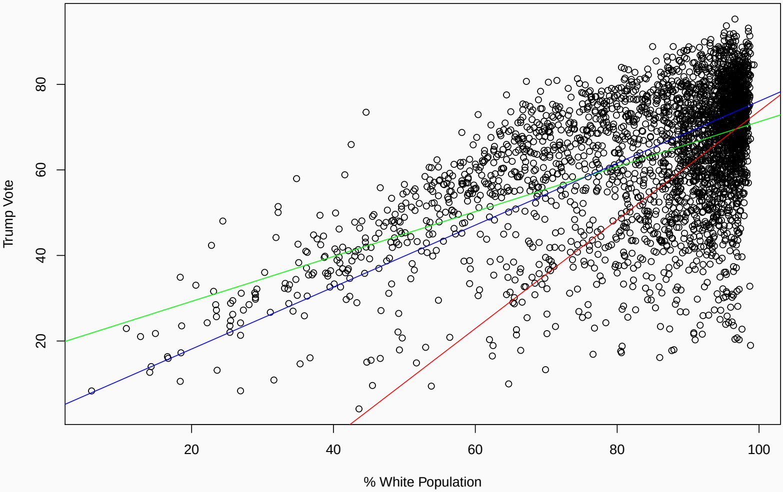
```r
penn = lm(Trump ~ White, data=votes,
          subset = state_abbr == 'PA')
coef(penn)
```

```
## (Intercept)       White
##   -53.30359     1.27029
```

```r
florida = lm(Trump ~ White, data=votes,
             subset = state_abbr == 'FL')
coef(florida)
```

```
## (Intercept)       White
##    3.631840    0.724387
```

# Breaking it down by State



# Why do we care about prediction?

- Prediction is broadly across different fields.
- Policy:
  - Can policymakers predict where crime is likely occur in a city to deploy police resources?
  - Can a school district predict which students will drop out of school to target counseling interventions?
- Business:
  - Can Amazon predict what product a customer is going to buy based on their past purchases (Amazon)?
  - Can Netflix/YouTube/Spotify predict what movies/TV show/song a person will like based on what they have viewed/listened to in the past?
- Linear regression often used to do these predictions, but how well does our model predict the data?

# Racial identity or Education?

- Does counties' racial compositon or education better predict vote for Trump?

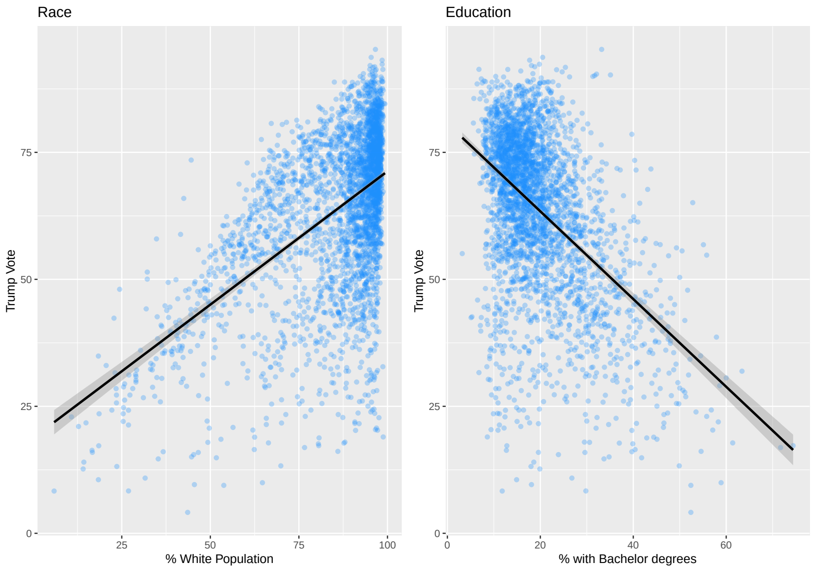
```r
# Race
race = lm(Trump ~ White, data=votes)
race
```

```
##
## Call:
## lm(formula = Trump ~ White, data = votes)
##
## Coefficients:
## (Intercept)        White
##      18.779        0.525
```

```r
# Education
educ = lm(Trump ~ educ_bach, data=votes)
educ
```

```
##
## Call:
## lm(formula = Trump ~ educ_bach, data = votes)
##
## Coefficients:
## (Intercept)    educ_bach
##     80.6644      -0.8636
```

# Comparing Models



# Model Fit

- How well does the model "fit the data"?
  - More specifically, how well does the model predict the outcome variable in the data?
- **Coefficient of determination** or $R^2$ ("R-squared"):
  - $R^2 =$ Explained variation / Total variation
  - R-squared gives you the percentage variation in y explained by x-variables.
  - The range is 0 to 1 (i.e. 0% to 100% of the variation in y can be explained by the x-variables.

## Correlation and R-Squared

- The coefficient of determination, $R^2$, is similar to the correlation coefficient, $R$
    - The correlation coefficient formula will tell you how strong of a linear relationship there is between two variables.
    - R Squared is the square of the correlation coefficient, r (hence the term r squared).
    - The more variance that is accounted for by the regression model the closer the data points will fall to the fitted regression line

# How to calculate R-squared

- The R-Squared formula compares our fitted regression line to a baseline model
  - The baseline model is a flat-line that predicts every value of y will be the mean value of y.
  - R-Squared checks to see if our fitted regression line will predict y better than the mean will

$$R^2 = 1 - \frac{SS_{RES}}{SS_{TOT}} = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \overline{y})^2}$$

# Model fit in R

- To access $R^2$ from the `lm()` output, first pass it to the `summary()` function:

```
# R-squared for race model
race.sum = summary(race)
race.sum$r.squared
```

```
## [1] 0.280542
```

```
# R-squared for educ model
educ.sum = summary(educ)
educ.sum$r.squared
```

```
## [1] 0.2374113
```

- Which does a better job predicting midterm election outcomes?

# Is R-squared useful?

- Does not prove causality.

# Are Low R-squared Values Always a Problem?

- Regression models with low R-squared values can be perfectly good models for several reasons
- Some thing are hard to predict (. . . human behavior)
- if you have a low R-squared value but the independent variables are statistically significant, you can still draw important conclusions about the relationships between the variables.

## Are High R-squared Values Always Great?

- You can be predicting one variable by unintentionally using a different form of the same variable
  - ex. Predict party affiliation with political ideology
- There are too many variables in your model compared to the number of observations.
  - This will lead to an **overfitted model**
  - Can predict the modeled data well BUT it will not predict new data well
- If you keep adding more and more independent variables, R-Squared will go up

# Adjusted R-squared

- **Adjusted R-Squared** takes into account the number of independent variables you employ in your model and can help indicate if a variable is useless or not
- The more variables you add to your model without predictive quality the lower your Adjusted R-Squared will be
- You can see that the number of independent variables, k, is included in the Adjusted R-Squared formula below

$$\text{Adjusted } R^2 = 1 - \frac{(n-1)}{[n-(k+1)]}(1-R^2)$$

Formula 9-6

where $n$ = sample size
$k$ = number of independent ($x$) variables

# In-sample/Out-of-sample Fit

- **In-sample fit:** how well your estimated model helps predict the data used to estimate the model.
  - $R^2$ is a measure of in-sample fit.
- **Out-of-sample fit:** how well your estimated model help predict outcomes outside of the sample used to fit the model.

# Overfitting

- **Overfitting:** OLS and other statistical procedures designed to predict in-sample outcomes really well, but may do really poorly out of sample.
  - Example: predicting winner of Democratic presidential primary with gender of the candidate.
  - Until 2016, gender of the canidate was a *perfect* predictor of who wins the primary.
  - Prediction for 2016 based on this: Bernie Sanders as Dem. nominee.
  - Bad out-of-sample prediction due to overfitting!
- Could waste tons of governmental or corporate resources with a bad prediction model!

## Avoiding overfitting

- Several procedure exist to guard against overfitting.
- **Cross validation** is the most popular:
  - Randomly choose half the sample to set aside (**test set**)
  - Estimate the coefficients with the remaining half of the units (**training set**)
  - Assess the model fit on the held out test set.
  - Switch the test and training set and repeat, average the results.
- Congrats, you know machine learning/artificial intelligence!

# Multiple Predictors

- What if we want to predict Y as a function of many variables

$$Y = \alpha + \beta_1 X_{i1} + \beta_2 X_{i2} + ...\beta_k X_{ik} + \epsilon_i$$

- Why include more than one predictor?
  - Better predictions
  - Better interpretation: $\beta_1$ is the effect of $X_1$ holding all other independent variables constant (**ceteris paribus**)

# Multiple Regression in R

```
mult.fit = lm(Trump ~ White + educ_bach, data=votes)
summary(mult.fit)
```

```
##
## Call:
## lm(formula = Trump ~ White + educ_bach, data = votes)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -61.283  -6.569   0.614   7.306  37.391
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 35.85222    1.15817   30.96   <2e-16 ***
## White        0.52450    0.01235   42.47   <2e-16 ***
## educ_bach   -0.86257    0.02208  -39.06   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.88 on 3109 degrees of freedom
## Multiple R-squared:  0.5174, Adjusted R-squared:  0.5171
## F-statistic:  1667 on 2 and 3109 DF,  p-value: < 2.2e-16
```

# Interpreting the output

- $\hat{\alpha} = 35.9$: percent vote for Trump in a county that is 0% white and 0% have a bachelor degree (does this exist?)
- $\hat{\beta}_1 = 0.52$: percent vote *increase* for Trump for additional percentage point of white population, **holding education fixed**
- $\hat{\beta}_2 = -0.86$: percent vote *decrease* for Trump for additional percentage point of bachelor degrees, **holding racial composition fixed**

# Least squares with multiple regression

- How do we estimate the coefficients?
- The same exact way as before: minimize prediction error!
- Residuals (aka prediction error) with multiple predictors:
- Find the coefficients that minimizes the sum of the squared residuals:

$$SSR = \sum_{i=1}^{n} \hat{\epsilon}_i^2 = (Y_i - \alpha - \beta_1 X_{i1} - \beta_2 X_{i2})^2$$

# Model fit with multiple predictors

- $R^2$ mechanically increases when you add a variables to the regression.
  - But this could be overfitting!!
- Solution: penalize regression models with more variables.
  - **Occam's razor:** simpler models are preferred
- **Adjusted $R^2$:** lowers regular $R^2$ for each additional covariate.
  - If the added covariates doesn't help predict, adjusted $R^2$goes down!

# Outputting regression results

```r
library(stargazer)
race.fit = lm(Trump ~ White, data=votes)
educ.fit = lm(Trump ~ educ_bach, data=votes)
mult.fit = lm(Trump ~ White + educ_bach, data=votes)
stargazer(race.fit, educ.fit, mult.fit, header=FALSE)
```

# Table

**Table 1:** Regression Results

| | *Dependent variable:* | | |
|---|---|---|---|
| | Percentage Vote for Trump | | |
| | (1) | (2) | (3) |
| Percent White Population | 0.525*** | | 0.525*** |
| | (0.015) | | (0.012) |
| Percent with Bachelor Degree | | −0.864*** | −0.863*** |
| | | (0.028) | (0.022) |
| Constant | 18.779*** | 80.664*** | 35.852*** |
| | (1.309) | (0.600) | (1.158) |
| Observations | 3,112 | 3,112 | 3,112 |
| $R^2$ | 0.281 | 0.237 | 0.517 |
| Adjusted $R^2$ | 0.280 | 0.237 | 0.517 |
| *Note:* | *p<0.1; **p<0.05; ***p<0.01 | | |

# Other resources...

- https://data.library.virginia.edu/is-r-squared-useless/
- https://medium.com/@erika.dauria/looking-at-r-squared-721252709098
- https://medium.com/@vince.shields913/why-we-dont-really-care-about-the-r-squared-in-econometrics-social-science-593e2db0391f
- http://svmiller.com/blog/2014/08/reading-a-regression-table-a-guide-for-students/
- https://scholar.princeton.edu/sites/default/files/bstewart/files/lecture8slides.pdf
- https://cran.r-project.org/web/packages/stargazer/vignettes/stargazer.pdf