STATS 305A HW # 2

Adhitya Venkatesh

October 10, 2017

Written Problems:

1.)

Distribution of $Y = [Y_1, Y_2]^T$ given by

$$N([\mu_1, \mu_2]^T, \sum)$$

Where $\sum$ is given by:

$$\sum = \begin{bmatrix} \sigma_1 & \sigma_2 \\ \sigma_3 & \sigma_4 \end{bmatrix}$$

We write out the characteristic function of $Y, Y_1, and Y_2$:

$$\phi_Y(t) = \exp(i(t_1\mu_1 + t_2\mu_2) - (\sigma_1 t_1^2 + \sigma_2 t_2^2 + 2\sigma t_1 t_2)/2)$$

Where $\sigma_2 = \sigma_3 = \sigma$.

Similarly,

$$\phi_{Y_1}(t) = \exp(it_1\mu_1 - \sigma_1 t_1^2/2)$$

and

$$\phi_{Y_2}(t) = \exp(it_2\mu_2 - \sigma_2 t_2^2/2)$$

If Y_1 and Y_2 are independent, then

$$\phi_Y(t) = \phi_{Y_1}(t) * \phi_{Y_2}(t)$$

After performing the algebra, we arrive at the following condition for the equation to hold: $\sigma = 0$.

2.)

a.) WLOG, suppose n = 4. The resulting matrix analysis is easily generalized to any dimension: $Y = AX$ where A is given by

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

From here, we merely apply the relationship $Cov(Y) = Cov(AX) = ACov(X)A^T$:

Note that Cov(X) is given by

$$\mathbf{cov(X)} = \begin{bmatrix} v_1 & 0 & 0 & 0 \\ 0 & v_2 & 0 & 0 \\ 0 & 0 & v_3 & 0 \\ 0 & 0 & 0 & v_4 \end{bmatrix}$$

Thus, upon matrix multiplication, we obtain

$$\mathbf{cov(Y)} = \begin{bmatrix} v_1 & -v_1 & 0 & 0 \\ -v_1 & v_1 + v_2 & -v_2 & 0 \\ 0 & -v_2 & v_2 + v_3 & -v_3 \\ 0 & 0 & -v_3 & v_3 + v_4 \end{bmatrix}$$

This result generalizes for any “n” as

$$Cov(Y)_{ij} = \begin{cases} \sum_{m=1}^{k} v_m & i = j = k \\ -v_j & i = j + 1 \\ -v_i & i = j - 1 \\ 0 & otherwise \end{cases}$$

b.)

Proceeding in the same fashion as the previous part, $X = BY$ where B is given by (again WLOG, n = 4):

$$\mathbf{B} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Again we define cov(Y) as

$$\mathbf{cov(Y)} = \begin{bmatrix} u_1 & 0 & 0 & 0 \\ 0 & u_2 & 0 & 0 \\ 0 & 0 & u_3 & 0 \\ 0 & 0 & 0 & u_4 \end{bmatrix}$$

Hence, $cov(X) = cov(BY) = Bcov(Y)B^T$, meaning for the case n = 4,

$$\mathbf{cov(X)} = \begin{bmatrix} u_1 & u_1 & u_1 & u_1 \\ u_1 & u_1+u_2 & u_1+u_2 & u_1+u_2 \\ u_1 & u_1+u_2 & u_1+u_2+u_3 & u_1+u_2+u_3 \\ u_1 & u_1+u_2 & u_1+u_2+u_3 & u_1+u_2+u_3+u_4 \end{bmatrix}$$

Thus, to generalize,

$$Cov(X)_{ij} = \sum_{k=1}^{min(i,j)} u_k$$

3.)

$\sum_{i=1}^{n-1}(X_{i+1}-X_i)^2 = \sum_{i=1}^{n-1}(X_{i+1}^2 + X_i^2 - 2X_{i+1}X_i)$

Thus, $E[Q] = \sum_{i=1}^{n-1}(E[X_{i+1}^2] + E[X_i^2] - 2E[X_{i+1}X_i])$

but as $E[X_{i+1}X_i] = E[X_{i+1}]E[X_i]$ since all X_i are i.i.d and $E[X_j] = \mu$ and $E[X_j^2] = Var[X_j] + (E[X_j])^2 = ^2 + ^2 $

Thus,

$$= \sum_{i=1}^{n-1}(2\mu^2 + 2\sigma^2 - 2\mu^2)$$

$$= 2\sum_{i=1}^{n-1}\sigma^2 = 2(n-1)\sigma^2$$

Therefore,

$$E[Q/2(n-1)] = E[Q]/2(n-1) = \sigma^2$$

Thus, by definition, Q/2(n-1) is an unbiased estimator of the variance.

4.)

a.) Taking variances on both sides of the AR(1) equation, we have $Var(x_t) = ^2Var(x_{t-1}) + 1 $. Thus, $Var(x_2) = \phi^2/(1-\phi^2) + 1 = 1/(1-\phi^2)$. Thus, by induction, we see $Var(x_t) = 1/(1-\phi^2)$ for all t. Now, assuming x_t, ϵ_j are independent for all $t \le j$, we can establish $cov(x_t, \epsilon_j) = 0$.

Now suppose WLOG $j \ge i$, then we have

$$x_j = \phi^{j-i}x_i + \sum_{k=i}^{j-1}\epsilon_k\phi^{j-k-1}$$

Now, we expand

$$cov(x_i, x_j) = cov(x_i, \phi^{j-i}x_i + \sum_{k=i}^{j-1}\epsilon_k\phi^{j-k-1})$$

= $\phi^{j-i}var(x_i) + \sum_{k=i}^{j-1}\phi^{j-k-1}cov(x_i, \epsilon_k) = \phi^{j-i}var(x_i)$ since $cov(x_i, \epsilon_j) = 0$.

Thus,

$$\sum_{ij} = \phi^{|i-j|}/(1-\phi^2)$$

b.)

We determine the precision matrix $Q = \sum^{-1}$ by determining the RREF of an augmented $\sum | I_n$ where I_n is the n x n identity matrix. Using row operations to convert the LHS to I_n, the RHS (Q) is given by

$$Cov(Q)_{ij} = \begin{cases} 1 & i = j = 1, n \\ -\phi & i = j+1, i = j-1 \\ 1+\phi^2 & 2 \le i = j \le n \\ 0 & otherwise \end{cases}$$

5.)

$ z_i = ax_i + b$ and $y_i = \beta_0^* + \beta_1^* z_i + \epsilon_i$

$$y_i = \beta_0^* + \beta_1^*(ax_i + b) + \epsilon_i$$

$$= (\beta_0^* + \beta_1^* b) + (\beta_1^* a)x_i + \epsilon_i$$

Thus, by equating terms, we can recover the original coefficients $[\beta_0, \beta_1]$:

$$[\beta_0, \beta_1] = [\beta_0^* + \beta_1^* b, \beta_1^* a]$$

Computational Problems:

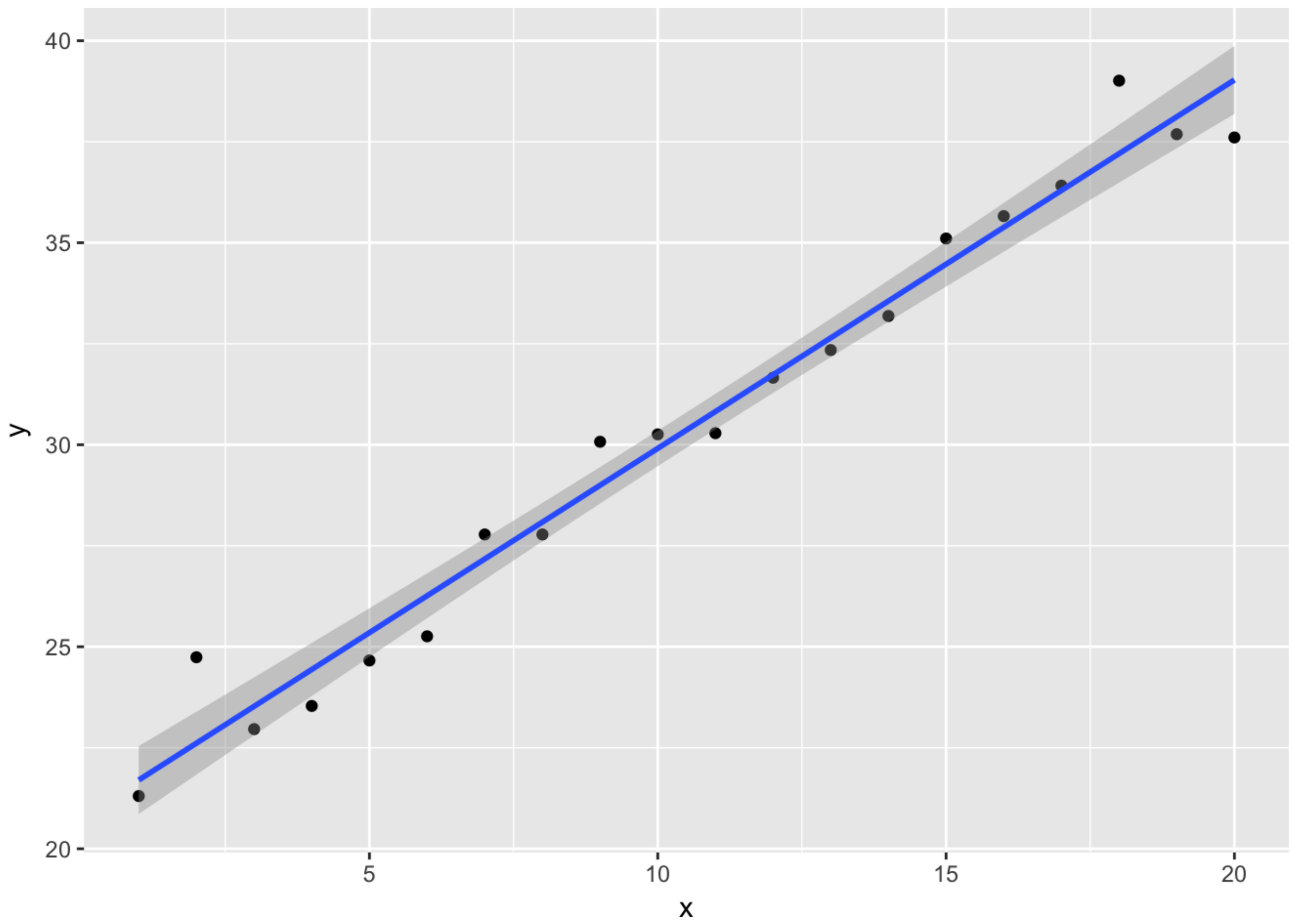
1.)

```
#Generating the data as given
x = 1:20
y = x + rnorm(20,20,1)
myData = as.data.frame(cbind(x,y))
require(ggplot2)
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 3.3.2
```

```
ggplot(myData, aes(x,y)) + geom_point() + stat_smooth(method = "lm")
```




```r
  #Computing the linear model via lm(), displaying the results with a scatteplot,
and its summarizing the model

  mod_1 = lm(y ~ x)
  summary(mod_1)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.4241 -0.5420 -0.3035  0.4121  2.1271
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 20.78901    0.43138   48.19  < 2e-16 ***
## x            0.91203    0.03601   25.33 1.58e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9286 on 18 degrees of freedom
## Multiple R-squared:  0.9727, Adjusted R-squared:  0.9712
## F-statistic: 641.4 on 1 and 18 DF,  p-value: 1.58e-15
```

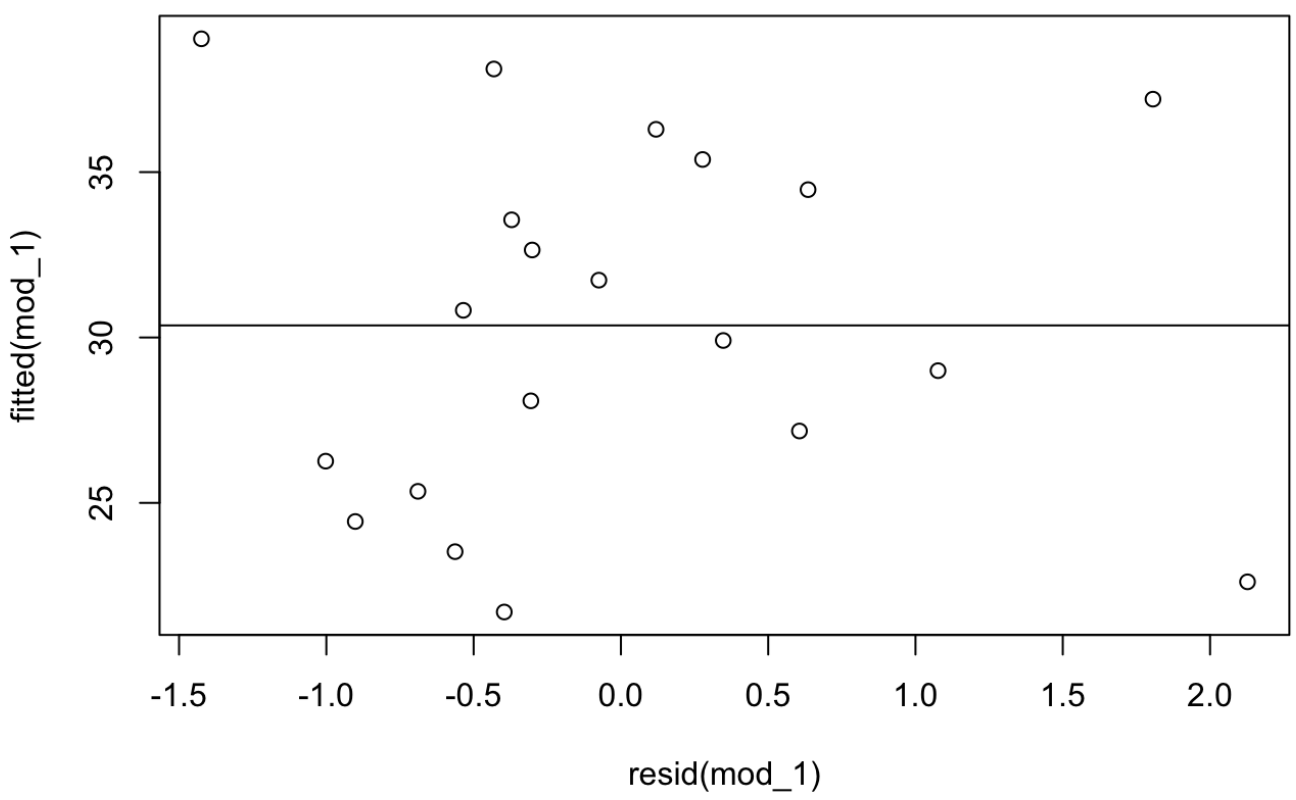
```r
#Computing the linear model via direct calculation

result = list()
result$B_slope = cov(x,y)/var(x) #formula for simple regression
result$B_intercept = mean(y) - (result$B_slope)*mean(x)
print(result)
```

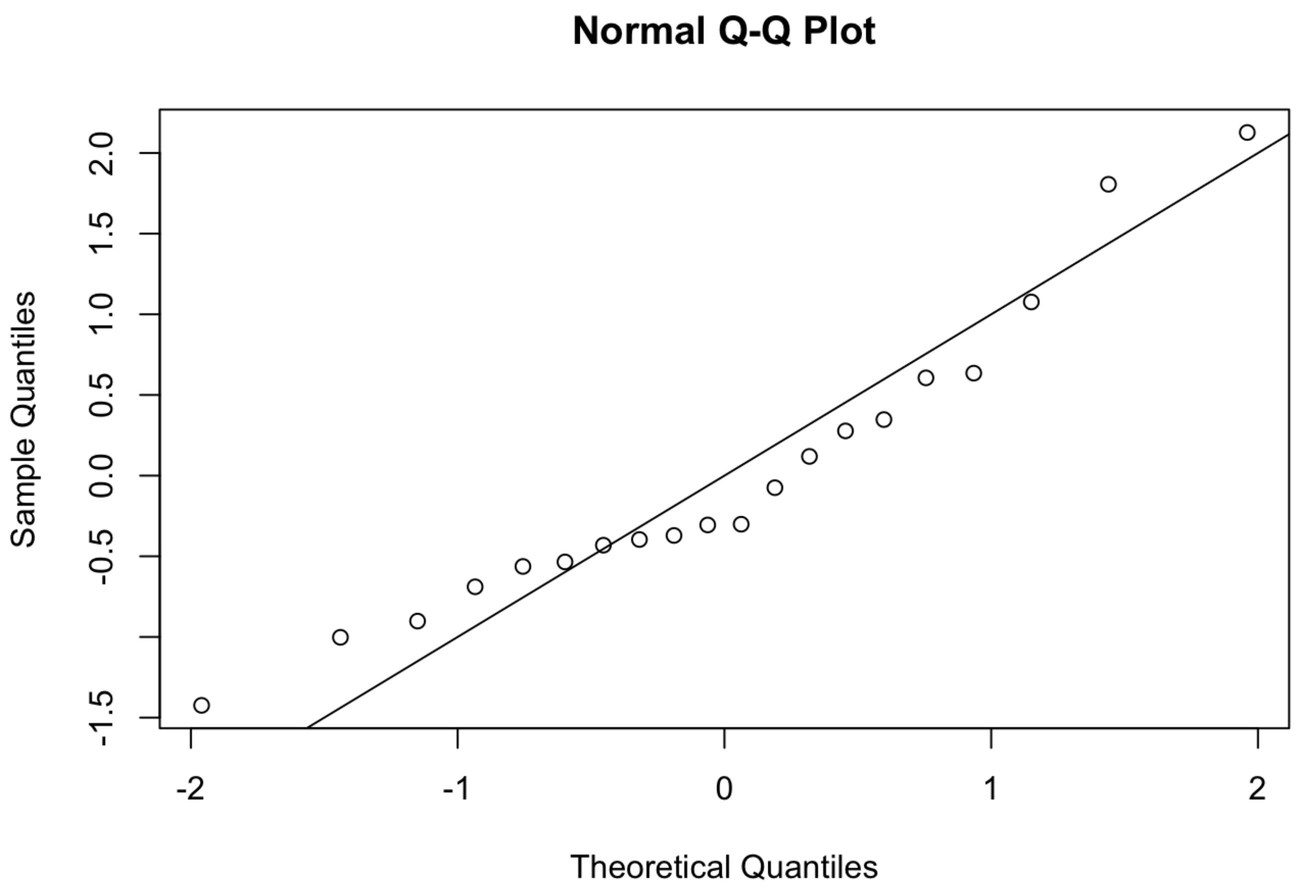
```
## $B_slope
## [1] 0.9120256
##
## $B_intercept
## [1] 20.78901
```

```r
#Residual and Normality plots

#X_mod = model.matrix(x)
plot(resid(mod_1), fitted(mod_1))
abline(h = mean(fitted(mod_1)))
```




```r
qqnorm(resid(mod_1))
abline(0,1)
```



The plots indicate that the regression model is indeed a good fit as the residuals are randomly scattered and roughly centering around 0. In addition, the proximity of the QQ plot to the line y=x (as expected considering how the data was generated) indicates that the residuals are normally distributed.

2.

a.)

```
library(HistData)
```

```
## Warning: package 'HistData' was built under R version 3.3.2
```

```
data("Galton")

#Linear model with child as the dependent variable
height_mod1 = lm(Galton$child ~ Galton$parent)
summary(height_mod1)
```

```
##
## 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
```

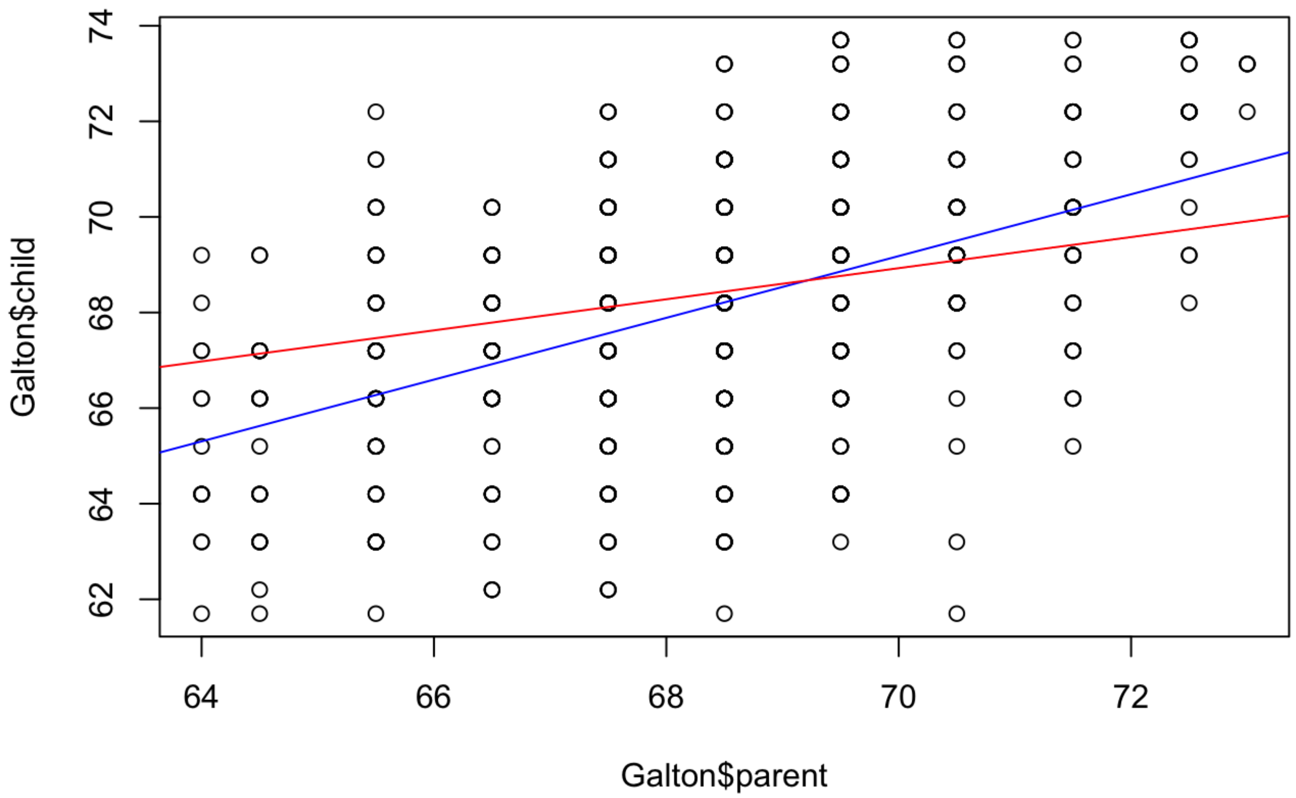
b.)

```
height_mod2 = lm(Galton$parent ~ Galton$child)
summary(height_mod2)
```

```
##
## Call:
## lm(formula = Galton$parent ~ Galton$child)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.6702 -1.1702 -0.1471  1.1324  4.2722
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  46.13535    1.41225   32.67   <2e-16 ***
## Galton$child  0.32565    0.02073   15.71   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.589 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
```

c.)

```
plot(Galton$parent, Galton$child)
abline(lm(Galton$child ~ Galton$parent), col = "blue") #Model from part a
abline(lm(Galton$parent ~ Galton$child), col = "red") #Model from part b
```



We can clearly see that the lines are not the same and in fact the 1st model has a larger slope estimate and smaller y-intercept estimate. In general, x regressed on y and y regressed on x do not yield the same model because for the parameter estimate of the slope in the case of y ~ x,

$$\beta_1 = cov(x, y)/var(y)$$

Whereas in the case x ~ y,

$$\beta_2 = cov(y, x)/var(x)$$

Thus, unless var(x) = var(y), the resulting models will not be equal.

3.

a.)

```
#library(GGaly)

sol_data = read.csv("/Users/Adi/Documents/COTERM_CLASSES/Fall_17/STATS 305/solubili
ty.csv")
summary(sol_data) #summarizing statistics of vatiables
```

```
##     NumAtoms      NumNonHAtoms      NumBonds      NumNonHBonds
##  Min.   : 5.00   Min.   : 2.00   Min.   : 4.00   Min.   : 1.00
##  1st Qu.:17.00   1st Qu.: 8.00   1st Qu.:17.00   1st Qu.: 8.00
##  Median :22.00   Median :12.00   Median :23.00   Median :12.00
##  Mean   :25.28   Mean   :13.05   Mean   :25.68   Mean   :13.45
##  3rd Qu.:30.50   3rd Qu.:17.00   3rd Qu.:31.00   3rd Qu.:18.00
##  Max.   :94.00   Max.   :47.00   Max.   :97.00   Max.   :50.00
##   NumMultBonds     NumRotBonds      NumDblBonds     NumAromaticBonds
##  Min.   : 0.000   Min.   : 0.000   Min.   :0.0000   Min.   : 0.00
##  1st Qu.: 1.000   1st Qu.: 0.000   1st Qu.:0.0000   1st Qu.: 0.00
##  Median : 6.000   Median : 2.000   Median :1.0000   Median : 6.00
##  Mean   : 6.189   Mean   : 2.176   Mean   :0.9771   Mean   : 5.19
##  3rd Qu.:10.000   3rd Qu.: 3.000   3rd Qu.:2.0000   3rd Qu.: 6.00
##  Max.   :27.000   Max.   :16.000   Max.   :7.0000   Max.   :27.00
##   NumHydrogen      NumCarbon       NumNitrogen       NumOxygen
##  Min.   : 0.00   Min.   : 1.000   Min.   :0.0000   Min.   : 0.000
##  1st Qu.: 7.00   1st Qu.: 6.000   1st Qu.:0.0000   1st Qu.: 0.000
##  Median :11.00   Median : 9.000   Median :0.0000   Median : 1.000
##  Mean   :12.23   Mean   : 9.866   Mean   :0.7869   Mean   : 1.528
##  3rd Qu.:16.00   3rd Qu.:12.000   3rd Qu.:1.0000   3rd Qu.: 2.000
##  Max.   :47.00   Max.   :33.000   Max.   :6.0000   Max.   :13.000
##    NumSulfur        NumChlorine        NumHalogen         NumRings
##  Min.   :0.0000   Min.   : 0.0000   Min.   : 0.0000   Min.   :0.000
##  1st Qu.:0.0000   1st Qu.: 0.0000   1st Qu.: 0.0000   1st Qu.:0.000
##  Median :0.0000   Median : 0.0000   Median : 0.0000   Median :1.000
##  Mean   :0.1484   Mean   : 0.5564   Mean   : 0.7009   Mean   :1.401
##  3rd Qu.:0.0000   3rd Qu.: 0.0000   3rd Qu.: 1.0000   3rd Qu.:2.000
##  Max.   :4.0000   Max.   :10.0000   Max.   :10.0000   Max.   :7.000
##    Solubility
##  Min.   :-11.620
##  1st Qu.: -3.955
##  Median : -2.490
##  Mean   : -2.738
##  3rd Qu.: -1.360
##  Max.   :  1.580
```

```
#ggpairs(sol_data) #visuals of pairplots illustrating relationship of variables
```

b.)

The ‘NA’ indicates that the corresponding predictor is linearly dependent on other predictors in the data. In layman terms, a variable with a coefficient ‘NA’ provides redundant information already captured via other variables and thus R solves this problem by dropping the redundant variable and re-fitting the regression model with the remaining predictors.