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?

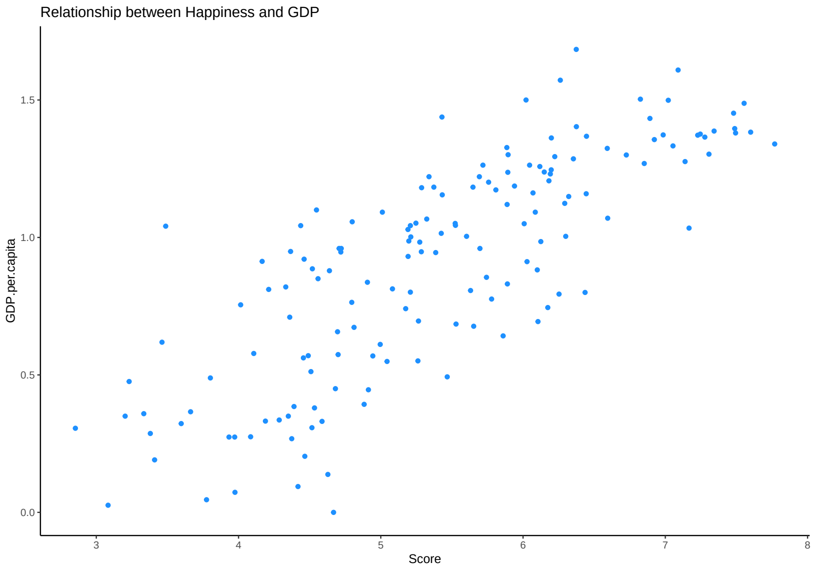
```r
# Read Happiness Data
happ2019 = read.csv("C:/Users/afisher/Documents/R Code/Resources/Data/Happiness/2019.csv")

# 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



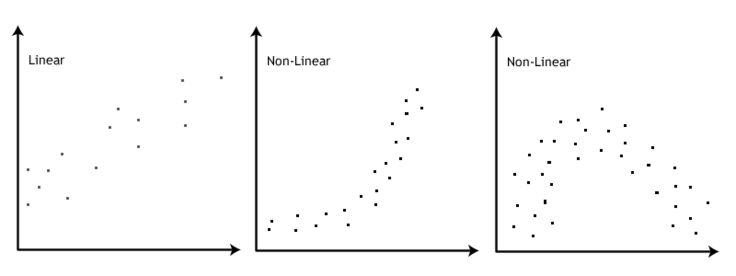
Correlation and scatter-plots:

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.

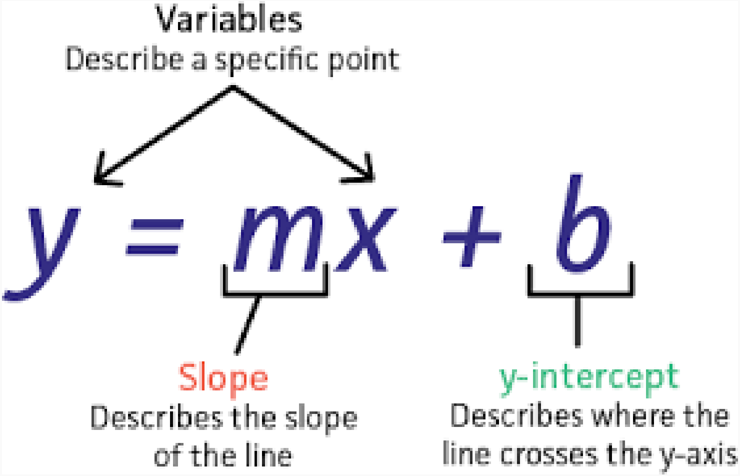
$$y = mx + b$$

- 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(α, β):** true unknown intercept/slope of the line of best fit.
- **Chance error (ϵ):** 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

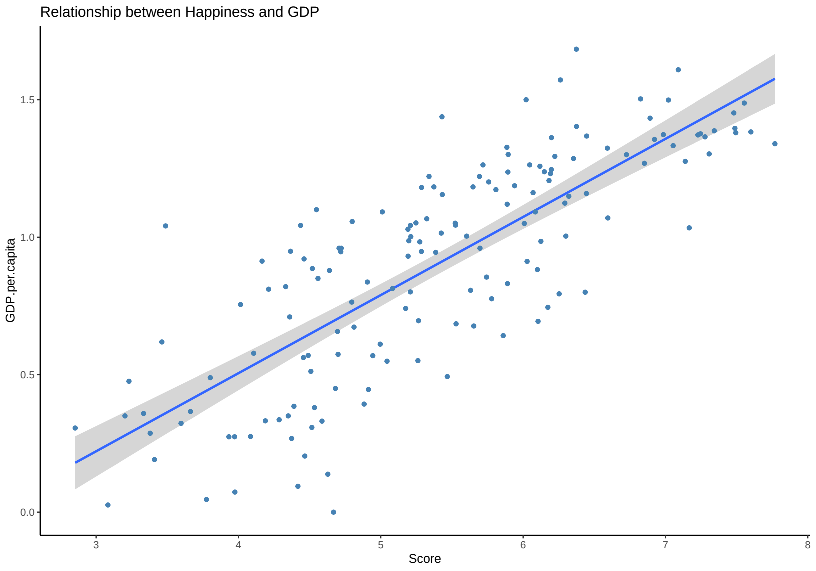
$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$$

- **Intercept** α: average of Y when X is 0
  - Average happiness when I GDP is 0.
- **Slope** β: 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 α or β is. How do we estimate it?

Estimated Coefficients

- Parameters: α, β
  - 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



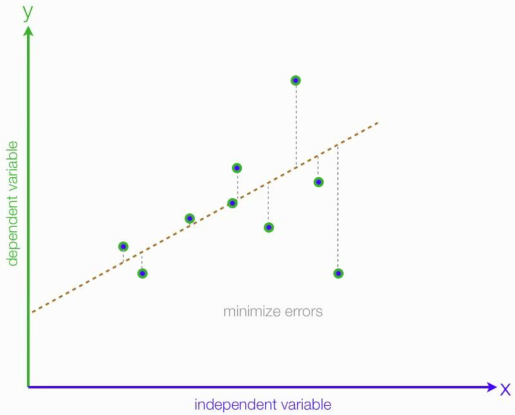
Least Squares

- 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

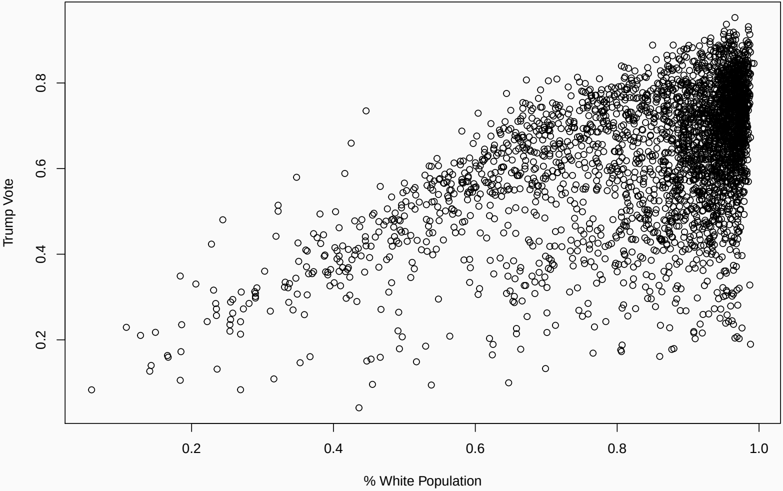
- Least squares line always goes throught $(\bar{X}, \bar{Y})$
- Estimated slop is relatd to correlation

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

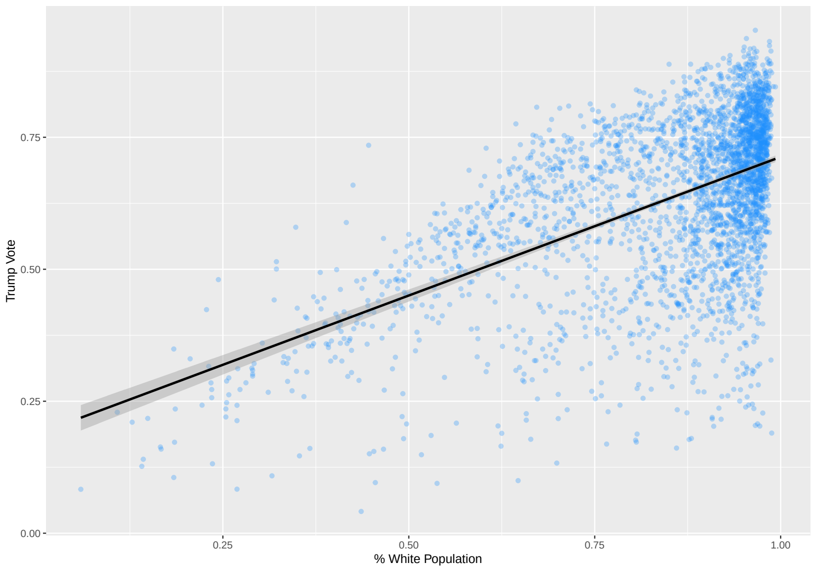
- mean of residuals is always 0

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
##      0.1878       0.5250
```

```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

```
coef(m1)
```

```
## (Intercept)       White
##   0.1877885   0.5250146
```

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

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

```
## (Intercept)
##   0.3452929
```

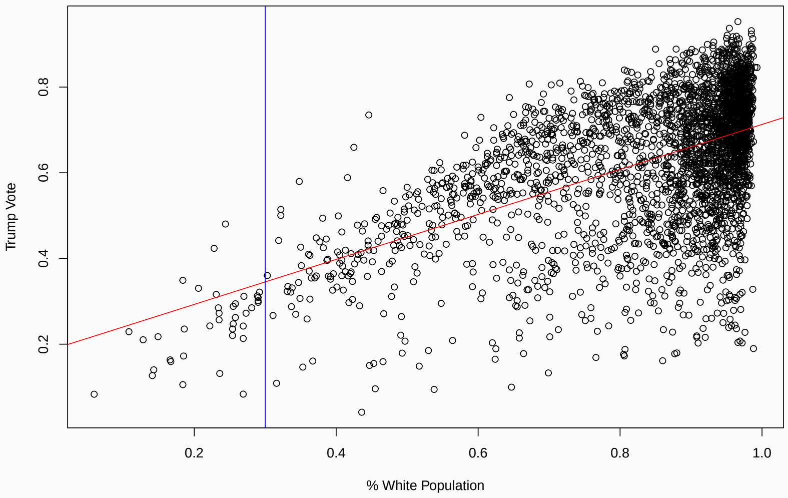
Making Predictions

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

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

```
## (Intercept)
##   0.6078002
```

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?

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

```
## (Intercept)       White
##  -0.5330359   1.2702904
```

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

```
## (Intercept)       White
##   0.0363184   0.7243870
```

Breaking it down by State