Gov 2000 - 6. What is Regression?

Matthew Blackwell
Harvard University
mblackwell@gov.harvard.edu

Where are we? Where are we going?

- What we've been up to: estimating parameters of population distributions. Generally we've been learning about a single variable.
- This week and for the rest of the term, we'll be interested in the relationships between variables. How does one variable change we change the values of another variable? These will be the bread and butter of the class moving forward.

RELATIONSHIPS BETWEEN TWO VARIABLES

What is a relationship and why do we care?

- Most of what we want to do in the social science is learn about how two variables are related
- Examples:
  - Does turnout vary by types of mailers received?
  - Is the quality of political institutions related to average incomes?
  - Does conflict mediation help reduce civil conflict?

Notation and conventions

- Y - the dependent variable or outcome or regressand or left-hand-side variable or response
  - Voter turnout
  - Log GDP per capita
  - Number of battle deaths
- X - the independent variable or explanatory variable or regressor or right-hand-side variable or treatment or predictor
  - Social pressure mailer versus Civic Duty Mailer
  - Average Expropriation Risk
  - Presence of conflict mediation
- Generally our goal is to understand how Y varies as a function of X:

$$Y = f(X) + \text{error}$$

Three uses of regression

1. **Description** - parsimonious summary of the data
2. **Prediction/Estimation/Inference** - learn about parameters of the joint distribution of the data
3. **Causal Inference** - evaluate counterfactuals

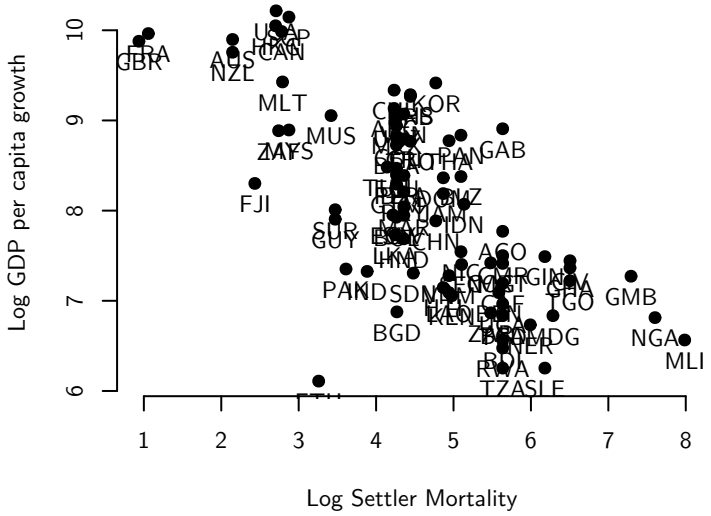
Describing relationships

- Remember that we had ways to summarize the relationship between variables in the population.
- Joint densities, covariance, and correlation were all ways to summarize the relationship between two variables.
- But these were population quantities and we only have samples, so we may want to estimate these quantities using their sample analogs

Scatterplots

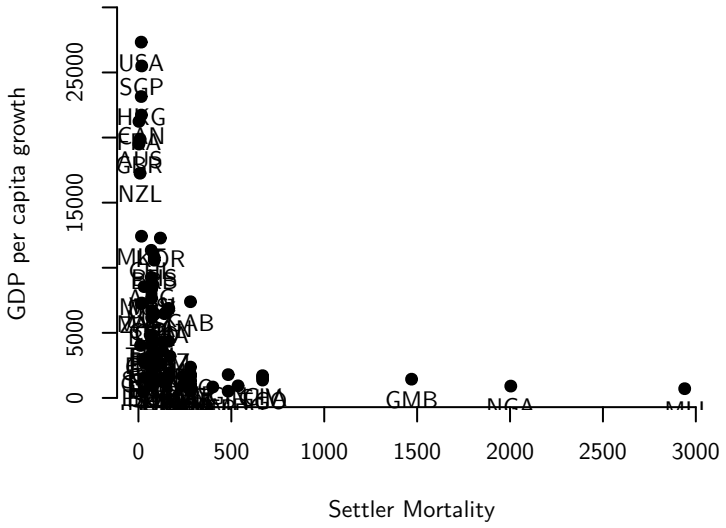
- Sample version of joint probability density.
- Shows graphically how two variables are related (positive, negative, linear, non-linear, etc)
- Use the `plot()` command in R.

```
ajr <- foreign::read.dta("ajr.dta")
plot(ajr$logem4, ajr$logpgp95, xlab = "Log Settler Mortality", ylab = "Log GDP per capita growth",
    pch = 19, bty = "n")
text(ajr$logem4, ajr$logpgp95, ajr$shortnam, pos = 1)
```



- Example of a non-linear relationship, where we use the unlogged version of GDP and settler mortality:

```
plot(exp(ajr$logem4), exp(ajr$logpgp95), xlab = "Settler Mortality", ylab = "GDP per capita growth",
    pch = 19, bty = "n")
text(exp(ajr$logem4), exp(ajr$logpgp95), ajr$shortnam, pos = 1)
```



Sample covariance

- The sample version of population covariance, $\sigma_{XY} = \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])]$.
- **Defintion** The **sample covariance** between Y_i and X_i is

$$S_{XY} = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{X}_n)(Y_i - \overline{Y}_n)$$

```
## cov() gets confused when you give it missing data
cov(ajr$logem4, ajr$logpgp95)
```

```
## [1] NA
```

```
## tell cov() to use only the pairwise complete observations:
cov(ajr$logem4, ajr$logpgp95, use = "pair")
```

```
## [1] -0.9881104
```

Sample correlation

- The sample version of population correlation, $\rho = \sigma_{XY}/\sigma_X\sigma_Y$.
- **Defintion** The **sample correlation** between Y_i and X_i is

$$\hat{\rho} = r = \frac{S_{XY}}{S_X S_Y} = \frac{\sum_{i=1}^{n}(X_i - \overline{X}_n)(Y_i - \overline{Y}_n)}{\sqrt{\sum_{i=1}^{n}(X_i - \overline{X}_n)^2 \sum_{i=1}^{n}(Y_i - \overline{Y}_n)^2}}$$

```
## cor() is very similar to cov()
cor(ajr$logem4, ajr$logpgp95)
```

```
## [1] NA
```

```
## and has the same solution to NAs:
cor(ajr$logem4, ajr$logpgp95, use = "pair")
```

```
## [1] -0.7047632
```

CONDITIONAL EXPECTATION

Conditional expectation review

- When we turn to predicting an outcome or estimating the effect of a covariate on an outcome, one way to describe relationships takes center stage: the conditional expectation function.
- **Definition** The **conditional expectation function** (CEF) or the **regression function** of Y given X, denoted $\mathbb{E}[Y|X = x]$ is the function that gives the mean of Y at various values of x.
- Note that this is a function of the *population* distributions.
- Regression at its most fundamental is about how the mean of Y changes as a function of X

Binary covariate: Difference in means as difference in conditional expectation

- We've been writing μ_y and μ_x for the means in different groups.
- Note that these are just conditional expectations. Define Y to be the loan amount, $X = 1$ to indicate a man, and $X = 0$ to indicate a woman and then we have:

$$\mu_m = E[Y|X = 1]$$
$$\mu_w = E[Y|X = 0]$$

- Notice here that since X can only take on two values, 0 and 1, then these two conditional means completely summarize the CEF.

- How do we calculate this? We've already done this: it's just the usual sample mean among the men and then the usual sample mean among the women:

$$\widehat{\mathbb{E}}[Y_i|X_i = 1] = \frac{1}{n_1} \sum_{i:X_i=1} Y_i$$
$$\widehat{\mathbb{E}}[Y_i|X_i = 0] = \frac{1}{n_0} \sum_{i:X_i=0} Y_i$$

- Here we have $n_1 = \sum_{i=1}^{n} X_i$ is the number of men in the sample and $n_0 = n - n_1$ is the number of women.
- The sum here $\sum_{i:X_i=1}$ is just summing only over the observations i such that have $X_i = 1$, meaning that i is a man.
- This is very straightforward: estimate the mean of Y conditional on X by just estimating the means within each group of X.

Binary covariate example

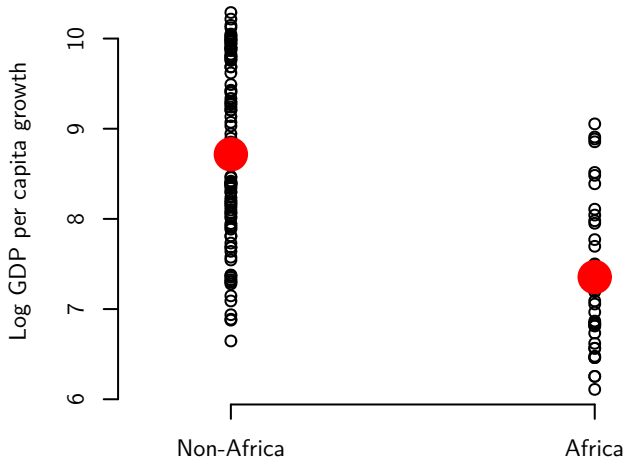
```
## mean of log GDP among non-African countries
mean(ajr$logpgp95[ajr$africa == 0], na.rm = TRUE)
```

```
## [1] 8.716383
```

```
## mean of log GDP among African countries
mean(ajr$logpgp95[ajr$africa == 1], na.rm = TRUE)
```

```
## [1] 7.355197
```

```
plot(ajr$africa, ajr$logpgp95, xlab = "", ylab = "Log GDP per capita growth",
    xaxt = "n", xlim = c(-0.25, 1.25), bty = "n")
axis(side = 1, at = c(0, 1), labels = c("Non-Africa", "Africa"))
points(x = 0, y = mean(ajr$logpgp95[ajr$africa == 0], na.rm = TRUE), pch = 19,
    col = "red", cex = 3)
points(x = 1, y = mean(ajr$logpgp95[ajr$africa == 1], na.rm = TRUE), pch = 19,
    col = "red", cex = 3)
```



Discrete covariate: sample conditional expectations

- In the last section we had a binary covariate. What if X is discrete?
- The same logic applies, we can still estimate $\mathbb{E}[Y|X = x]$ with the sample mean among those who have $X_i = x$:

$$\widehat{\mathbb{E}}[Y_i|X_i = x] = \frac{1}{n_x} \sum_{i:X_i=x} Y_i$$

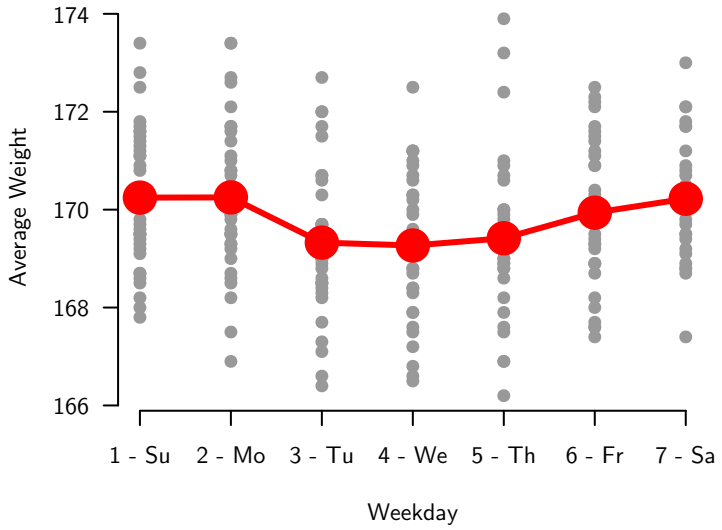
- For example, I've been collecting data on my own weight for a little under a year. What if I wanted to know how my weight (Y) varied by the day of the week (X)? This is a discrete, ordered variable.
- Well, we could just calculate the mean weight for each day of the week:

```
weight <- read.csv("weight.csv", stringsAsFactors = FALSE)
weight$weekday <- as.numeric(format(as.Date(weight$date, format = "%m/%d/%y%n%H:%M"),
    "%w")) + 1
weight$date <- as.Date(weight$date, format = "%m/%d/%y%n%H:%M")
day.means <- rep(NA, times = 7)
names(day.means) <- c("1 - Su", "2 - Mo", "3 - Tu", "4 - We", "5 - Th", "6 - Fr",
    "7 - Sa")
for (i in 1:7) {
    day.means[i] <- mean(weight$weight[weight$weekday == i])
}
day.means
```

```
##   1 - Su   2 - Mo   3 - Tu   4 - We   5 - Th   6 - Fr   7 - Sa
```

```
## 170.2457 170.2500 169.3265 169.2676 169.4156 169.9375 170.2231
```

```
plot(x = weight$weekday, y = weight$weight, bty = "n", xaxt = "n", xlab = "Weekday",
    ylab = "Average Weight", las = 1, pch = 19, col = "grey60")
points(x = 1:7, y = day.means, pch = 19, col = "red", cex = 3)
lines(x = 1:7, y = day.means, pch = 19, col = "red", lwd = 3)
axis(side = 1, at = 1:7, labels = names(day.means))
```



Moving to continuous covariates

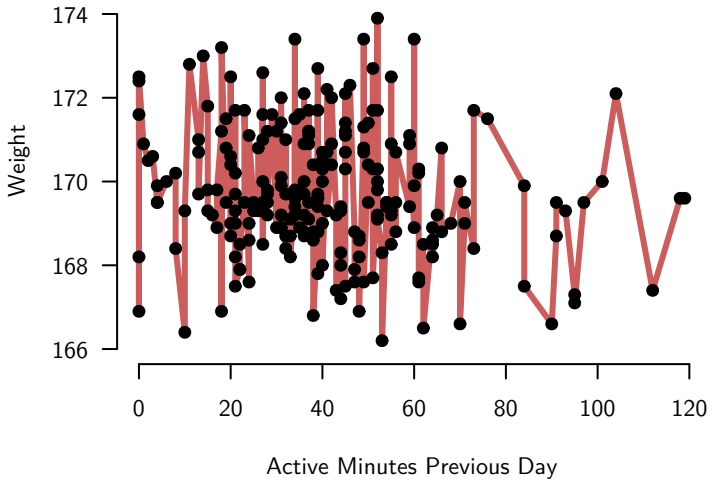
Continuous covariate (I): each unique value gets a mean

- What if X is continuous? Can we calculate a mean for every value of X?
- Not really, because remember the probability that two values will be the same in a continuous variable is 0.
- Thus, we'll end up with a very "jumpy" function, $\widehat{\mathbb{E}}[Y_i|X_i = x]$, since n_x will be at most 1 for any value of x.
- Let's look at the relationship between my weight and my active minutes in the previous day using this approach:

```
fitbit <- read.csv("fitbit.csv", stringsAsFactors = FALSE)
fitbit$date <- as.Date(fitbit$date, format = "%m/%d/%y")
## lag fitbit by one day
```

```
fitbit$date <- fitbit$date + 1
## merge fitbit and weight data
weight <- merge(weight, fitbit, by = "date")

plot(weight$active.mins[order(weight$active.mins)], weight$weight[order(weight$active.mins)],
    type = "l", lwd = 3, pch = 19, col = "indianred", las = 1, bty = "n", xlab = "Active Minutes Previous Day",
    ylab = "Weight", ylim = c(166, 175))
points(weight$active.mins, weight$weight, pch = 19)
```



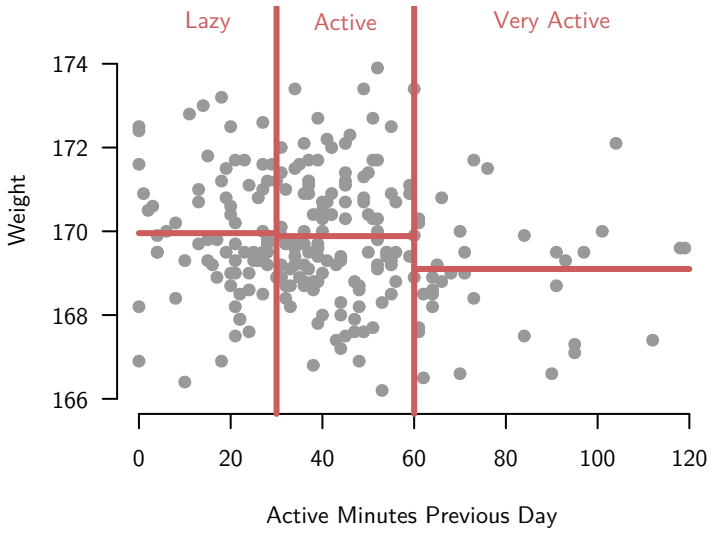
- You can imagine that this will jump around a lot from sample to sample. The estimates, $\widehat{\mathbb{E}}[Y_i|X_i = x]$, will have high sampling variance.

Continuous covariate (II): stratify and take means

- So, that seems like each value of X won't work, but maybe we can take the continuous variable and turn it into a discrete variable. We call this **stratification**.
- Once it's discrete, we can just calculate the means within each **strata**.
- For instance, we could break up the "Active Minutes" variable into 3 categories: lazy (< 30mins), active (30-60mins), and very active (>60min).

```
lowactivity.mean <- mean(weight$weight[weight$active.mins < 30])
medactivity.mean <- mean(weight$weight[weight$active.mins >= 30 & weight$active.mins <
    60])
hiactivity.mean <- mean(weight$weight[weight$active.mins >= 60])
```

```
plot(weight$active.mins, weight$weight, pch = 19, las = 1, bty = "n", xlab = "Active Minutes Previous Day",
    ylab = "Weight", ylim = c(166, 175), col = "grey60")
abline(v = c(30, 60), col = "indianred", lwd = 3)
text(x = c(15, 45, 90), y = c(175, 175, 175), c("Lazy", "Active", "Very Active"),
    col = "indianred")
segments(x0 = c(0, 30, 60), x1 = c(30, 60, 120), y0 = c(lowactivity.mean, medactivity.mean,
    hiactivity.mean), col = "indianred", lwd = 3)
```



- Now we're starting to see that there seems to be a negative relationship.
- But can we make this even more simple?

Continuous covariate (III): model relationship as a line

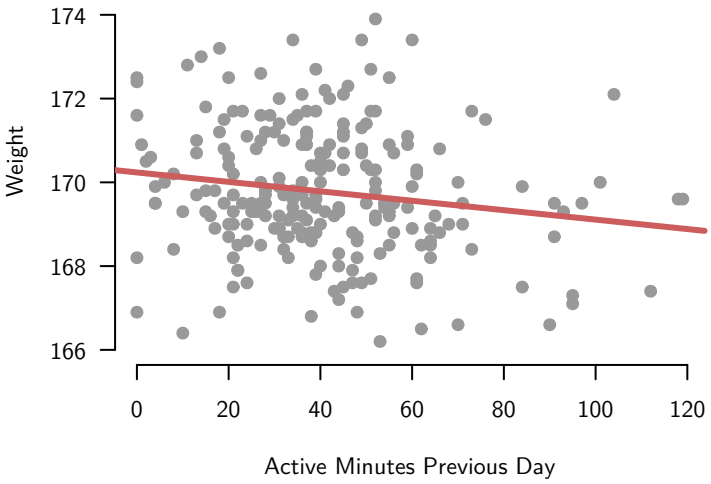
- The stratification approach was fairly crude: it assumed that means were constant within strata, but that seems wrong.
- Can we get a more global model for the regression function? Well, maybe we could assume that it is linear:

$$\mathbb{E}[Y_i|X_i = x] = \beta_0 + \beta_1 x$$

- Why might we do this? Parsimony, first and foremost: 2 numbers to predict any value.
- Some other nice properties we'll talk about in the coming weeks.

- Here is the linear regression function for the weight-active minutes relationships:

```
plot(weight$active.mins, weight$weight, pch = 19, las = 1, bty = "n", xlab = "Active Minutes Previous Day",
    ylab = "Weight", ylim = c(166, 175), col = "grey60")
abline(lm(weight ~ active.mins, data = weight), col = "indianred", lwd = 3)
```



- We'll see soon how we estimate this line. It's a bit more complicated that the stratify and calculate means.

Interpretation of the regression slope

- When we model the regression function as a line, we can interpret the parameters of the line in appealing ways:

  1. Intercept: the average outcome among units with $X = 0$ is β_0:

$$\mathbb{E}[Y|X = 0] = \beta_0 + \beta_1 0 = \beta_0$$

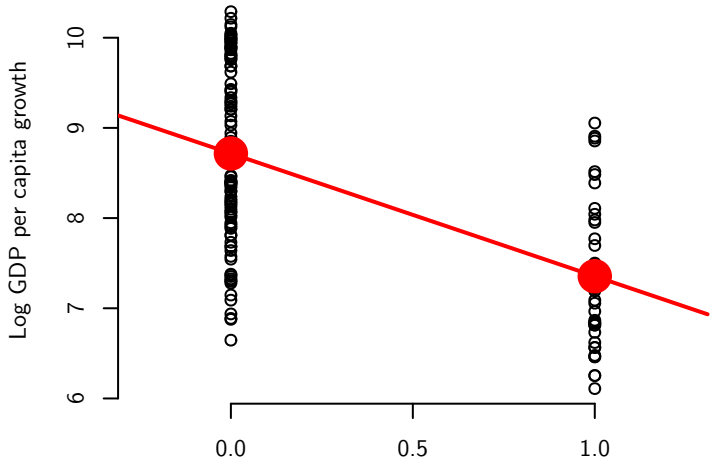
  2. Slope: a one-unit change in X is associated with a β_1 change in Y

$$\begin{aligned}\mathbb{E}[Y|X = x + 1] - \mathbb{E}[Y|X = x] &= (\beta_0 + \beta_1(x + 1)) - (\beta_0 + \beta_1 x) \\ &= \beta_0 + \beta_1 x + \beta_1 - \beta_0 - \beta_1 x \\ &= \beta_1\end{aligned}$$

Linear regression with a binary variable

- Using the two facts above, it's easy to see that when X is binary, then we have the following:
  1. Intercept: $\mathbb{E}[Y|X = 0] = \beta_0$
  2. Slope: average difference between $X = 1$ group and $X = 0$ group: $\beta_1 = \mathbb{E}[Y|X = 1] - \mathbb{E}[Y|X = 0]$
- Thus, we can read off the difference in means between two groups as the slope coefficient on a linear regression

```
plot(ajr$africa, ajr$logpgp95, xlab = "", ylab = "Log GDP per capita growth",
    xlim = c(-0.25, 1.25), bty = "n")
points(x = 0, y = mean(ajr$logpgp95[ajr$africa == 0], na.rm = TRUE), pch = 19,
    col = "red", cex = 3)
points(x = 1, y = mean(ajr$logpgp95[ajr$africa == 1], na.rm = TRUE), pch = 19,
    col = "red", cex = 3)
abline(lm(logpgp95 ~ africa, data = ajr), col = "red", lwd = 2)
```



Parametric vs. nonparametric models

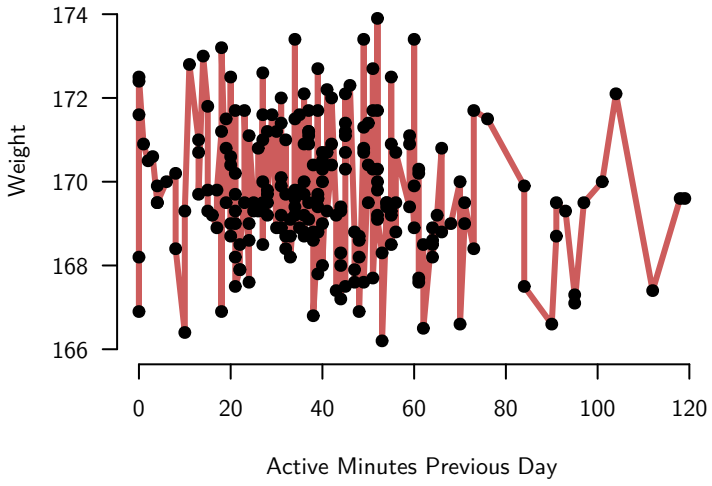
- The conditional mean approach for discrete independent variables are **nonparametric** because they make no assumptions about the functional form of $\mathbb{E}[Y_i|X_i = x]$.
- We just estimate the mean among each value of x.
- With continuous independent variables, this approach breaks down because of the number of values.

- Need to make **parametric** assumptions about the functional form of $\mathbb{E}[Y_i|X_i = x]$ in order to make progress
- These are parametric because they involve writing the functional form in terms of parameters, like the linear model.

Bias-variance tradeoff

- How we model the regression function, $\mathbb{E}[Y_i|X_i = x]$, affects our the behavior of our estimates:

```
plot(weight$active.mins[order(weight$active.mins)], weight$weight[order(weight$active.mins)],
    type = "l", lwd = 3, pch = 19, col = "indianred", las = 1, bty = "n", xlab = "Active Minutes Previous Day",
    ylab = "Weight", ylim = c(166, 175))
points(weight$active.mins, weight$weight, pch = 19)
```

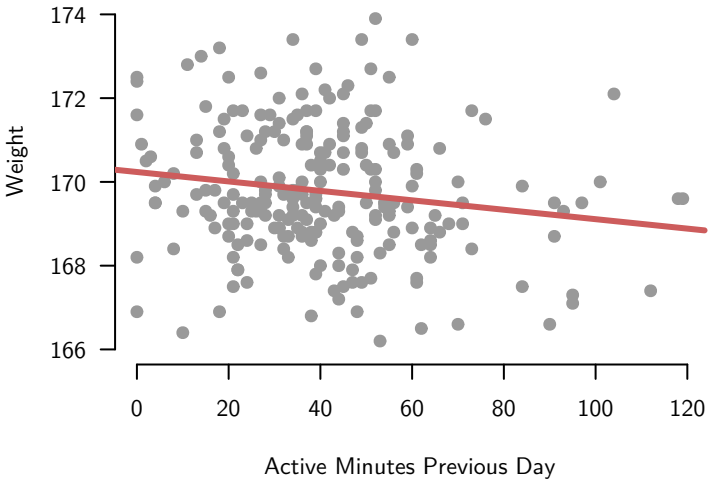


- Low bias (function "nails" every point)
- High variance (drastic changes from sample to sample)

Bias-variance tradeoff

- How we model the regression function, $\mathbb{E}[Y_i|X_i = x]$, affects our the behavior of our estimates:

```
plot(weight$active.mins, weight$weight, pch = 19, las = 1, bty = "n", xlab = "Active Minutes Previous Day",
    ylab = "Weight", ylim = c(166, 175), col = "grey60")
abline(lm(weight ~ active.mins, data = weight), col = "indianred", lwd = 3)
```



- Higher bias (misses "local" variation)
- Low variance (slope and intercept will only change slightly from sample to sample)

LEAST SQUARES

Back up and review

- To review our approach:
  - We wanted to estimate the CEF/regression function $\mathbb{E}[Y|X = x]$, but found that it was hard to do nonparametrically
  - So we're going to *model* it: place restrictions on its functional.
  - Easiest functional form is a line:

$$\mathbb{E}[Y_i|X_i = x] = \beta_0 + \beta_1 x$$

- β_0 and β_1 are population parameters just like μ or σ^2!
- Need to estimate them in our samples! But how?

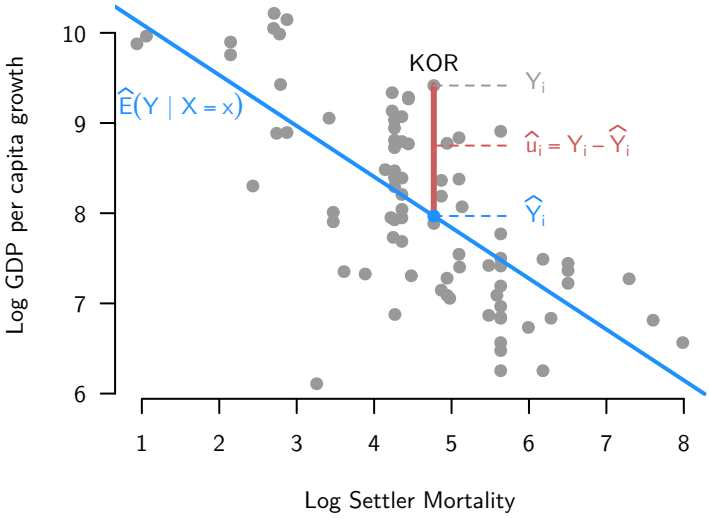
Simple linear regression model

- We'll need some terms and concepts first. Let's write our model:

$$Y_i = \mathbb{E}[Y_i|X_i] + u_i$$
$$Y_i = \beta_0 + \beta_1 X_i + u_i$$

- Now, suppose we have some estimates of the slope, $\hat{\beta}_1$, and the intercept, $\hat{\beta}_0$. Then the fitted or sample regression line is

$$\widehat{\mathbb{E}}[Y_i|X_i = x] = \widehat{\beta}_0 + \widehat{\beta}_1 x$$



- **Definition** A **fitted value** or **predicted value** is the estimated conditional mean of Y_i for a particular observation with independent variable X_i:

$$\widehat{Y}_i = \widehat{\mathbb{E}}[Y_i|X_i] = \widehat{\beta}_0 + \widehat{\beta}_1 X_i$$

- **Definition** The **residual** is the difference between the actual value of Y_i and the predicted value, $\widehat{Y}_i$:

$$\widehat{u}_i = Y_i - \widehat{Y}_i = Y_i - \widehat{\beta}_0 - \widehat{\beta}_1 X_i$$

Minimize the residuals

- The residuals, $\widehat{u}_i = Y_i - \widehat{\beta}_0 - \widehat{\beta}_1 X_i$, tell us how well the line fits the data.
  - Larger magnitude residuals means that points are very far from the line
  - Residuals close to 0 mean points very close to the line
- The smaller the magnitude of the residuals, the better we are doing at predicting Y
- Choose the line that minimizes the residuals

Minimizing the residuals

- Let $\widetilde{\beta}_0$ and $\widetilde{\beta}_1$ be possible values of the intercept and slope
- **Least absolute deviations** (LAD) regression:

$$(\widehat{\beta}_0^{LAD}, \widehat{\beta}_1^{LAD}) = \underset{\widetilde{\beta}_0, \widetilde{\beta}_1}{\arg\min} \sum_{i=1}^{n} |Y_i - \widetilde{\beta}_0 - \widetilde{\beta}_1 X_i|$$

- **Least squares** (LS) regression:

$$(\widehat{\beta}_0, \widehat{\beta}_1) = \underset{\widetilde{\beta}_0, \widetilde{\beta}_1}{\arg\min} \sum_{i=1}^{n} (Y_i - \widetilde{\beta}_0 - \widetilde{\beta}_1 X_i)^2$$

- Sometimes called **ordinary least squares** (OLS)

Why least squares?

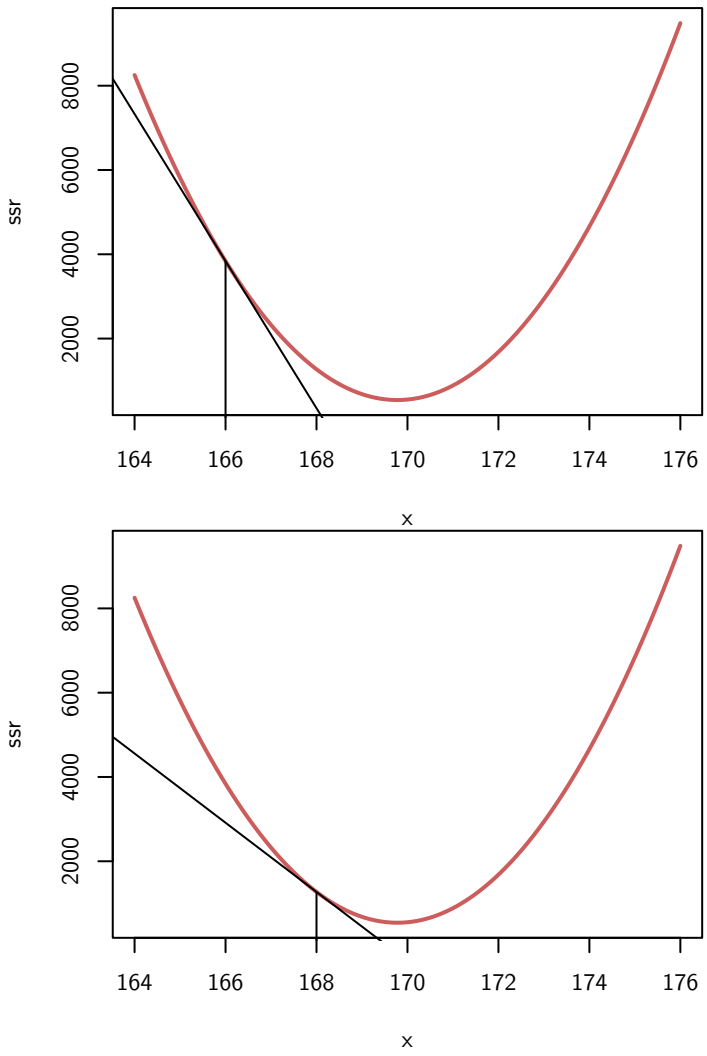
- Easy to derive the least squares estimator
- East to investigate the properties of the least squares estimator
- Least squares is optimal in a certain sense that we'll see in the coming weeks

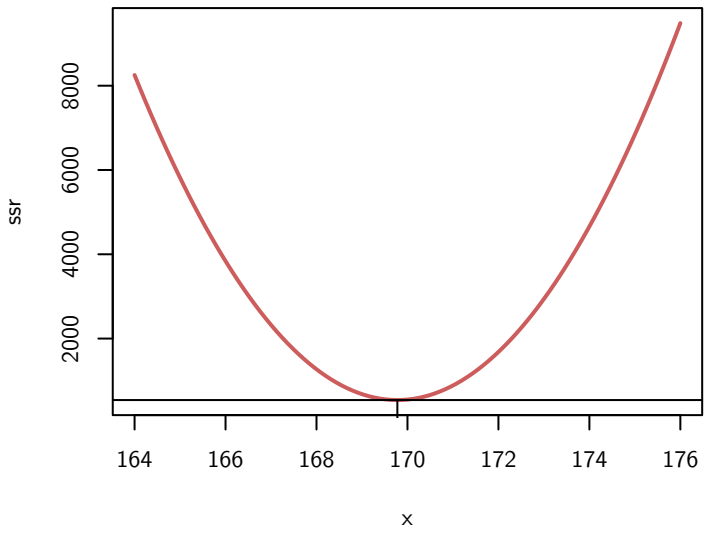
Least squares and the mean

- Let's understand how to calculate least squares analytically
- Consider the simplest case: minimizing the least squares for a single variable Y
- n sample observations: $Y_1, Y_2, Y_3, \ldots, Y_n$.
- What is a single number $\tilde{\mu}$ that summarizes all Y's?
- Find $\tilde{\mu}$ that minimizes the **sum of squared residuals** (SSR)

$$S(\tilde{\mu}) = \sum_{i=1}^{n} (Y_i - \tilde{\mu})^2.$$

- How do we solve this?
    1. Calculate the derivative of S with respect to $\tilde{\mu}$
    2. Set the derivative equal to 0
    3. Solve for $\tilde{\mu}$ and replace $\tilde{\mu}$ with the solution
- What does the sum of the squared residuals (SSR) function look like?


ssr
2000
4000
6000
8000
164
166
168
170
172
174
176
x
ssr
2000
4000
6000
8000
164
166
168
170
172
174
176
x



Minimize the SSR

1. Calculate the derivative

$$\begin{aligned} S(\tilde{\mu}) &= \sum_{i=1}^{n} (Y_i - \tilde{\mu})^2 \\ &= \sum_{i=1}^{n} (Y_i^2 - 2Y_i\tilde{\mu} + \tilde{\mu}^2) \end{aligned}$$

$$\frac{\partial S(\tilde{\mu})}{\partial \tilde{\mu}} = \sum_{i=1}^{n} (-2Y_i + 2\tilde{\mu})$$

2. Setting it to zero:

$$0 = \sum_{i=1}^{n} (-2Y_i + 2\tilde{\mu})$$

3. And solve:

$$\begin{aligned} 0 &= \sum_{i=1}^{n}(-2Y_i + 2\widehat{\mu}) \\ 0 &= -2\left(\sum_{i=1}^{n} Y_i\right) + 2n\widehat{\mu} \\ -2n\widehat{\mu} &= -2\sum_{i=1}^{n} Y_i \\ \widehat{\mu} &= \frac{1}{n}\sum_{i=1}^{n} Y_i \end{aligned}$$

- Therefore, the sample average is the least squares estimator.

Deriving the OLS estimator

- Now we look at two variables, Y and X
- n pairs of sample observations: $(Y_1, X_1), (Y_2, X_2), \ldots, (Y)_n, X_n)$
- Let $\{\tilde{\beta}_0, \tilde{\beta}_1\}$ be possible values for $\{\beta_0, \beta_1\}$
- Define the least squares objective function:

$$S(\tilde{\beta}_0, \tilde{\beta}_1) = \sum_{i=1}^{n}(Y_i - \tilde{\beta}_0 - X_i\tilde{\beta}_1)^2.$$

- How do we derive the LS estimators for β_0 and β_1?

    1. Take partial derivatives of S with respect to $\tilde{\beta}_0$ and $\tilde{\beta}_1$.
    2. Set each of the partial derivatives to 0
    3. Solve for $\{\tilde{\beta}_0, \tilde{\beta}_1\}$ and replace them with the solutions

Taking the partial derivatives

$$\begin{aligned} S(\tilde{\beta}_0, \tilde{\beta}_1) &= \sum_{i=1}^{n}(Y_i - \tilde{\beta}_0 - X_i\tilde{\beta}_1)^2 \\ &= \sum_{i=1}^{n}(Y_i^2 - 2Y_i\tilde{\beta}_0 - 2Y_i\tilde{\beta}_1 X_i + \tilde{\beta}_0^2 + 2\tilde{\beta}_0\tilde{\beta}_1 X_i + \tilde{\beta}_1^2 X_i^2) \end{aligned}$$

$$\frac{\partial S(\tilde{\beta}_0, \tilde{\beta}_1)}{\partial \tilde{\beta}_0} = \sum_{i=1}^{n}(-2Y_i + 2\tilde{\beta}_0 + 2\tilde{\beta}_1 X_i)$$
$$\frac{\partial S(\tilde{\beta}_0, \tilde{\beta}_1)}{\partial \tilde{\beta}_1} = \sum_{i=1}^{n}(-2Y_i X_i + 2\tilde{\beta}_0 X_i + 2\tilde{\beta}_1 X_i^2)$$

First order conditions

- The first order conditions are:

$$0 = \sum_{i=1}^{n}(-2Y_i + 2\tilde{\beta}_0 + 2\tilde{\beta}_1 X_i)$$
$$0 = \sum_{i=1}^{n}(-2Y_i X_i + 2\tilde{\beta}_0 X_i + 2\tilde{\beta}_1 X_i^2)$$

now solving for $\tilde{\beta}_0$ and $\tilde{\beta}_1$ yields the **normal equations**:

$$\widehat{\beta}_0 n = \left(\sum_{i=1}^{n} Y_i\right) - \widehat{\beta}_1 \left(\sum_{i=1}^{n} X_i\right)$$

$$\widehat{\beta}_1 \sum_{i=1}^{n} X_1^2 = \left(\sum_{i=1}^{n} X_i Y_i\right) - \widehat{\beta}_0 \left(\sum_{i=1}^{n} X_i\right)$$

rearranged yield the **OLS estimators**:

$$\widehat{\beta}_0 = \overline{Y} - \widehat{\beta}_1 \overline{X}$$

$$\widehat{\beta}_1 = \frac{\sum_{i=1}^{n}(X_i - \overline{X})(Y_i - \overline{Y})}{\sum_{i=1}^{n}(X_i - \overline{X})^2}$$

- Note that:

$$\widehat{\beta}_1 = \frac{\sum_{i=1}^{n}(X_i - \overline{X})(Y_i - \overline{Y})}{\sum_{i=1}^{n}(X_i - \overline{X})^2} = \frac{\text{Sample Covariance between } X \text{ and } Y}{\text{Sample Variance of } X}$$

Mechanical properties of least squares

- The residuals will be 0 on average:

$$\sum_{i=1}^{n} \widehat{u}_i = 0$$

- The residuals will be uncorrelated with the predictor:

$$\sum_{i=1}^{n} X_i \widehat{u}_i = 0$$

- The residuals will be uncorrelated with the fitted values:

$$\sum_{i=1}^{n} \widehat{Y}_i \widehat{u}_i = \sum_{i=1}^{n} \widehat{Y}_i \widehat{u}_i - n \sum_{i=1}^{n} \widehat{Y}_i \sum_{i=1}^{n} \widehat{u}_i = (n-1)\text{Cov}(\widehat{Y}_i, \widehat{u}_i) = 0$$

AJR Example in R

- Let's use those simple formulas we just learned:

```
cov.xy <- cov(ajr$logem4, ajr$logpgp95, use = "pair")
var.x <- var(ajr$logem4, na.rm = TRUE)
cov.xy/var.x
```

```
## [1] -0.5816937
```

```
mean(ajr$logpgp95, na.rm = TRUE) - cov.xy/var.x * mean(ajr$logem4, na.rm = TRUE)
```

```
## [1] 10.97596
```

- Compare it to what `lm()`, the OLS function in R produces:

```
coef(lm(logpgp95 ~ logem4, data = ajr))
```

```
## (Intercept)      logem4
##  10.6602465  -0.5641215
```

- Why aren't these equal?

- Note that when one variable is o on average, then multiplication and adding is equal to the covariance

$$\begin{aligned}
\text{Cov}(X_i, \widehat{u}_i) &= \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{X})(\widehat{u}_i - \overline{\widehat{u}}) \\
&= \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{X})\widehat{u}_i \\
&= \frac{1}{n-1}\sum_{i=1}^{n}X_i\widehat{u}_i - \frac{1}{n-1}\sum_{i=1}^{n}\overline{X}\widehat{u}_i \\
&= \frac{1}{n-1}\sum_{i=1}^{n}X_i\widehat{u}_i - \frac{1}{n-1}\overline{X}\sum_{i=1}^{n}\widehat{u}_i \\
&= \frac{1}{n-1}\sum_{i=1}^{n}X_i\widehat{u}_i \\
&= 0
\end{aligned}$$

- The same principal applies to the relationship between the residuals and the fitted values.

Mechanical properties of least squares in R

```
mod <- lm(logpgp95 ~ logem4, data = ajr)
mean(residuals(mod))
```

```
## [1] -2.623502e-18
```

```
## mod$model is the data used in the lm() call
cor(mod$model$logem4, residuals(mod))
```

```
## [1] -3.184875e-17
```

```
cor(fitted(mod), residuals(mod))
```

```
## [1] -1.160489e-16
```

APPENDIX

Normal equations to OLS estimators

- For the intercept, just divide by n:

$$\widehat{\beta}_0 n = \left(\sum_{i=1}^{n} Y_i\right) - \widehat{\beta}_1 \left(\sum_{i=1}^{n} X_i\right)$$
$$\widehat{\beta}_0 = \overline{Y} - \widehat{\beta}_1 \overline{X}$$

Normal equations to OLS estimators

- Now, rearrange this equation a bit:

$$\widehat{\beta}_0 n = \left(\sum_{i=1}^{n} Y_i\right) - \widehat{\beta}_1 \left(\sum_{i=1}^{n} X_i\right)$$
$$\widehat{\beta}_0 = \overline{Y} - \widehat{\beta}_1 \overline{X}$$
$$\widehat{\beta}_0 \left(\sum_{i=1}^{n} X_i\right) = \overline{Y} \left(\sum_{i=1}^{n} X_i\right) - \widehat{\beta}_1 \overline{X} \left(\sum_{i=1}^{n} X_i\right)$$
$$\widehat{\beta}_0 \left(\sum_{i=1}^{n} X_i\right) = \left(\sum_{i=1}^{n} \overline{Y} X_i\right) - \widehat{\beta}_1 \left(\sum_{i=1}^{n} \overline{X} X_i\right)$$

- Plug this into the second normal equation:

$$\begin{aligned}
\widehat{\beta}_1 \sum_{i=1}^{n} X_i^2 &= \left(\sum_{i=1}^{n} X_i Y_i\right) - \widehat{\beta}_0 \left(\sum_{i=1}^{n} X_i\right) \\
\widehat{\beta}_1 \sum_{i=1}^{n} X_i^2 &= \left(\sum_{i=1}^{n} X_i Y_i\right) - \left(\sum_{i=1}^{n} \overline{Y} X_i\right) + \widehat{\beta}_1 \left(\sum_{i=1}^{n} \overline{X} X_i\right) \\
\widehat{\beta}_1 \left[\sum_{i=1}^{n} \left(X_i^2 - \overline{X} X_i\right)\right] &= \left(\sum_{i=1}^{n} X_i Y_i - \overline{Y} X_i\right) \\
\widehat{\beta}_1 \sum_{i=1}^{n} X_i \left(X_i - \overline{X}\right) &= \sum_{i=1}^{n} X_i (Y_i - \overline{Y}) \\
\widehat{\beta}_1 \sum_{i=1}^{n} X_i \left(X_i - \overline{X}\right) - \widehat{\beta}_1 \overline{X} \sum_{i=1}^{n} \left(X_i - \overline{X}\right) &= \sum_{i=1}^{n} X_i (Y_i - \overline{Y}) - \overline{X} \sum_{i=1}^{n} (Y_i - \overline{Y}) \\
\widehat{\beta}_1 \sum_{i=1}^{n} \left(X_i (X_i - \overline{X}) - \overline{X} (X_i - \overline{X})\right) &= \sum_{i=1}^{n} \left(X_i (Y_i - \overline{Y}) - \overline{X} (Y_i - \overline{Y})\right) \\
\widehat{\beta}_1 \sum_{i=1}^{n} \left(X_i - \overline{X}\right)^2 &= \sum_{i=1}^{n} (X_i - \overline{X}_i)(Y_i - \overline{Y})
\end{aligned}$$

- **Question** Here in the second to last line, we used the fact that the sum of deviations from the mean is equal to o: $\sum_{i=1}^{n}(Y_i - \overline{Y})$. Can you show that this is true?

- And rearrange them to get the **OLS estimators**:

$$\widehat{\beta}_1 = \frac{\sum_{i=1}^{n}(X_i - \overline{X})(Y_i - \overline{Y})}{\sum_{i=1}^{n}(X_i - \overline{X})^2}$$