STAT*3240 F19
Some Helpful SAS and R Codes for Assignment #4

You can carry out the computational part of this assignment in either R SAS.

The purpose of this document is to give some R and SAS commands that may be helpful on Assignment #4. There may be more efficient ways of going about some of the analysis.

R commands

An easy way to import a data set into R is by using the file.choose() command:

```
mydata<-file.choose()
```

then browse for the file.

`attach(mydata)` puts the object `mydata` into R's search path, so the variables contained in `mydata` are found when the following commands are used.

The Box-Cox transformation is available in the MASS library. You will need to make appropriate adjustments to the following commands.

```
library(MASS)
pH.fit<-lm(pH~time)
boxcox(pH.fit,lambda=seq(-3,3,.05))
```

For more information on the `boxcox` command in `R`, use `?boxcox` or `help(boxcox)`

The `car` package contains many useful regression functions, including one that plots added variable plots:

(after installing the `car` package):

```
library(car)
```

```
avPlots(pH.fit)
```

plots added variable plots for the regressors in pH.fit.

It will also calculate variance inflation factors:

```
vif(pH.fit)
```

The `stats` package contains many useful measures:

http://stat.ethz.ch/R-manual/R-patched/library/stats/html/00Index.html.

including the listed influence measures:

https://stat.ethz.ch/R-manual/R-devel/library/stats/html/influence.measures.html

e.g.

```
influence.measures(pH.fit)
rstudent(pH.fit)
```

```
cooks.distance(pH.fit)
hatvalues(pH.fit)
```

The `qpcR` package (https://mran.microsoft.com/package/qpcR/) has a function that will calculate $R^2_{Prediction}$.

After installing this package:

```
library(qpcR)
```

```
PRESS(pH.fit)
```

SAS commands

All of the options in the SAS model statement can be found here:

http://support.sas.com/documentation/cdl/en/statug/63033/HTML/default/viewer.htm#statug_reg_sect013.htm

```
model y = x / lackfit;
```

carries out a lack of fit test.

To use a transformed variable value in PROC REG, we'll have to create the transformed variable ourselves in the data step, something like:

```
data steer;
infile 'thisiswheremyfileis' firstobs=2;
input obs time pH;
lntime = log(time);
run;

proc reg data = steer;
model whatevertheappropriatemodelis;
run;
```

To carry out the Box-Cox transformation, make the appropriate changes to the commands used in the example in class:

```
data dot;
infile 'F:\dot_red_sas.txt' firstobs=2;
input contract cost dotest status;
lndotest=log(dotest);
run;
proc transreg details data=dot;
      model boxcox(cost / convenient lambda=-2 to 2 by 0.01) = identity(lndotest) identity(status);
      output out=trans;
      run;
```

```
model y = x / stb;
```

prints standardized coefficients.

```
model y = x / VIF;
```

prints variance inflation factors.

```
model y = x / influence;
```

prints a variety of diagnostic and influence measures (leverage, Cook's D, studentized residuals, the PRESS statistic, etc.)

```
model y = x1 x2 x3 /partial;
```

prints partial regression plots (added variable plots) for x1, x2, and x3.