Lab 4

Juan D Astudillo

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:

```
rm(list=ls())
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.

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

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)
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 using the `data` command:

```
data("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`.

```
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
```

```r
table(Galton)
```

```
##       child
## parent 61.7 62.2 63.2 64.2 65.2 66.2 67.2 68.2 69.2 70.2 71.2 72.2 73.2
##   64      1    0    2    4    1    2    2    1    1    0    0    0    0
##   64.5    1    1    4    4    1    5    5    0    2    0    0    0    0
##   65.5    1    0    9    5    7   11   11    7    7    5    2    1    0
##   66.5    0    3    3    5    2   17   17   14   13    4    0    0    0
##   67.5    0    3    5   14   15   36   38   28   38   19   11    4    0
##   68.5    1    0    7   11   16   25   31   34   48   21   18    4    3
##   69.5    0    0    1   16    4   17   27   20   33   25   20   11    4
##   70.5    1    0    1    0    1    1    3   12   18   14    7    4    3
##   71.5    0    0    0    0    1    3    4    3    5   10    4    9    2
##   72.5    0    0    0    0    0    0    0    1    2    1    2    7    2
##   73      0    0    0    0    0    0    0    0    0    0    0    1    3
##       child
## parent 73.7
##   64      0
##   64.5    0
##   65.5    0
##   66.5    0
##   67.5    0
##   68.5    0
##   69.5    5
##   70.5    3
##   71.5    2
##   72.5    4
##   73      0
```

```r
str(Galton)
```

```
## 'data.frame':    928 obs. of  2 variables:
##  $ parent: num  70.5 68.5 65.5 64.5 64 67.5 67.5 67.5 66.5 66.5 ...
##  $ child : num  61.7 61.7 61.7 61.7 61.7 62.2 62.2 62.2 62.2 62.2 ...
```

TO-DO

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

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

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.

```r
mod=lm(Galton$child~Galton$parent)
mod
```

```
##
## Call:
## lm(formula = Galton$child ~ Galton$parent)
```

```
##
## Coefficients:
##   (Intercept)  Galton$parent
##       23.9415         0.6463
```

```r
summary(mod)
```

```
##
## Call:
## lm(formula = Galton$child ~ Galton$parent)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.8050 -1.3661  0.0487  1.6339  5.9264
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   23.94153    2.81088   8.517   <2e-16 ***
## Galton$parent  0.64629    0.04114  15.711   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.239 on 926 degrees of freedom
## Multiple R-squared:  0.2105, Adjusted R-squared:  0.2096
## F-statistic: 246.8 on 1 and 926 DF,  p-value: < 2.2e-16
```

```r
b_0 = coef(mod)[1]
b_1 = coef(mod)[2]

names(summary(mod))
```

```
##  [1] "call"          "terms"         "residuals"     "coefficients"
##  [5] "aliased"       "sigma"         "df"            "r.squared"
##  [9] "adj.r.squared" "fstatistic"    "cov.unscaled"
```

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

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

```r
summary(mod)$sigma #the RMSE
```

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

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

b_0 is the intercept and b_1 is the slope of our linear model. RMSE indicates (how far off is our prediction to y) the average difference between the actual child's height and the predicted child's height. R^2 is the difference of the sample variance to the null model. Here R^2 is 0.20, indicating the sample variance of errors is consideration to the null model.

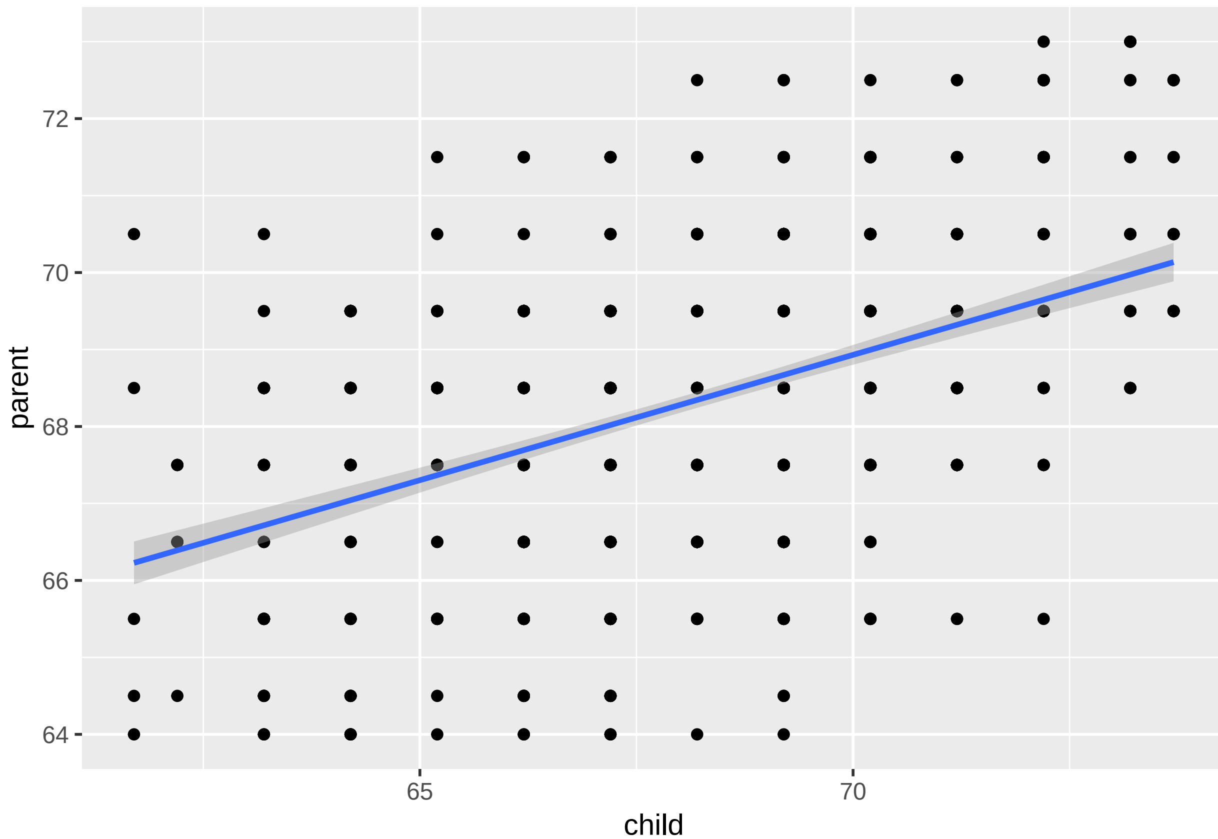
How good is this model? How well does it predict? Discuss.

The low R^2 shows that the model works poorly. RMSE, shows the measurement of the errors. both together imply that this is not a good model because f the big sample variance.

Now use the code from practice lecture 8 to plot the data and a best fit line using package `ggplot2`. Don't forget to load the library.

```r
library(ggplot2)
ggplot(Galton, aes(child, parent)) +
```

```r
  geom_point() +
  geom_smooth(method = 'lm')
```



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

yes it could be reasonable to assume the parents and their children have the same height since the parents will transfer some of their characteristics via DNA.

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?

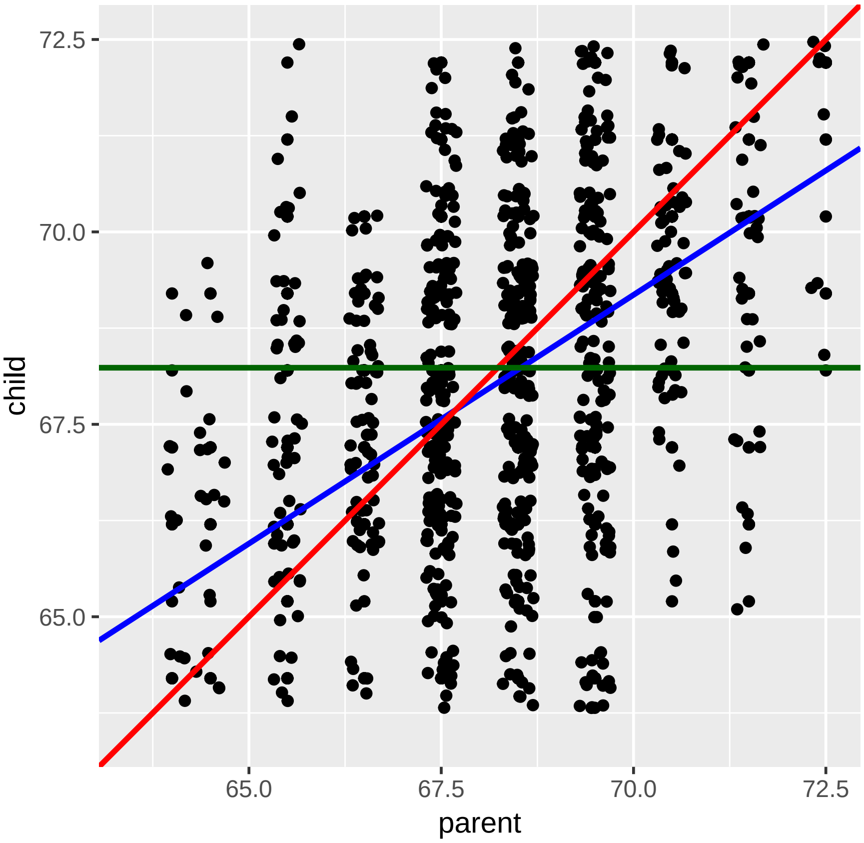
If they were to have the same height and the differences were just random noise, then we can say that y=x 100% correlation, the intersept would be b_0=0, and the slope b_1 would be equal to 1.

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
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 85 rows containing missing values (geom_point).
```



Fill in the following sentence:

TO-DO: Children of short parents became (taller than their parents) on average and children of tall parents became (shorter than their parents) 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 shows a relationship children-parent that is passsed hereditary which show a linear relationship.

Why should this effect be real?

it should be real. In reality parents pass its characteristics to their children by the information in the genes, the stronger characteristics will prevail, and tis will happend from generation to generation.

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.

TO-DO Galton called regression since as opposed to progressing, we are falling back to the mean. I would called the best match model, because we use the information from our old data set to create a linear model that matches the best to our data.

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(2,2,3,4,5,6,1)
y = c(1,2,2,0,0,1,3)
Xy = data.frame(x = x, y = y)
```

```r
mod=lm(Xy$y~Xy$x)
mod
```

```
##
## Call:
## lm(formula = Xy$y ~ Xy$x)
##
## Coefficients:
## (Intercept)         Xy$x
##      2.7353      -0.4412
```

```r
#summary(mod)
#b_0 = coef(mod)[1]
#b_1 = coef(mod)[2]
#names(summary(mod))
summary(mod)$r.squared #the R^2
```

```
## [1] 0.5090498
```

```r
summary(mod)$sigma #the RMSE
```

```
## [1] 0.8540561
```

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.

```r
x = rep( 1 : 5, 2)
y = rep( 2 : 3, 5)
Xy = data.frame(x = x, y = y)
mod=lm(Xy$y~Xy$x)
mod
```

```
##
## Call:
## lm(formula = Xy$y ~ Xy$x)
##
## Coefficients:
## (Intercept)         Xy$x
##   2.500e+00   -2.483e-17
```

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

```
## [1] 7.888609e-32
```

```r
summary(mod)$sigma #the RMSE
```

```
## [1] 0.559017
```

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

```r
data("iris")
newiris = 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
##
##
##
```

```
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.

```
#meanVersicolor = mean(newiris$Species == 'versicolor')
#meanSetosa =  mean(newiris$Species == 'setosa')
#meany = mean(newiris$Species)
#b_1 = (meanVersicolor - meanSetosa)
#b_0 = (meanSetosa)
#g_Petal_Length = b_0 + b_1

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

sumVersicolor = 0
sumSetosa = 0
n = numeric()
for(i in 1:length(x)){
  if(x[i] == 'setosa'){
    sumSetosa = sumSetosa + y[i]
    n = i
  } else{
    sumVersicolor = sumVersicolor +y[i]
  }
}

b_0 = sumVersicolor/n
b_1 = sumSetosa/(length(x) - n) - b_0

b_0
```

```
## [1] 4.26
```

```r
b_1
```

```
## [1] -2.798
```

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
reg1 <- lm(Petal.Length ~ newSpecies, newiris)
```

```
## Error in eval(predvars, data, env): object 'newSpecies' not found
```

```r
reg1
```

```
## Error in eval(expr, envir, enclos): object 'reg1' not found
```

```r
summary(reg1)
```

```
## Error in summary(reg1): object 'reg1' not found
```

```r
predict(reg1)
```

```
## Error in predict(reg1): object 'reg1' not found
```