Chapter 1 (AEPV)

DJM, Revised: NAK

22 January 2019

The Linear Model

In a purely vector/matrix form, the linear model can be written as:

$$\underline{y} = \underline{X}\underline{\beta} + \underline{\epsilon}$$

- $\underline{y} = [y_i]$ is an $n \times 1$ vector of the observed responses.
- $\underline{X} = [x_{ij}]$ is the $n \times p$ 'design matrix'. Column $j = 1, \ldots, p$ is the observed values of the j^{th} 'predictor' variable. Each row is the the set of observed values of all p 'predictor' variables.
- $\underline{\beta} = [\beta_j]$ is the $p \times 1$ vector of the p parameters or coefficients associated with each predictor variable.
- $\underline{\epsilon}$ is the $n \times 1$ error vector.

If we write $\underline{x}_i^\top$ as the i^{th} row of $\underline{X}$, we can look at the model in terms of individual y observations.

$$y_i = \underline{x}_i^\top \underline{\beta} + \epsilon_i$$

1. What are all of these things?
2. What is the mean of y_i?
3. What is the distribution of ϵ_i?

Simulating The Model

$$y_i = \underline{x}_i^\top \underline{\beta} + \epsilon_i$$

We can break down the model in to two components:

- a deterministic component
- a random component

This gives us the form for how we could picture the data produced by the system we try to model.

```
n=50

# Need x values
x <- rnorm(n, 33, 5) #n, mu, sigma

#need a value for coefficients.
beta <- c(300, -5) # don't forget the y-intercept.

# Create the design matrix
X <- cbind(1, x) # Pastes 'columns' side-by-side together. Why the '1'?

# Deterministic portion
mu_y <- X%*%beta

# Going to create side-by-side plots
```

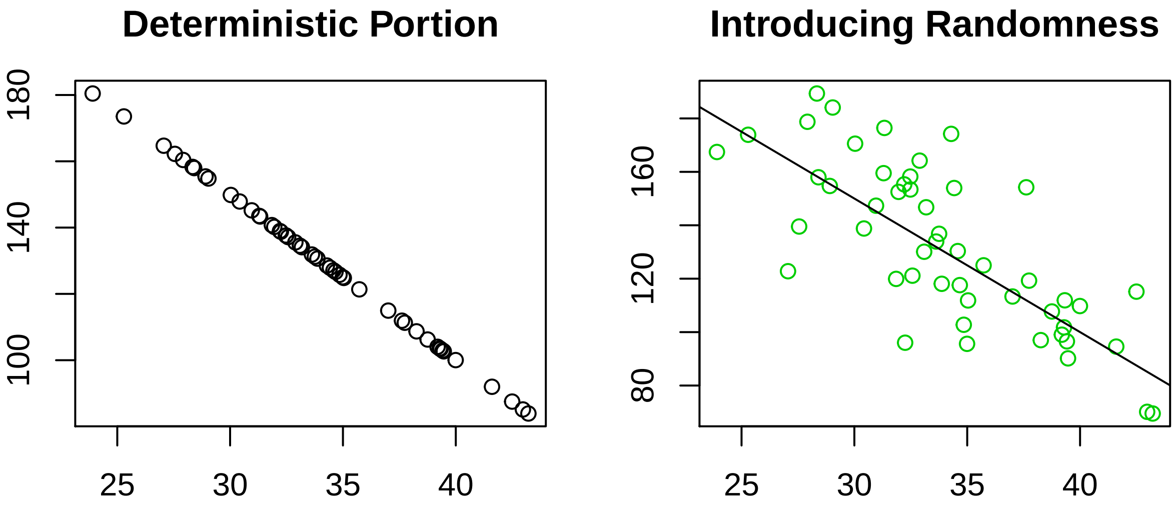
```r
opar <- par() # save current R settings
par(mfrow = c(1,2), mar = c(3,3,3,1)) # Change plot() grid, and margins

plot(x, mu_y, main = "Deterministic Portion")

# Introducing error
err <- rnorm(length(x), 0, 20)

y <- mu_y + err

plot(x, y, main = "Introducing Randomness", col = 3)
abline(coef = beta)
```



```r
dev.off() # This turns off the graphic device and resets par()
```

```
## null device
##           1
```

What needs to be changed here if we want to simulate a multiple regression situation?

Will this work?

```r
# Need SOMETHING for X
X <- cbind(1, rnorm(n, 37, 5), rnorm(n, 82, 20), rnorm(n, 2.52, .5))

# Parameter vector
beta <- c(-10, 1.5, -3, 5.2)
```

How do we estimate $\underline{\beta}$?

1. Ordinary least squares (OLS).
2. Maximum likelihood.
3. Do something more creative.

Method 1. OLS

Suppose I want to find an estimator $\underline{\widehat{\beta}}$ which makes small errors on my data.

I measure errors with the difference between predictions $\underline{X}\underline{\widehat{\beta}}$ and the responses $\underline{y}$.

I don't care if the differences are positive or negative, so I try to measure the total error with

$$\sum_{i=1}^{n} \left| y_i - \underline{x}_i^\top \underline{\widehat{\beta}} \right| .$$

This is fine, but hard to minimize (what is the derivative of $|\cdot|$?)

So I use

$$\sum_{i=1}^{n} (y_i - \underline{x}_i^\top \underline{\widehat{\beta}})^2 .$$

OLS solution

We write this as

$$\underline{\widehat{\beta}} = \arg\min_{\underline{\beta}} \sum_{i=1}^{n} (y_i - \underline{x}_i^\top \underline{\beta})^2 .$$

"Find the β which minimizes the sum of squared errors."

Note that this is the same as

$$\underline{\widehat{\beta}} = \arg\min_{\underline{\beta}} \frac{1}{n} \sum_{i=1}^{n} (y_i - \underline{x}_i^\top \underline{\beta})^2 .$$

"Find the beta which minimizes the mean squared error."

Optimize = Calculus

We differentiate and set to zero

$$\begin{aligned}
& \frac{\partial}{\partial \underline{\beta}} \frac{1}{n} \sum_{i=1}^{n} (y_i - \underline{x}_i^\top \underline{\beta})^2 \\
&= \frac{2}{n} \sum_{i=1}^{n} \underline{x}_i (y_i - \underline{x}_i^\top \underline{\beta}) \\
&= \frac{2}{n} \sum_{i=1}^{n} -\underline{x}_i \underline{x}_i^\top \underline{\beta} + \underline{x}_i y_i \\
0 &\equiv \sum_{i=1}^{n} -\underline{x}_i \underline{x}_i^\top \underline{\beta} + \underline{x}_i y_i \\
&\Rightarrow \sum_{i=1}^{n} \underline{x}_i \underline{x}_i^\top \underline{\beta} = \sum_{i=1}^{n} \underline{x}_i y_i \\
&\Rightarrow \underline{\beta} = \left(\sum_{i=1}^{n} \underline{x}_i \underline{x}_i^\top \right)^{-1} \sum_{i=1}^{n} \underline{x}_i y_i
\end{aligned}$$

Matrix OLS Solution

Very often, it is said that the OLS solution is:

$$\underline{\widehat{\beta}} = (X^\top X)^{-1} X^\top Y.$$

A more general solution is the following:

$$\underline{\widehat{\beta}} = \underline{X}^- \underline{y} + \left(\underline{I} - \underline{X}^- \underline{X}\right) \underline{h}$$

- Here X^- is the Moore-Penrose Generalized Inverse.
- It is used when all columns of X are not linearly independent.
- This usually arises in 'Effects Model' representations of Analysis of Variance models.

Method 2: Maximum Likelihood Estimation, MLE

Method 1 didn't use anything about the distribution of ϵ.

But if we know that ϵ has a normal distribution, we can write down the joint distribution of $Y = (y_1, \dots, y_n)$:

$$\begin{aligned} f_Y(y; \underline{\beta}) &= \prod_{i=1}^n f_{y_i;\underline{\beta}}(y_i) \\ &= \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(y_i - \underline{x}_i^\top \underline{\beta})^2\right) \\ &= \left(\frac{1}{2\pi\sigma^2}\right)^{n/2} \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^n (y_i - \underline{x}_i^\top \underline{\beta})^2\right) \end{aligned}$$

We initially learn to think of f_Y as a function of y with $\underline{\beta}$ fixed:

1. If we integrate over y from $-\infty$ to ∞, it's 1.
2. If we want the probability of (a, b), we integrate from a to b.
3. etc.

Likelihood Functions

Instead, think of it as a function of $\underline{\beta}$.

We call this "the likelihood" of beta: $\mathcal{L}(\underline{\beta})$.

Given some data, we can evaluate the likelihood for any value of $\underline{\beta}$ (assuming σ is known).

It won't integrate to 1 over $\underline{\beta}$.

But it is "convex", meaning we can maximize it (the second derivative wrt $\underline{\beta}$ is everywhere negative).

Another Round of Optimization: Log Likelihood Functions

The derivative of $L(\underline{\beta})$ tractable but a pain to work with. (Why is it so bad?)

- If we're trying to maximize over $\underline{\beta}$, we can take the log of $L(\underline{\beta})$, and maximize over the log function instead.
- We will get the same solution for $\underline{\beta}$. Why?
- It will be easier too. Again... Why?

$$\mathcal{L}(\underline{\beta}) = \left(\frac{1}{2\pi\sigma^2}\right)^{n/2} \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \underline{x}_i^\top \underline{\beta})^2\right)$$

$$\ell(\underline{\beta}) = -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \underline{x}_i^\top \underline{\beta})^2$$

But we can ignore constants (assume sigma is constant for simplicity), so this gives

$$\underline{\widehat{\beta}} = \arg\max_{\underline{\beta}} -\sum_{i=1}^{n}(y_i - \underline{x}_i^\top \underline{\beta})^2$$

The same as before!

Other Methods

Weighted Least Squares (Make OLS More Complex)

- OLS treats each observation equally.
- Some observations may be more reliable than others for different reasons.
- We can weight each i^{th} observation by a 'weight', w_i.

$$\underline{\widehat{\beta}} = \arg_{\underline{\beta}} \min \sum_{i=1}^{n} w_i \left(y_i - \underline{x}_i^\top \underline{\beta}\right)^2$$

Options Beyond "Least Squares"

In general, we want to minimize some form of the Sum of Squared Error (SSE).

SSE is just one of what are referred to as **Loss Functions**.

Loss functions are functions that measure the cost of a predicted value $\widehat{y}_i$ when it is compared to the observed value y_i.

$L(y_i, \widehat{y}_i) =$

- $(y_i - \widehat{y}_i)^2$
- $|y_i - \widehat{y}_i|$
- $I(y_i \neq \widehat{y}_i)$
- And many more depending on the problem at hand.

What ever the loss function is, we seek to minimize it across the sample.

Mean Square Error (MSE): The L_2 Loss

Forget about the linear model. We can get more general.

Suppose we think that there is **some** function which relates y and x.

Let's call this function g for the moment: $Y = g(X) + \epsilon$

How do we estimate g?

What is g?

Minimizing MSE

Let's try to minimize the **expected** sum of squared errors (MSE)

$$\begin{aligned}
\mathbb{E}\left[(Y-g(X))^2\right] &= \mathbb{E}\left[\mathbb{E}\left[(Y-g(X))^2 \mid X\right]\right] \\
&= \mathbb{E}\left[\mathrm{Var}\left[Y \mid X\right] + \mathbb{E}\left[(Y-g(X)) \mid X\right]^2\right] \\
&= \mathbb{E}\left[\mathrm{Var}\left[Y \mid X\right]\right] + \mathbb{E}\left[\mathbb{E}\left[(Y-g(X)) \mid X\right]^2\right]
\end{aligned}$$

The first part doesn't depend on g, it's constant, and we toss it.

To minimize the rest, take derivatives and set to 0.

$$\begin{aligned}
0 &= \frac{\partial}{\partial g}\mathbb{E}\left[\mathbb{E}\left[(Y-g(X))^2 \mid X\right]\right] \\
&= -\mathbb{E}\left[\mathbb{E}\left[2(Y-g(X) \mid X\right]\right] \\
&\Rightarrow 2\mathbb{E}\left[g(X) \mid X\right] = 2\mathbb{E}\left[Y \mid X\right] \\
&\Rightarrow g(X) = \mathbb{E}\left[Y \mid X\right]
\end{aligned}$$

The regression function

We call this solution:

$$\mu(X) = \mathbb{E}\left[Y \mid X\right]$$

the **regression function**.

If we **assume** that $\mu(x) = \mathbb{E}\left[Y \mid X = x\right] = x^\top \underline{\beta}$, then we get back exactly OLS.

But why should we assume $\mu(x) = x^\top \underline{\beta}$?

Estimating The Regression Function

In mathematics: $\mu(x) = \mathbb{E}\left[Y \mid X = x\right]$.

In words: Regression is really about estimating the mean.

1. If $Y \sim \mathrm{N}(\mu, 1)$, our best guess for a **new** Y is μ.

2. For regression, we let the mean (μ) **depend** on X.

3. Think of $Y \sim \mathrm{N}(\mu(X), 1)$, then conditional on $X = x$, our best guess for a **new** Y is $\mu(x)$ [whatever this function μ is]

Causality

For any two variables Y and X, we can **always** write

$$Y \mid X = \mu(X) + \eta(X)$$

such that $\mathbb{E}\left[\eta(X)\right] = 0$.

- Suppose, $\mu(X) = \mu_0$ (constant in X), are Y and X independent?
- Suppose Y and X are independent, is $\mu(X) = \mu_0$?

Previews of future chapters

Linear smoothers

What is a linear smoother?

1. Suppose I observe $y_1, \ldots, y_n$.
2. A linear smoother is any **prediction function** that's linear in $\underline{y}$.
   - Linear functions of $\underline{y}$ are simply premultiplications by a matrix, i.e. $\underline{\hat{y}} = \underline{W}\underline{y}$ for any matrix $\underline{W}$.
3. Examples:
   - $\overline{y} = \frac{1}{n}\sum y_i = \frac{1}{n}\begin{bmatrix}1 & 1 & \cdots & 1\end{bmatrix}\underline{y}$
   - OLS Regression: $\widehat{y} = \underline{X}\underline{\widehat{\beta}} = X(\underline{X}\top\underline{X})^{-1}\underline{X}\top\underline{y}$
   - You will see many other smoothers in this class

k-nearest neighbors (kNN)

(We will see **smoothers** in more detail in Ch. 4)

1. For kNN, consider a particular pair (Y_i, X_i)
2. Find the k covariates X_j which are closest to X_i
3. Predict Y_i with the average of those X_j's
4. This turns out to be a linear smoother

- How would you specify $\underline{W}$?

Kernels

(Again, more info in Ch. 4)

Kernel Regression is a linear smoothing technique that is similar in nature to kNN.

First, the definition of a "kernel" function, $K(u)$.

1. $K(u) \geq 0$
2. $\int uK(u)\,du = 0$
3. $0 < \int u^2 K(u)\,du < \infty$

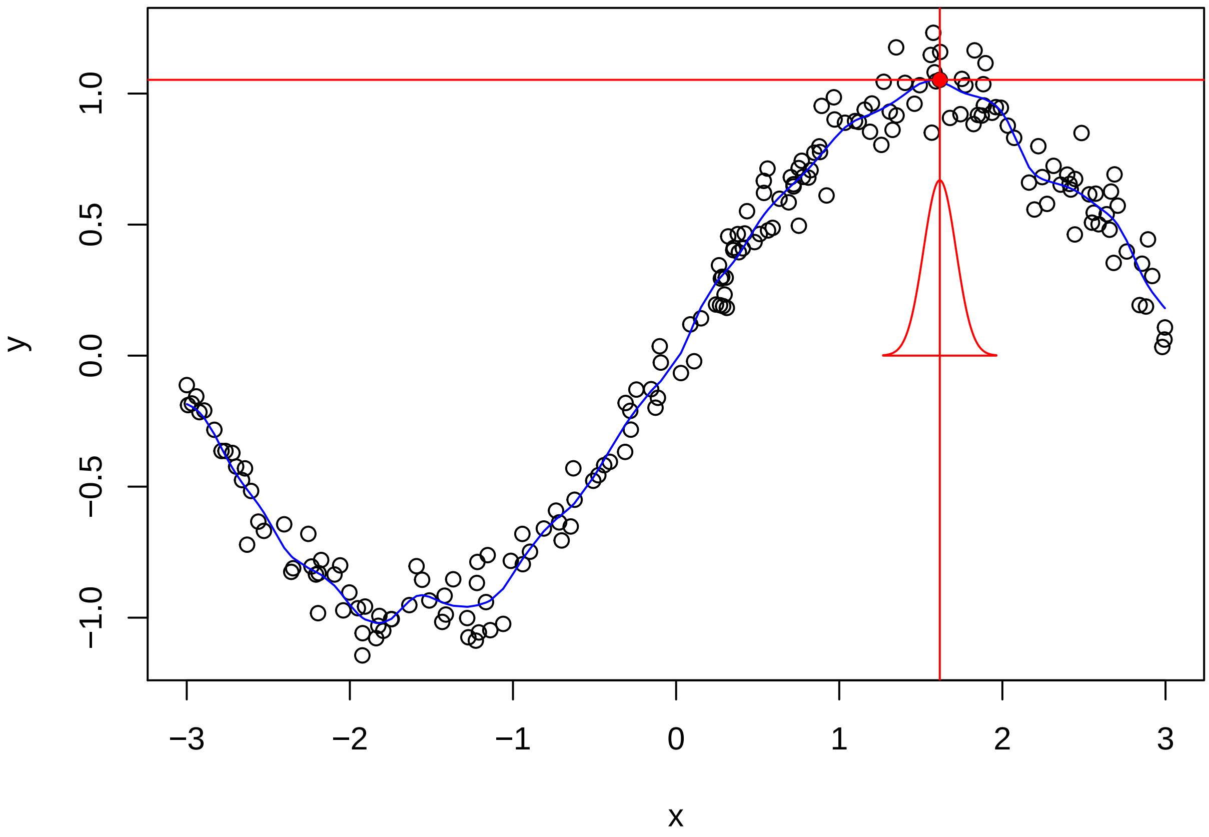
Usually, it's a density function with a finite variance and mean of 0.

To predict at a point x we look at an average the of the y_i centered around x, like kNN.

However, we do not restrict ourselves to estimating the mean with a limited number of observations.

We use a weighted average where the weight of a y_i is determined by its horizontal distance from the point x via the kernel K.

$$\frac{1}{\sum_{i=1}^{n} K(\frac{x-x_i}{h})} \sum_{i=1}^{n} K(\frac{x-x_i}{h}) y_i$$


1.0
0.5
0.0
−0.5
−1.0
y
−3
−2
−1
0
1
2
3
x