STAT340 Lecture 08: Prediction

Brian Powers

10/31/2024

Introduction

In these notes, we will introduce the task of prediction and begin talking about some of the fundamental tools for performing prediction, focusing on linear regression.

The task of prediction will be a theme for much of the remainder of the course, and it is not an exaggeration to say that prediction is the fundamental task that lies at the heart of machine learning.

Tasks like image classification (e.g., "does this image contain a cat or not?") are very naturally cast as prediction problems, as are many of the most basic problems in machine learning (e.g., predict how likely a person is to engage with the next piece of content in their feed).

Learning objectives

After this lesson, you will be able to

- Explain both simple and multiple linear regression
- Use R to run linear regression on a given data set and interpret the resulting coefficient estimates
- Explain what it means to associate a p-value to an estimated coefficient

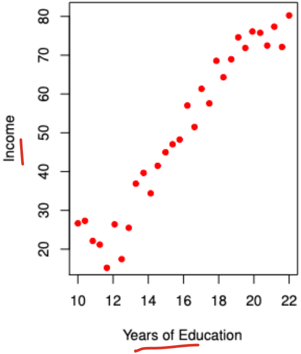
Prediction: an overview I

In a prediction problem, we are given data pairs $(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)$ and we want to use X_i to predict Y_i.

We call the X_i values the *predictors* (also called the *independent variables*), and we call our Y_i values the *responses* (or the *dependent variables* or the *outcomes*).

Example - education and income

Let's look at an example that we discussed in our very first lecture.



Here, our (X_i, Y_i) pairs correspond to years of education (X_i) and income (Y_i). That is, our predictors are years of education, and our responses are income.

Our goal is to use this data to learn a function that maps years of education to income. That is, we want a function that takes years of education as input and outputs a prediction as to how much income we predict for a person with that income.

Writing down the model I

To recap, a simple linear model has an outcome y and single predictor x. It is defined by the following equation:

intercept

slope coefficient

$$\begin{aligned} y_i &= \beta_0 + \beta_1 x_i + \epsilon_i \\ &= \hat{y}_i + \epsilon_i \end{aligned}$$

Y_i = this is r.v Y_i

$\hat{y}_i$ our predictor / prediction of y_i

where $i = 1, 2, \ldots, n$, and the error terms $\epsilon_i \sim N(0, \sigma^2)$ are independent over $i = 1, 2, \ldots, n$. The $\hat{y}_i$ notation is to stress that once we have chosen values for the coefficients β_0 and β_1, our prediction of the response of the i-th data point is $\hat{y}_i = \beta_0 + \beta_1 x_i$.

This equation represents our *model*, not the truth! We want to choose β_0 and β_1 so that this model describes our observed data as well as possible, but we have to bear in mind that this linearity assumption, that y_i is (exactly or approximately) expressible as $\beta_0 + \beta_1 x_i$, is an *assumption*. Our model will be good at predicting outcomes only in so far as this model agrees with reality.

Writing down the model II

The subscript i in our regression equation indexes the n observations in the dataset. Think of i as a row number. So another way to think about our model

$$\boxed{y_i = \beta_0 + \beta_1 x_i + \epsilon_i,}$$

Parameters

input var

r.v.

i index from 1, ..., n

is as a system of n equations,

$$\left\{\begin{aligned}
y_1 &= \beta_0 + \beta_1 x_1 + \epsilon_1 \\
y_2 &= \beta_0 + \beta_1 x_2 + \epsilon_2 \\
&\vdots \\
y_i &= \beta_0 + \beta_1 x_i + \epsilon_i \\
&\vdots \\
y_{n-1} &= \beta_0 + \beta_1 x_{n-1} + \epsilon_{n-1} \\
y_n &= \beta_0 + \beta_1 x_n + \epsilon_n.
\end{aligned}\right.$$

Writing down the model III

The error terms ϵ_i in a linear model correspond, essentially, to the part of the variation in the data that remains unexplained by the deterministic portion of the model (encoded in the linear function $\beta_0 + \beta_1 x$).

One of the key assumptions of a linear model is that the residuals are independent and have mean zero, $\mathbb{E}\epsilon_i = 0$. Most typically, we further assume that they are normally distributed with mean 0 and variance σ^2. We'll do that here in these notes, but this choice can sometimes be "relaxed", in the sense that we may not need to assume that the errors are normal for linear regression to work, depending on what we want to do downstream. Later in your studies, when you take your mathematical statistics course, you'll put that statement on firmer ground; for now, we'll have to leave it vague.)

Interpreting simple linear regression I

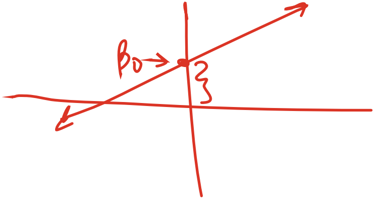
So let's suppose that we're using the linear model

$$\boxed{Y_i = \underline{\beta_0} + \underline{\beta_1} X_i + \epsilon_i.}$$

To simplify things, let's just ignore the error term for a moment. After all, ϵ_i just captures uncertainty in our measurements. In the ideal world of no measurement error, our model predicts that for a particular choice of predictor x, we will measure the response

$$\underline{y} = \underline{\beta_0} + \underline{\beta_1}\underline{x}.$$

Now, let's first consider what happens when $x = 0$. Then $y = \beta_0$. Said another way, if we plotted the line $y = \beta_0 + \beta_1 x$, β_0 would be the intercept of our model.

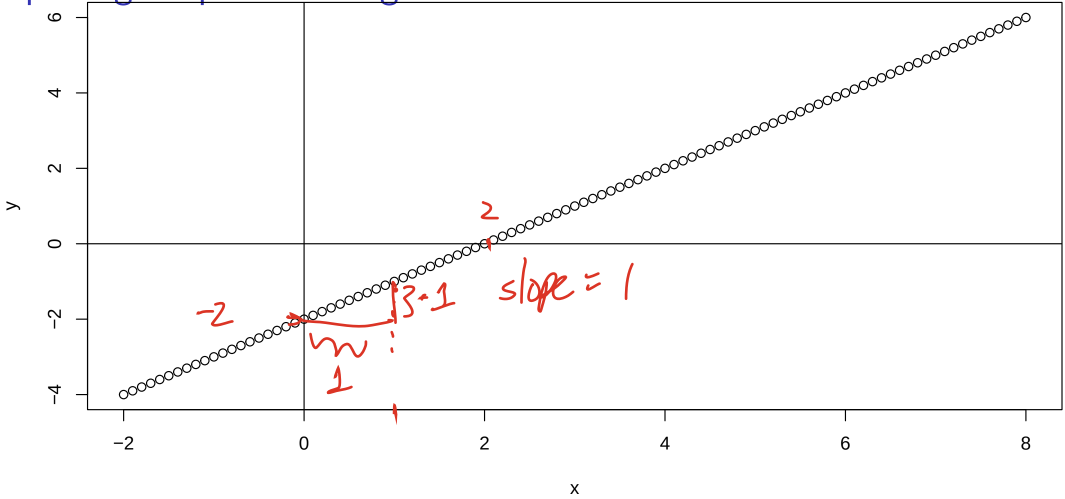


Interpreting simple linear regression II

For example, here's the function with $\beta_0 = -2$ and $\beta_1 = 1$.

```
x <- seq(-2,8,0.1)
beta0 <- -2;
beta1 <- 1;
y <- beta0 + beta1 * x;
plot(x,y); abline(h=0); abline(v=0);
```

Interpreting simple linear regression III

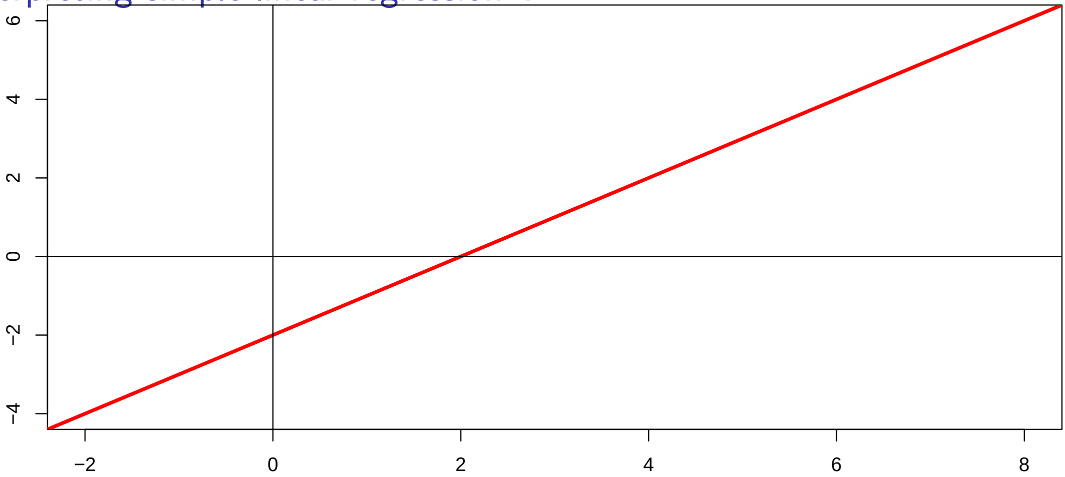


Interpreting simple linear regression IV

Equivalently, since we know that this function forms a line:

```r
# Pass NULL to plot to create an empty plot with axes.
plot(NULL, xlab="", ylab="", xlim=c(-2, 8), ylim=c(-4, 6))
abline(a=-2, b=1, col='red', lw=3);
abline(h=0);
abline(v=0);
```

Interpreting simple linear regression V



Interpreting simple linear regression VI

Looking at those two plots, it's clear that $\beta_0 = -2$ is indeed the intercept of our function. But to reiterate, the "typical" interpretation of the parameter β_0 is as describing what would happen if we observed a data point for which our predictor x were equal to zero.

Now, it's pretty obvious that $\underline{\beta_1}$ is the slope of our function. But how do we interpret it? Well, let's suppose that we take one measurement with predictor x. Our model says that (again, ignoring the error term for now) we will see a response

$$y = \beta_0 + \beta_1 x.$$

Interpreting simple linear regression VII

Now, let's suppose we take another measurement, this time at predictor value $x + 1$. Our model predicts that we will measure the response

$$y' = \beta_0 + \beta_1(x + 1).$$

If we subtract one from the other, we have

$$y' - y = \beta_0 + \beta_1(x + 1) - (\beta_0 + \beta_1 x) = \beta_1.$$

In other words, β_1 is the change in response that our model predicts if we increase the value of our predictor by one unit.

Interpreting simple linear regression VIII

Example: income and education

Let's come back to our model predicting income (in tens of thousands of dollars) from education, and suppose that we have fit a model of the form

Y in $10,000

$$y = 20 + 4x.$$

So our coefficients are $\beta_0 = 20$ and $\beta_1 = 4$. Thus, our model predicts that an increase in education by 1 year is associated with in an increase of \$40K in salary (4 times our unit of measurement, \$10K/year).

Similarly, since $\beta_0 = 20$, our model "predicts" that a person with zero years of education will receive a salary of \$~~20~~K per year.

\$200

Interpreting simple linear regression IX

Example: a cautionary tale

Interpreting the intercept as describing the response at $x = 0$ can get a little bit weird if we push the idea too far. Let's consider a similar problem, this time of predicting income from height. Suppose that we fit a model that predicts income (in thousands of dollars) from height (in centimeters),

$$y = 10 + 0.4x,$$

$(.4)(10{,}000) = 4000$

where x is height in centimeters (note that the units on this example don't really make sense–don't let that bother you; it's not the point).

The intercept of this model is $\beta_0 = 10$. So our model "predicts" that a person with height $x = 0$ would make a salary of \$10,000 per year. Now, that's all fine and good, except that I, for one, have never encountered a person with height 0 cm.

So our model makes a prediction, but it is making a prediction on an input that we don't really every expect to encounter in the real word.

Interpreting simple linear regression X

The high-level point is that often our linear regression model really only makes sense over a certain range of values, and we should be careful when using our model to extrapolate to "strange" values of x.

Even though we might be able to associate a response with any particular input x, that doesn't mean that every such input is realistic. These matters will mostly have to wait for later courses on modeling, but it's a point we'll come back to a couple of times over the next few weeks, and it's a common pitfall in interpreting linear regression models, so it's worth bringing it to your attention now.

Caution: causality I

It is always tempting in talking about models like this to say, having fit a model, that "increasing the value of the predictor by one unit causes an increase in the response by one unit" or that "increasing the predicor by one unit results in an increase of such and such amount". Indeed, many statisticians will say something like this when speaking informally. Still, we should be careful to avoid giving a causal interpretation to our findings.

$$\hat{y} = 37 + 4.3x$$

" a 1 unit increase in x is associated with an average increase of 4.3 in y "

Caution: causality II

For example, suppose that we fit a linear regression model to predict the number of cancer cases per year in Wisconsin based on pollution levels (measured in, say, PM2.5), and we estimate $\beta_1 = 10.2$. We might be tempted to say that a unit increase of PM2.5 *causes*, on average*, an additional 10.2 cancer cases.

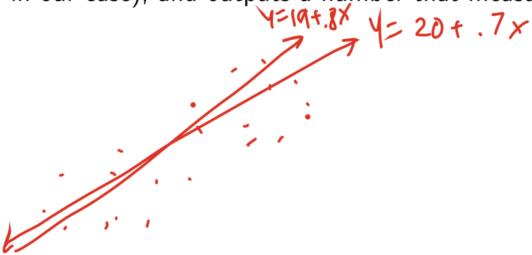
For better or worse, this is a stronger statement than what we can conclude from a linear model fitted in this way. We can only say that a unit increase in PM2.5 is *associated* with an increase of 10.2 cancer cases. This is the old "correlation is not causation" saying, wearing a slightly different hat.

There is a whole area in statistics called *causal inference* that attempts to use statistics to make causal statements, but it is, unfortunately, outside the scope of the course.

Fitting the model

Suppose that we have chosen values of our coefficients β_0 and β_1 in our regression model. How do we decide how "good" or "bad" this choice of coefficients is? We need a function that takes a particular choice of β_0 and β_1 and outputs a number that measures how well or poorly the resulting model describes our data.

In the setting where larger values of this function correspond to worse model fit, we call this kind of a function a *loss function*: it takes a choice of model parameters (i.e., coefficients β_0 and β_1, in our case), and outputs a number that measures how poorly our model fits the data.



Choosing a loss: sum of squares I

There are lots of functions we could choose to use as our loss function, but by far the most common choice is the *residual sum of squares* (RSS), sometimes called the *sum of squared errors* (SSE):

$$\ell(\beta_0, \beta_1) = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} (y_i - (\beta_0 + \beta_1 x_i))^2. \qquad \geq 0$$

The terms $y_i - \hat{y}_i = y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i)$ are called the *residuals*. The word *residual* comes from the word *residue* (cue flashback to chem lab?), which refers to something that is left over. The residuals are what is left over after we try to predict the responses from the predictors $x_1, x_2, \ldots, x_n$.

$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$ actual Y value

$\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 X_i$ our predicted Y value

Choosing a loss: sum of squares II

Our goal is then to choose our coefficients β_0 and β_1 to minimize the sum of squared residuals loss in the equation above. We call this *ordinary least squares* (OLS) regression. "Least squares" because, well, we're minimizing the sum of squares. "Ordinary" because there are other sums of squares we could look at that would be a little less ordinary (see here for details, if you're curious).

Let's note that the sum of squared errors is not the only possible loss we could choose. For example, we might try to minimize the sum of absolute deviations,

$$\sum_{i=1}^{n} |y_i - \hat{y}_i| = \sum_{i=1}^{n} |y_i - (\beta_0 + \beta_1 x_i)| \, .$$

As we've seen in recent lectures, though, trying to minimize this loss with respect to our coefficients β_0 and β_1 can be challenging.

Minimizing the loss I

So we have a loss function

$$\ell(\beta_0, \beta_1) = \sum_{i=1}^{n} (y_i - (\beta_0 + \beta_1 x_i))^2,$$

and we want to choose β_0 and β_1 to minimize this quantity.

To do that, we are going to dust off our calculus textbooks, take derivatives, set those derivatives equal to zero, and solve for β_0 and β_1. That is, we want to solve

$$\frac{\partial \ell(\beta_0, \beta_1)}{\partial \beta_0} = 0 \quad \text{and} \quad \frac{\partial \ell(\beta_0, \beta_1)}{\partial \beta_1} = 0.$$

Minimizing the loss II

I'll spare you the mathematical details; if you're curious, you can find a derivation of the solution in any introductory regression book, or here

The important point is that we find that our estimates should be

$$\left[\begin{aligned}\hat{\beta}_0 &= \bar{y} - \hat{\beta}_1 \bar{x} \\ \hat{\beta}_1 &= \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2},\end{aligned}\right.$$

where $\bar{x}$ and $\bar{y}$ are the means of the predictors and responses, respectively:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \quad \text{and} \quad \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i.$$

Interpreting the estimated slope I

By our definition of $\hat{\beta}_1$, we have

$$\hat{\beta}_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2} = \frac{\frac{1}{n}\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\frac{1}{n}\sum_{i=1}^n (x_i - \bar{x})^2}, = \frac{\frac{1}{n}\sum \sim}{s_x^2}$$

where we multiplied the numerator and denominator by $1/n$.

Now, let's notice that the denominator is just the (uncorrected) sample variance of the predictors:

$$s_x^2 = \frac{1}{n}\sum_{i=1}^n (x_i - \bar{x})^2.$$

If we define the analogous quantity for the predictors,

Interpreting the estimated slope II

$$s_y^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2,$$

we have

$$\hat{\beta}_1 = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{s_x^2} = \frac{s_y}{s_x}\frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{s_x s_x}.$$

Now, let's look at the other sum,

$$\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}),$$

and notice that it is the sample covariance of the predictors and responses.

Interpreting the estimated slope III

Recalling our definition of the correlation as

$$\rho_{x,y} = \frac{\mathbb{E}(X - \mathbb{E}X)(Y - \mathbb{E}Y)}{\sqrt{(\mathrm{Var}\,X)(\mathrm{Var}\,Y)}},$$

we notice that

$$\hat{\rho}_{x,y} = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{s_x s_x} = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{s_x^2 s_y^2}}$$

is the *sample correlation* between our predictors and responses– we plugged in the sample versions of the covariance and the variances.

Interpreting the estimated slope IV

So, our estimated coefficient $\hat{\beta}_1$ can be expressed as

$$\hat{\beta}_1 = \underbrace{\frac{s_y}{s_x}}_{\frac{\text{pos}}{\text{pos}}} \underbrace{\hat{\rho}_{x,y}}_{[-1, +1]}.$$

In other words, the slope of our model is the ratio of the standard deviations, scaled by the correlation between our predictors and responses.

Interpreting the estimated slope V

An interesting case to think about is when the predictors and responses are perfectly correlated (i.e., the predictors and responses form a perfect line, with no "jitter"). Then our estimated slope is $\hat{\beta}_1 = \sqrt{s_y^2/s_x^2} = s_y/s_x$. In other words, the slope of our model is just the ratio of the standard deviation of the responses to that of the predictors. Think of this as like a "change of units" from predictors to responses. If our predictors are measured in, say, years of education, and our responses are measured in dollars per year, then the ratio of the standard deviations has units

$$\begin{matrix} Y \rightarrow \\ X \rightarrow \end{matrix} \qquad \frac{\text{dollars per year}}{\text{years of education}},$$

and multiplying this by our predictor, which is measured in "years of education", we get

$$\underline{\text{response}} = \underbrace{\frac{\text{dollars per year}}{\cancel{\text{years of education}}}}_{\beta_1} \cdot \underbrace{\cancel{\text{years of education}}}_{X} = \text{dollars per year},$$

which is what we expect.

Interpreting the intercept term $\hat{\beta}_0$ I

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

predict y for $X_i = \bar{X}$

$$= (\bar{y} - \cancel{\hat{\beta}_1 \bar{x}}) + \cancel{\hat{\beta}_1 \bar{x}} = \bar{y}$$

Turning our attention to $\hat{\beta}_0$, we have

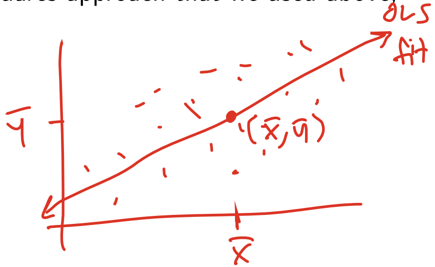
$$\hat{\beta}_0 = \underline{\bar{y} - \hat{\beta}_1 \bar{x}}.$$

Why does this choice make sense?

Well, let's suppose that we decided to make our lives even harder by restricting our choice of prediction function to be a constant. That is, suppose we wanted to choose a prediction function that returns the same output, say, $\hat{y}$, no matter the input x.

If we wanted to choose this output using the same least-squares approach that we used above, we would want to choose $\hat{y}$ so that it minimizes

$$\sum_{i=1}^{n} (y_i - y)^2.$$



Interpreting the intercept term $\hat{\beta}_0$ II

A little bit of calculus (seriously, this one's easy– try it!) shows that the way to minimize this is to choose the output

$$\hat{y} = \frac{1}{n}\sum_{i=1}^{n} y_i = \bar{y}.$$

Now, thankfully, we are not actually trying to predict our data with a constant function. We are allowed to choose a slope!

Having chosen our slope $\hat{\beta}_1$, our model *without* an intercept term predicts that the *i*-th observation should have response $\hat{\beta}_1 x_i$. If we add an intercept term to the model, sticking with our least squares loss, we would like to choose β_0 so as to minimize

$$\sum_{i=1}^{n}\left(y_i - \hat{\beta}_1 x_i - \beta_0\right)^2.$$

The exact same kind of calculus argument (taking a derivative with respect to β_0, this time– again, give it a try!), gets us

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1\bar{x}.$$

Variance of estimates I

After fitting, we can find our predicted $\hat{y}_i$, i.e. the y values on the line.

$$\hat{y}_i = \underline{\hat{\beta}_0} + \underline{\hat{\beta}_1} x_i$$

as well as our model residuals $\hat{\epsilon}_i$

$$\varepsilon_i \sim N(0, \underline{\sigma^2})$$

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$$

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$$

$$\hat{\epsilon}_i = y_i - \hat{y}_i$$

From this, we also get for free an estimate of the variance of the residuals σ^2, which happens to be very useful in computing other statistics. The reason is that **the larger the residuals' variance, the less precisely we can estimate our regression coefficients**, which should make a lot of sense.

$$\underline{\hat{\sigma}^2} = \underline{\text{mean squared error}} = \frac{SSE}{n-2} = \frac{1}{n-2}\sum_i (y_i - \hat{y}_i)^2$$

Variance of estimates II

We can also easily derive the variance of the slope. First, observe that

$$\begin{aligned}
\sum_i (x_i - \bar{x})\bar{y} &= \bar{y} \sum_i (x_i - \bar{x}) && (1)\\
&= \bar{y} \left(\left(\sum_i x_i \right) - n\bar{x} \right) && (2)\\
&= \bar{y} \left(n\bar{x} - n\bar{x} \right) && (3)\\
&= 0 && (4)
\end{aligned}$$

This means that

Variance of estimates III

$$\begin{align}
\sum_i (x_i - \bar{x})(y_i - \bar{y}) &= \sum_i (x_i - \bar{x})y_i - \sum_i (x_i - \bar{x})\bar{y} \tag{5}\\
&= \sum_i (x_i - \bar{x})y_i \tag{6}\\
&= \sum_i (x_i - \bar{x})(\beta_0 + \beta_1 x_i + \epsilon_i) \tag{7}\\
& \tag{8}
\end{align}$$

Variance of estimates IV

Using this, we can easily derive $\text{Var}(\hat{\beta}_1)$ as follows:

$$\text{Var}(\hat{\beta}_1) = \text{Var}\left(\frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sum_i (x_i - \bar{x})^2}\right) \tag{9}$$

$$= \text{Var}\left(\frac{\sum_i (x_i - \bar{x})(\beta_0 + \beta_1 x_i + \epsilon_i)}{\sum_i (x_i - \bar{x})^2}\right), \quad \text{substituting in the above} \tag{10}$$

$$= \text{Var}\left(\frac{\sum_i (x_i - \bar{x})\epsilon_i}{\sum_i (x_i - \bar{x})^2}\right), \quad \text{noting only } \epsilon_i \text{ is a random variable} \tag{11}$$

$$= \frac{\sum_i (x_i - \bar{x})^2 \text{Var}(\epsilon_i)}{\left(\sum_i (x_i - \bar{x})^2\right)^2}, \quad \text{independence of } \epsilon_i \text{ and, } \text{Var}(cX) = c^2\text{Var}(X) \tag{12}$$

$$= \frac{\sigma^2}{\sum_i (x_i - \bar{x})^2} \tag{13}$$

$$\tag{14}$$

$$\text{Var}(\hat{\beta}_1) = \frac{\sigma^2}{(n-1)s_x^2}$$

Running simple linear regression I

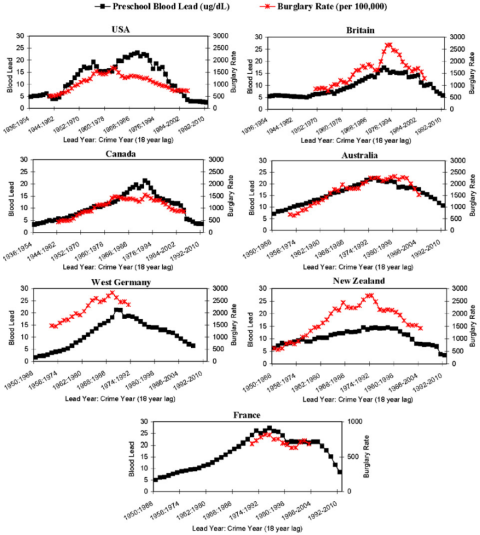
Okay, that's enough abstraction. Let's apply this to some real data and see how things go.

In the 1920s, Thomas Midgley Jr. discovered that adding tetraethyllead to gasoline decreased engine knocking (i.e., when fuel doesn't fully ignite in an engine cylinder, which may damage the engine). He won the 1923 Nichols medal, a prestigious prize in chemistry, for his discovery.

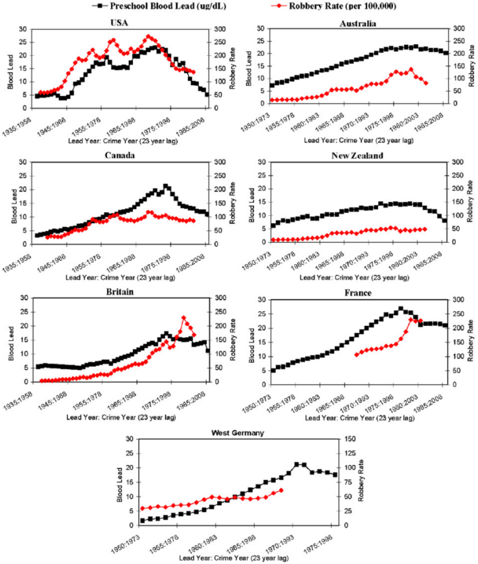
The result of burning tetraethyllead in gasoline resulted in high levels of lead in the atmostphere. By the 1950s to 70s, researchers started to suspect that increased lead levels in the atmosphere was causing widespread lead poisoning, with symptoms ranging from depression, loss of appetite, and amnesia to anemia, insomnia, slurred speech, and cognitive impairment. Starting in the 1980s, the use of tetraethyllead in gasoline started to be phased out. At most gas stations in the United States, you'll notice that gasoline is still marked as being "unleaded", just in case you were worried!

In more recent years, a more controversial theory has emerged, suggesting that exposure to lead (be it in the atmosphere or in paint in older buildings) correlates with incidents of violent crime later in life [1] [2]. This study was first conducted in the US, but it was soon replicated in other countries and the similar results have been found elsewhere in the world.

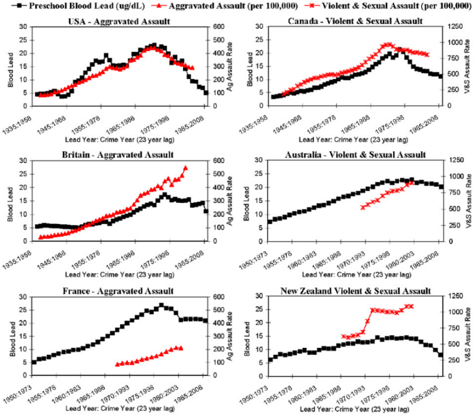
Running simple linear regression II



Running simple linear regression III



Running simple linear regression IV



USA - Aggravated Assault

Canada - Violent & Sexual Assault

Britain - Aggravated Assault

Australia - Violent & Sexual Assault

France - Aggravated Assault

New Zealand Violent & Sexual Assault

Running simple linear regression V

Let's look at a dataset that contains atmospheric lead content levels and aggravated assault rates for several cities in the US and see if we can build a simple linear regression model to explain the trend and make predictions.

```
lead <- read.csv('lead.csv')
# First things first: let's look at the data.
head(lead)
##      city air.pb.metric.tons aggr.assault.per.million
## 1 Atlanta                421                     1029
## 2 Atlanta                429                      937
## 3 Atlanta                444                      887
## 4 Atlanta                457                      533
## 5 Atlanta                461                     1012
## 6 Atlanta                454                      848
```

The variables we are interested in are lead levels in the atmosphere (measured in metric tons of lead emitted) and the aggravated assault rate per million 22 years later.

Running simple linear regression VI

We want to predict the assault rate from lead levels, so our predictor (or explanatory variable or independent variable, if you prefer) is lead levels, and our response (or dependent variable) is the assault rate. This data is available for a number of cities:

```
levels( as.factor(lead$city) )
## [1] "Atlanta"      "Chicago"      "Indianapolis" "Minneapolis"  "New Orleans"
## [6] "~~San Diego~~"
```

Running simple linear regression VII

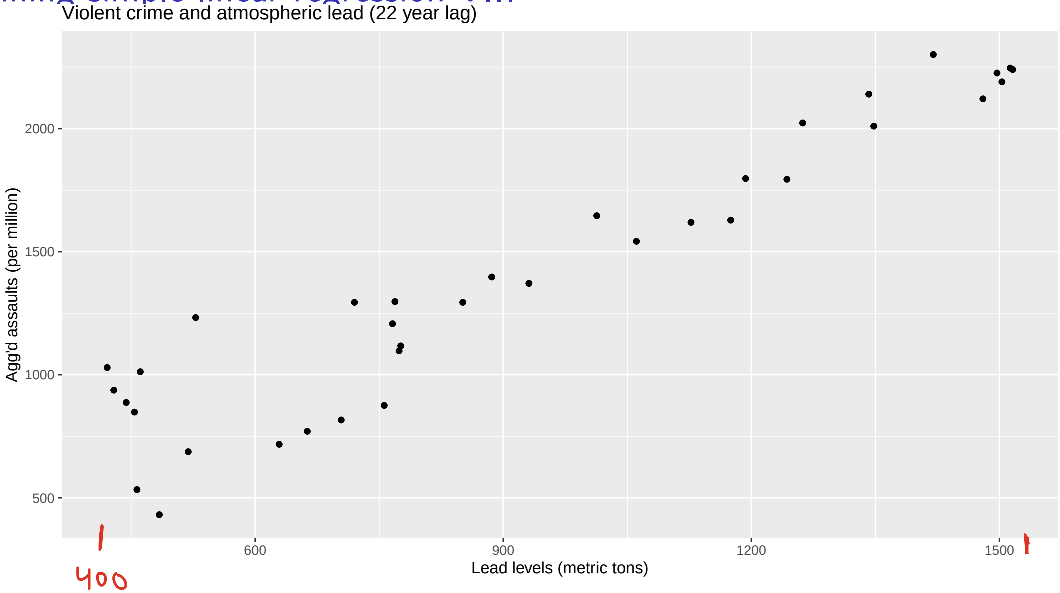
For simplicity, let's focus on the city of Atlanta.

or subset(lead, city=="Atlanta")

```
atlanta_lead <- lead[ lead$city=='Atlanta', ];
## Alternative approach, using filter in dplyr:
# library(dplyr)
# atlanta_lead  lead %>% filter(city == "Atlanta")

# Plot the data to get a look at what's going on.
library(ggplot2)
pp <- ggplot( atlanta_lead,
          aes(x=air.pb.metric.tons, y=aggr.assault.per.million));
pp <- pp + geom_point();
pp <- pp + labs( x="Lead levels (metric tons)",
        y="Agg'd assaults (per million)",
       title="Violent crime and atmospheric lead (22 year lag)" )
```

Running simple linear regression VIII



Running simple linear regression IX

Visually, it's quite clear that assaults are well-predicted as a linear function of lead levels. One thing that is already perhaps of concern is that the data appears to be a bit more "spread out" in the vertical direction for lower lead levels. This is a bit concerning in light of our assumption that the error terms were all distributed according to a normal with mean 0 and variance σ^2. We'll come back to this point below. For now, let's press on.

To fit a linear model in R, we use the `lm()` function (`lm` for "linear model"). The syntax is as simple as `lm(y ~ 1 + x, data=dframe)`, where `dframe` is the data frame containing our data, and `y ~ 1 + x` means to regress the variable `y` (i.e., the column `y` in the dataframe `dframe`) against the variable `x` and an intercept term (that's the `1` in the model formula):

$$y = \beta_1 x + \underbrace{\beta_0}_{\text{intercept}}$$

Note that the `1` in the model formula `y ~ 1 + x` is completely optional– R will include an intercept term automatically.

Running simple linear regression X

The function `lm` returns an object of the class `lm`. This is an object that contains a bunch of information about our fitted model. We'll see some of that information below.

So in our case, we want to regress `aggr.assault.per.million` against `air.pb.metric.tons`. That is, we want a model like

$$\underline{\text{agg.assault}} = \underline{\beta_0} + \underline{\beta_1} \cdot \underline{\text{air.pb}}$$

Running simple linear regression XI

So we'll write that as `aggr.assault.per.million ~ 1+ air.pb.metric.tons`. Let's try fitting the model and then we'll ask R to summarize our model. Running `summary()` on the model object gives us a variety of useful summary statistics and other information about the fitted model.

```
atlanta_lead_lm <- lm(aggr.assault.per.million ~ 1 + air.pb.metric.tons,
                      data=atlanta_lead)
```

specify data   no need to use "$" notation

e.g. lead$air.pb.metric.tons

Running simple linear regression XII

```
summary(atlanta_lead_lm)
##
## Call:
## lm(formula = aggr.assault.per.million ~ 1 + air.pb.metric.tons,
##     data = atlanta_lead)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -356.36  -84.55    6.89  122.93  382.88
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         107.94276   80.46409   1.342    0.189
## air.pb.metric.tons    1.40375    0.08112  17.305   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 180.6 on 34 degrees of freedom
## Multiple R-squared:  0.898,  Adjusted R-squared:  0.895
## F-statistic: 299.4 on 1 and 34 DF,  p-value: < 2.2e-16
```

note: mean is omitted

stars ↓

Running simple linear regression XIII

For now, let's notice in particular that we have estimated coefficients, accessible in the `coefficients` attribute of our model object:

```
atlanta_lead_lm$coefficients
##       (Intercept) air.pb.metric.tons
##        107.942757           1.403746
```

So our model predicts that in Atlanta, an increase of one metric ton of lead in the atmosphere is associated with an average increase of about 1.4 aggravated assaults per million people. Similarly, the intercept indicates that our model predicts that in the absence of any lead in the atmosphere, there would be about 108 aggravated assaults per million people.

[1] https://ir.lawnet.fordham.edu/ulj/vol20/iss3/1

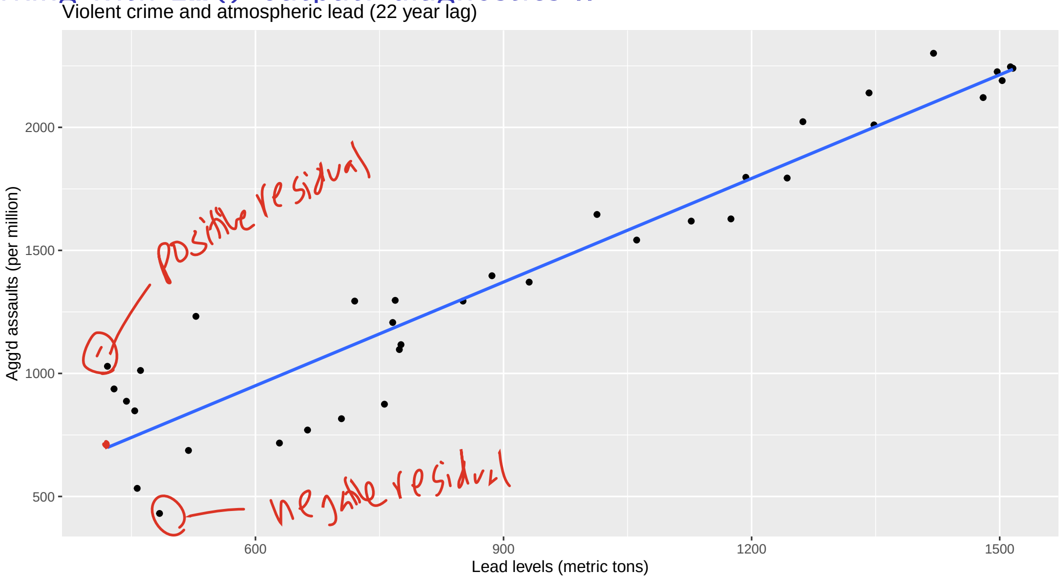
[2] https://doi.org/10.1016/j.envres.2007.02.008

Working with `lm()` output: diagnostics I

Let's look at how our fitted model tracks the data by overlaying the fitted line on our scatterplot above. One way to do this would be to use `abline` with the entries of `atlanta_lead_lm$coefficients` to specify the slope and intercept, but `lm` is so common that `ggplot2` has this same basic functionality built-in, in the form of `geom_smooth()`.

```r
pp <- ggplot( atlanta_lead,
              aes(x=air.pb.metric.tons,y=aggr.assault.per.million));
# The argument `se` specifies whether or not to include a
# confidence interval around the plotted line.
# We'll talk about that later.
# For now we'll just suppress the CI with se=FALSE
pp <- pp +geom_point() + geom_smooth(method="lm",
                                     formula="y~x",
                                     se=FALSE);
pp <- pp + labs( x="Lead levels (metric tons)",
            y="Agg'd assaults (per million)",
            title="Violent crime and atmospheric lead (22 year lag)" )
pp
```

Working with `lm()` output: diagnostics II



Working with lm() output: diagnostics III

Looks like a pretty good fit! Let's look at some of the other information included in the output of lm().

```
# gets fitted y-values (i.e., points on line of best fit)
fitted(atlanta_lead_lm)
##         1         2         3         4         5         6         7         8
##  698.9200  710.1499  731.2061  749.4548  755.0698  745.2436  787.3560  836.4871
##         9        10        11        12        13        14        15        16
##  849.1208  990.8992 1038.6266 1096.1802 1118.6401 1169.1750 1183.2125 1187.4237
##        17        18        19        20        21        22        23        24
## 1194.4424 1197.2499 1302.5309 1351.6620 1414.8306 1529.9378 1597.3176 1689.9649
##        25        26        27        28        29        30        31        32
## 1757.3447 1782.6122 1852.7995 1879.4707 1991.7704 2000.1929 2101.2626 2185.4874
##        33        34        35        36
## 2209.3511 2231.8110 2217.7735 2236.0222
```

Working with `lm()` output: diagnostics IV

```
# residuals( model ) gets residuals (the difference between the
# observed response y and the response predicted by our model)
residuals(atlanta_lead_lm)  # We can also use resid()
##           1           2           3           4           5           6
##  330.080025  226.850054  155.793859 -216.454844  256.930171  102.756395
##           7           8           9          10          11          12
## -356.355996 -149.487118  382.879164 -273.899218 -268.626594 -280.180195
##          13          14          15          16          17          18
##  175.359863 -294.175006   23.787530  109.576291  -97.442440  -80.249933
##          19          20          21          22          23          24
##   -8.530910   45.337967  -43.830619  116.062179  -55.317646  -70.964906
##          25          26          27          28          29          30
## -129.344731   14.387834  -58.799484  143.529335  148.229626    9.807148
##          31          32          33          34          35          36
##  199.737410  -64.487372   16.648940   14.188998  -27.773538    3.977759
```

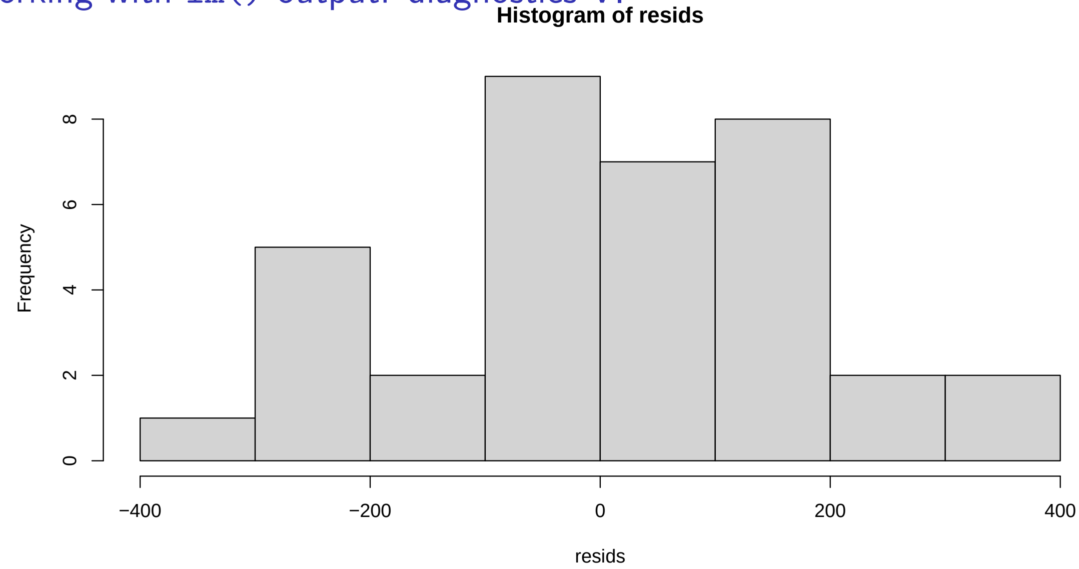
Working with `lm()` output: diagnostics V

These residuals reflect the error between our model's prediction and the true observations, and they are often quite informative.

Recall that our model assumes that the observation errors ϵ_i are normally distributed about zero, with a shared variance σ^2. To check that this assumption is (approximately) true, we can plot the residuals:

```r
resids <- residuals(atlanta_lead_lm)
hist(resids)
```

Working with `lm()` output: diagnostics VI



Working with `lm()` output: diagnostics VII

That looks... okay, at any rate. The residuals are (approximately) symmetric about zero, and the histogram looks normal-ish to me. We'll come back to this point, and later in your studies (e.g., if you take our department's regression course) you'll learn lots of ways for assessing model fit (e.g., checking if the normal errors assumption is correct), but for the time being, we'll be satisfied with the "ocular inspection" method.

Homoscedasticity I

Another important point, far more important that the normality assumption, is that the variance of the errors ϵ_i does not depend on the predictor X_i. This is referred to as "homogeneity of variance", more commonly called homoscedasticity. Its absence, *heteroscedasticity*, wherein the variance of the error terms varies with X_i, can be a big problem for linear regression.

So let's check for it, just visually for now. We want to plot the residuals as a function of their predictor values. If our errors are homoscedastic, we should observe the variance of the residuals about the horizontal line $y = 0$ to be more or less constant along the x-axis. R will do this for us automatically if we call `plot` on our model object. In fact, R will make several plots for us automatically, and return those plots in a list-like object. The residuals as a function of the x values is the first of these, and we can access it with the `which` keyword to `plot()`.

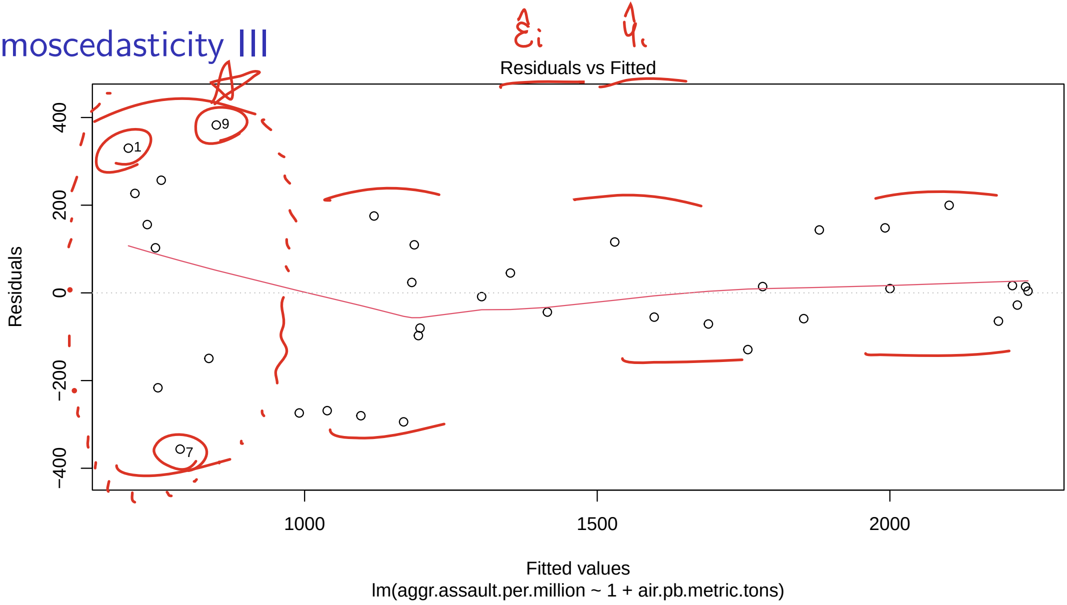
$\varepsilon_i \sim N(0, \sigma^2)$

Variance should not depend on X value

Homoscedasticity II

```
plot(atlanta_lead_lm, which=1)
```

Homoscedasticity III



Homoscedasticity IV

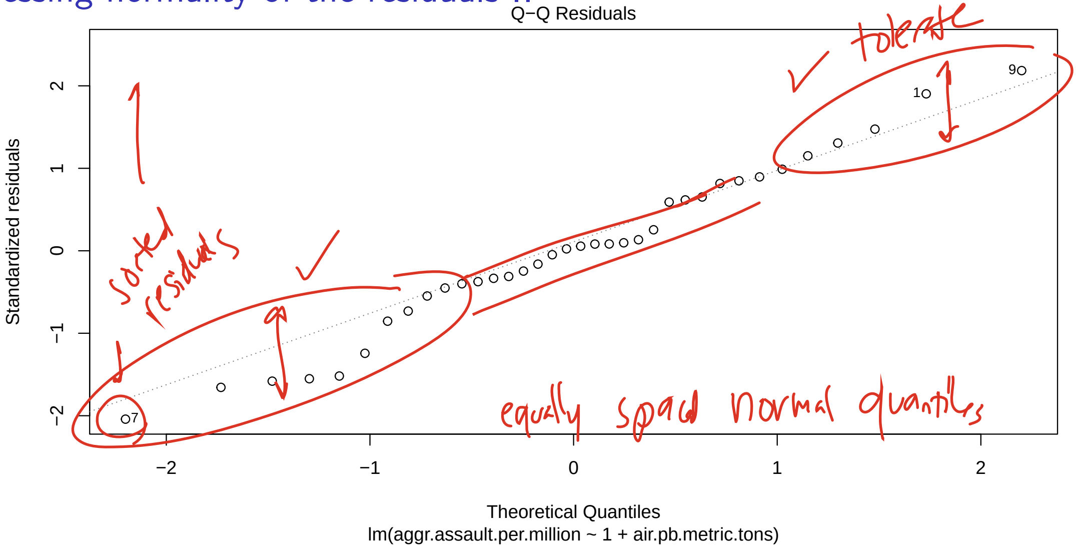
The red line is fitted by R; if our residuals are reasonably well-behaved, this line should be horizontal. Inspecting this plot, it looks as we suspected– the residuals for smaller lead atmospheric levels have slightly higher variance, and tend to be biased toward positive values. Still (and this is an intuition that you'll develop as you perform more analyses), this doesn't look especially extreme.

Assessing normality of the residuals I

The stronger assumption, not required by linear regression per se, but a good assumption to check for use in downstream testing procedures (we'll talk about that soon!), is that the errors are normal with mean zero. We saw that their histogram above looked pretty reasonable. A better check for fit is to construct a Q-Q plot (still no relation to the Chinese restaurant on University Ave, sadly).

```r
plot(atlanta_lead_lm, which=2)
```

Assessing normality of the residuals II



tolerate

sorted residuals

equally spaced normal quantiles

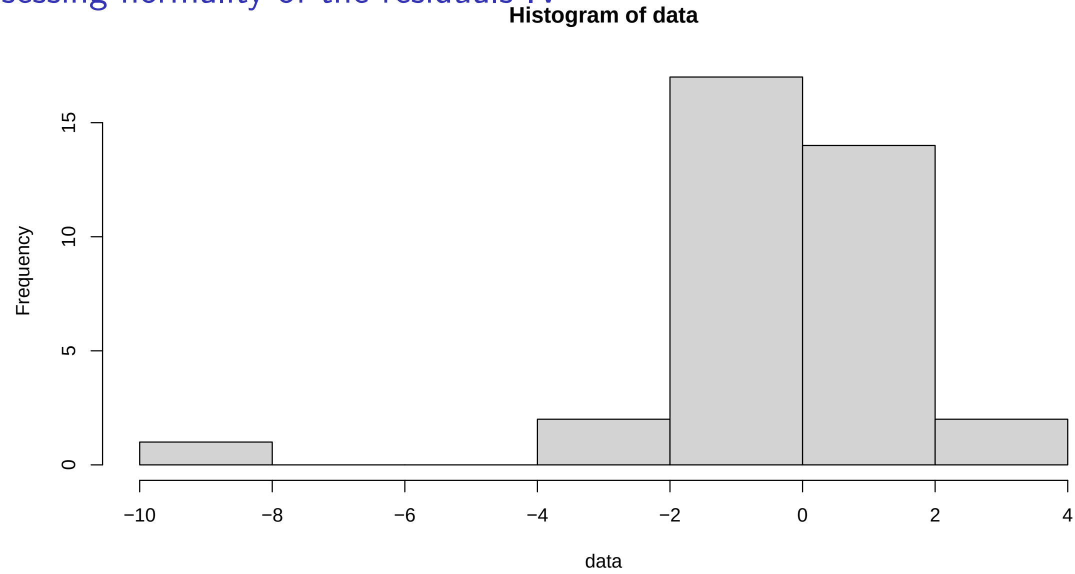
Assessing normality of the residuals III

Let's recall that a Q-Q-plot displays the quantiles (i.e., the percentiles) of our observed data against the quantiles of the normal distribution. If our data were perfectly normally distributed, then the Q-Q plot would look like a straight line with slope 1 (up to randomness in the data, of course). If our data is not normal, the Q-Q plot will look far different.

Just to see an example of this, let's generate some data from a t-distribution (which *looks* normal, but has "heavier tails"), and look at the Q-Q plot.

```r
data <- rt(n=36, df=3, ncp=0);
hist(data)
```

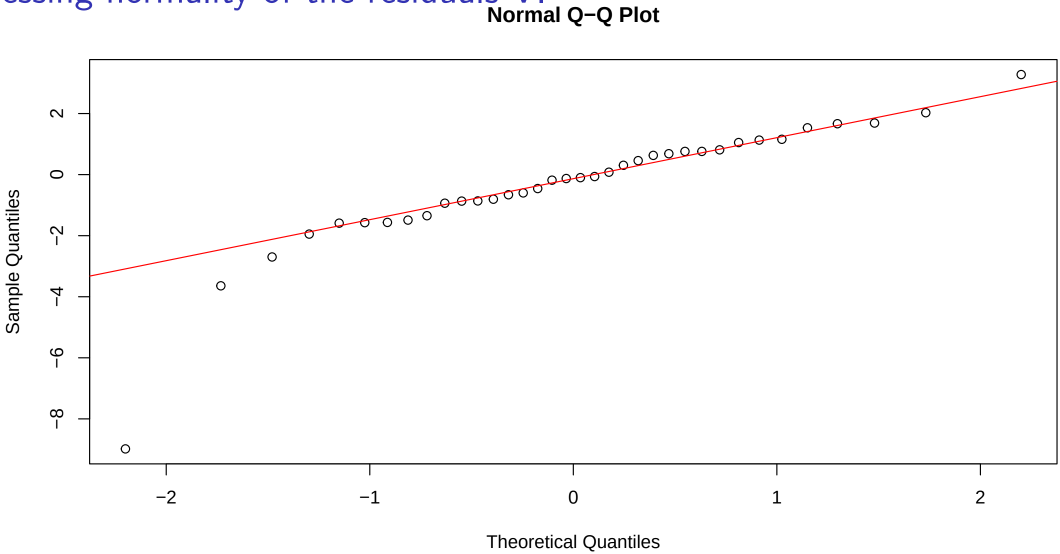
Assessing normality of the residuals IV



Assessing normality of the residuals V

```r
qqnorm(data);
# Add a line to the plot to indicate the behavior we would
# expect to see if the data were normal.
qqline(data, col='red');
```

Assessing normality of the residuals VI

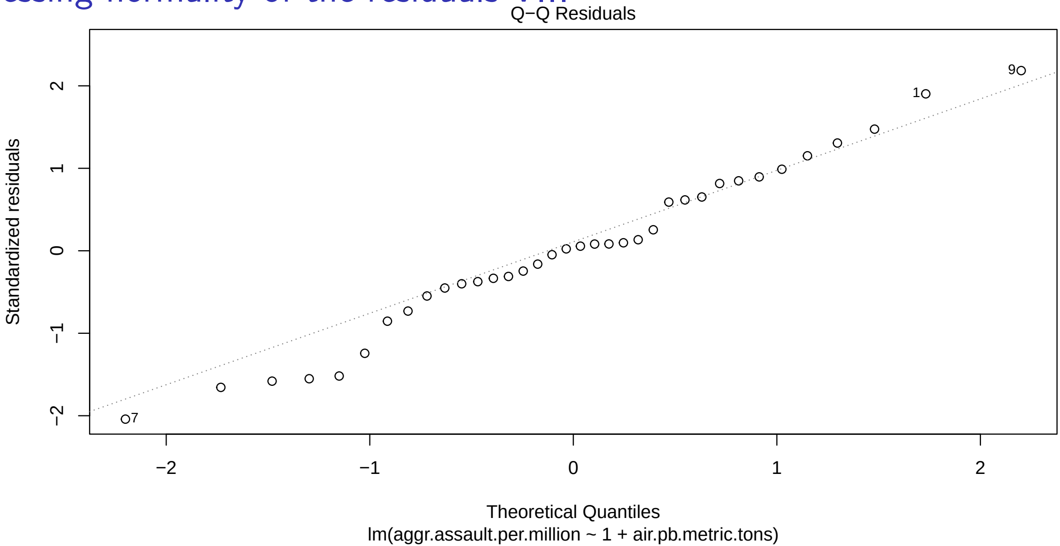


Assessing normality of the residuals VII

So this is an example of the kind of behavior we would expect to see if our data were *not* well-described by a normal. Here's our lead level data again.

```r
# We could also call qqnorm(atlanta_lead_lm$residuals)
plot(atlanta_lead_lm, which=2)
```

Assessing normality of the residuals VIII



Assessing normality of the residuals IX

There are some slightly concerning spots there, especially in the bottom-left, but it’s not too extreme (in my opinion, anyway). Once again, later on you’ll learn more rigorous ways of checking model assumptions like this. We’re just trying to get an intuition for now.

Testing and confidence intervals for coefficients I

So we've established that our model is a reasonably good fit for the lead data, at least in the sense that the trend in the data follows our plotted line and such.

Can we conclude from this that the association between lead and aggravated assault rate is "real"? It's possible, after all, that the observed association is merely due to chance.

Well, in our discussions of hypothesis testing we saw a number of tools for checking if observations were merely due to chance or not. Linear regression has its own set of tools for checking whether observed coefficient estimates are "merely due to chance".

Let's look back at our model summary and let's pay particular attention to the "coefficients" part of the output.

Testing and confidence intervals for coefficients II

```
summary(atlanta_lead_lm)
##
## Call:
## lm(formula = aggr.assault.per.million ~ 1 + air.pb.metric.tons,
##     data = atlanta_lead)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -356.36  -84.55    6.89  122.93  382.88
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         107.94276   80.46409   1.342    0.189
## air.pb.metric.tons    1.40375    0.08112  17.305   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 180.6 on 34 degrees of freedom
## Multiple R-squared:  0.898,  Adjusted R-squared:  0.895
## F-statistic: 299.4 on 1 and 34 DF,  p-value: < 2.2e-16
```

$\hat{\beta}_0$ is a r.v.

Testing and confidence intervals for coefficients III

The `Coefficients` table includes our estimates for the coefficients, standard errors for those estimates, t-values (i.e., a test statistic) for those statistics and, at the far right of the table, a column headed `Pr[>|t|]`. Hey... that's a p-value!

Notice that in this case, the p-value associated to our estimate of the coefficient of lead levels is quite small. That indicates that an estimate this extreme (or more extreme) is highly unlikely to have arisen entirely by chance.

Now, it is in dealing with these p-values that we need to be a bit careful about our model assumptions. If the assumption of normal, homoscedastic errors is violated, things can go wrong with these p-values. But since our Q-Q plot indicated that our residuals were reasonably normal-looking, we can be somewhat confident that this is reflecting a real effect (well, and it's pretty clear just from the plot that there's a linear relationship...).

This p-value arises, in essence, from a *t*-test. This t-test is designed to test the null hypothesis

$$t = \frac{\hat{\beta}_1 - \beta_1^{0 \leftarrow \text{null}}}{se(\hat{\beta}_1)} \qquad ☆ \boxed{H_0 : \beta_1 = 0.} \qquad H_A : \beta_1 \neq 0$$

ie $\frac{\hat{\beta}_1}{se(\hat{\beta}_1)}$

Testing and confidence intervals for coefficients IV

In this case, our p-value associated with β_1 is quite small, indicating a correlation between lead levels and aggravated assault levels. It does **not** imply causation, though, as you well know by now. Nonetheless, this seems to be fairly convincing evidence that there is an association.

Thinking back to our brief discussion of the connection between confidence intervals and testing, you won't be surprised to learn that we can also compute confidence intervals for the true value of the coefficients.

```
confint(atlanta_lead_lm, level=0.95)
##                          2.5 %     97.5 %
## (Intercept)         -55.579958 271.465472
## air.pb.metric.tons    1.238891   1.568602
```

$1-\alpha$ CI does not contain 0 $\longleftrightarrow$ p-value $< \alpha$

Of course, we also tested the hypothesis $H_0 : \beta_0 = 0$, and the p-value is not especially small (also reflected in the fact that the confidence interval includes 0). This indicates that our intercept term was not statistically significantly different from zero.

Note, however, that just because our p-value associated to the intercept term isn't especially small, that doesn't mean that the intercept isn't useful for prediction. Let's turn our attention to that matter.

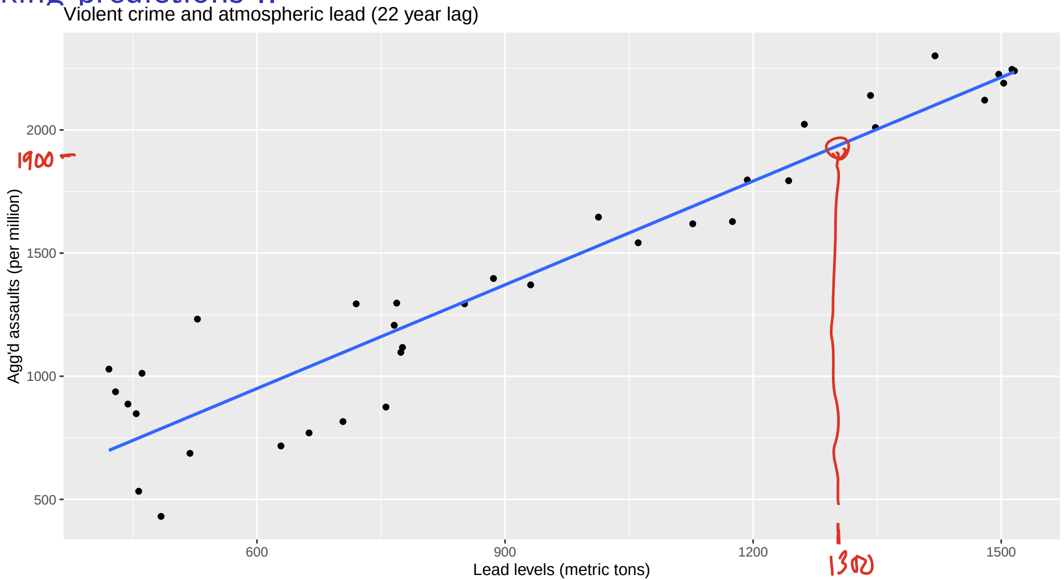
Making predictions I

Suppose that tomorrow a chemical company near Atlanta has an incident, and lead is released into the atmosphere. Suppose that the new atmospheric levels of lead are found to be 1300 metric tons. What would you predict the approximate aggravated assault rate to be 22 years later?

Well, let's start by just looking at a plot of our model again.

```
pp <- ggplot( atlanta_lead,
              aes(x=air.pb.metric.tons,y=aggr.assault.per.million));
# The argument `se` specifies whether or not to include a
# confidence interval around the plotted line.
# We'll talk about that later.
# For now we'll just suppress the CI with se=FALSE
pp <- pp +geom_point() + geom_smooth(method="lm",
                                     formula="y~x",
                                     se=FALSE);
pp <- pp + labs( x="Lead levels (metric tons)",
          y="Agg'd assaults (per million)",
          title="Violent crime and atmospheric lead (22 year lag)" )
```

Making predictions II



Making predictions III

Just looking at the plot, we see that at x-value 1300, our fitted line is just about exactly at 2000 aggravated assaults per million people.

Hopefully you can imagine how annoying it would be to perform this exercise by hand every time we need a new prediction. Luckily, R model objects (including linear regression) support a function called `predict`, which does exactly what it sounds like. We pass our model, and some data (i.e., x-values), and `predict()` outputs our model's predicted responses at those values.

```
predict(atlanta_lead_lm, newdata=data.frame(air.pb.metric.tons=1300))
##        1
## 1932.813
```

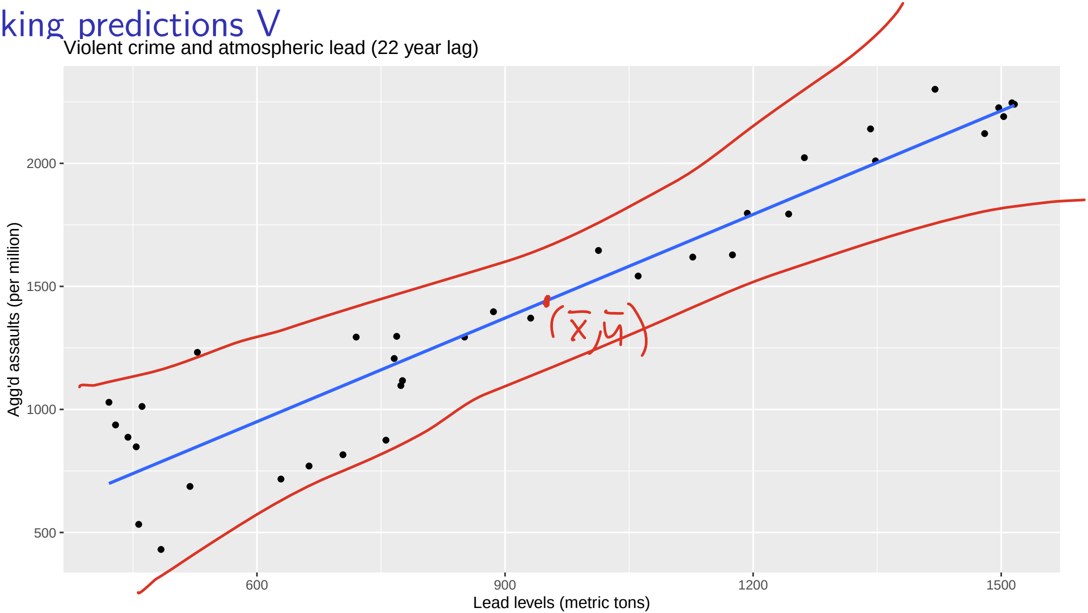
Suppose the company continues to release more lead into the atmosphere, and next year, the levels are measured to be 2000 metric tons. Can we use our model to predict what aggravated assault rates might look like 22 years later?

Making predictions IV

Well, looking at the plot again, 2000 metric tons is rather far outside the range of our observed predictor values.

```r
pp <- ggplot( atlanta_lead,
              aes(x=air.pb.metric.tons,y=aggr.assault.per.million));
# The argument `se` specifies whether or not to include a
# confidence interval around the plotted line.
# We'll talk about that later.
# For now we'll just suppress the CI with se=FALSE
pp <- pp +geom_point() + geom_smooth(method="lm",
                                     formula="y~x",
                                     se=FALSE);
pp <- pp + labs( x="Lead levels (metric tons)",
          y="Agg'd assaults (per million)",
          title="Violent crime and atmospheric lead (22 year lag)" )
```

Making predictions V



Making predictions VI

As we've alluded to earlier in these lecture notes, predictions made far outside the range of observed predictors have to be treated carefully They may be reliable, but they also may not. The reliability of a prediction usually decreases the further away it is from your data.

For example, perhaps once lead levels reach a certain point, there just isn't much more damage they can do to human development. Then we would see the linear trend flatten out at higher lead levels. Our function would cease to be linear, and naively applying our linear models to those values would result in poor prediction performance.

Still, just to show that we can do it, let's see what our model predicts.

```r
predict(atlanta_lead_lm, newdata=data.frame(air.pb.metric.tons=2000))
##        1
## 2915.435
```

Multiple regression I

Everything so far is likely (mostly) familiar to you from STAT240: regressing a single variable against a single other variable. What happens, however, when we want to incorporate multiple different predictors in our model?

For example, to improve the descriptive power of or model for predicting salary at age 30 based on years of education, we might want to add additional information (e.g., demographic information, parents' level of education, etc).

How do we go about adding more variables to our model?

Well, it's basically as simple as you would imagine. We just add more predictors (and more coefficients) to our linear function. If we have p predictors plus an intercept, we predict the response y according to

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_{p-1} x_{p-1} + \beta_p x_p,$$

where $x_1, x_2, \ldots, x_p \in \mathbb{R}$ are predictors.

Multiple regression II

Similarly, our model now takes the form that for each $i = 1, 2, \ldots, n$, we observe

$$Y_i = \beta_0 + \beta_1 X_{i,1} + \beta_2 X_{i,2} + \cdots + \beta_{p-1} X_{i,p-1} + \beta_p X i, p + \epsilon_i,$$

where ϵ_i is an error (again, assumed normally distributed, independent over i, etc) and $X_i = (X_{i,1}, X_{i,2}, X_{i,3}, \ldots, X_{i,p})^T \in \mathbb{R}^p$ is a vector of predictors. If you haven't taken linear algebra or you just don't like vectors, not to worry– it's perfectly safe to think of this as just a list of numbers in this class.

Note: some resources will use this notation "backward", instead writing $(X_{1,i}, X_{2,i}, \ldots, X_{p-1,i}, X_{p,i})$ for the predictors. The distinction doesn't matter much, so long as you're consistent.

Example: the mtcars dataset I

Let's recall the mtcars dataset, which includes a number of variables describing the specifications and performance of a collection of car brands.

```
head(mtcars)
##                    mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

Let's suppose that we are interested in predicting the quarter mile time (qsec, the time it takes the car to go 1/4 mile from a dead stop) based on its engine displacement (disp, measured in cubic inches), horsepower (hp, measured in... horsepower) and weight (wt, measured in 1000s of pounds).

Example: the `mtcars` dataset II

That is, we want to build a multiple linear regression model of the form

$$\mathsf{qsec} = \beta_0 + \beta_1 \mathsf{disp} + \beta_2 \mathsf{hp} + \beta_3 \mathsf{wt} + \epsilon$$

To fit such a model in R, the syntax is quite similar to the simple linear regression case. The only thing that changes is that we need to specify this model in R's notation. We do that as `qsec ~ 1 + disp + hp + wt`.

Example: the `mtcars` dataset III

Let's fit the model in R and see what happens.

Example: the mtcars dataset IV

```
mtc_model <- lm( qsec ~ 1 + disp + hp + wt, data=mtcars);
summary(mtc_model)
##
## Call:
## lm(formula = qsec ~ 1 + disp + hp + wt, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.8121 -0.3125 -0.0245  0.3544  3.3693
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 17.965050   0.849663  21.144  < 2e-16 ***
## disp        -0.006622   0.004166  -1.590  0.12317
## hp          -0.022953   0.004603  -4.986 2.88e-05 ***
## wt           1.485283   0.429172   3.461  0.00175 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.062 on 28 degrees of freedom
## Multiple R-squared:  0.6808, Adjusted R-squared:  0.6466
## F-statistic: 19.91 on 3 and 28 DF,  p-value: 4.134e-07
```

Assessing model fit I

Once we've fit a model to the data, how do we tell if our model is good or not? We started talking about this above, and it is a trickier question that it might seem at first. We'll have lots more to say about the problem in coming weeks. For now, though, let's consider the most obvious answer to this question.

We fit our model to data by minimizing the sum of squares (we're sticking with simple linear regression here for simplicity– this idea extends to multiple linear regression in the obvious way),

$$\ell(\beta_0, \beta_1) = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} \left(y_i - (\beta_0 + \beta_1 x_i)\right)^2 .$$

So what better way to measure how good our model is than using precisely this quantity?

Assessing model fit II

We define the residual sum of squares (RSS; also called the sum of squared errors, SSE) to be the sum of squared errors of our model. That is, letting $\hat{\beta}_0$ and $\hat{\beta}_1$ be our estimates of the coefficients,

$$\mathrm{RSS} = \mathrm{SSE} = \sum_{i=1}^{n} \epsilon_i^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} \left(y_i - (\hat{beta}_0 + \hat{\beta}_1 x_i) \right)^2$$

The number of degrees of freedom will have bearing on the distribution of this error term- Under the model where the errors are indeed normally distributed, the residual sum of squares (RSS) will have an F-distribution with degrees of freedom given by the number of observations minus the number of parameters (like the t-distribution, the F-distribution has the degrees of freedom as one of its parameters; the other, the 3 in the summary above, comes from the number of coefficients less one). Knowing this fact lets us build a rejection region for using the RSS as a test statistic, and that's exactly where the overall p-value at the bottom of the summary comes from.

Assessing model fit III

```
##
## Call:
## lm(formula = qsec ~ 1 + disp + hp + wt, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.8121 -0.3125 -0.0245  0.3544  3.3693
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 17.965050   0.849663  21.144  < 2e-16 ***
## disp        -0.006622   0.004166  -1.590  0.12317
## hp          -0.022953   0.004603  -4.986 2.88e-05 ***
## wt           1.485283   0.429172   3.461  0.00175 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 1.062 on 28 degrees of freedom
## Multiple R-squared:  0.6808, Adjusted R-squared:  0.6466
## F-statistic: 19.91 on 3 and 28 DF,  p-value: 4.134e-07
```

Assessing model fit IV

Another useful quantity in describing how well our model describes the data is the **Coefficient of Determination**, or R-squared, which can be interpreted as measuring the proportion (between 0 and 1) of the variation in Y that is explained by the variation in X.

$$R^2 = \frac{\text{TSS} - \text{RSS}}{\text{TSS}} = 1 - \frac{\text{RSS}}{\text{TSS}},$$

where

$$\text{TSS} = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

is the **total sum of squares**.

Assessing model fit V

In the case of simple linear regression, things simplify so that $R^2 = r^2$, where r is the correlation coefficient between X and Y,

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

When this quantity is close to 1, we can be confident that our linear model is accurately capturing a trend in the data.

Looking ahead: model selection

One important point that we've ignored in our discussion above is how we go about choosing what predictors to include in our model. For example, the `mtcars` data set has columns

```
names(mtcars)
##  [1] "mpg"  "cyl"  "disp" "hp"   "drat" "wt"   "qsec" "vs"   "am"   "gear"
## [11] "carb"
```

In our example above, we just chose a few of these to use as predictors. But suppose that we didn't know ahead of time which predictors to use. How do we choose which ones to include in our model? Are there downsides to just including all of them?

Review I

- The simple linear regression model
- predictor / response variables
- assumptions of SLR model
- interpretation of SLR components
- correlation is not causation
- residuals
- loss function of OLS: Sum of squared residuals

Review II

- slope estimate formula and interpretation
- intercept estimate formula and interpretation
- units of β estimates
- mean squared error, estimate of σ_ϵ^2
- variance of β estimates
- OLS in R, and summary output
- OLS diagnostics (residual plot, residual QQ plot)
- inference and confidence intervals for coefficients
- predictions & prediction intervals
- R^2 and r