Gov 50: 11. Linear Regression

Matthew Blackwell

Harvard University

Fall 2018

1/ Today's agenda

Logistics

- Mid-semester evaluation out—please respond!

Logistics

- Mid-semester evaluation out—please respond!
- DataCamp 4 due Thursday.

Logistics

- Mid-semester evaluation out—please respond!
- DataCamp 4 due Thursday.
- HW 3 going out today, due next Thursday.

Logistics

- Mid-semester evaluation out—please respond!
- DataCamp 4 due Thursday.
- HW 3 going out today, due next Thursday.
- Matt’s OH moved to Fri, 10:30am-12:00pm this week only.

Final project

- Final project:

Final project

- Final project:
  - Short report that states a research question and answers it using a data set that you find.

Final project

- Final project:
  - Short report that states a research question and answers it using a data set that you find.
  - A few pages long.

Final project

- Final project:
  - Short report that states a research question and answers it using a data set that you find.
  - A few pages long.
- Group project:

Final project

- Final project:
  - Short report that states a research question and answers it using a data set that you find.
  - A few pages long.
- Group project:
  - No more than 4 people in a group.

Final project

- Final project:
  - Short report that states a research question and answers it using a data set that you find.
  - A few pages long.
- Group project:
  - No more than 4 people in a group.
  - Due to feedback on the surveys, we have decided to allow for people to work individually.

Final project

- Final project:
  - Short report that states a research question and answers it using a data set that you find.
  - A few pages long.
- Group project:
  - No more than 4 people in a group.
  - Due to feedback on the surveys, we have decided to allow for people to work individually.
  - Graded the same, no matter the group size.

Final project

- Final project:
  - Short report that states a research question and answers it using a data set that you find.
  - A few pages long.
- Group project:
  - No more than 4 people in a group.
  - Due to feedback on the surveys, we have decided to allow for people to work individually.
  - Graded the same, no matter the group size.
- Timeline:

Final project

- Final project:
  - Short report that states a research question and answers it using a data set that you find.
  - A few pages long.
- Group project:
  - No more than 4 people in a group.
  - Due to feedback on the surveys, we have decided to allow for people to work individually.
  - Graded the same, no matter the group size.
- Timeline:
  - Fill out surveys on Canvas (under “Final Project”) by Nov. 1.

Final project

- Final project:
  - Short report that states a research question and answers it using a data set that you find.
  - A few pages long.
- Group project:
  - No more than 4 people in a group.
  - Due to feedback on the surveys, we have decided to allow for people to work individually.
  - Graded the same, no matter the group size.
- Timeline:
  - Fill out surveys on Canvas (under “Final Project”) by Nov. 1.
  - Paragraph describing data, research questions due Nov. 21.

Final project

- Final project:
  - Short report that states a research question and answers it using a data set that you find.
  - A few pages long.
- Group project:
  - No more than 4 people in a group.
  - Due to feedback on the surveys, we have decided to allow for people to work individually.
  - Graded the same, no matter the group size.
- Timeline:
  - Fill out surveys on Canvas (under "Final Project") by Nov. 1.
  - Paragraph describing data, research questions due Nov. 21.
  - Rmd file with analyses due Nov. 30.

Final project

- Final project:
  - Short report that states a research question and answers it using a data set that you find.
  - A few pages long.
- Group project:
  - No more than 4 people in a group.
  - Due to feedback on the surveys, we have decided to allow for people to work individually.
  - Graded the same, no matter the group size.
- Timeline:
  - Fill out surveys on Canvas (under "Final Project") by Nov. 1.
  - Paragraph describing data, research questions due Nov. 21.
  - Rmd file with analyses due Nov. 30.
  - Final report due Dec. 10.

Where are we? Where are going?

- Last time: used sample means to make prediction about future events based on the past.

Where are we? Where are going?

- Last time: used sample means to make prediction about future events based on the past.
- Now: how can we use one variable to predict another?

Where are we? Where are going?

- Last time: used sample means to make prediction about future events based on the past.
- Now: how can we use one variable to predict another?
- Big technical tool: **linear regression**

Where are we? Where are going?

- Last time: used sample means to make prediction about future events based on the past.
- Now: how can we use one variable to predict another?
- Big technical tool: **linear regression**
  - Now: how to fit, get predictions

2/ Prediction using a second variable

Predicting my weight

- I’ve been tracking my physical activity and weight for a few years now.

Predicting my weight

- I've been tracking my physical activity and weight for a few years now.
- Can we use my activity to predict my weight on a day-to-day basis?

Predicting my weight

- I've been tracking my physical activity and weight for a few years now.
- Can we use my activity to predict my weight on a day-to-day basis?

Name	Description
`date`	date of measurements
`active.calories`	calories burned
`steps`	number of steps taken (in 1,000s)
`weight`	weight (lbs)
`steps.lag`	steps on day before (in 1,000s)
`calories.lag`	calories burned on day before

Predicting using bivariate relationship

- Goal: what's our best guess about Y_i if we know what X_i is?

Predicting using bivariate relationship

- Goal: what's our best guess about Y_i if we know what X_i is?
  - what's our best guess about my weight this morning if I know how many steps I took yesterday?

Predicting using bivariate relationship

- Goal: what's our best guess about Y_i if we know what X_i is?
  - what's our best guess about my weight this morning if I know how many steps I took yesterday?
- Terminology:

Predicting using bivariate relationship

- Goal: what's our best guess about Y_i if we know what X_i is?
  - what's our best guess about my weight this morning if I know how many steps I took yesterday?
- Terminology:
  - **Dependent/outcome variable**: the variable we want to predict (weight).

Predicting using bivariate relationship

- Goal: what's our best guess about Y_i if we know what X_i is?
  - what's our best guess about my weight this morning if I know how many steps I took yesterday?
- Terminology:
  - **Dependent/outcome variable**: the variable we want to predict (weight).
  - **Independent/explanatory variable**: the variable we're using to predict (steps).

- Load the data:

- Load the data:

```
health <- read.csv("data/health.csv")
health <- na.omit(health)
```

- Load the data:

```
health <- read.csv("data/health.csv")
health <- na.omit(health)
```

- Plot the data:

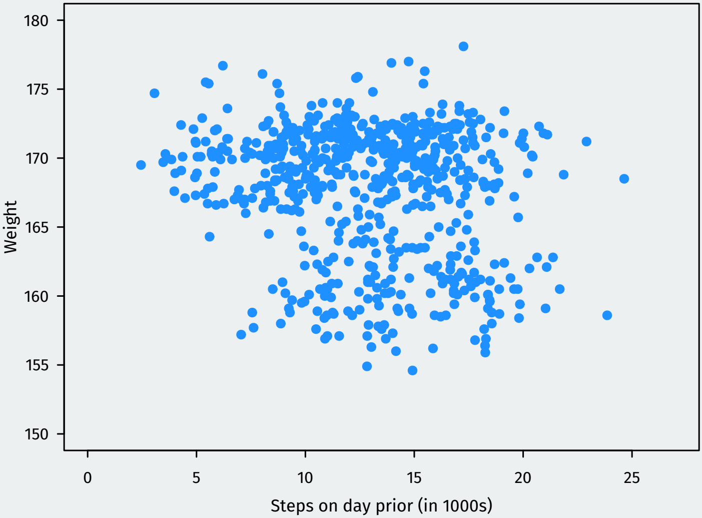
- Load the data:

```
health <- read.csv("data/health.csv")
health <- na.omit(health)
```

- Plot the data:

```
plot(health$steps.lag, health$weight, pch = 19,
     col =  "dodgerblue",
     xlim = c(0, 27), ylim = c(150, 180),
     xlab = "Steps on day prior (in 1000s)",
     ylab = "Weight",
     main = "Weight and Steps")
```

Weight and Steps



Correlation and scatterplots

- Recall the definition of correlation:

$$\frac{1}{n-1}\sum_{i=1}^{n}\left[(\text{z-score for } x_i) \times (\text{z-score for } y_i)\right]$$

Correlation and scatterplots

- Recall the definition of correlation:

$$\frac{1}{n-1}\sum_{i=1}^{n}\left[(\text{z-score for } x_i) \times (\text{z-score for } y_i)\right]$$

- Correlation between lagged steps and weight:

Correlation and scatterplots

- Recall the definition of correlation:

$$\frac{1}{n-1}\sum_{i=1}^{n}\left[(\text{z-score for } x_i) \times (\text{z-score for } y_i)\right]$$

- Correlation between lagged steps and weight:

```
cor(health$steps.lag, health$weight)
```

```
## [1] -0.191
```

Correlation and scatterplots

- Recall the definition of correlation:

$$\frac{1}{n-1}\sum_{i=1}^{n}\left[(\text{z-score for } x_i) \times (\text{z-score for } y_i)\right]$$

- Correlation between lagged steps and weight:

```
cor(health$steps.lag, health$weight)
```

```
## [1] -0.191
```

- Correlation and scatter-plots:

Correlation and scatterplots

- Recall the definition of correlation:

$$\frac{1}{n-1}\sum_{i=1}^{n}[(\text{z-score for } x_i) \times (\text{z-score for } y_i)]$$

- Correlation between lagged steps and weight:

```
cor(health$steps.lag, health$weight)
```

```
## [1] -0.191
```

- Correlation and scatter-plots:
  1. positive correlation $\rightsquigarrow$ upward slope

Correlation and scatterplots

- Recall the definition of correlation:

$$\frac{1}{n-1}\sum_{i=1}^{n}\left[(\text{z-score for } x_i) \times (\text{z-score for } y_i)\right]$$

- Correlation between lagged steps and weight:

```
cor(health$steps.lag, health$weight)
```

```
## [1] -0.191
```

- Correlation and scatter-plots:
  1. positive correlation $\rightsquigarrow$ upward slope
  2. negative correlation $\rightsquigarrow$ downward slope

Correlation and scatterplots

- Recall the definition of correlation:

$$\frac{1}{n-1}\sum_{i=1}^{n}\left[(\text{z-score for } x_i) \times (\text{z-score for } y_i)\right]$$

- Correlation between lagged steps and weight:

```
cor(health$steps.lag, health$weight)
```

```
## [1] -0.191
```

- Correlation and scatter-plots:
    1. positive correlation $\rightsquigarrow$ upward slope
    2. negative correlation $\rightsquigarrow$ downward slope
    3. high correlation $\rightsquigarrow$ tighter, closer to a line

Correlation and scatterplots

- Recall the definition of correlation:

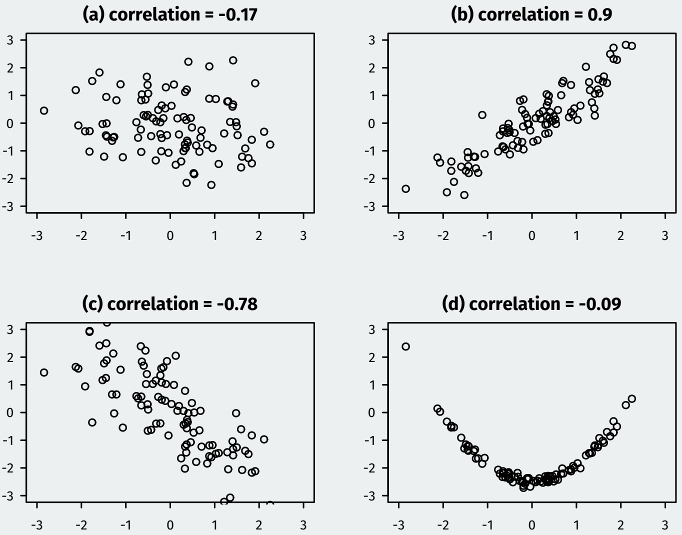
$$\frac{1}{n-1}\sum_{i=1}^{n}\left[(\text{z-score for } x_i) \times (\text{z-score for } y_i)\right]$$

- Correlation between lagged steps and weight:

```
cor(health$steps.lag, health$weight)
```

```
## [1] -0.191
```

- Correlation and scatter-plots:
    1. positive correlation $\rightsquigarrow$ upward slope
    2. negative correlation $\rightsquigarrow$ downward slope
    3. high correlation $\rightsquigarrow$ tighter, closer to a line
    4. correlation cannot capture nonlinear relationship.


(a) correlation = -0.17
(b) correlation = 0.9
(c) correlation = -0.78
(d) correlation = -0.09
3
2
1
0
-1
-2
-3
-3 -2 -1 0 1 2 3

3/ Linear regression

Using a line to predict

- Prediction: for any value of X, what's the best guess about Y?

Using a line to predict

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

$$y = mx + b$$

Using a line to predict

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

$$y = mx + b$$

- Problem: for any line we draw, not all the data is on the line.

Using a line to predict

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

$$y = mx + b$$

- Problem: for any line we draw, not all the data is on the line.
  - Some weights will be above the line, some below.

Using a line to predict

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

$$y = mx + b$$

- 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.

Linear regression model

- Model for the line of best fit:

Linear regression model

- Model for the line of best fit:

$$Y_i = \underbrace{\alpha}_{\text{intercept}} + \underbrace{\beta}_{\text{slope}} \cdot X_i + \underbrace{\epsilon_i}_{\text{error term}}$$

Linear regression model

- Model for the line of best fit:

$$Y_i = \underbrace{\alpha}_{\text{intercept}} + \underbrace{\beta}_{\text{slope}} \cdot X_i + \underbrace{\epsilon_i}_{\text{error term}}$$

- **Coefficients/parameters** (α, β): true unknown intercept/slope of the line of best fit.

Linear regression model

- Model for the line of best fit:

$$Y_i = \underbrace{\alpha}_{\text{intercept}} + \underbrace{\beta}_{\text{slope}} \cdot X_i + \underbrace{\epsilon_i}_{\text{error term}}$$

- **Coefficients/parameters** (α, β): true unknown intercept/slope of the line of best fit.
- **Chance error** ϵ_i: accounts for the fact that the line doesn't perfectly fit the data.

Linear regression model

- Model for the line of best fit:

$$Y_i = \underbrace{\alpha}_{\text{intercept}} + \underbrace{\beta}_{\text{slope}} \cdot X_i + \underbrace{\epsilon_i}_{\text{error term}}$$

- **Coefficients/parameters** (α, β): true unknown intercept/slope of the line of best fit.
- **Chance error** ϵ_i: accounts for the fact that the line doesn't perfectly fit the data.
  - Each observation allowed to be off the regression line.

Linear regression model

- Model for the line of best fit:

$$Y_i = \underbrace{\alpha}_{\text{intercept}} + \underbrace{\beta}_{\text{slope}} \cdot X_i + \underbrace{\epsilon_i}_{\text{error term}}$$

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

$$Y_i = \alpha + \beta \cdot X_i + \epsilon_i$$

- **Intercept** α: average value of Y when X is 0

Interpreting the regression line

$$Y_i = \alpha + \beta \cdot X_i + \epsilon_i$$

- **Intercept** α: average value of Y when X is 0
  - Average weight when I take 0 steps the day prior.

Interpreting the regression line

$$Y_i \quad = \quad \alpha + \beta \cdot X_i + \epsilon_i$$

- **Intercept** α: average value of Y when X is 0
  - Average weight when I take 0 steps the day prior.
- **Slope** β: average change in Y when X increases by one unit.

Interpreting the regression line

$$Y_i = \alpha + \beta \cdot X_i + \epsilon_i$$

- **Intercept** α: average value of Y when X is 0
  - Average weight when I take 0 steps the day prior.
- **Slope** β: average change in Y when X increases by one unit.
  - Average decrease in weight for each additional 1,000 steps.

Interpreting the regression line

$$Y_i \quad = \quad \alpha + \beta \cdot X_i + \epsilon_i$$

- **Intercept** α: average value of Y when X is 0
  - Average weight when I take 0 steps the day prior.
- **Slope** β: average change in Y when X increases by one unit.
  - Average decrease in weight for each additional 1,000 steps.
- But we don't know α or β. How can we estimate them?

Estimated coefficients

- Parameters: α, β

Estimated coefficients

- Parameters: α, β
    - Unknown features of the **data-generating process**.

Estimated coefficients

- Parameters: α, β
  - Unknown features of the **data-generating process**.
  - Chance error makes these impossible to observe directly.

Estimated coefficients

- Parameters: α, β
  - Unknown features of the **data-generating process**.
  - Chance error makes these impossible to observe directly.
- Estimates: $\widehat{\alpha}, \widehat{\beta}$

Estimated coefficients

- Parameters: α, β
  - Unknown features of the **data-generating process**.
  - Chance error makes these impossible to observe directly.
- Estimates: $\widehat{\alpha}, \widehat{\beta}$
  - An **estimate** is a function of the data that is our best guess about some parameter.

Estimated coefficients

- Parameters: α, β
  - Unknown features of the **data-generating process**.
  - Chance error makes these impossible to observe directly.
- Estimates: $\widehat{\alpha}, \widehat{\beta}$
  - An **estimate** is a function of the data that is our best guess about some parameter.
- **Regression line**: $\widehat{Y} = \widehat{\alpha} + \widehat{\beta} \cdot x$

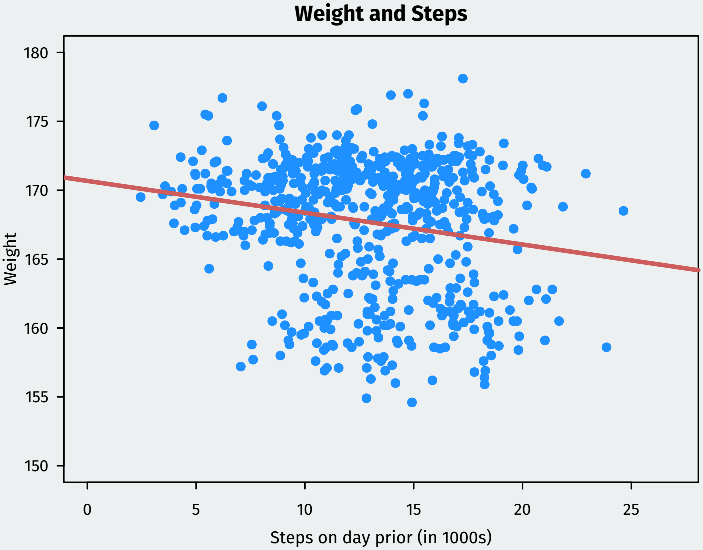
Estimated coefficients

- Parameters: α, β
  - Unknown features of the **data-generating process**.
  - Chance error makes these impossible to observe directly.
- Estimates: $\widehat{\alpha}, \widehat{\beta}$
  - An **estimate** is a function of the data that is our best guess about some parameter.
- **Regression line**: $\widehat{Y} = \widehat{\alpha} + \widehat{\beta} \cdot x$
  - Average value of Y when X is equal to x.

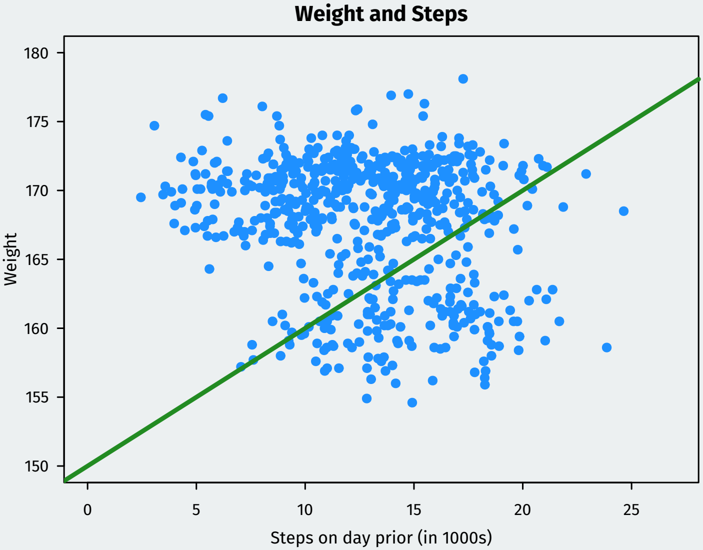
Estimated coefficients

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

Line of best fit



Why not this line?



4/ Ordinary least squares

Least squares

- How do we figure out the best line to draw?

Least squares

- How do we figure out the best line to draw?
  - **Fitted/predicted value** for each observation: $\widehat{Y}_i = \widehat{\alpha} + \widehat{\beta} X_i$

Least squares

- How do we figure out the best line to draw?
  - **Fitted/predicted value** for each observation: $\widehat{Y}_i = \widehat{\alpha} + \widehat{\beta} X_i$
  - **Residual/prediction error**: $\widehat{\epsilon}_i = Y_i - \widehat{Y}$

Least squares

- How do we figure out the best line to draw?
  - **Fitted/predicted value** for each observation: $\widehat{Y}_i = \widehat{\alpha} + \widehat{\beta} X_i$
  - **Residual/prediction error**: $\widehat{\epsilon}_i = Y_i - \widehat{Y}$
- Get these estimates by the **least squares method**.

Least squares

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

$$\text{SSR} = \sum_{i=1}^{n} \widehat{\epsilon}_i^2 = \sum_{i=1}^{n} (Y_i - \widehat{\alpha} - \widehat{\beta} X_i)^2$$

Least squares

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

$$\text{SSR} = \sum_{i=1}^{n} \widehat{\epsilon}_i^2 = \sum_{i=1}^{n} (Y_i - \widehat{\alpha} - \widehat{\beta} X_i)^2$$

- This finds the line that minimizes the magnitude of the prediction errors!

Linear regression in R

- R will calculate least squares line for a data set using `lm()`.

Linear regression in R

- R will calculate least squares line for a data set using `lm()`.
  - Jargon: “fit the model”

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)`

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 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

```
fit <- lm(weight ~ steps.lag, data = health)
fit
```

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

```
fit <- lm(weight ~ steps.lag, data = health)
fit
```

```
##
## Call:
## lm(formula = weight ~ steps.lag, data = health)
##
## Coefficients:
## (Intercept)    steps.lag
##     170.675       -0.231
```

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

```
fit <- lm(weight ~ steps.lag, data = health)
fit
```

```
##
## Call:
## lm(formula = weight ~ steps.lag, data = health)
##
## Coefficients:
## (Intercept)    steps.lag
##     170.675       -0.231
```

- Interpretation?

Coefficients and fitted values

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

Coefficients and fitted values

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

```
coef(fit)
```

Coefficients and fitted values

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

```
coef(fit)
```

```
## (Intercept)   steps.lag
##     170.675      -0.231
```

Coefficients and fitted values

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

```
coef(fit)
```

```
## (Intercept)   steps.lag
##     170.675      -0.231
```

- R can show you each of the fitted values as well:

Coefficients and fitted values

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

```
coef(fit)
```

```
## (Intercept)    steps.lag
##     170.675       -0.231
```

- R can show you each of the fitted values as well:

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

Coefficients and fitted values

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

```
coef(fit)
```

```
## (Intercept)   steps.lag
##     170.675      -0.231
```

- R can show you each of the fitted values as well:

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

```
##   2   3   4   5   6   7
## 167 166 166 168 166 169
```

Properties of least squares

- Least squares line always goes through $(\overline{X}, \overline{Y})$.

Properties of least squares

- Least squares line always goes through $(\overline{X}, \overline{Y})$.
- Estimated slope is related to correlation:

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

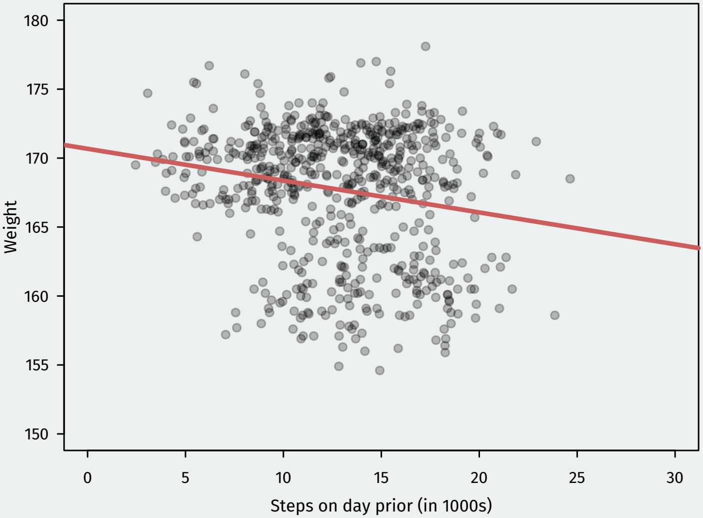
Properties of least squares

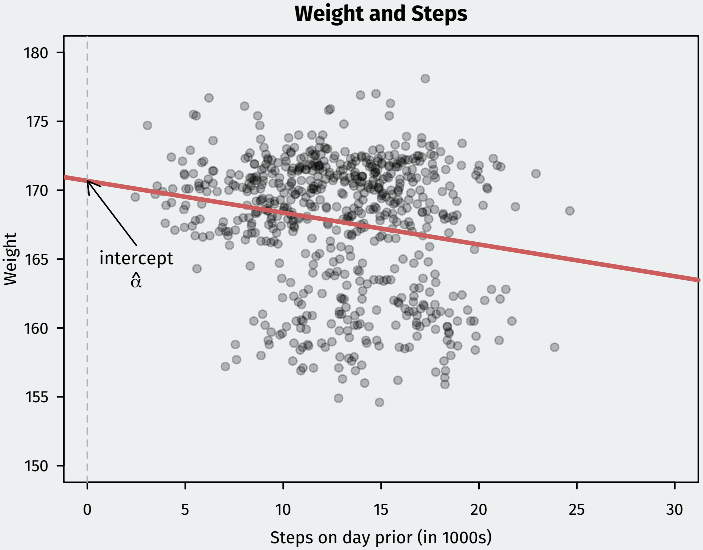
- Least squares line always goes through $(\overline{X}, \overline{Y})$.
- Estimated slope is related to correlation:

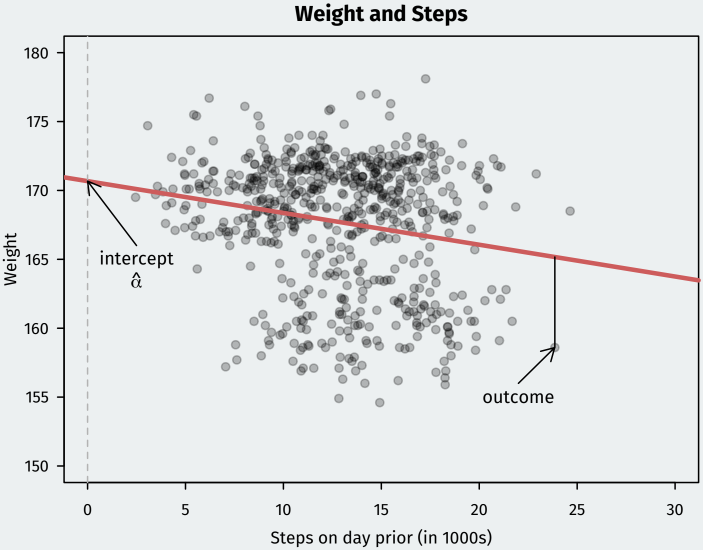
$$\widehat{\beta} = (\text{correlation of } X \text{ and } Y) \times \frac{\text{SD of } Y}{\text{SD of } X}$$

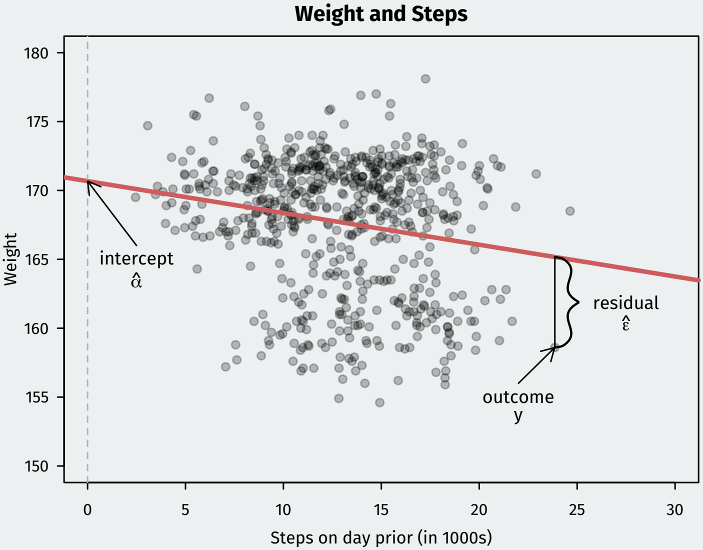
- Mean of residuals is always 0.

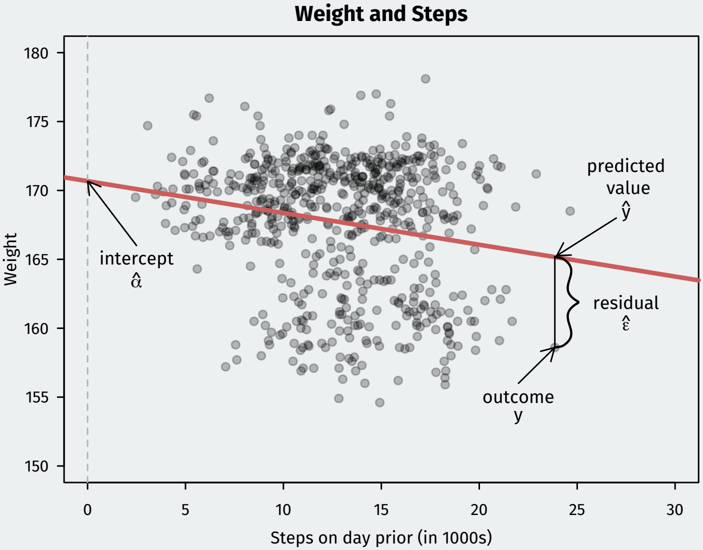
Weight and Steps

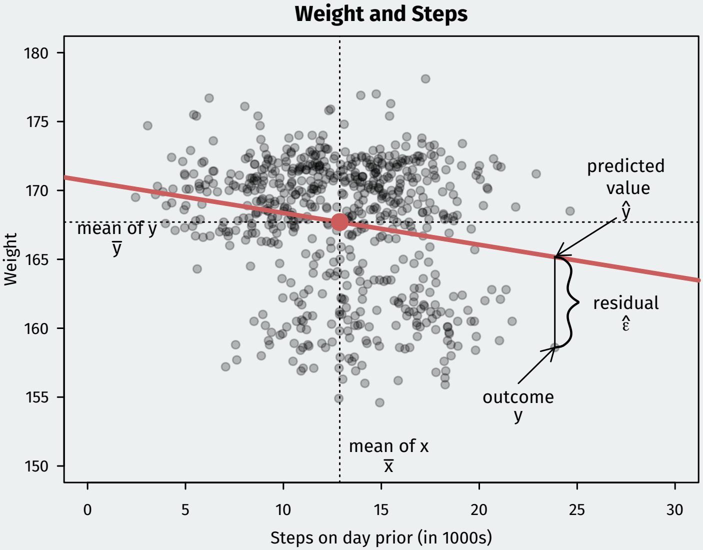



Weight and Steps
180
175
170
165
160
155
150
Weight
intercept
$\hat{\alpha}$
0
5
10
15
20
25
30
Steps on day prior (in 1000s)


Weight and Steps
180
175
170
165
160
155
150
Weight
intercept
$\hat{\alpha}$
outcome
0
5
10
15
20
25
30
Steps on day prior (in 1000s)


Weight and Steps
180
175
170
165
160
155
150
Weight
intercept
α̂
residual
ε̂
outcome
y
0
5
10
15
20
25
30
Steps on day prior (in 1000s)


Weight and Steps
Weight
180
175
170
165
160
155
150
0
5
10
15
20
25
30
Steps on day prior (in 1000s)
intercept
α̂
predicted
value
ŷ
residual
ε̂
outcome
y


Weight and Steps
180
175
170
165
160
155
150
Weight
0
5
10
15
20
25
30
Steps on day prior (in 1000s)
mean of y
$\bar{y}$
mean of x
$\bar{x}$
predicted
value
$\hat{y}$
residual
$\hat{\varepsilon}$
outcome
y

5/ Prediction midterm elections

Presidential popularity and the midterms

- How does the popularity of a president predict how well their party will do in the midterm elections?

Presidential popularity and the midterms

- How does the popularity of a president predict how well their party will do in the midterm elections?
- Small dataset with information on approval and midterm election outcomes:

Presidential popularity and the midterms

- How does the popularity of a president predict how well their party will do in the midterm elections?
- Small dataset with information on approval and midterm election outcomes:

Name	Description
`year`	midterm election year
`president`	name of president
`party`	Democrat or Republican
`approval`	Gallup approval rating at midterms
`seat.change`	change in the number of House seat's for the president's party

Loading the data

```r
midterms <- read.csv("data/midterms.csv")
head(midterms)
```

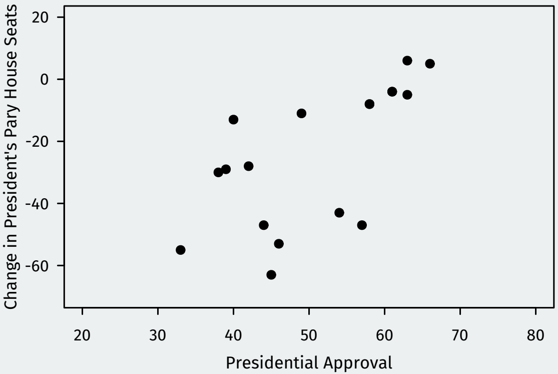
Loading the data

```r
midterms <- read.csv("data/midterms.csv")
head(midterms)
```

```
##   year  president party approval seat.change
## 1 1946     Truman     D       33         -55
## 2 1950     Truman     D       39         -29
## 3 1954 Eisenhower     R       61          -4
## 4 1958 Eisenhower     R       57         -47
## 5 1962    Kennedy     D       61          -4
## 6 1966    Johnson     D       44         -47
```

Scatterplot

```
plot(midterms$approval, midterms$seat.change, xlim = c(20, 80),
     ylim = c(-70, 20), pch = 19, xlab = "Presidential Approval",
     ylab = "Change in President's Pary House Seats")
```



Running a regression

- Run the regression with `seat.change` as dependent variable and `approval` as independent variable:

Running a regression

- Run the regression with `seat.change` as dependent variable and `approval` as independent variable:

```
appseats <- lm(seat.change ~ approval, data = midterms)
appseats
```

Running a regression

- Run the regression with `seat.change` as dependent variable and `approval` as independent variable:

```
appseats <- lm(seat.change ~ approval, data = midterms)
appseats
```

```
##
## Call:
## lm(formula = seat.change ~ approval, data = midterms)
##
## Coefficients:
## (Intercept)     approval
##      -96.84         1.42
```

Running a regression

- Run the regression with `seat.change` as dependent variable and `approval` as independent variable:

```
appseats <- lm(seat.change ~ approval, data = midterms)
appseats
```

```
##
## Call:
## lm(formula = seat.change ~ approval, data = midterms)
##
## Coefficients:
## (Intercept)     approval
##      -96.84         1.42
```

- Intercept: predicted seat change when presidential approval is 0.

Running a regression

- Run the regression with `seat.change` as dependent variable and `approval` as independent variable:

```
appseats <- lm(seat.change ~ approval, data = midterms)
appseats
```

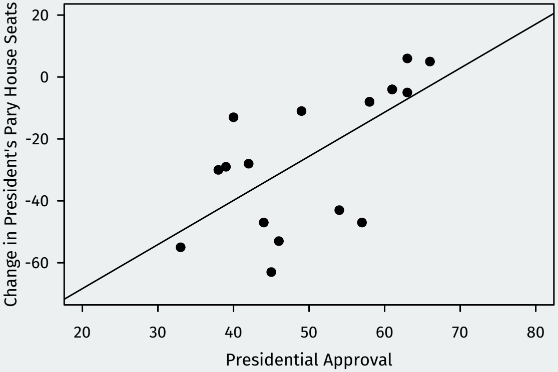
```
##
## Call:
## lm(formula = seat.change ~ approval, data = midterms)
##
## Coefficients:
## (Intercept)     approval
##      -96.84         1.42
```

- Intercept: predicted seat change when presidential approval is 0.
- Slope: a one-percentage point increase in approval ≈ 1.42 increase in House seats

Scatterplot

```
plot(midterms$approval, midterms$seat.change, xlim = c(20, 80),
     ylim = c(-70, 20), pch = 19, xlab = "Presidential Approval",
     ylab = "Change in President's Pary House Seats")

abline(appseats)  ## appseats is call to lm() from above
```



Predicting the next midterm

- Can we get a prediction for Republicans in 2018?

Predicting the next midterm

- Can we get a prediction for Republicans in 2018?

```
tail(midterms)
```

Predicting the next midterm

- Can we get a prediction for Republicans in 2018?

```r
tail(midterms)
```

```
##    year president party approval seat.change
## 14 1998   Clinton     D       66           5
## 15 2002   W. Bush     R       63           6
## 16 2006   W. Bush     R       38         -30
## 17 2010     Obama     D       45         -63
## 18 2014     Obama     D       40         -13
## 19 2018     Trump     R       38          NA
```

Predicting 2018

- We can use the `coef()` function to access the estimated slope and intercept:

Predicting 2018

- We can use the `coef()` function to access the estimated slope and intercept:

```
coef(appseats)
```

Predicting 2018

- We can use the `coef()` function to access the estimated slope and intercept:

```
coef(appseats)
```

```
## (Intercept)    approval
##      -96.84        1.42
```

Predicting 2018

- We can use the `coef()` function to access the estimated slope and intercept:

```
coef(appseats)
```

```
## (Intercept)    approval
##      -96.84        1.42
```

- Select the estimates and save them:

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

Predicting 2018

- We can use the `coef()` function to access the estimated slope and intercept:

```
coef(appseats)
```

```
## (Intercept)    approval
##      -96.84        1.42
```

- Select the estimates and save them:

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

- Use these to create prediction, $\widehat{Y} = \widehat{\alpha} + \widehat{\beta} \cdot x$:

Predicting 2018

- We can use the `coef()` function to access the estimated slope and intercept:

```
coef(appseats)
```

```
## (Intercept)    approval
##      -96.84        1.42
```

- Select the estimates and save them:

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

- Use these to create prediction, $\widehat{Y} = \widehat{\alpha} + \widehat{\beta} \cdot x$:

```
pred2018 <- a.hat + b.hat * 38
pred2018
```

Predicting 2018

- We can use the `coef()` function to access the estimated slope and intercept:

```
coef(appseats)
```

```
## (Intercept)    approval
##      -96.84        1.42
```

- Select the estimates and save them:

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

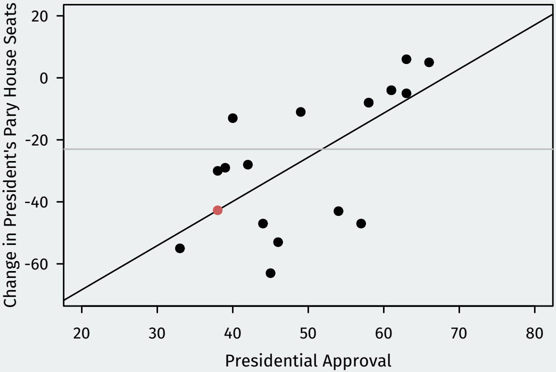
- Use these to create prediction, $\widehat{Y} = \widehat{\alpha} + \widehat{\beta} \cdot x$:

```
pred2018 <- a.hat + b.hat * 38
pred2018
```

```
## (Intercept)
##       -42.7
```

Scatterplot

```
plot(midterms$approval, midterms$seat.change, xlim = c(20, 80),
     ylim = c(-70, 20), pch = 19, xlab = "Presidential Approval",
     ylab = "Change in President's Pary House Seats")
abline(appseats)  ## appseats is call to lm() from above
points(x = 38, y = pred2018, col = "indianred", pch = 19)
abline(h = -23, col = "grey") ## flips the House
```



Regressions on subsets

- We can run regressions on subsets using the `subset` argument:

Regressions on subsets

- We can run regressions on subsets using the `subset` argument:

```
regR <- lm(seat.change ~ approval, data = midterms, subset = party == "R")
coef(regR)
```

Regressions on subsets

- We can run regressions on subsets using the `subset` argument:

```
regR <- lm(seat.change ~ approval, data = midterms, subset = party == "R")
coef(regR)
```

```
## (Intercept)    approval
##      -81.58        1.15
```

Regressions on subsets

- We can run regressions on subsets using the `subset` argument:

```
regR <- lm(seat.change ~ approval, data = midterms, subset = party == "R")
coef(regR)
```

```
## (Intercept)    approval
##      -81.58        1.15
```

```
regD <- lm(seat.change ~ approval, data = midterms, subset = party == "D")
coef(regD)
```

Regressions on subsets

- We can run regressions on subsets using the `subset` argument:

```
regR <- lm(seat.change ~ approval, data = midterms, subset = party == "R")
coef(regR)
```

```
## (Intercept)    approval
##      -81.58        1.15
```

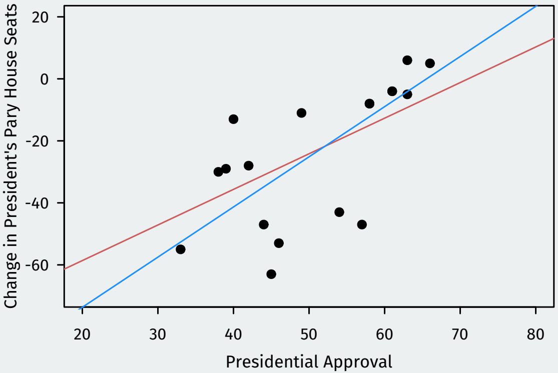
```
regD <- lm(seat.change ~ approval, data = midterms, subset = party == "D")
coef(regD)
```

```
## (Intercept)    approval
##     -106.03        1.62
```

Scatterplot

```
plot(midterms$approval, midterms$seat.change, xlim = c(20, 80),
     ylim = c(-70, 20), pch = 19, xlab = "Presidential Approval",
     ylab = "Change in President's Pary House Seats")

abline(regR, col = "indianred")
abline(regD, col = "dodgerblue")
```



On deck

- Mid-semester evaluation: please respond!

On deck

- Mid-semester evaluation: please respond!
- DataCamp assignment 4: due this Thursday.

On deck

- Mid-semester evaluation: please respond!
- DataCamp assignment 4: due this Thursday.
- Homework 3: Out today, due next Thursday.

On deck

- Mid-semester evaluation: please respond!
- DataCamp assignment 4: due this Thursday.
- Homework 3: Out today, due next Thursday.
- Start thinking about groups for final project.