Lab 4

Adriana Sham

11:59PM March 9, 2019

Note: the content of this lab is on the midterm exam (March 5) even though the lab itself is due after the midterm exam.

We now move on to simple linear modeling using the ordinary least squares algorithm.

Let's quickly recreate the sample data set from practice lecture 7:

```
n = 20
x = runif(n)
beta_0 = 3
beta_1 = -2
y = beta_0 + beta_1 * x + rnorm(n, mean = 0, sd = 0.33)
```

Solve for the least squares line by computing b_0 and b_1 *without* using the functions `mean`, `cor`, `cov`, `var`, `sd` but instead computing it from the x and y quantities manually using base function such as `sum` and other basic operators. See the class notes.

```
b_1 = (n * sum(x * y) - (sum (x) * sum (y) )) / ((n*sum(x^2))-(sum(x))^2)
b_0 = ( sum(y) / n) - ( b_1 * ( sum(x) / n))
```

Verify your computations are correct using the `lm` function in R:

```
lm_mod = lm( y ~ x)
b_vec = coef(lm_mod)
pacman::p_load(testthat)
library(testthat)
expect_equal(b_0, as.numeric(b_vec[1]), tol = 1e-4)
expect_equal(b_1, as.numeric(b_vec[2]), tol = 1e-4)
```

6. We are now going to repeat one of the first linear model building exercises in history — that of Sir Francis Galton in 1886. First load up package `HistData`.

```
library("HistData")
```

In it, there is a dataset called `Galton`. Load it up.

```
Galton = HistData::Galton
```

You now should have a data frame in your workspace called `Galton`. Summarize this data frame and write a few sentences about what you see. Make sure you report n, p and a bit about what the columns represent and how the data was measured. See the help file `?Galton`.

```
#n is the number of the obsevations in 928 and p is the number of characteristics is two. For each obse

summary(Galton)
```

```
##      parent          child      
##  Min.   :64.00   Min.   :61.70  
##  1st Qu.:67.50   1st Qu.:66.20  
##  Median :68.50   Median :68.20  
##  Mean   :68.31   Mean   :68.09  
##  3rd Qu.:69.50   3rd Qu.:70.20  
##  Max.   :73.00   Max.   :73.70  
```

Find the average height (include both parents and children in this computation).

```
avg_height = sum(c(Galton$parent)*2/3 + c(Galton$child)*1/3)/928
```

If you were to use the null model, what would the RMSE be of this model be?

```
y_bar = mean(c(Galton$child))
sse = sum((c(Galton$child)*1/3 - y_bar)^2)
mse = sse/(length(Galton$parent)-2)
rmse = sqrt(mse)
rmse
```

```
## [1] 45.44907
```

Note that in Math 241 you learned that the sample average is an estimate of the "mean", the population expected value of height. We will call the average the "mean" going forward since it is probably correct to the nearest tenth of an inch with this amount of data.

Run a linear model attempting to explain the childrens' height using the parents' height. Use `lm` and use the R formula notation. Compute and report b_0, b_1, RMSE and R^2. Use the correct units to report these quantities.

```
summary(Galton)
```

```
##      parent          child
##  Min.   :64.00   Min.   :61.70
##  1st Qu.:67.50   1st Qu.:66.20
##  Median :68.50   Median :68.20
##  Mean   :68.31   Mean   :68.09
##  3rd Qu.:69.50   3rd Qu.:70.20
##  Max.   :73.00   Max.   :73.70
```

```
y = Galton$child
x = Galton$parent
lmode_galton= lm( y ~ x)
r=cor(x, y)
s_x = sd(x)
s_y = sd(y)
y_bar = mean(y)
x_bar = mean(x)
b_1 = r * s_y / s_x
b_1
```

```
## [1] 0.6462906
```

```
b_0 = y_bar - b_1 * x_bar
b_0
```

```
## [1] 23.94153
```

```
#y_hat = b_0 + b_1 *x
#residual = y - y_hat
#sse = sum (residual^2)
#mse = sse / length(y)
#rmse = sqrt(mse)
#Rsqr=((s_y^2) - sd(residual)^2)/ s_y^2
Rsqr=summary(lmode_galton)$r.squared
Rsqr
```

```
## [1] 0.2104629
```

```r
rmse=summary(lmode_galton)$sigma
rmse
```

```
## [1] 2.238547
```

Interpret all four quantities: b_0, b_1, RMSE and R^2.

b_0 is the intercept of the line it is in childs, units are in inches. b1 is slope of the line child over parent. RMSE is 95% predicitive errors plus/minus 2*rmse= 4.472, units are in inches. R2 is the percent of the variance of the child height explained by the parent.

How good is this model? How well does it predict? Discuss.

This is not a good model because it does not tell all the variantions in a childs height based on their parents. In this case, RMSE is big and R^2 is low which indicates huge errors in our prediction.

It is reasonable to assume that parents and their children have the same height? Explain why this is reasonable using basic biology and common sense.

The inheritance of these variants from one's parents helps explain why children usually grow to be approximately as tall as their parents. Assuming just two genes play role in height, calling them X and Y. X stands for tall and x stands for short. So there are 4 copies in total which will decide the height. If dad has like xxYY and mom has XxYy, there are 2 tall ones and 2 short ones, then the child could be average height.

If they were to have the same height and any differences were just random noise with expectation 0, what would the values of β_0 and β_1 be?

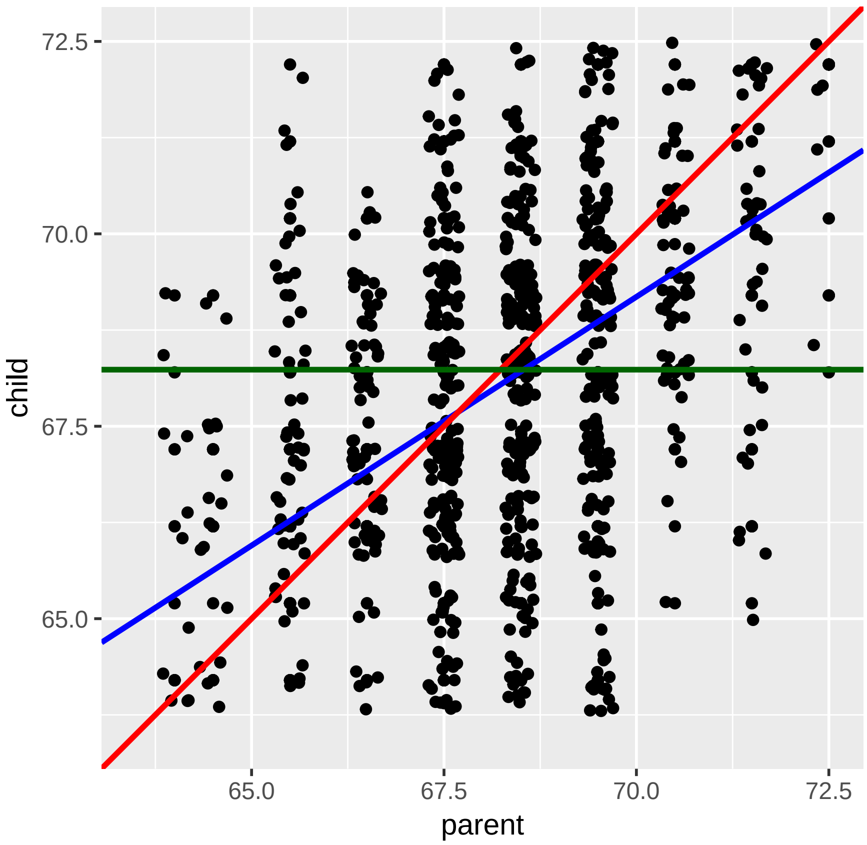
If they were to have the same height and the differences were really just random noise, then there would be a y = x realtionship. Hence the intercept, b_0 = 0 and the slope b_1 = 1.

Let's plot (a) the data in $\mathbb{D}$ as black dots, (b) your least squares line defined by b_0 and b_1 in blue, (c) the theoretical line β_0 and β_1 if the parent-child height equality held in red and (d) the mean height in green.

```r
pacman::p_load(ggplot2)
ggplot(Galton, aes(x = parent, y = child)) +
  geom_point() +
  geom_jitter() +
  geom_abline(intercept = b_0, slope = b_1, color = "blue", size = 1) +
  geom_abline(intercept = 0, slope = 1, color = "red", size = 1) +
  geom_abline(intercept = avg_height, slope = 0, color = "darkgreen", size = 1) +
  xlim(63.5, 72.5) +
  ylim(63.5, 72.5) +
  coord_equal(ratio = 1)
```

```
## Warning: Removed 76 rows containing missing values (geom_point).
```

```
## Warning: Removed 86 rows containing missing values (geom_point).
```



Fill in the following sentence:

Children of short parents became taller on average and children of tall parents became shorter on average.

Why did Galton call it "Regression towards mediocrity in hereditary stature" which was later shortened to "regression to the mean"?

Galton called it "Regression towards mediocrity in hereditary stature" because the data indicates the theoretical linear relationship is approaching an average height. We expect that most people would be around the mean.

Why should this effect be real?

This effect should be real since genetic makeup is passed down from parents to children, so on average we expect that the child will be closer to the mean.

You now have unlocked the mystery. Why is it that when modeling with y continuous, everyone calls it "regression"? Write a better, more descriptive and appropriate name for building predictive models with y continuous.

I would name this model estimated linear model. Galton used the term regression because he observed that the data was cluttering towards the average. Regression is a statistical process for estimating relationships between variables which implies the data will behave in a certain way which indicates that the term 'regression' will simplify our data into something, in this case a line, that can help us estimate the next child's height.

Create a dataset $\mathbb{D}$ which we call `Xy` such that the linear model as R^2 about 50% and RMSE approximately 1.

```
x = c(1, 4, 3, 0)
y = c(5, 6, 4, 3)
Xy = data.frame(x = x, y = y)
mod = lm(Xy$y~Xy$x)
```

```
summary(mod)$r.squared #the R^2 about 50%
```

```
## [1] 0.5
```

```
summary(mod)$sigma #the RMSE approx. 1
```

```
## [1] 1.118034
```

Create a dataset $\mathbb{D}$ which we call `Xy` such that the linear model as R^2 about 0% but x, y are clearly associated.

```
x = c(4, 2, 3, 1, 2, 1, 3, 5, 5, 4)
y = c(4, 4, 2, 3, 5, 2, 1, 1, 3, 5)
Xy = data.frame(x = x, y = y)
mod = lm(Xy$y~Xy$x)
summary(mod)$r.squared #the R^2 about 0%
```

```
## [1] 0.01
```

Load up the famous iris dataset and drop the data for Species "virginica".

```
data(iris)
class(iris)
```

```
## [1] "data.frame"
```

```
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

```
str(iris)
```

```
## 'data.frame':    150 obs. of  5 variables:
##  $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##  $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
##  $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
##  $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
##  $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
newiris = as.data.frame(iris[iris$Species != "virginica", ])
summary(iris)
```

```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width   
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100  
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300  
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300  
##  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199  
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800  
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500  
##        Species  
##  setosa    :50  
##  versicolor:50  
##  virginica :50  
## 
```

```
##
##
```

```r
summary(newiris)
```

```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
##  1st Qu.:5.000   1st Qu.:2.800   1st Qu.:1.500   1st Qu.:0.200
##  Median :5.400   Median :3.050   Median :2.450   Median :0.800
##  Mean   :5.471   Mean   :3.099   Mean   :2.861   Mean   :0.786
##  3rd Qu.:5.900   3rd Qu.:3.400   3rd Qu.:4.325   3rd Qu.:1.300
##  Max.   :7.000   Max.   :4.400   Max.   :5.100   Max.   :1.800
##        Species
##  setosa    :50
##  versicolor:50
##  virginica : 0
##
##
##
```

If the only input x is Species and you are trying to predict y which is Petal.Length, what would a reasonable, naive prediction be under both Species? Hint: it's what we did in class.

```r
x = newiris$Species
y = newiris$Petal.Length
```

Prove that this is the OLS model by fitting an appropriate `lm` and then using the predict function to verify you get the same answers as you wrote previously.

```r
simple_linear_model = lm(newiris$Petal.Length ~ newiris$Species)
#predict( )
```