Homework 3

Ayush Kumar

2/12/2021

1. *Proof.* Notice that $\epsilon_i = y_i - \beta_0$, thus $\epsilon_i^2 = (y_i - \beta_0)^2 = y_i^2 - 2y_i\beta_0 + \beta_0$. Now to minimize the sum of squared errors we differentiate with respect to β_0 to find $\frac{\partial \epsilon_i^2}{\partial \beta_0} = -2y_i + 2\beta_0$, consequently $\sum \frac{\partial \epsilon_i^2}{\partial \beta_0} = -\sum 2y_i + 2n\beta_0$ supposing there are $n \in \mathbb{N}$ entries. Now to minimize $\sum \epsilon_i^2$ we set its derivative with respect to β_0 to zero and obtain $2\sum y_i = 2n\beta_0$ thus $\beta_0 = \frac{\sum y_i}{n} = \bar{y}$. □

2. We load the data in sat and then we fit our model and load it to *sat*.1 now we observe the following

```
sat<-read.table('http://math.uttyler.edu/nathan/data/sat.data',header=T)
sat.1<-lm(math~expend,data = sat)
summary(sat.1)
```

```
##
## Call:
## lm(formula = math ~ expend, data = sat)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -77.20 -28.91   0.93  18.78  73.78
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   569.65      24.17   23.57   <2e-16 ***
## expend        -10.31       3.99   -2.58    0.013 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 38.1 on 48 degrees of freedom
## Multiple R-squared:  0.122,  Adjusted R-squared:  0.104
## F-statistic: 6.68 on 1 and 48 DF,  p-value: 0.0129
```

Now notice that the intercept and the coefficient are 569.6527, -10.3082, respectively. This indicates that per-unit increase in expend we expect math to go down 10.31 SAT math scores. Now observe the following:

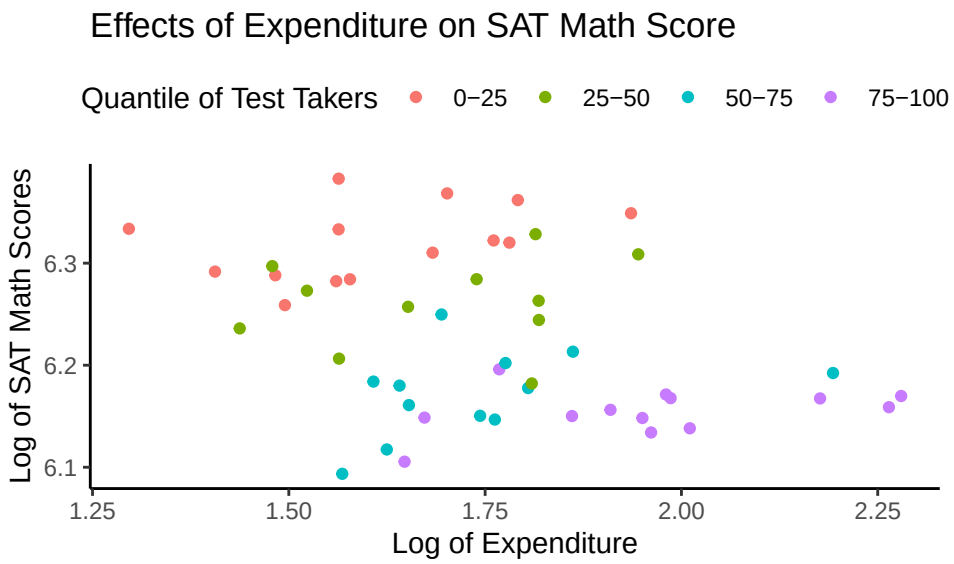
```
sat.2<-lm(math~expend+takers,data = sat)
summary(sat.2)
```

```
##
## Call:
## lm(formula = math ~ expend + takers, data = sat)
```

```
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -54.27 -10.37  -1.59   9.08  45.37
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  518.301     12.404   41.79   <2e-16 ***
## expend         7.539      2.400    3.14   0.0029 **
## takers        -1.534      0.122  -12.55   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.4 on 47 degrees of freedom
## Multiple R-squared:  0.798,  Adjusted R-squared:  0.79
## F-statistic:   93 on 2 and 47 DF,  p-value: <2e-16
```

Now notice that per-unit increase in expenditure assuming no change in exam takers results in 7.54 units increase in SAT math scores, and per-unit increase in test-takers assuming no change in expenditure to result in 1.53 units decrease in SAT math scores.

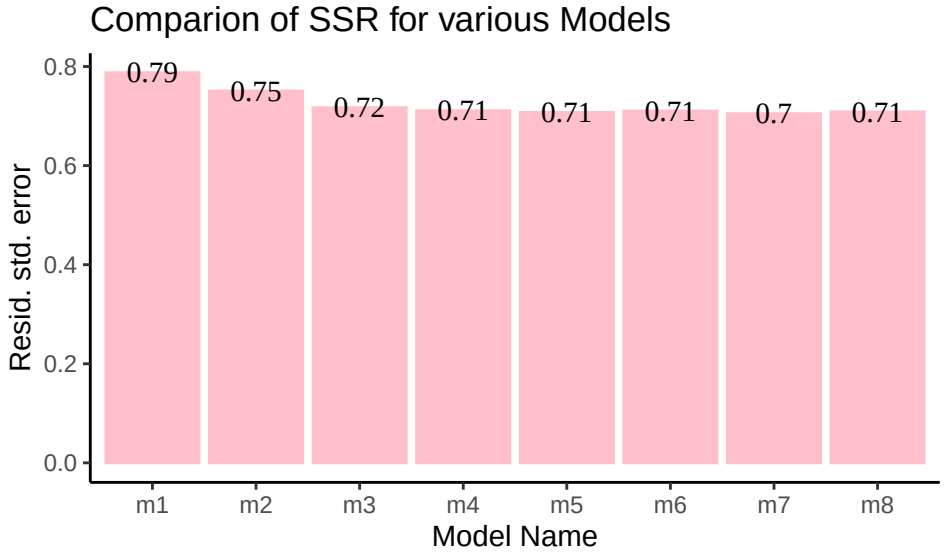
The difference in the partial effect of expenditure variable in the models suggests that not accounting for test-takers would makes our results to be underweight by the fact that states with high test-takers have lower SAT math score. As the less number of people taking the test may result in bias because of a specialized population, this effect in part may be responsible for the change in size. Below we divide the states in four categories based on the number of test-takers the figure does indicate some-evidence for our hypothesis above. Thus, accounting for test-takers the data indicates increasing expenditure is positively related with SAT math scores. Additionally we notice that states with high number of test-takers may already have high per-student expenditure thus, marginal increase may result in large changes in math SAT scores.



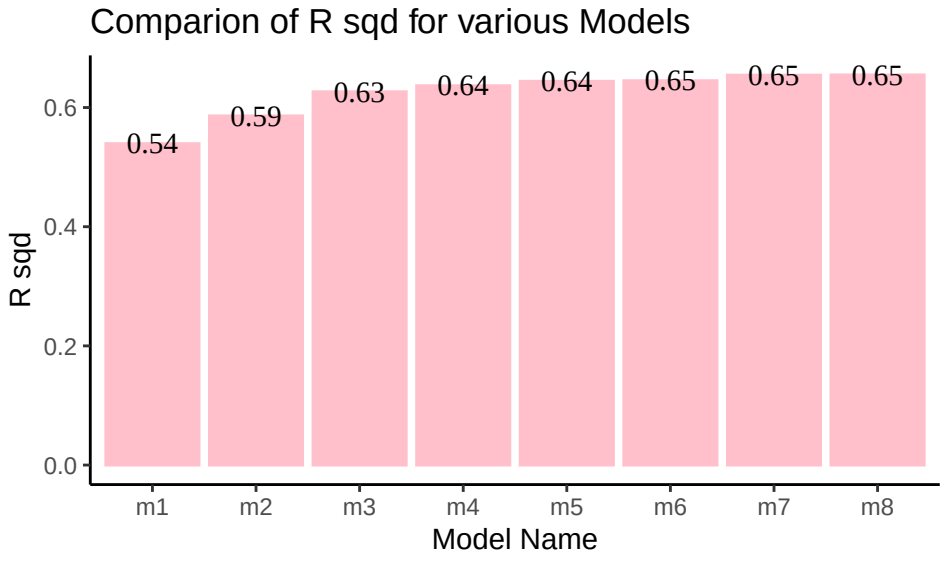
3. The following is the solution to the problem:

```
#Loading Data
pros<-read.table('http://math.uttyler.edu/nathan/data/prostate.data',header=T)

#Generating Models
```

```
m1 <- lm(lpsa ~ lcavol,data=pros)
m2 <- lm(lpsa ~ lcavol+lweight,data = pros)
m3<-lm(lpsa~lcavol+lweight+svi,data=pros)
m4<-lm(lpsa~lcavol+lweight+svi+lbph,data=pros)
m5<-lm(lpsa~lcavol+lweight+svi+lbph+age,data=pros)
m6<-lm(lpsa~lcavol+lweight+svi+lbph+age+lcp,data=pros)
m7<-lm(lpsa~lcavol+lweight+svi+lbph+age+lcp+pgg45,data=pros)
m8<-lm(lpsa~lcavol+lweight+svi+lbph+age+lcp+pgg45+gleason,data=pros)

#Generate Graphs
list<-rep(0,8)
list_2<-rep(0,8)
list_3<-rep(0,8)
for(i in 1:8) {
  list[i]<-sqrt(sum(residuals(eval(parse(text=paste0("m",i))))^2)/(97-(1+i)))
  list_2[i]<-paste0("m",i)
  list_3[i]<-summary(eval(parse(text=paste0("m",i))))$r.squared
}
tibble(
  "Resid. std. error"=list,
  "Model Name"=list_2,
  "R sqd"=list_3
)->models
ggplot(data=models)+
  geom_col(aes(x=`Model Name`,y=`Resid. std. error`),col="pink",fill="pink")+
  geom_text(aes(label=round(`Resid. std. error`,2),x=`Model Name`,y=`Resid. std. error`),family="serif")
  labs(title ="Comparion of SSR for various Models")+
  theme_classic()
```




```
ggplot(data=models)+
geom_col(aes(x=`Model Name`,y=`R sqd`),col="pink",fill="pink")+
geom_text(aes(label=round(`R sqd`,2),x=`Model Name`,y=`R sqd`)
            ,family="serif")+
  labs(title ="Comparion of R sqd for various Models")+
  theme_classic()
```



4. We construct the models as indicated by the text of the question below. Now for each of the models n's(where $n \in \mathbb{N} \wedge n < 9$).

```
# Generating the required models
m2.xpart <- lm(lweight ~ lcavol,data=pros)
m2.part <- lm(m1$residuals ~ m2.xpart$residuals)
m3.xpart<-lm(svi~lcavol+lweight,data=pros)
m3.part<-lm(m2$resid~m3.xpart$resid)
m4.xpart<-lm(lbph~svi+lcavol+lweight,data=pros)
m4.part<-lm(m3$resid~m4.xpart$resid)
m5.xpart<-lm(age~lbph+svi+lcavol+lweight,data=pros)
m5.part<-lm(m4$resid~m5.xpart$resid)
m6.xpart<-lm(lcp~lcavol+lweight+svi+lbph+age,data=pros)
m6.part<-lm(m5$resid~m6.xpart$resid)
m7.xpart<-lm(pgg45~lcp+lcavol+lweight+svi+lbph+age,data=pros)
m7.part<-lm(m6$resid~m7.xpart$resid)
m8.xpart<-lm(gleason~pgg45+lcp+lcavol+lweight+svi+lbph+age,data=pros)
m8.part<-lm(m7$resid~m8.xpart$resid)

# Observing that the respective residual is always equal to the slope of the additional variable,
#for example for m2.part the coefficient of m2.xpart$residuals is exactly equal to the

#coefficient of lweight, and so on.
for(i in 2:8){
  print(summary(eval(parse(text=paste0('m',i)))))
  print(summary(eval(parse(text=paste0('m',i,'.part')))))
}
```

```
##
## Call:
## lm(formula = lpsa ~ lcavol + lweight, data = pros)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.6196 -0.5078 -0.0209  0.5229  1.8988
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.3026     0.5690   -0.53   0.5961
## lcavol        0.6775     0.0663   10.22   <2e-16 ***
## lweight       0.5109     0.1573    3.25   0.0016 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.751 on 94 degrees of freedom
## Multiple R-squared:  0.586,  Adjusted R-squared:  0.577
## F-statistic: 66.5 on 2 and 94 DF,  p-value: <2e-16
##
##
## Call:
## lm(formula = m1$residuals ~ m2.xpart$residuals)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.6196 -0.5078 -0.0209  0.5229  1.8988
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -2.25e-16   7.58e-02    0.00   1.0000
## m2.xpart$residuals  5.11e-01   1.56e-01    3.27   0.0015 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.747 on 95 degrees of freedom
## Multiple R-squared:  0.101,  Adjusted R-squared:  0.0915
## F-statistic: 10.7 on 1 and 95 DF,  p-value: 0.00152
##
##
## Call:
## lm(formula = lpsa ~ lcavol + lweight + svi, data = pros)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.7296 -0.4576  0.0281  0.4640  1.5701
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.2681     0.5435   -0.49    0.623
## lcavol        0.5516     0.0747    7.39  6.3e-11 ***
## lweight       0.5085     0.1502    3.39    0.001 **
## svi           0.6662     0.2098    3.18    0.002 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.717 on 93 degrees of freedom
## Multiple R-squared:  0.626,  Adjusted R-squared:  0.614
## F-statistic:   52 on 3 and 93 DF,  p-value: <2e-16
##
##
```

```
## Call:
## lm(formula = m2$resid ~ m3.xpart$resid)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.7296 -0.4576  0.0281  0.4640  1.5701
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    2.25e-16   7.20e-02    0.00   1.0000
## m3.xpart$resid 6.66e-01   2.08e-01    3.21   0.0018 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.709 on 95 degrees of freedom
## Multiple R-squared:  0.0978, Adjusted R-squared:  0.0883
## F-statistic: 10.3 on 1 and 95 DF,  p-value: 0.00181
##
##
## Call:
## lm(formula = lpsa ~ lcavol + lweight + svi + lbph, data = pros)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.8265 -0.4227  0.0436  0.4704  1.4853
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.1455     0.5975    0.24    0.808
## lcavol        0.5496     0.0741    7.42  5.6e-11 ***
## lweight       0.3909     0.1660    2.35    0.021 *
## svi           0.7117     0.2100    3.39    0.001 **
## lbph          0.0901     0.0562    1.60    0.112
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.711 on 92 degrees of freedom
## Multiple R-squared:  0.637,  Adjusted R-squared:  0.621
## F-statistic: 40.3 on 4 and 92 DF,  p-value: <2e-16
##
##
## Call:
## lm(formula = m3$resid ~ m4.xpart$resid)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.8265 -0.4227  0.0436  0.4704  1.4853
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.60e-17   7.10e-02    0.00     1.00
## m4.xpart$resid 9.01e-02   5.53e-02    1.63     0.11
##
## Residual standard error: 0.7 on 95 degrees of freedom
```

```
## Multiple R-squared:  0.0272, Adjusted R-squared:  0.017
## F-statistic: 2.66 on 1 and 95 DF,  p-value: 0.106
##
##
## Call:
## lm(formula = lpsa ~ lcavol + lweight + svi + lbph + age, data = pros)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.8350 -0.3940  0.0041  0.4634  1.5789
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.9510     0.8317    1.14  0.25588
## lcavol        0.5656     0.0746    7.58  2.8e-11 ***
## lweight       0.4237     0.1669    2.54  0.01281 *
## svi           0.7210     0.2090    3.45  0.00085 ***
## lbph          0.1118     0.0581    1.93  0.05716 .
## age          -0.0149     0.0108   -1.38  0.16953
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.707 on 91 degrees of freedom
## Multiple R-squared:  0.644,  Adjusted R-squared:  0.625
## F-statistic: 32.9 on 5 and 91 DF,  p-value: <2e-16
##
##
## Call:
## lm(formula = m4$resid ~ m5.xpart$resid)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.8350 -0.3940  0.0041  0.4634  1.5789
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -2.24e-16   7.03e-02    0.00     1.00
## m5.xpart$resid -1.49e-02   1.05e-02   -1.41     0.16
##
## Residual standard error: 0.692 on 95 degrees of freedom
## Multiple R-squared:  0.0206, Adjusted R-squared:  0.0103
## F-statistic:    2 on 1 and 95 DF,  p-value: 0.16
##
##
## Call:
## lm(formula = lpsa ~ lcavol + lweight + svi + lbph + age + lcp,
##     data = pros)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.8285 -0.4074  0.0169  0.4716  1.5904
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)   0.9349     0.8358    1.12   0.2663
## lcavol        0.5876     0.0866    6.78  1.2e-09 ***
## lweight       0.4181     0.1679    2.49   0.0146 *
## svi           0.7826     0.2426    3.23   0.0018 **
## lbph          0.1138     0.0584    1.95   0.0545 .
## age          -0.0151     0.0108   -1.40   0.1655
## lcp          -0.0412     0.0814   -0.51   0.6139
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.71 on 90 degrees of freedom
## Multiple R-squared:  0.645,  Adjusted R-squared:  0.621
## F-statistic: 27.3 on 6 and 90 DF,  p-value: <2e-16
##
##
## Call:
## lm(formula = m5$resid ~ m6.xpart$resid)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.8285 -0.4074  0.0169  0.4716  1.5904
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     2.13e-17   7.02e-02    0.00      1.0
## m6.xpart$resid -4.12e-02   7.92e-02   -0.52      0.6
##
## Residual standard error: 0.691 on 95 degrees of freedom
## Multiple R-squared:  0.00284,    Adjusted R-squared:  -0.00766
## F-statistic: 0.271 on 1 and 95 DF,  p-value: 0.604
##
##
## Call:
## lm(formula = lpsa ~ lcavol + lweight + svi + lbph + age + lcp +
##     pgg45, data = pros)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.7312 -0.3814 -0.0173  0.4336  1.6351
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.95393    0.82944    1.15   0.2532
## lcavol       0.59161    0.08600    6.88  8.1e-10 ***
## lweight      0.44829    0.16777    2.67   0.0090 **
## svi          0.75773    0.24128    3.14   0.0023 **
## lbph         0.10767    0.05811    1.85   0.0672 .
## age         -0.01934    0.01107   -1.75   0.0840 .
## lcp         -0.10448    0.09048   -1.15   0.2513
## pgg45        0.00532    0.00343    1.55   0.1249
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.705 on 89 degrees of freedom
```

```
## Multiple R-squared:  0.654,  Adjusted R-squared:  0.627
## F-statistic: 24.1 on 7 and 89 DF,  p-value: <2e-16
##
##
## Call:
## lm(formula = m6$resid ~ m7.xpart$resid)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.7312 -0.3814 -0.0173  0.4336  1.6351
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -1.31e-16   6.93e-02     0.0     1.00
## m7.xpart$resid  5.32e-03   3.32e-03     1.6     0.11
##
## Residual standard error: 0.682 on 95 degrees of freedom
## Multiple R-squared:  0.0263, Adjusted R-squared:  0.016
## F-statistic: 2.56 on 1 and 95 DF,  p-value: 0.113
##
##
## Call:
## lm(formula = lpsa ~ lcavol + lweight + svi + lbph + age + lcp +
##     pgg45 + gleason, data = pros)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -1.733 -0.371 -0.017  0.414  1.638
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.66934    1.29639    0.52   0.6069
## lcavol       0.58702    0.08792    6.68  2.1e-09 ***
## lweight      0.45447    0.17001    2.67   0.0090 **
## svi          0.76616    0.24431    3.14   0.0023 **
## lbph         0.10705    0.05845    1.83   0.0704 .
## age         -0.01964    0.01117   -1.76   0.0823 .
## lcp         -0.10547    0.09101   -1.16   0.2496
## pgg45        0.00453    0.00442    1.02   0.3089
## gleason      0.04514    0.15746    0.29   0.7750
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.708 on 88 degrees of freedom
## Multiple R-squared:  0.655,  Adjusted R-squared:  0.623
## F-statistic: 20.9 on 8 and 88 DF,  p-value: <2e-16
##
##
## Call:
## lm(formula = m7$resid ~ m8.xpart$resid)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -1.733 -0.371 -0.017  0.414  1.638
```

```
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    6.75e-17   6.92e-02     0.0     1.00
## m8.xpart$resid 4.51e-02   1.52e-01     0.3     0.77
##
## Residual standard error: 0.682 on 95 degrees of freedom
## Multiple R-squared:  0.000933,   Adjusted R-squared:  -0.00958
## F-statistic: 0.0887 on 1 and 95 DF,  p-value: 0.766
```

5. The following is the code required for the exercise:

   a.

```
set.seed(1)
x<-rnorm(100,0,1)
```

   b.

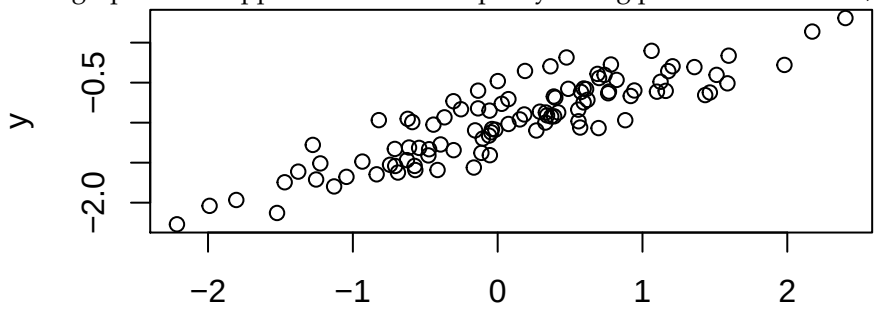
```
eps<-rnorm(100,0,0.25)
```

   c. The length of $y = 100$, while the value of $\beta_0 = -1$, and $\beta_1 = 0.5$ in the model.

```
y<-(-1+0.5*x+eps)
length(y)
```

```
## [1] 100
```

   d. From the graph below appears to indicate a pretty strong positive correlation, with intercept close to -1.



   e. We notice $\hat{\beta}_0 = -1.0094$, while $\hat{\beta}_1 = 0.4997$, while $\beta_0 = -1$, and $\beta_1 = 0.5$, they are pretty much the same value.

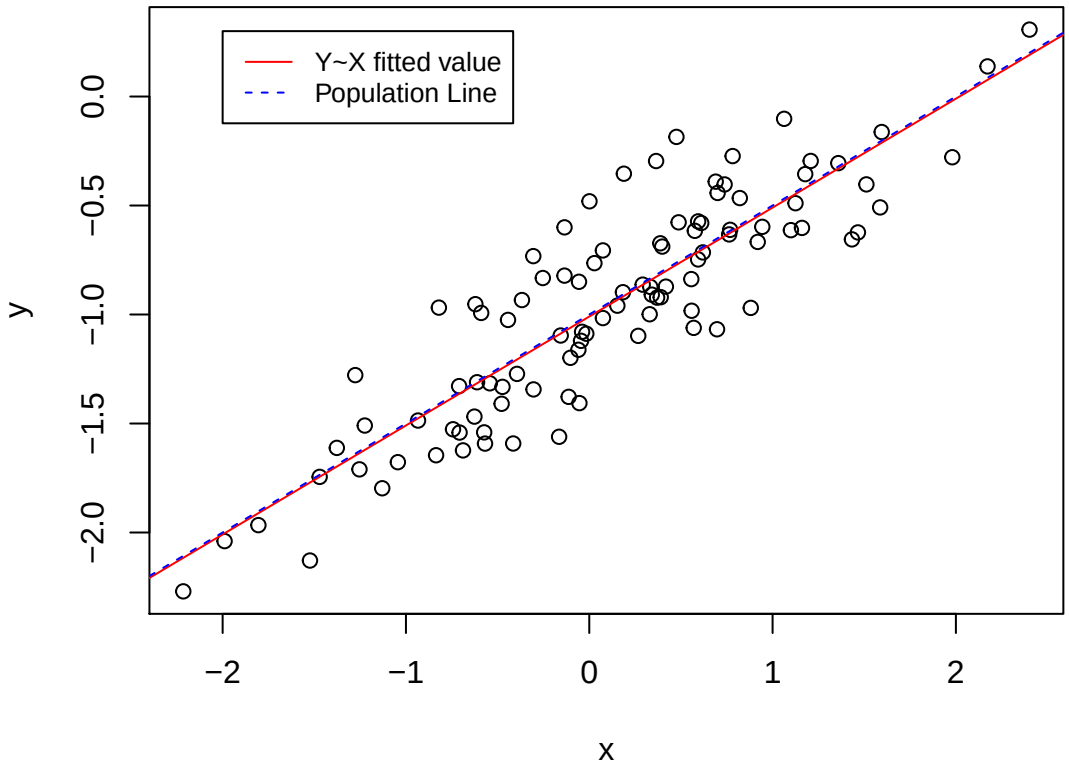
```
summary(lm(y~x))
```

```
##
## Call:
## lm(formula = y ~ x)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.4692 -0.1534 -0.0349  0.1349  0.5865
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.0094     0.0242   -41.6   <2e-16 ***
## x             0.4997     0.0269    18.6   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.241 on 98 degrees of freedom
## Multiple R-squared:  0.778,  Adjusted R-squared:  0.776
## F-statistic:  344 on 1 and 98 DF,  p-value: <2e-16
```

f. Below is the graph describing the difference between population model, and the sample least square model.

```
knitr::opts_chunk$set(fig.width=6, fig.height=5)
plot(x,y)
abline(lm(y~x),col="red")
abline(a=-1,b=0.5,col="blue",lty=2)
legend(-2, 0.3, legend=c("Y~X fitted value", "Population Line"),
       col=c("red", "blue"), lty=1:2, cex=0.8)
```



g. As discussed in class, we note though marginally, it is true that the sum of square residuals is lower for the quadratic model. However we note that the variable x^2 in itself is non-significant at 5 confidence level.

```
sum(residuals(lm(y~x+I(x^2)))^2)
```

```
## [1] 5.564
```

```
sum(residuals(lm(y~x))^2)
```

```
## [1] 5.677
```

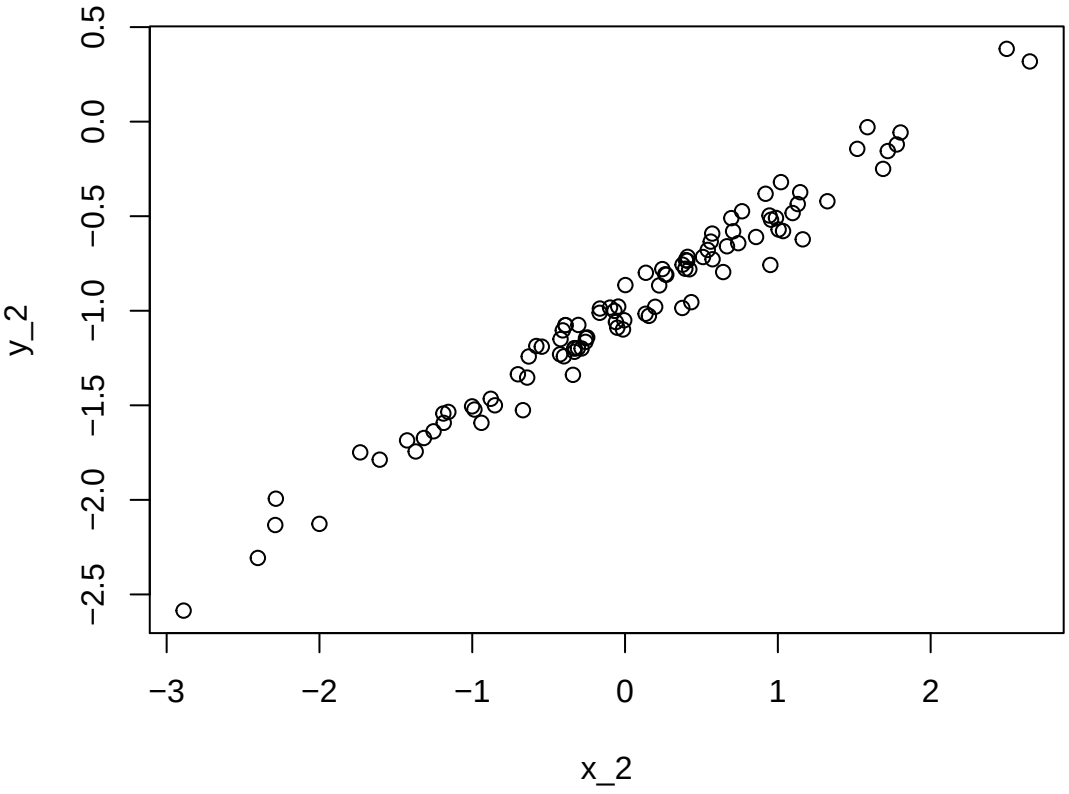
h. Below is the for ϵ with a standard deviation of 0.09.

```
x_2<-rnorm(100,0,1)

eps_2<-rnorm(100,0,0.09)

y_2<-(-1+0.5*x_2+eps_2)

length(y_2)
```

```
## [1] 100
```

We notice that $\beta_0 = -1$, and $\beta_1 = 0.5$, and the figure below shows that the sample's correlation is significantly lower than what we had above.

```
plot(x_2,y_2)
```

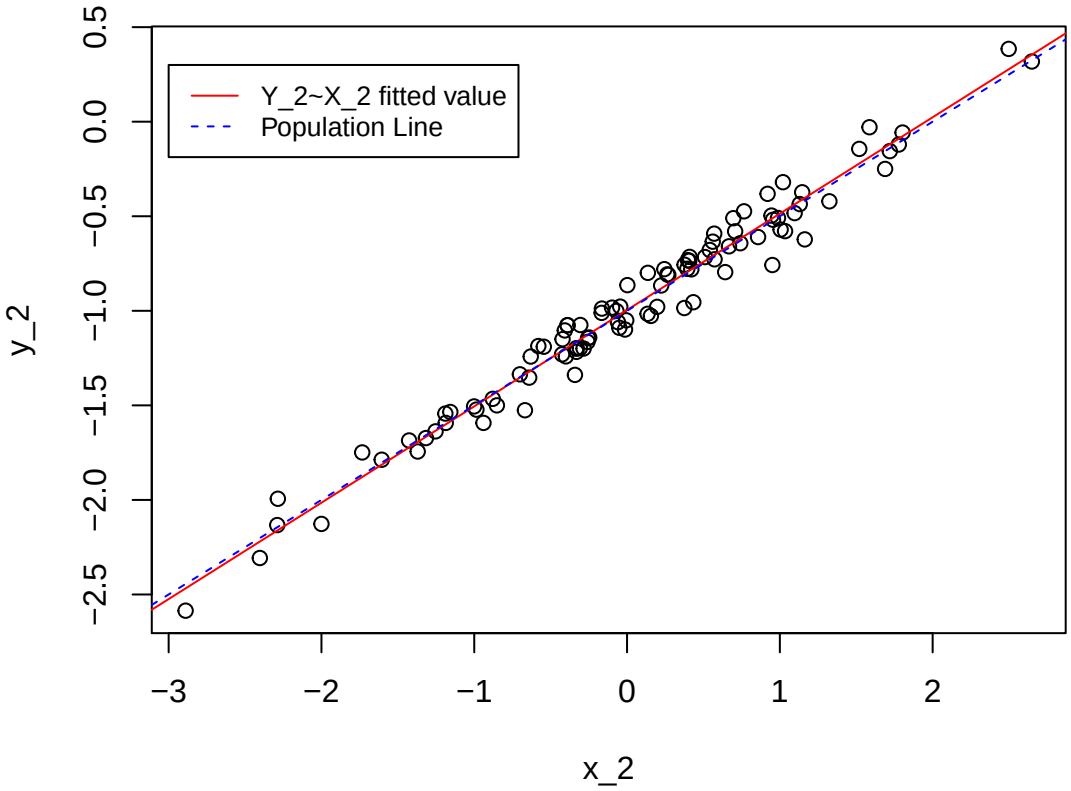


```
summary(lm(y_2~x_2))
```

```
##
## Call:
## lm(formula = y_2 ~ x_2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.24676 -0.05053 -0.00157  0.06118  0.16636
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.99564    0.00892  -111.6   <2e-16 ***
## x_2          0.50956    0.00866    58.8   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0892 on 98 degrees of freedom
## Multiple R-squared:  0.972,  Adjusted R-squared:  0.972
## F-statistic: 3.46e+03 on 1 and 98 DF,  p-value: <2e-16
```

Notice that $\hat{\beta}_0$=-0.9956, while $\hat{\beta}_1$=0.5096, we note that $\hat{\beta}_0$, and $\hat{\beta}_1$ are closer to population parameters when compared to previous model.Notice that the graph below also indicates that the best-fit line appears to be pretty much exactly equal to the population model.

```
plot(x_2,y_2)
abline(lm(y_2~x_2),col="red")
abline(a=-1,b=0.5,col="blue",lty=2)
legend(-3, 0.3, legend=c("Y_2~X_2 fitted value", "Population Line"),
       col=c("red", "blue"), lty=1:2, cex=0.8)
```



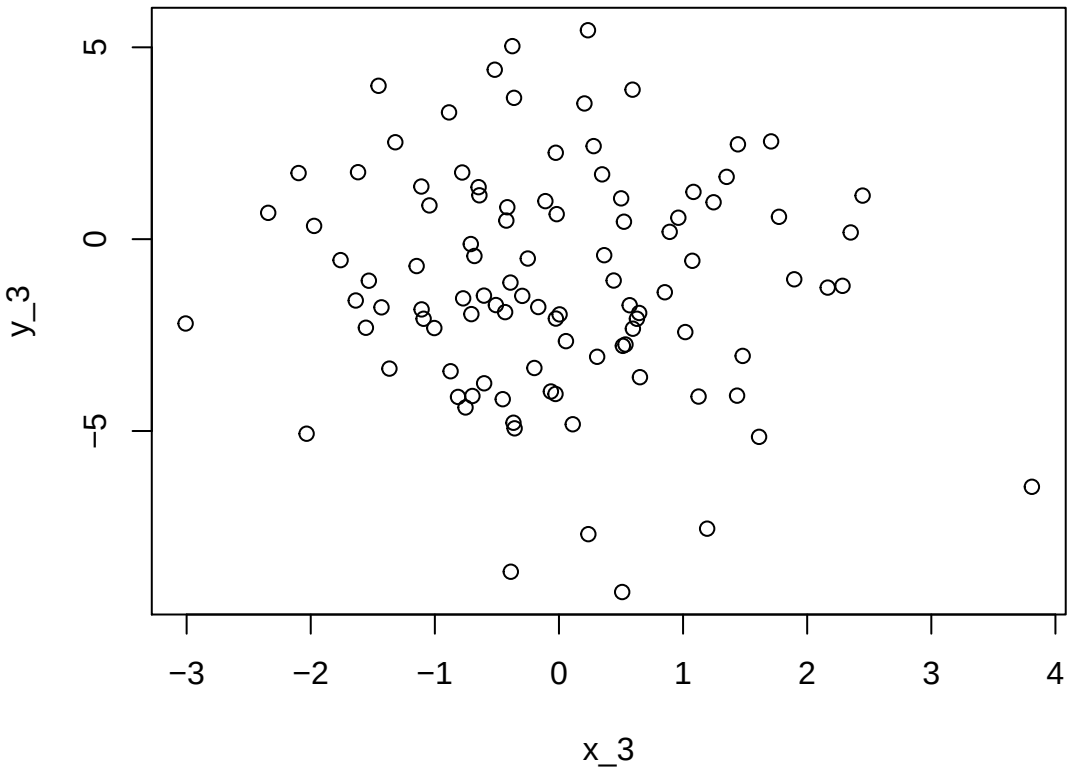
i. We now consider the case where the standard deviation of ϵ to 3.

```
x_3<-rnorm(100,0,1)

eps_3<-rnorm(100,0,3)

y_3<-(-1+0.5*x_2+eps_3)

length(y_3)
```

```
## [1] 100
```

We notice that $\beta_0 = -1$, and $\beta_1 = 0.5$, and the figure below shows that the sample's correlation is significantly lower than what we had above.

```
plot(x_3,y_3)
```



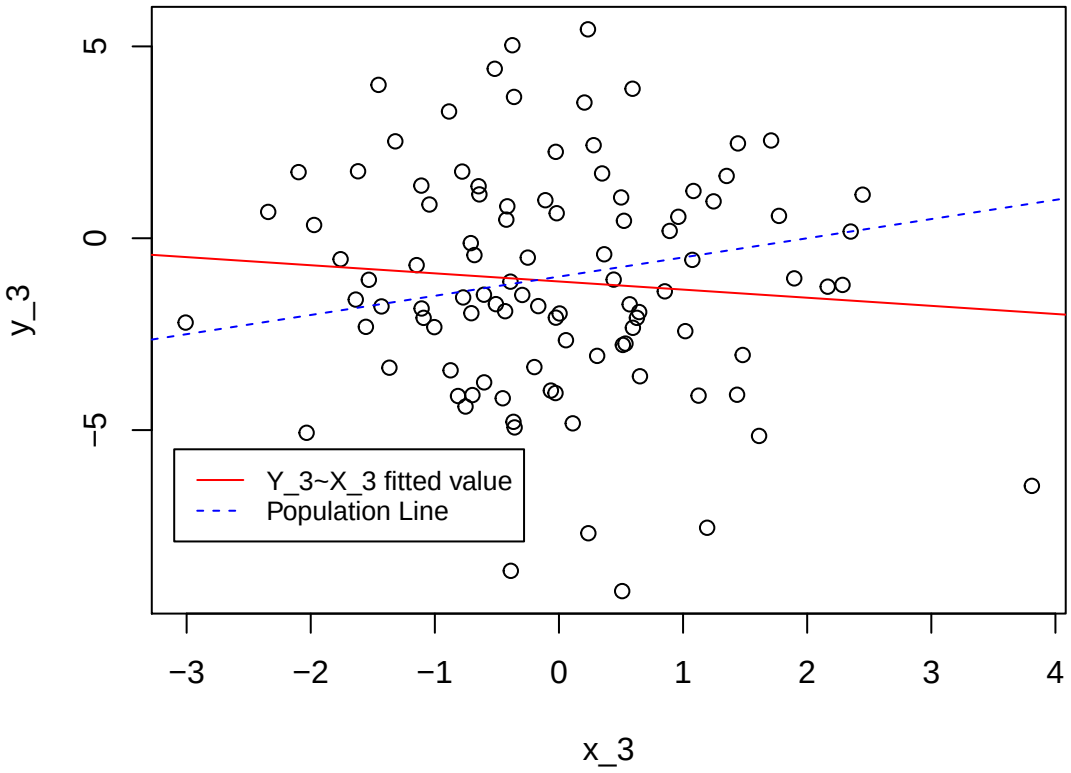

```
summary(lm(y_3~x_3))
```

```
##
## Call:
## lm(formula = y_3 ~ x_3)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.962 -1.629 -0.184  2.105  6.621
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -1.127      0.293   -3.85  0.00021 ***
## x_3           -0.212      0.252   -0.84  0.40201
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.93 on 98 degrees of freedom
## Multiple R-squared:  0.00718,    Adjusted R-squared:  -0.00295
## F-statistic: 0.708 on 1 and 98 DF,  p-value: 0.402
```

Notice that $\hat{\beta}_0$=-1.127, while $\hat{\beta}_1$=-0.2117, we note that both the slope and the intercept in this model to be both significantly different from the population model, and to be insignificantly different from 0. Notice that the graph below also indicates that the best-fit line appears to be largely different from the population model.

```
plot(x_3,y_3)
abline(lm(y_3~x_3),col="red")
abline(a=-1,b=0.5,col="blue",lty=2)
```

```
legend(-3.1, -5.5, legend=c("Y_3~X_3 fitted value", "Population Line"),
       col=c("red", "blue"), lty=1:2, cex=0.8)
```



j. Now we compare the confidence intervals for these models, which are:

```
confint(lm(y~x))
```

```
##                2.5 %  97.5 %
## (Intercept) -1.0575 -0.9613
## x            0.4463  0.5532
```

```
confint(lm(y_2~x_2))
```

```
##                2.5 %  97.5 %
## (Intercept) -1.0133 -0.9779
## x_2          0.4924  0.5268
```

```
confint(lm(y_3~x_3))
```

```
##                2.5 %  97.5 %
## (Intercept) -1.7082 -0.5458
## x_3         -0.7109  0.2874
```

Notice that the variable $\sigma_{\hat{\beta}} = (X'X)^{-1}\sigma$ consequently as σ^2 increases we expect $\sigma^2_{\hat{\beta}}$ to increase and vice versa. We note the σ^2 for the respective models are 0.0579, 0.0079, and 8.5687. The SSR is largest for models 3, followed by model 2, and model 1, this can be attributed to the choice of variance for our ϵ's above.

6. We begin with loading data.

a. We then define x,y, and $\hat{\beta}$.

```
sat<-read.table('http://math.uttyler.edu/nathan/data/sat.data',header=T)
x<-matrix(c(rep(1,50),sat$expend,sat$takers),ncol = 3)
y<-matrix(sat$math)
beta_hat<-solve(t(x)%*%x)%*%t(x)%*%y
beta_hat
```

```
##          [,1]
## [1,] 518.301
## [2,]   7.539
## [3,]  -1.534
```

b. We then define $\hat{\epsilon}$, and $\hat{\sigma}$.

```
epsilon_hat<-y-x%*%beta_hat
sigma_hat<-sqrt(sum(epsilon_hat^2))/(47)
sigma_hat
```

```
## [1] 2.69
```

c. We then define SSE, SSM, and SST.

```
sse<-t(epsilon_hat)%*%epsilon_hat
ssm<-t(y-x%*%beta_hat)%*%(y-x%*%beta_hat)
sst<-ssm+sse
print(c(sse,ssm,sst))
```

```
## [1] 15983 15983 31967
```

d. We compute V(B), and then find standard errors for betas.

```
XtXinv<-solve(t(x)%*%x)
beta_0_sd_error<-sqrt(XtXinv[1,1]) * sqrt(t(epsilon_hat) %*% epsilon_hat) /sqrt(47)
beta_1_sd_error<-sqrt(XtXinv[2,2]) * sqrt(t(epsilon_hat) %*% epsilon_hat) /sqrt(47)
beta_2_sd_error<-sqrt(XtXinv[3,3]) * sqrt(t(epsilon_hat) %*% epsilon_hat) /sqrt(47)
print(c(beta_0_sd_error,beta_1_sd_error,beta_2_sd_error))
```

```
## [1] 12.4040  2.3999  0.1222
```

e. We find the t-statistic.

```
t_stat_beta_0<-beta_hat[1]/(sqrt(XtXinv[1,1]) * sqrt(t(epsilon_hat) %*% epsilon_hat) /sqrt(47))
t_stat_beta_1<-beta_hat[2]/(sqrt(XtXinv[2,2]) * sqrt(t(epsilon_hat) %*% epsilon_hat) /sqrt(47))
t_stat_beta_2<-beta_hat[3]/(sqrt(XtXinv[3,3]) * sqrt(t(epsilon_hat) %*% epsilon_hat) /sqrt(47))
print(c(t_stat_beta_0,t_stat_beta_1,t_stat_beta_2))
```

```
## [1]  41.785   3.142 -12.549
```

f. We find the corresponding p-values.

```
round(pt(t_stat_beta_0,df=49,lower.tail = F)*2,6)
```

```
##      [,1]
## [1,]    0
```

```
round(pt(t_stat_beta_1,df=49,lower.tail = F)*2,6)
```

```
##          [,1]
## [1,] 0.002849
```

```
round(pt(abs(t_stat_beta_2),df=49,lower.tail = F)*2,6)
```

```
##      [,1]
## [1,]    0
```