```r
## Homework 4 Wallace O'Rear
## chapter 6 1,3,4,5

setwd("C:/code/Math550/")

## 3 Cars

library(readr)
library(data.table)
cars <- data.table(read_csv("Data/cars04.csv"))

### a

# get smaller dataframe for the predictors we want in the problem
carsSub <- cars[,.(SuggestedRetailPrice,EngineSize,Cylinders,Horsepower,HighwayMPG,Weight,WheelBase,Hybrid)]

pairs(carsSub[,-c("SuggestedRetailPrice")])

cars.lm <- lm(SuggestedRetailPrice ~ ., data=carsSub)
summary(cars.lm)

# Call:
#   lm(formula = SuggestedRetailPrice ~ ., data = carsSub)
#
# Residuals:
#   Min     1Q Median     3Q    Max
# -17436  -4134    173   3561  46392
#
# Coefficients:
#   Estimate Std. Error t value Pr(>|t|)
# (Intercept) -68965.793  16180.381  -4.262 2.97e-05 ***
#   EngineSize   -6957.457   1600.137  -4.348 2.08e-05 ***
#   Cylinders     3564.755    969.633   3.676 0.000296 ***
#   Horsepower     179.702     16.411  10.950  < 2e-16 ***
#   HighwayMPG     637.939    202.724   3.147 0.001873 **
#   Weight          11.911      2.658   4.481 1.18e-05 ***
#   WheelBase       47.607    178.070   0.267 0.789444
# Hybrid         431.759   6092.087   0.071 0.943562
# ---
#   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# Residual standard error: 7533 on 226 degrees of freedom
# Multiple R-squared:  0.7819,  Adjusted R-squared:  0.7751
# F-statistic: 115.7 on 7 and 226 DF,  p-value: < 2.2e-16

par(mfrow=c(2,2))
plot(cars.lm,which = c(1,2,3,4))

## there is a pattern of the residuals in this model, so I don't think this is a valid model fit
par(mfrow=c(1,1))
plot(y=carsSub$SuggestedRetailPrice,x=cars.lm$fitted.values)

### b
## also looking at the plot of predicted vs. actuals you can still see a pattern.
## therefore, there is still something not linear about the data un-transformed.

### c

# point 223 is a quite a big leverage point
cars[223,] ## it's this car Mercedes-Benz CL600 2dr so expensive
# Vehicle Name Hybrid SuggestedRetailPrice DealerCost EngineSize Cylinders Horsepower CityMPG HighwayMPG Weight
WheelBase Length
#  Mercedes-Benz CL600 2dr      0               128420     119600        5.5        12        493      13         19
4473       114    196
# Width
#  73
cars.lm$fitted.values[223]
# 223
# 94963.28
### d fitting a new model

#let's just create new columns with the transformed values
carsSub[,lSRP:=log(SuggestedRetailPrice)]
carsSub[,tEngine:=EngineSize^.25]
carsSub[,lCyl:=log(Cylinders)]
carsSub[,lHP:=log(Horsepower)]
carsSub[,tHwyMPG:=HighwayMPG^-1]
carsSub[,lWB:=log(WheelBase)]

cars.tran.lm <- lm(lSRP ~ tEngine + lCyl + lHP + tHwyMPG + Weight + lWB + Hybrid, data = carsSub)
summary(cars.tran.lm)

# Call:
#   lm(formula = lSRP ~ tEngine + lCyl + lHP + tHwyMPG + Weight +
#        lWB + Hybrid, data = carsSub)
```

```
#
# Residuals:
#   Min       1Q   Median       3Q      Max
# -0.42288 -0.10983 -0.00203  0.10279  0.70068
#
# Coefficients:
#   Estimate Std. Error t value Pr(>|t|)
# (Intercept)  5.703e+00  2.010e+00   2.838  0.00496 **
#   tEngine     -1.575e+00  3.332e-01  -4.727 4.01e-06 ***
#   lCyl         2.335e-01  1.204e-01   1.940  0.05359 .
#   lHP          8.992e-01  8.876e-02  10.130  < 2e-16 ***
#   tHwyMPG      8.029e-01  4.758e+00   0.169  0.86614
#   Weight       5.043e-04  6.367e-05   7.920 1.07e-13 ***
#   lWB         -6.385e-02  4.715e-01  -0.135  0.89240
#   Hybrid       6.422e-01  1.150e-01   5.582 6.78e-08 ***
#   ---
#   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# Residual standard error: 0.1789 on 226 degrees of freedom
# Multiple R-squared:  0.8621,  Adjusted R-squared:  0.8578
# F-statistic: 201.8 on 7 and 226 DF,  p-value: < 2.2e-16

par(mfrow=c(2,2))
plot(cars.tran.lm,which = c(1,2,3,4))

par(mfrow=c(1,1))
plot(y=carsSub$lSRP,x=cars.tran.lm$fitted.values)

## looking at the residuals and the actual vs. fitted, the model definitely looks more valid.
## However, there is some huge leverage points.

cars.tran.lm.red <- update(cars.tran.lm, . ~ . - tHwyMPG - lWB)
summary(cars.tran.lm.red)

# Call:
#   lm(formula = lSRP ~ tEngine + lCyl + lHP + Weight + Hybrid, data = carsSub)
#
# Residuals:
#   Min       1Q   Median       3Q      Max
# -0.42224 -0.11001 -0.00099  0.10191  0.70205
#
# Coefficients:
#   Estimate Std. Error t value Pr(>|t|)
# (Intercept)  5.422e+00  3.291e-01  16.474  < 2e-16 ***
#   tEngine     -1.591e+00  3.157e-01  -5.041 9.45e-07 ***
#   lCyl         2.375e-01  1.186e-01   2.003   0.0463 *
#   lHP          9.049e-01  8.305e-02  10.896  < 2e-16 ***
#   Weight       5.029e-04  5.203e-05   9.666  < 2e-16 ***
#   Hybrid       6.340e-01  1.080e-01   5.870 1.53e-08 ***
#   ---
#   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# Residual standard error: 0.1781 on 228 degrees of freedom
# Multiple R-squared:  0.862,   Adjusted R-squared:  0.859
# F-statistic: 284.9 on 5 and 228 DF,  p-value: < 2.2e-16

## e
anova(cars.tran.lm.red,cars.tran.lm)

# Analysis of Variance Table
#
# Model 1: lSRP ~ tEngine + lCyl + lHP + Weight + Hybrid
# Model 2: lSRP ~ tEngine + lCyl + lHP + tHwyMPG + Weight + lWB + Hybrid
# Res.Df    RSS Df Sum of Sq     F Pr(>F)
# 1    228 7.2358
# 2    226 7.2337  2 0.0021769 0.034 0.9666

## the F-test shows that removing the 2 was probably a good idea.

## f
## The manufacturer of the car is in the data set, so you can just strip the name of the
## car maker from that and make it a column on which you predict.

library(alr3)

### 4
krafft <- read.delim("C:/code/Math550/Data/krafft.txt")
pairs(krafft)

krafft.lm <- lm(KPOINT ~ RA + HEAT + VTINV + DIPINV, data=krafft)

attach(krafft)
par(mfrow=c(2,2))
mmp(krafft.lm,RA)
mmp(krafft.lm,HEAT)
```

```
mmp(krafft.lm,VTINV)
mmp(krafft.lm,DIPINV)

avPlot(krafft.lm,variable=RA,ask=FALSE,identify.points=TRUE, main="")
avPlot(krafft.lm,variable=HEAT,ask=FALSE,identify.points=TRUE, main="")
avPlot(krafft.lm,variable=VTINV,ask=FALSE,identify.points=TRUE, main="")
avPlot(krafft.lm,variable=DIPINV,ask=FALSE,identify.points=TRUE, main="")

detach(krafft)

plot(krafft.lm,which = c(1,2,3,4))

## a
## I can't find a patern in the residuals anywhere. So, I would be inclined to say that this might
## be a valid model.

## b
## Yeah, those values against each other produced a little curve, so there might be some correlation between them,
but
## against the response variable, they look linear to me.

## c
## I feel like a lot more go into selecting between models than just 4 things. While all those listed are important,
there are things
## you can see by observing the data or knowing something about what you are trying to model that would lead you to
choose
## one model over another. Interpretability of a model might be more important than having the absolute best fit.


### 5
golf <- read.csv("C:/code/Math550/Data/pgatour2006.csv")

golfDf <- data.table(golf[,c(3,5,6,7,8,9,10,12)]) # pull out the columns I need

golf.lm <- lm(PrizeMoney ~ ., golfDf)
summary(golf.lm)

# Call:
#   lm(formula = PrizeMoney ~ ., data = golfDf)
#
# Residuals:
#   Min     1Q Median     3Q    Max
# -81239 -26260  -6521  17539 420230
#
# Coefficients:
#   Estimate Std. Error t value Pr(>|t|)
# (Intercept)      -1165233.1   587382.9  -1.984 0.048737 *
#   DrivingAccuracy     -1835.8      889.2  -2.065 0.040326 *
#   GIR                  9671.3     3309.4   2.922 0.003899 **
#   PuttingAverage     -47435.3   521566.4  -0.091 0.927631
# BirdieConversion    10426.0     3049.6   3.419 0.000771 ***
#   SandSaves            1182.1      744.8   1.587 0.114184
# Scrambling           4741.3     2400.8   1.975 0.049749 *
#   PuttsPerRound        5267.5    35765.7   0.147 0.883070
# ---
#   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# Residual standard error: 50140 on 188 degrees of freedom
# Multiple R-squared:  0.4064,  Adjusted R-squared:  0.3843
# F-statistic: 18.39 on 7 and 188 DF,  p-value: < 2.2e-16

par(mfrow=c(2,2))
plot(golf.lm,which = c(1,2,3,4))

golf.logY.lm <- lm(log(PrizeMoney) ~ ., golfDf)
summary(golf.logY.lm)

# Call:
#   lm(formula = log(PrizeMoney) ~ ., data = golfDf)
#
# Residuals:
#   Min       1Q   Median       3Q      Max
# -1.71949 -0.48608 -0.09172  0.44561  2.14013
#
# Coefficients:
#   Estimate Std. Error t value Pr(>|t|)
# (Intercept)       0.194300   7.777129   0.025 0.980095
# DrivingAccuracy  -0.003530   0.011773  -0.300 0.764636
# GIR               0.199311   0.043817   4.549 9.66e-06 ***
#   PuttingAverage   -0.466304   6.905698  -0.068 0.946236
# BirdieConversion  0.157341   0.040378   3.897 0.000136 ***
#   SandSaves         0.015174   0.009862   1.539 0.125551
# Scrambling        0.051514   0.031788   1.621 0.106788
# PuttsPerRound    -0.343131   0.473549  -0.725 0.469601
# ---
```

```
#   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# Residual standard error: 0.6639 on 188 degrees of freedom
# Multiple R-squared:  0.5577,  Adjusted R-squared:  0.5412
# F-statistic: 33.87 on 7 and 188 DF,  p-value: < 2.2e-16

plot(golf.logY.lm,which = c(1,2,3,4))

## a
## I would say, after looking at the non-transformed response vs. the transformed response, the recommendation of
## applying log to PrizeMoney is valid.

library(MASS)
library(car)
par(mfrow=c(1,1))
powerTransform(golfDf)
# Estimated transformation parameters
# PrizeMoney  DrivingAccuracy              GIR   PuttingAverage BirdieConversion        SandSaves       Scrambling
PuttsPerRound
# 0.03656263       0.27563813       1.52277672       1.00596706       0.89094560       0.99276862       0.67459642
-0.03445974
## b

## doing the boxcox test, you can see that the notion of applying log on PrizeMoney is verified. You could also apply
log to PuttsPerRound and .25 power to
## driving accuracy, the rest probably keep un-transformed.

golf.try.lm <- lm(log(PrizeMoney) ~ I(DrivingAccuracy^.25) + GIR + PuttingAverage + BirdieConversion
                  + SandSaves + Scrambling + log(PuttsPerRound), data = golfDf)
summary(golf.try.lm)

# Call:
#   lm(formula = log(PrizeMoney) ~ I(DrivingAccuracy^0.25) + GIR +
#        PuttingAverage + BirdieConversion + SandSaves + Scrambling +
#        log(PuttsPerRound), data = golfDf)
#
# Residuals:
#   Min       1Q   Median       3Q      Max
# -1.71954 -0.48331 -0.09046  0.44650  2.13980
#
# Coefficients:
#   Estimate Std. Error t value Pr(>|t|)
# (Intercept)             24.043954  35.885781   0.670 0.503671
# I(DrivingAccuracy^0.25) -0.268263   1.059174  -0.253 0.800332
# GIR                      0.198490   0.043946   4.517 1.11e-05 ***
#   PuttingAverage          -0.475990   6.901283  -0.069 0.945086
# BirdieConversion         0.158179   0.040363   3.919 0.000124 ***
#   SandSaves                0.015249   0.009859   1.547 0.123622
# Scrambling               0.051556   0.031816   1.620 0.106817
# log(PuttsPerRound)      -9.868262  13.847835  -0.713 0.476964
# ---
#   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# Residual standard error: 0.664 on 188 degrees of freedom
# Multiple R-squared:  0.5576,  Adjusted R-squared:  0.5411
# F-statistic: 33.85 on 7 and 188 DF,  p-value: < 2.2e-16

anova(golf.logY.lm,golf.try.lm)

#Analysis of Variance Table
#
# Model 1: log(PrizeMoney) ~ DrivingAccuracy + GIR + PuttingAverage + BirdieConversion +
#   SandSaves + Scrambling + PuttsPerRound
# Model 2: log(PrizeMoney) ~ I(DrivingAccuracy^0.25) + GIR + PuttingAverage +
#   BirdieConversion + SandSaves + Scrambling + log(PuttsPerRound)
# Res.Df    RSS Df Sum of Sq F Pr(>F)
# 1    188 82.866
# 2    188 82.888  0 -0.022661

par(mfrow=c(2,2))
plot(golf.try.lm,which = c(1,2,3,4))
## so my try and the untransformed didn't really do much difference

## c

## point 185 should be investigated as it has high leverage and it shows up on all the residual plots
# > golf[185,]
#           Name TigerWoods PrizeMoney AveDrivingDistance DrivingAccuracy   GIR PuttingAverage BirdieConversion
SandSaves Scrambling BounceBack PuttsPerRound
# 185 Tom Lehman          0      84604              286.6           60.96 65.93          1.827            24.37
39.36      54.89      16.48         30.08

par(mfrow=c(3,3))
attach(golfDf)
mmp(golf.logY.lm,DrivingAccuracy,key=NULL)
```

```
mmp(golf.logY.lm,GIR,key=NULL)
mmp(golf.logY.lm,PuttingAverage,key=NULL)
mmp(golf.logY.lm,BirdieConversion,key=NULL)
mmp(golf.logY.lm,SandSaves,key=NULL)
mmp(golf.logY.lm,Scrambling,key=NULL)
mmp(golf.logY.lm,PuttsPerRound,key=NULL)
mmp(golf.logY.lm,golf.logY.lm$fitted.values,xlab="Fitted Values",key=NULL)

par(mfrow=c(2,4))
avPlot(golf.logY.lm,variable=DrivingAccuracy,ask=FALSE,identify.points=TRUE, main="")
avPlot(golf.logY.lm,variable=GIR,ask=FALSE,identify.points=TRUE, main="")
avPlot(golf.logY.lm,variable=PuttingAverage,ask=FALSE,identify.points=FALSE, main="")
avPlot(golf.logY.lm,variable=BirdieConversion,ask=FALSE,identify.points=FALSE, main="")
avPlot(golf.logY.lm,variable=SandSaves,ask=FALSE,identify.points=FALSE, main="")
avPlot(golf.logY.lm,variable=Scrambling,ask=FALSE,identify.points=FALSE, main="")
avPlot(golf.logY.lm,variable=PuttsPerRound,ask=FALSE,identify.points=FALSE, main="")
detach(golfDf)

## after looking at the added value plots, you can tell why BirdieConversion and GIR are significant
## while the others aren't.

## d
## I probably wouldn't just drop all the insignificant variables at once. You should remove just one at a time and
check to see if
## the removal changes effect on another variable.
```