Homework 4, MATH455: Due Monday, 03/05/2018

Your Name: (replace this)

March 9, 2018

Instructions: The homework assignment editing this LaTeX document. Download the LaTeX source from the class web page and study it to learn more about LaTeX. Replace the text with appropriate information. Run "pdflatex" on this document.

You will submit this assignment in two parts:

1. Print out the PDF file and bring it to class, and
2. Send an e-mail to:

   gang@math.binghamton.edu

   before class on the due date with two attachments:

   - The LaTeX source file, and
   - The generated PDF document.

Please complete the following:

1. Read chapter 3 and finish questions 3.2, 3.4 (on pages 49-50) in this chapter.

```
data(cheddar)
cheeseMod = lm(taste~Acetic+H2S+Lactic,cheddar)
summary(cheeseMod)
Call:
lm(formula = taste ~ Acetic + H2S + Lactic, data = cheddar)

Residuals:
Min      1Q  Median      3Q     Max
-17.390  -6.612  -1.009   4.908  25.449

Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept) -28.8768    19.7354  -1.463  0.15540
Acetic        0.3277     4.4598   0.073  0.94198
H2S           3.9118     1.2484   3.133  0.00425 **
Lactic       19.6705     8.6291   2.280  0.03108 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 10.13 on 26 degrees of freedom
Multiple R-squared:  0.6518,Adjusted R-squared:  0.6116
F-statistic: 16.22 on 3 and 26 DF,  p-value: 3.81e-06
```

Accordingly, H2S and Lactic are the two parameters significant at the 5 percent level.
After applying the exponential function to both Acetic and H2S, we get the following results

```
> cheeseModP = lm(taste~exp(Acetic)+exp(H2S)+Lactic,cheddar)
> summary(cheeseModP)

Call:
lm(formula = taste ~ exp(Acetic) + exp(H2S) + Lactic, data = cheddar)

Residuals:
Min      1Q  Median      3Q     Max
-16.209  -7.266  -1.651   7.385  26.335

Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.897e+01  1.127e+01  -1.684   0.1042
exp(Acetic)  1.891e-02  1.562e-02   1.210   0.2371
exp(H2S)     7.668e-04  4.188e-04   1.831   0.0786 .
```

```
Lactic       2.501e+01  9.062e+00   2.760   0.0105 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 11.19 on 26 degrees of freedom
Multiple R-squared:  0.5754,Adjusted R-squared:  0.5264
F-statistic: 11.75 on 3 and 26 DF,  p-value: 4.746e-05
```

Thus, only Lactic remains statistically significant.
We can not operate the f-test on these data sets

```
anova(cheeseMod,cheeseModP)Analysis of Variance Table

Model 1: taste ~ Acetic + H2S + Lactic
Model 2: taste ~ exp(Acetic) + exp(H2S) + Lactic
Res.Df    RSS Df Sum of Sq F Pr(>F)
1     26 2668.4
2     26 3253.6  0    -585.2
```

This is because our degrees of freedom are the same, and thus we are dividing by zero and will be unable to compute anything.
According to our summary, H2S=3.9118, thus for every increase of .01, we increase taste by .039 approximately.

```
> log(10)
[1] 2.302585
> log(10.01)
[1] 2.303585
> log(10.01)/log(10)
[1] 1.000434
```

So about a .04 percent increase given an additive of .01 on the log scale.

```
> scores = lm(total~expend+ratio+salary,sat)
> scoresSZ = lm(total~expend+ratio,sat)
> scoresNull=lm(total~1,sat)
> anova(scores,scoresSZ)
Analysis of Variance Table

Model 1: total ~ expend + ratio + salary
Model 2: total ~ expend + ratio
Res.Df    RSS Df Sum of Sq      F  Pr(>F)
1     46 216812
2     47 233443 -1    -16631 3.5285 0.06667 .
---
```

```
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
> anova(scores,scoresNull)
Analysis of Variance Table

Model 1: total ~ expend + ratio + salary
Model 2: total ~ 1
Res.Df    RSS Df Sum of Sq      F  Pr(>F)
1     46 216812
2     49 274308 -3    -57496 4.0662 0.01209 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

Accordingly, it appears that with $H_0 : \beta_{salary} = 0$, we can not reject that null hypothesis and salary may not be indicative. Meanwhile, all three parameters do seem to have some indication on total score.

```
> anova(tscores,scores)
Analysis of Variance Table

Model 1: total ~ expend + ratio + salary + takers
Model 2: total ~ expend + ratio + salary
Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1     45  48124
2     46 216812 -1   -168688 157.74 2.607e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
> summary(tscores)

Call:
lm(formula = total ~ expend + ratio + salary + takers, data = sat)

Residuals:
Min      1Q  Median      3Q     Max
-90.531 -20.855  -1.746  15.979  66.571

Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept) 1045.9715    52.8698  19.784  < 2e-16 ***
expend         4.4626    10.5465   0.423    0.674
ratio         -3.6242     3.2154  -1.127    0.266
salary         1.6379     2.3872   0.686    0.496
takers        -2.9045     0.2313 -12.559 2.61e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

```
Residual standard error: 32.7 on 45 degrees of freedom
Multiple R-squared:  0.8246,Adjusted R-squared:  0.809
F-statistic: 52.88 on 4 and 45 DF,  p-value: < 2.2e-16
```

as we can see, the t-value demonstrated in summary is the same as the F value provided by anova, which demonstrates their equivalence

2. Read chapter 4 and finish questions 4.1, 4.5 (on pages 56-58) in this chapter.

```
> proModel = lm(lpsa~lcavol+lweight+age+lbph+svi+lcp+gleason+pgg45,pros
> summary(proModel)

Call:
lm(formula = lpsa ~ lcavol + lweight + age + lbph + svi + lcp +
gleason + pgg45, data = prostate)

Residuals:
Min      1Q  Median      3Q     Max
-1.7331 -0.3713 -0.0170  0.4141  1.6381

Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.669337   1.296387   0.516  0.60693
lcavol       0.587022   0.087920   6.677 2.11e-09 ***
lweight      0.454467   0.170012   2.673  0.00896 **
age         -0.019637   0.011173  -1.758  0.08229 .
lbph         0.107054   0.058449   1.832  0.07040 .
svi          0.766157   0.244309   3.136  0.00233 **
lcp         -0.105474   0.091013  -1.159  0.24964
gleason      0.045142   0.157465   0.287  0.77503
pgg45        0.004525   0.004421   1.024  0.30886
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.7084 on 88 degrees of freedom
Multiple R-squared:  0.6548,Adjusted R-squared:  0.6234
F-statistic: 20.86 on 8 and 88 DF,  p-value: < 2.2e-16
```

So, for a)

```
> predict(proModel,new=data.frame(lcavol=1.44692,lweight=3.62301,age=65
fit       lwr      upr
1 2.389053 0.9646584 3.813447
```

and for b) we get

```
> predict(proModel,new=data.frame(lcavol=1.44692,lweight=3.62301,age=20
fit      lwr      upr
1 3.272726 1.538744 5.006707
```

This is best explained by the median of this data taking place in the mid 60s,

```
> median(prostate$age)
[1] 65
```

thus the prediction interval is narrower for the 65 year old as its closer to our data.

```
> newModel = lm(lpsa~lcavol+lweight+svi,prostate)
> predict(newModel,new=data.frame(lcavol=1.44692,lweight=3.62301,age=65
fit       lwr      upr
1 2.372534 0.9383436 3.806724
> predict(newModel,new=data.frame(lcavol=1.44692,lweight=3.62301,age=20
fit       lwr      upr
1 2.372534 0.9383436 3.806724
```

I would choose the simpler model as our confidence interval does not change very much and we have a simpler model.

```
> summary(fatModel)

Call:
lm(formula = brozek ~ age + weight + height + abdom, data = fat)

Residuals:
Min       1Q   Median       3Q      Max
-11.5105  -2.9346   0.0087   2.8942   9.4179

Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept) -32.769636   6.541902  -5.009 1.04e-06 ***
age          -0.007051   0.024342  -0.290    0.772
weight       -0.123722   0.025046  -4.940 1.44e-06 ***
height       -0.116694   0.082727  -1.411    0.160
abdom         0.889704   0.067267  13.226  < 2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 4.126 on 247 degrees of freedom
Multiple R-squared:  0.7211,Adjusted R-squared:  0.7166
F-statistic: 159.7 on 4 and 247 DF,  p-value: < 2.2e-16
```

```
> summary(lm(brozek~.,fat))

Call:
lm(formula = brozek ~ ., data = fat)

Residuals:
Min       1Q   Median       3Q      Max
-1.11191 -0.04847  0.00277  0.04625  1.47542

Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept) 12.1524013  4.1718589   2.913  0.00393 **
siri          0.8884085  0.0111341  79.792  < 2e-16 ***
density      -9.8456305  3.7471770  -2.627  0.00917 **
age          -0.0005268  0.0012935  -0.407  0.68421
weight        0.0084855  0.0036200   2.344  0.01991 *
height       -0.0005459  0.0044439  -0.123  0.90234
adipos       -0.0153248  0.0124778  -1.228  0.22062
free         -0.0097388  0.0044270  -2.200  0.02880 *
neck          0.0005002  0.0094279   0.053  0.95773
chest         0.0021454  0.0043013   0.499  0.61840
abdom         0.0014464  0.0044217   0.327  0.74388
hip          -0.0044514  0.0058941  -0.755  0.45087
thigh         0.0156926  0.0059507   2.637  0.00892 **
knee         -0.0252126  0.0098531  -2.559  0.01113 *
ankle         0.0027790  0.0089580   0.310  0.75667
biceps       -0.0147134  0.0069201  -2.126  0.03454 *
forearm       0.0149983  0.0080832   1.855  0.06478 .
wrist         0.0326518  0.0218000   1.498  0.13554
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.1706 on 234 degrees of freedom
Multiple R-squared:  0.9995,Adjusted R-squared:  0.9995
F-statistic: 3.046e+04 on 17 and 234 DF,  p-value: < 2.2e-16
```

Looking at the two, the simpler model should only be used as a very rough model for those who want a very ballparked prediction interval. It is not a terrible model but it does lack a lot of the confidence the fuller model has

```
> vals = model.matrix(fatModel)
> medians = apply(vals,2,median)
> predict(fatModel,new=data.frame(t(medians)),interval = "prediction")
fit      lwr      upr
```

```
1 17.84028 9.696631 25.98392
```

Thus, the predicted interval is actually not that far off from the full model, despite the lower R squared. This is still just about as trustworthy.

```
> smallData = data.frame(fat$age,fat$weight,fat$height,fat$abdom)
> smallData[25:50,]
fat.age fat.weight fat.height fat.abdom
25      28     151.25      67.75      76.3
26      27     159.25      71.50      79.7
27      34     131.50      67.50      74.6
28      31     148.00      67.50      88.7
29      27     133.25      64.75      73.9
30      29     160.75      69.00      83.5
31      32     182.00      73.75      88.7
32      29     160.25      71.25      84.5
33      27     168.00      71.25      79.1
34      41     218.50      71.00     100.5
35      41     247.25      73.50     115.6
36      49     191.75      65.00     113.1
37      40     202.25      70.00     100.9
38      50     196.75      68.25      98.8
39      46     363.15      72.25     148.1
40      50     203.00      67.00     108.1
41      45     262.75      68.75     126.2
42      44     205.00      29.50     104.3
43      48     217.00      70.00     111.2
44      41     212.00      71.50     104.3
45      39     125.25      68.00      76.0
46      43     164.25      73.25      81.5
47      40     133.50      67.50      73.7
48      39     148.50      71.25      79.5
49      45     135.75      68.50      83.4
50      47     127.50      66.75      70.4
```

To me, it appears cases 39 and 41 may be the outliers as we have 363lb and 262 lb, both of which have a very high abdominal measure.

```
> newfatModel = lm(brozek~age+weight+height+abdom,data=fat[-c(39,41),])
> vals = model.matrix(newfatModel)
> medians = apply(vals,2,median)
> predict(newfatModel,new=data.frame(t(medians)),interval = "prediction
fit       lwr       upr
1 17.89765 9.925792 25.86951
```

Thus there is a minor narrowing inside the interval, but it isnt that significant

3. Read chapter 7.3 and finish question 7.8 (on page 111) in this chapter.

```
> vif(fat)
brozek         siri      density          age       weight       height
2214.080613 2112.104021   45.153243    2.293310   99.902769    2.285116
adipos         free         neck        chest        abdom          hip
17.985147   57.342442    4.529877   11.352040   19.614474   15.413920
thigh         knee        ankle       biceps      forearm        wrist
8.667135    5.006429    1.988803    3.842645    2.334644    3.606408
> fullFatModel = lm(brozek~.,fat)
> fatMatrix = model.matrix(fullFatModel)[,-1]
#NOTE: this -1 is to remove intercept value of 1
> eVals = eigen(t(fatMatrix)%*%fatMatrix)
> sqrt(eVals$val[1]/eVals$val)
[1]     1.00000    19.37610    21.16725    36.01034
85.84167    94.71785
[7]   121.84306   160.76414   196.59630   212.67150
234.59368   245.36144
[13]   280.69168   300.69906   406.32695   643.19610
18326.65827
```

compared to this

```
> vif(fat[-c(39,42)])
brozek         siri      density          age       weight       height
2214.080613 2112.104021   45.153243    2.293310   99.902769    2.285116
adipos         free         neck        chest        abdom          hip
17.985147   57.342442    4.529877   11.352040   19.614474   15.413920
thigh         knee        ankle       biceps      forearm        wrist
8.667135    5.006429    1.988803    3.842645    2.334644    3.606408
> fullFatModel = lm(brozek~.,fat[-c(39,42)])
> fatMatrix = model.matrix(fullFatModel)[,-1] #NOTE: this -1 is to remo
> eVals = eigen(t(fatMatrix)%*%fatMatrix)
> sqrt(eVals$val[1]/eVals$val)
[1]     1.00000    19.37610    21.16725    36.01034    85.84167    94.7
[7]   121.84306   160.76414   196.59630   212.67150   234.59368   245.3
[13]   280.69168   300.69906   406.32695   643.19610 18326.65827
```

There is no difference, which makes sense as how linearly related our parameters are , in a theoretical thought, are not dependent on outliers

```
> vif(data.frame(fat$age,fat$weight,fat$height))
fat.age fat.weight fat.height
1.032253   1.107050   1.140470
> fatMatrix = model.matrix(adjustedModel)[,-1] #NOTE: this -1 is to rem
```

```
> eVals = eigen(t(fatMatrix)%*%fatMatrix)
> sqrt(eVals$val[1]/eVals$val)
[1]  1.00000 13.51194 22.67250
```

we can see that our variance inflation factors are much lower that previous, and that our condition numbers is not absurdly large.

```
> vals = model.matrix(adjustedModel)
> medians = apply(vals,2,median)[2:4]
> medians
age weight height
43.0  176.5   70.0
> predict(adjustedModel,new=data.frame(t(medians)),interval = "predicti
fit      lwr      upr
1 18.28132 7.659609 28.90304
> predict(adjustedModel,new=data.frame(age=40,weight=200,height=73),int
fit      lwr      upr
1 20.47854 9.837784 31.11929
> predict(adjustedModel,new=data.frame(age=40,weight=130,height=73),int
fit       lwr     upr
1 7.617419 -3.101062 18.3359
```

comparing the first prediction and the second, it is relatively close to the median, so its interval is similar to the median prediction's interval. The second prediction has a larger interval as it is further away from our median.