Lab 4

Tziporah Horowitz

11:59PM Feb 29, 2020

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:

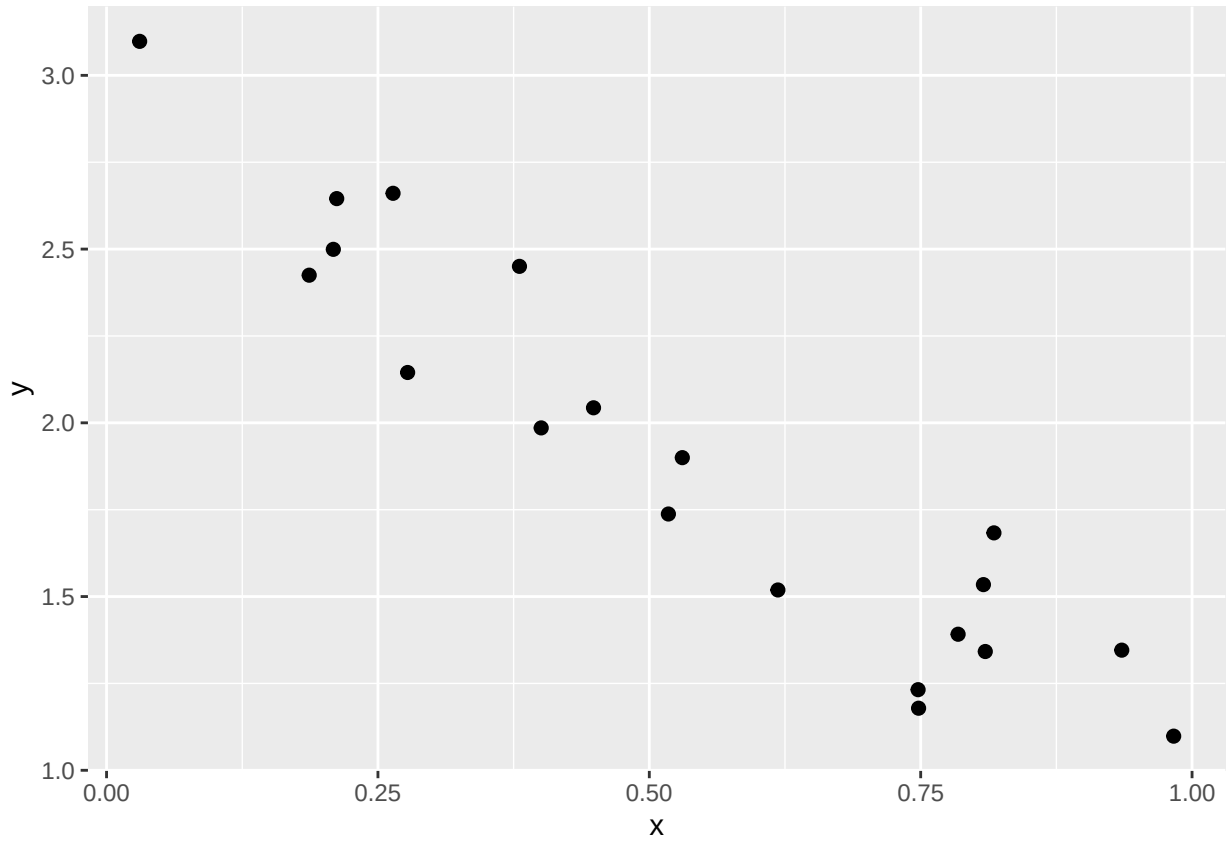
```
n = 20
x = runif(n)
beta_0 = 3
beta_1 = -2
y = beta_0 + beta_1 * x + rnorm(n, mean = 0, sd = 0.33)
```

Rewrite the computation of y so that it is h*(x) + epsilon.

```
h_star_x = beta_0 + beta_1 * x
epsilon = rnorm(n, mean = 0, sd = 0.33)
y = h_star_x + epsilon
```

Graph the data by running the following chunk:

```
pacman::p_load(ggplot2)
simple_df = data.frame(x = x, y = y)
simple_viz_obj = ggplot(simple_df, aes(x, y)) +
  geom_point(size = 2)
simple_viz_obj
```



Does this make sense given the values of $beta_0$ and $beta_1$?

Yes.

Write a function `my_simple_ols` that takes in a vector `x` and vector `y` and returns a list that contains the `b_0` (intercept), `b_1` (slope), `yhat` (the predictions), `e` (the residuals), `SSE`, `SST`, `MSE`, `RMSE` and `Rsq` (for the R-squared metric). Internally, you can only use the functions `sum` and `length` and other basic arithmetic operations. You should throw errors if the inputs are non-numeric and not the same length. You should also name the class of the return value 'my_simple_ols_obj`by using the`class' function as a setter. No need to create ROxygen documentation here.

```
my_simple_ols = function(x,y){
 if (class(x) != "numeric" | class(y) != "numeric") {stop("argument x or y is not numeric")}
 n = length(x)
  if (n != length(y)){stop("x and y must be same length")}

  y_bar = sum(y)/length(y)
  x_bar = sum(x)/length(x)
  s_x_squared = (1/(n-1) * sum((x - x_bar)^2))
  s_xy = (1/(n-1)) * sum((x - x_bar)*(y - y_bar))
  b1= s_xy/s_x_squared
  b0= y_bar - b1*x_bar
  y_hat = b0 + b1*x

  e = y - y_hat
  SSE = sum(e^2)
  SST = sum((y-y_bar)^2)
```

```
  Rsq = 1 - SSE/SST # 1 - NA
  MSE = SSE / (n-2)
  RMSE = sqrt(MSE)

  mod = list(b_0 = b0, b_1 = b1, y_hat = y_hat, e = e,
             SSE = SSE, SST = SST, Rsq = Rsq, MSE = MSE,
             RMSE = RMSE)
  class(mod) = "my_simple_ols_obj"
  mod
}
```

Verify your computations are correct for the vectors `x` and `y` from the first chunk using the `lm` function in R:

```
lm_mod = lm(y ~ x)
my_lm_mod = my_simple_ols(x, y)
#run the tests to ensure the function is up to spec
pacman::p_load(testthat)
expect_equal(my_lm_mod$b_0, as.numeric(coef(lm_mod)[1]), tol = 1e-4)
expect_equal(my_lm_mod$b_1, as.numeric(coef(lm_mod)[2]), tol = 1e-4)
expect_equal(my_lm_mod$RMSE, summary(lm_mod)$sigma, tol = 1e-4)
expect_equal(my_lm_mod$Rsq, summary(lm_mod)$r.squared, tol = 1e-4)
```

Verify that the average of the residuals is 0.

```
expect_equal(mean(my_lm_mod$e), 0, tol = 1e-4)
```

Create the X matrix for this data example.

```
X = cbind(1, x)
X
```

```
##                   x
##  [1,] 1 0.27735190
##  [2,] 1 0.80951920
##  [3,] 1 0.53041606
##  [4,] 1 0.03047193
##  [5,] 1 0.21202984
##  [6,] 1 0.74761795
##  [7,] 1 0.26392672
##  [8,] 1 0.93524838
##  [9,] 1 0.18667318
## [10,] 1 0.51757907
## [11,] 1 0.40040007
## [12,] 1 0.78447311
## [13,] 1 0.74806699
## [14,] 1 0.81748282
## [15,] 1 0.61847028
## [16,] 1 0.38038890
## [17,] 1 0.20891174
## [18,] 1 0.44860507
## [19,] 1 0.98307853
## [20,] 1 0.80780011
```

Use the `model.matrix` function to compute the matrix `X` and verify it is the same as your manual construction.

```
model.matrix(~ x)
```

```
##    (Intercept)          x
## 1            1 0.27735190
## 2            1 0.80951920
## 3            1 0.53041606
## 4            1 0.03047193
## 5            1 0.21202984
## 6            1 0.74761795
## 7            1 0.26392672
## 8            1 0.93524838
## 9            1 0.18667318
## 10           1 0.51757907
## 11           1 0.40040007
## 12           1 0.78447311
## 13           1 0.74806699
## 14           1 0.81748282
## 15           1 0.61847028
## 16           1 0.38038890
## 17           1 0.20891174
## 18           1 0.44860507
## 19           1 0.98307853
## 20           1 0.80780011
## attr(,"assign")
## [1] 0 1
```

Using matrix algebra, verify the OLS estimate is the same as you computed from the `my_simple_ols` function.

```
XtXinvX = solve(t(X) %*% X) %*% t(X)
b = XtXinvX %*% y
b
```

```
##        [,1]
##    2.929805
## x -1.931370
```

Find the hat matrix H.

```
H = X %*% XtXinvX
H
```

```
##               [,1]         [,2]       [,3]        [,4]         [,5]
##  [1,]  0.093629065  0.003662661 0.05084689  0.13536576  0.104672193
##  [2,]  0.003662661  0.099213728 0.04910053 -0.04066484 -0.008065969
##  [3,]  0.050846894  0.049100535 0.05001644  0.05165706  0.051061255
##  [4,]  0.135365758 -0.040664836 0.05165706  0.21702885  0.156973028
##  [5,]  0.104672193 -0.008065969 0.05106126  0.15697303  0.118510491
##  [6,]  0.014127477  0.088099309 0.04930367 -0.02018911  0.005047639
```

```
##  [7,]  0.095898682  0.001252158 0.05089095  0.13980655  0.107516282
##  [8,] -0.017592688  0.121788503 0.04868794 -0.08225360 -0.034701345
##  [9,]  0.108958904 -0.012618777 0.05114447  0.16536052  0.123882229
## [10,]  0.053017073  0.046795642 0.05005857  0.05590328  0.053780736
## [11,]  0.072826958  0.025756059 0.05044310  0.09466382  0.078604781
## [12,]  0.007896868  0.094716683 0.04918273 -0.03238008 -0.002760024
## [13,]  0.014051564  0.088179935 0.04930220 -0.02033764  0.004952511
## [14,]  0.002316359  0.100643602 0.04907440 -0.04329905 -0.009753038
## [15,]  0.035960746  0.064910740 0.04972748  0.02253043  0.032407215
## [16,]  0.076209980  0.022163036 0.05050877  0.10128312  0.082844093
## [17,]  0.105199327 -0.008625825 0.05107149  0.15800443  0.119171051
## [18,]  0.064677585  0.034411305 0.05028491  0.07871854  0.068392687
## [19,] -0.025678693  0.130376446 0.04853098 -0.09807489 -0.044834030
## [20,]  0.003953285  0.098905065 0.04910618 -0.04009620 -0.007701784
##              [,6]          [,7]         [,8]         [,9]      [,10]
##  [1,]  0.014127477  0.0958986820 -0.017592688  0.108958904 0.05301707
##  [2,]  0.088099309  0.0012521581  0.121788503 -0.012618777 0.04679564
##  [3,]  0.049303670  0.0508909506  0.048687942  0.051144466 0.05005857
##  [4,] -0.020189107  0.1398065492 -0.082253604  0.165360518 0.05590328
##  [5,]  0.005047639  0.1075162816 -0.034701345  0.123882229 0.05378074
##  [6,]  0.079494969  0.0122613614  0.105575802  0.001523036 0.04751931
##  [7,]  0.012261361  0.0982863658 -0.021108911  0.112025990 0.05317402
##  [8,]  0.105575802 -0.0211089107  0.154718529 -0.041342567 0.04532578
##  [9,]  0.001523036  0.1120259905 -0.041342567  0.129675151 0.05407717
## [10,]  0.047519314  0.0531740232  0.045325775  0.054077174 0.05020864
## [11,]  0.031231306  0.0740144339  0.014635147  0.080847611 0.05157855
## [12,]  0.084617876  0.0057066320  0.115228623 -0.006896807 0.04708845
## [13,]  0.079557386  0.0121814990  0.105693411  0.001420449 0.04751406
## [14,]  0.089206261 -0.0001641795  0.123874273 -0.014438126 0.04670254
## [15,]  0.061543301  0.0352304139  0.071750429  0.031027808 0.04902915
## [16,]  0.028449731  0.0775734425  0.009393973  0.085419317 0.05181249
## [17,]  0.004614220  0.1080708378 -0.035518012  0.124594581 0.05381719
## [18,]  0.037931853  0.0654411236  0.027260631  0.069834812 0.05101500
## [19,]  0.112224245 -0.0296155548  0.167245837 -0.052269731 0.04476661
## [20,]  0.087860354  0.0015578999  0.121338253 -0.012226038 0.04681574
##           [,11]        [,12]        [,13]         [,14]      [,15]
##  [1,] 0.07282696  0.007896868  0.014051564  0.0023163593 0.03596075
##  [2,] 0.02575606  0.094716683  0.088179935  0.1006436019 0.06491074
##  [3,] 0.05044310  0.049182726  0.049302196  0.0490744012 0.04972748
##  [4,] 0.09466382 -0.032380077 -0.020337641 -0.0432990457 0.02253043
##  [5,] 0.07860478 -0.002760024  0.004952511 -0.0097530384 0.03240721
##  [6,] 0.03123131  0.084617876  0.079557386  0.0892062606 0.06154330
##  [7,] 0.07401443  0.005706632  0.012181499 -0.0001641795 0.03523041
##  [8,] 0.01463515  0.115228623  0.105693411  0.1238742727 0.07175043
##  [9,] 0.08084761 -0.006896807  0.001420449 -0.0144381259 0.03102781
## [10,] 0.05157855  0.047088450  0.047514064  0.0467025419 0.04902915
## [11,] 0.06194319  0.027971418  0.031191587  0.0250516662 0.04265459
## [12,] 0.02797142  0.090630568  0.084691134  0.0960158975 0.06354823
## [13,] 0.03119159  0.084691134  0.079619935  0.0892892287 0.06156773
## [14,] 0.02505167  0.096015898  0.089289229  0.1021150196 0.06534396
## [15,] 0.04265459  0.063548228  0.061567729  0.0653439621 0.05451765
## [16,] 0.06371320  0.024706719  0.028404126  0.0213542509 0.04156598
## [17,] 0.07888058 -0.003268722  0.004518175 -0.0103291609 0.03223759
## [18,] 0.05767939  0.035835766  0.037906314  0.0339583849 0.04527695
```

```
## [19,] 0.01040451  0.123031818  0.112355924  0.1327117327 0.07435240
## [20,] 0.02590811  0.094436224  0.087940474  0.1003259701 0.06481722
##             [,16]        [,17]      [,18]       [,19]        [,20]
##  [1,] 0.076209980  0.105199327 0.06467758 -0.02567869  0.003953285
##  [2,] 0.022163036 -0.008625825 0.03441130  0.13037645  0.098905065
##  [3,] 0.050508768  0.051071488 0.05028491  0.04853098  0.049106176
##  [4,] 0.101283125  0.158004433 0.07871854 -0.09807489 -0.040096195
##  [5,] 0.082844093  0.119171051 0.06839269 -0.04483403 -0.007701784
##  [6,] 0.028449731  0.004614220 0.03793185  0.11222424  0.087860354
##  [7,] 0.077573443  0.108070838 0.06544112 -0.02961555  0.001557900
##  [8,] 0.009393973 -0.035518012 0.02726063  0.16724584  0.121338253
##  [9,] 0.085419317  0.124594581 0.06983481 -0.05226973 -0.012226038
## [10,] 0.051812494  0.053817189 0.05101500  0.04476661  0.046815740
## [11,] 0.063713200  0.078880581 0.05767939  0.01040451  0.025908114
## [12,] 0.024706719 -0.003268722 0.03583577  0.12303182  0.094436224
## [13,] 0.028404126  0.004518175 0.03790631  0.11235592  0.087940474
## [14,] 0.021354251 -0.010329161 0.03395838  0.13271173  0.100325970
## [15,] 0.041565977  0.032237590 0.04527695  0.07435240  0.064817221
## [16,] 0.065745536  0.083160767 0.05881750  0.00453634  0.022337627
## [17,] 0.083160767  0.119837979 0.06857002 -0.04574839 -0.008258129
## [18,] 0.058817498  0.068570024 0.05493780  0.02454036  0.034509075
## [19,] 0.004536340 -0.045748394 0.02454036  0.18127177  0.129872333
## [20,] 0.022337627 -0.008258129 0.03450908  0.12987233  0.098598337
```

Verify that this specific hat matrix is symmetric.

```
expect_equal(H, t(H))
```

Using the `diag` function, find the trace of the hat matrix.

```
diag(H)
```

```
##  [1] 0.09362907 0.09921373 0.05001644 0.21702885 0.11851049 0.07949497
##  [7] 0.09828637 0.15471853 0.12967515 0.05020864 0.06194319 0.09063057
## [13] 0.07961993 0.10211502 0.05451765 0.06574554 0.11983798 0.05493780
## [19] 0.18127177 0.09859834
```

```
sum(diag(H))
```

```
## [1] 2
```

Create a prediction method `g` that takes in a vector `x_future` and `my_simple_ols_obj`, an object of type `my_simple_ols_obj` and predicts y values for each entry in `x_future`.

```
g = function(x_future, my_simple_ols_obj){
  my_simple_ols_obj$b_0 + my_simple_ols_obj$b_1 * x_future
}
```

Use this function to verify that when predicting for the average x, you get the average y.

```
expect_equal(g(mean(x), my_lm_mod), mean(y))
```

Create a prediction method `g` that takes in a vector `x_future` and the dataset $\mathbb{D}$ i.e. `X` where the first column is the one vector and `y` and returns the OLS predictions.

```
g = function(x_future, X, y){
b = solve(t(X) %*% X) %*% t(X) %*% y
b[1] +b[2]*x_future
}
```

In class we spoke about error due to ignorance, misspecification error and estimation error. Show that as n grows, estimation error shrinks. Let us define an error metric that is the difference between b_0 and b_1 and β_0 and β_1. How about $h = ||b - \beta||^2$ where the quantities are now the vectors of size two. Show as n increases, this shrinks.

```
ns = 10^(1:7)
errors = array(dim=length(ns))
beta = c(beta_0, beta_1)
for (i in 1:length(ns)) {
  n = ns[i]
  x = runif(n)
  h_star_x = beta_0 + beta_1 * x
  epsilon = rnorm(n, mean = 0, sd = 0.33)
  y = h_star_x + epsilon

  mod = lm(y ~ x)
  b = coef(mod)
  errors[i] = sum((beta - b)^2)
}
errors
```

```
## [1] 8.405344e-03 2.438902e-02 1.243378e-04 1.895339e-04 5.512073e-06
## [6] 2.816691e-07 1.901726e-08
```

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`.

```
pacman::p_load(HistData)
```

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

```
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`. p is 1 and n is 928 the number of observations

```
head(Galton)
```

```
##   parent child
## 1   70.5  61.7
```

```
## 2   68.5  61.7
## 3   65.5  61.7
## 4   64.5  61.7
## 5   64.0  61.7
## 6   67.5  62.2
```

```
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
```

```
nrow(Galton)
```

```
## [1] 928
```

```
#?Galton
```

Galton contains 928 rows ($n = 928$) of data that show childrens' height, ranging from 61.7 units to 73 units, and a combined metric of their parents' height, ranging from 64 units to 73 units. The dataset is used to model childrens' height based on parents' height ($p = 1$). Both children and parents have an average height of about 68 units.

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

```
avg_height = (mean(Galton$parent) + mean(Galton$child)) / 2
avg_height
```

```
## [1] 68.19833
```

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

```
rmse_null = sqrt(mean((Galton$child - avg_height) ^ 2))
```

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.

```
mod = lm(child ~ parent, Galton)
b_0 = mod$coefficients[1]
b_1 = mod$coefficients[2]
b_0
```

```
## (Intercept)
##    23.94153
```

```
b_1
```

```
##    parent
## 0.6462906
```

```
summary(mod)$sigma
```

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

```
summary(mod)$r.sq
```

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

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

b_0: Childrens' height is 23.94 units when parents' height is 0. b_1: For each unit of parents' height, children's height increases by 0.65 units. RMSE: There's 95% confidence that childrens' height will be within $\pm 2 \cdot 2.24$ units of their parents' height. R^2: 0.21 of childrens' height is explained by parents' height.

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

Given an R^2 of 0.21 and a RMSE of 2.24, the model does not predict well. This is likely because there are additional factors that determine a child's height.

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.

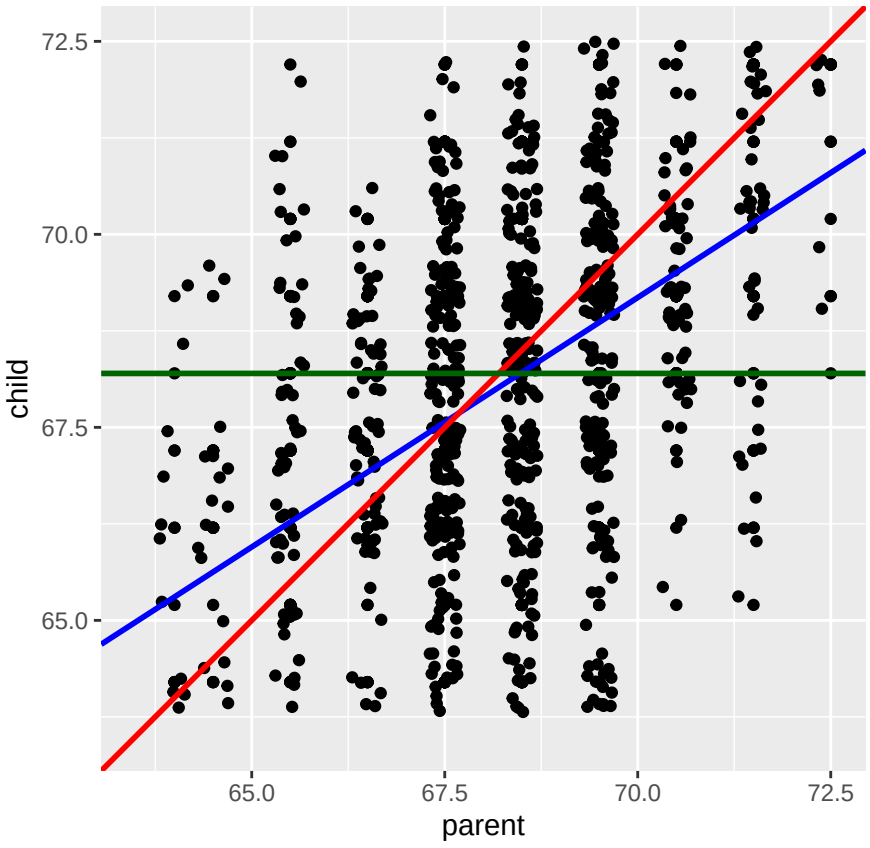
Because height is mostly genetic, it is reasonable to assume a child will have a close height to the parents, but because on other factors like nourishment, a child will not always have the same height as the parents.

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?

$\beta_0 = 0$ and $\beta_1 = 1$.

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.

```
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)
```



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"?

Because children were more often closer to the mean height than being the same height as their parents.

Why should this effect be real?

Because height is not solely based on parents' height.

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.

The regression line is the *mean* line of all data points.

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, 5, 1, 5, 1, 5)
y = c(1, 2, 2, 5, 2, 5)
Xy = data.frame(x = x, y = y)
mod1 = lm(y ~ x, data = Xy)
summary(mod1)$sigma
```

```
## [1] 1.290994
```

```
summary(mod1)$r.sq
```

```
## [1] 0.5505618
```

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(1, 2, 3, 4, 5, 6)
y = c(1, 2, 1, 2, 1, 2)
Xy = data.frame(x = x, y = y)
mod2 = lm(y ~ x, data = Xy)
summary(mod2)$sigma
```

```
## [1] 0.58554
```

```
summary(mod2)$r.sq
```

```
## [1] 0.08571429
```

Extra credit: create a dataset $\mathbb{D}$ and a model (hint: not a lienar model) that can give you R^2 arbitrarily close to 1 but RMSE arbitrarily high.

```
x = c(1, 2, 3, 4, 5, 6)
y = c(100^2, 200^2, 300^2, 400^2, 500^2, 600^2)
Xy = data.frame(x = x, y = y)
mod3 = lm(y ~ x, data = Xy)
summary(mod3)$sigma
```

```
## [1] 30550.5
```

```
summary(mod3)$r.sq
```

```
## [1] 0.958279
```

Load up the famous iris dataset. We are going to do a different prediction problem. Imagine 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 all Species? Hint: it's what we did in class.

```
pacman::p_load(iris)
```

```
## Installing package into 'C:/Users/Tziporah/Documents/R/win-library/3.6'
## (as 'lib' is unspecified)
```

```
data("iris")
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
```

```r
pl_setosa = mean(iris$Petal.Length[1:50])
pl_versicolor = mean(iris$Petal.Length[51:100])
pl_virginica = mean(iris$Petal.Length[101:150])

pl_setosa
```

```
## [1] 1.462
```

```r
pl_versicolor
```

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

```r
pl_virginica
```

```
## [1] 5.552
```

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. Show this by doing a linear regression with and without the intercept.

```r
mod = lm(Petal.Length ~ 0 + Species, data = iris)
predict(mod)
```

```
##     1     2     3     4     5     6     7     8     9    10    11    12
## 1.462 1.462 1.462 1.462 1.462 1.462 1.462 1.462 1.462 1.462 1.462 1.462
##    13    14    15    16    17    18    19    20    21    22    23    24
## 1.462 1.462 1.462 1.462 1.462 1.462 1.462 1.462 1.462 1.462 1.462 1.462
##    25    26    27    28    29    30    31    32    33    34    35    36
## 1.462 1.462 1.462 1.462 1.462 1.462 1.462 1.462 1.462 1.462 1.462 1.462
##    37    38    39    40    41    42    43    44    45    46    47    48
## 1.462 1.462 1.462 1.462 1.462 1.462 1.462 1.462 1.462 1.462 1.462 1.462
##    49    50    51    52    53    54    55    56    57    58    59    60
## 1.462 1.462 4.260 4.260 4.260 4.260 4.260 4.260 4.260 4.260 4.260 4.260
##    61    62    63    64    65    66    67    68    69    70    71    72
## 4.260 4.260 4.260 4.260 4.260 4.260 4.260 4.260 4.260 4.260 4.260 4.260
##    73    74    75    76    77    78    79    80    81    82    83    84
## 4.260 4.260 4.260 4.260 4.260 4.260 4.260 4.260 4.260 4.260 4.260 4.260
##    85    86    87    88    89    90    91    92    93    94    95    96
## 4.260 4.260 4.260 4.260 4.260 4.260 4.260 4.260 4.260 4.260 4.260 4.260
##    97    98    99   100   101   102   103   104   105   106   107   108
## 4.260 4.260 4.260 4.260 5.552 5.552 5.552 5.552 5.552 5.552 5.552 5.552
##   109   110   111   112   113   114   115   116   117   118   119   120
## 5.552 5.552 5.552 5.552 5.552 5.552 5.552 5.552 5.552 5.552 5.552 5.552
##   121   122   123   124   125   126   127   128   129   130   131   132
## 5.552 5.552 5.552 5.552 5.552 5.552 5.552 5.552 5.552 5.552 5.552 5.552
##   133   134   135   136   137   138   139   140   141   142   143   144
## 5.552 5.552 5.552 5.552 5.552 5.552 5.552 5.552 5.552 5.552 5.552 5.552
##   145   146   147   148   149   150
## 5.552 5.552 5.552 5.552 5.552 5.552
```

Use the `model.matrix` function to compute the matrix `X` for the regression with the intercept and without the intercept. What is different?

```
model.matrix(Petal.Length ~ Species, data = iris) # the intercept is setosa
```

```
##     (Intercept) Speciesversicolor Speciesvirginica
## 1             1                 0                0
## 2             1                 0                0
## 3             1                 0                0
## 4             1                 0                0
## 5             1                 0                0
## 6             1                 0                0
## 7             1                 0                0
## 8             1                 0                0
## 9             1                 0                0
## 10            1                 0                0
## 11            1                 0                0
## 12            1                 0                0
## 13            1                 0                0
## 14            1                 0                0
## 15            1                 0                0
## 16            1                 0                0
## 17            1                 0                0
## 18            1                 0                0
## 19            1                 0                0
## 20            1                 0                0
## 21            1                 0                0
## 22            1                 0                0
## 23            1                 0                0
## 24            1                 0                0
## 25            1                 0                0
## 26            1                 0                0
## 27            1                 0                0
## 28            1                 0                0
## 29            1                 0                0
## 30            1                 0                0
## 31            1                 0                0
## 32            1                 0                0
## 33            1                 0                0
## 34            1                 0                0
## 35            1                 0                0
## 36            1                 0                0
## 37            1                 0                0
## 38            1                 0                0
## 39            1                 0                0
## 40            1                 0                0
## 41            1                 0                0
## 42            1                 0                0
## 43            1                 0                0
## 44            1                 0                0
## 45            1                 0                0
## 46            1                 0                0
## 47            1                 0                0
## 48            1                 0                0
```

```
## 49            1               0               0
## 50            1               0               0
## 51            1               1               0
## 52            1               1               0
## 53            1               1               0
## 54            1               1               0
## 55            1               1               0
## 56            1               1               0
## 57            1               1               0
## 58            1               1               0
## 59            1               1               0
## 60            1               1               0
## 61            1               1               0
## 62            1               1               0
## 63            1               1               0
## 64            1               1               0
## 65            1               1               0
## 66            1               1               0
## 67            1               1               0
## 68            1               1               0
## 69            1               1               0
## 70            1               1               0
## 71            1               1               0
## 72            1               1               0
## 73            1               1               0
## 74            1               1               0
## 75            1               1               0
## 76            1               1               0
## 77            1               1               0
## 78            1               1               0
## 79            1               1               0
## 80            1               1               0
## 81            1               1               0
## 82            1               1               0
## 83            1               1               0
## 84            1               1               0
## 85            1               1               0
## 86            1               1               0
## 87            1               1               0
## 88            1               1               0
## 89            1               1               0
## 90            1               1               0
## 91            1               1               0
## 92            1               1               0
## 93            1               1               0
## 94            1               1               0
## 95            1               1               0
## 96            1               1               0
## 97            1               1               0
## 98            1               1               0
## 99            1               1               0
## 100           1               1               0
## 101           1               0               1
## 102           1               0               1
```

```
## 103           1                 0                 1
## 104           1                 0                 1
## 105           1                 0                 1
## 106           1                 0                 1
## 107           1                 0                 1
## 108           1                 0                 1
## 109           1                 0                 1
## 110           1                 0                 1
## 111           1                 0                 1
## 112           1                 0                 1
## 113           1                 0                 1
## 114           1                 0                 1
## 115           1                 0                 1
## 116           1                 0                 1
## 117           1                 0                 1
## 118           1                 0                 1
## 119           1                 0                 1
## 120           1                 0                 1
## 121           1                 0                 1
## 122           1                 0                 1
## 123           1                 0                 1
## 124           1                 0                 1
## 125           1                 0                 1
## 126           1                 0                 1
## 127           1                 0                 1
## 128           1                 0                 1
## 129           1                 0                 1
## 130           1                 0                 1
## 131           1                 0                 1
## 132           1                 0                 1
## 133           1                 0                 1
## 134           1                 0                 1
## 135           1                 0                 1
## 136           1                 0                 1
## 137           1                 0                 1
## 138           1                 0                 1
## 139           1                 0                 1
## 140           1                 0                 1
## 141           1                 0                 1
## 142           1                 0                 1
## 143           1                 0                 1
## 144           1                 0                 1
## 145           1                 0                 1
## 146           1                 0                 1
## 147           1                 0                 1
## 148           1                 0                 1
## 149           1                 0                 1
## 150           1                 0                 1
## attr(,"assign")
## [1] 0 1 1
## attr(,"contrasts")
## attr(,"contrasts")$Species
## [1] "contr.treatment"
```

```r
model.matrix(Petal.Length ~ 0 + Species, data = iris)
```

```
##     Speciessetosa Speciesversicolor Speciesvirginica
## 1               1                 0                0
## 2               1                 0                0
## 3               1                 0                0
## 4               1                 0                0
## 5               1                 0                0
## 6               1                 0                0
## 7               1                 0                0
## 8               1                 0                0
## 9               1                 0                0
## 10              1                 0                0
## 11              1                 0                0
## 12              1                 0                0
## 13              1                 0                0
## 14              1                 0                0
## 15              1                 0                0
## 16              1                 0                0
## 17              1                 0                0
## 18              1                 0                0
## 19              1                 0                0
## 20              1                 0                0
## 21              1                 0                0
## 22              1                 0                0
## 23              1                 0                0
## 24              1                 0                0
## 25              1                 0                0
## 26              1                 0                0
## 27              1                 0                0
## 28              1                 0                0
## 29              1                 0                0
## 30              1                 0                0
## 31              1                 0                0
## 32              1                 0                0
## 33              1                 0                0
## 34              1                 0                0
## 35              1                 0                0
## 36              1                 0                0
## 37              1                 0                0
## 38              1                 0                0
## 39              1                 0                0
## 40              1                 0                0
## 41              1                 0                0
## 42              1                 0                0
## 43              1                 0                0
## 44              1                 0                0
## 45              1                 0                0
## 46              1                 0                0
## 47              1                 0                0
## 48              1                 0                0
## 49              1                 0                0
## 50              1                 0                0
```

```
## 51              0               1               0
## 52              0               1               0
## 53              0               1               0
## 54              0               1               0
## 55              0               1               0
## 56              0               1               0
## 57              0               1               0
## 58              0               1               0
## 59              0               1               0
## 60              0               1               0
## 61              0               1               0
## 62              0               1               0
## 63              0               1               0
## 64              0               1               0
## 65              0               1               0
## 66              0               1               0
## 67              0               1               0
## 68              0               1               0
## 69              0               1               0
## 70              0               1               0
## 71              0               1               0
## 72              0               1               0
## 73              0               1               0
## 74              0               1               0
## 75              0               1               0
## 76              0               1               0
## 77              0               1               0
## 78              0               1               0
## 79              0               1               0
## 80              0               1               0
## 81              0               1               0
## 82              0               1               0
## 83              0               1               0
## 84              0               1               0
## 85              0               1               0
## 86              0               1               0
## 87              0               1               0
## 88              0               1               0
## 89              0               1               0
## 90              0               1               0
## 91              0               1               0
## 92              0               1               0
## 93              0               1               0
## 94              0               1               0
## 95              0               1               0
## 96              0               1               0
## 97              0               1               0
## 98              0               1               0
## 99              0               1               0
## 100             0               1               0
## 101             0               0               1
## 102             0               0               1
## 103             0               0               1
## 104             0               0               1
```

```
## 105            0                 0                 1
## 106            0                 0                 1
## 107            0                 0                 1
## 108            0                 0                 1
## 109            0                 0                 1
## 110            0                 0                 1
## 111            0                 0                 1
## 112            0                 0                 1
## 113            0                 0                 1
## 114            0                 0                 1
## 115            0                 0                 1
## 116            0                 0                 1
## 117            0                 0                 1
## 118            0                 0                 1
## 119            0                 0                 1
## 120            0                 0                 1
## 121            0                 0                 1
## 122            0                 0                 1
## 123            0                 0                 1
## 124            0                 0                 1
## 125            0                 0                 1
## 126            0                 0                 1
## 127            0                 0                 1
## 128            0                 0                 1
## 129            0                 0                 1
## 130            0                 0                 1
## 131            0                 0                 1
## 132            0                 0                 1
## 133            0                 0                 1
## 134            0                 0                 1
## 135            0                 0                 1
## 136            0                 0                 1
## 137            0                 0                 1
## 138            0                 0                 1
## 139            0                 0                 1
## 140            0                 0                 1
## 141            0                 0                 1
## 142            0                 0                 1
## 143            0                 0                 1
## 144            0                 0                 1
## 145            0                 0                 1
## 146            0                 0                 1
## 147            0                 0                 1
## 148            0                 0                 1
## 149            0                 0                 1
## 150            0                 0                 1
## attr(,"assign")
## [1] 1 1 1
## attr(,"contrasts")
## attr(,"contrasts")$Species
## [1] "contr.treatment"
```