STAT425_HW6_Jinran Yang

1. (1)Forward selection based on F-test statistics

```
#load data
misner<- read.csv("~/Desktop/hw6/misner(1).dat", sep="")
#forwad selection
mod<-lm(yield~1,data = misner)
indep.var<- ~ year+I(year^2)+rain+I(rain^2)+year*rain
add1(mod,indep.var,test = 'F')
```

```
## Single term additions
##
## Model:
## yield ~ 1
##           Df Sum of Sq    RSS    AIC F value  Pr(>F)
## <none>                 704.55 112.96
## year       1   101.580 602.97 109.04  6.0648 0.01871 *
## I(year^2)  1   101.387 603.16 109.06  6.0513 0.01883 *
## rain       1   114.215 590.34 108.24  6.9651 0.01221 *
## I(rain^2)  1    86.247 618.30 110.00  5.0216 0.03129 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
mod<-update(mod, .~. +rain)#add rain
add1(mod,indep.var,test = 'F')
```

```
## Single term additions
##
## Model:
## yield ~ rain
##           Df Sum of Sq    RSS    AIC F value  Pr(>F)
## <none>                 590.34 108.24
## year       1    95.994 494.34 103.49  6.7965 0.01333 *
## I(year^2)  1    95.827 494.51 103.51  6.7824 0.01342 *
## I(rain^2)  1    94.807 495.53 103.58  6.6964 0.01397 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
mod<-update(mod, .~. + year)#add year
add1(mod,indep.var,test = 'F')
```

```
## Single term additions
##
## Model:
## yield ~ rain + year
##           Df Sum of Sq    RSS     AIC F value   Pr(>F)
## <none>                 494.34 103.494
## I(year^2)  1    10.948 483.39 104.643  0.7700 0.386366
## I(rain^2)  1    83.349 410.99  98.478  6.8952 0.012862 *
## year:rain  1   130.400 363.94  93.858 12.1822 0.001357 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
mod<-update(mod, .~. + year*rain)#add interaction
add1(mod,indep.var,test = 'F')
```

```
## Single term additions
##
## Model:
## yield ~ rain + year + rain:year
##           Df Sum of Sq    RSS    AIC F value  Pr(>F)
## <none>                 363.94 93.858
## I(year^2)  1     0.730 363.21 95.781  0.0663 0.79841
## I(rain^2)  1    61.388 302.55 88.838  6.6956 0.01426 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
mod<-update(mod, .~. + I(rain^2))#add rain^2
add1(mod,indep.var,test = 'F')
```

```
## Single term additions
##
## Model:
## yield ~ rain + year + I(rain^2) + rain:year
##           Df Sum of Sq    RSS    AIC F value Pr(>F)
## <none>                 302.55 88.838
## I(year^2)  1    7.0766 295.48 89.938  0.7664 0.3879
```

```r
summary(mod)
```

```
##
## Call:
## lm(formula = yield ~ rain + year + I(rain^2) + rain:year, data = misner)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.2969 -2.5471  0.6011  1.9923  5.0204
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.909e+03  4.862e+02  -3.927 0.000414 ***
## rain         1.588e+02  4.457e+01   3.564 0.001138 **
## year         1.001e+00  2.555e-01   3.919 0.000423 ***
## I(rain^2)   -1.862e-01  7.198e-02  -2.588 0.014257 *
## rain:year   -8.064e-02  2.345e-02  -3.439 0.001599 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.028 on 33 degrees of freedom
## Multiple R-squared:  0.5706, Adjusted R-squared:  0.5185
## F-statistic: 10.96 on 4 and 33 DF,  p-value: 9.127e-06
```

Therefore, best model select by forward selection based on F-statistics is `yield ~ rain + year + I(rain^2) + rain:year`.

(2)Backward selection based on AIC

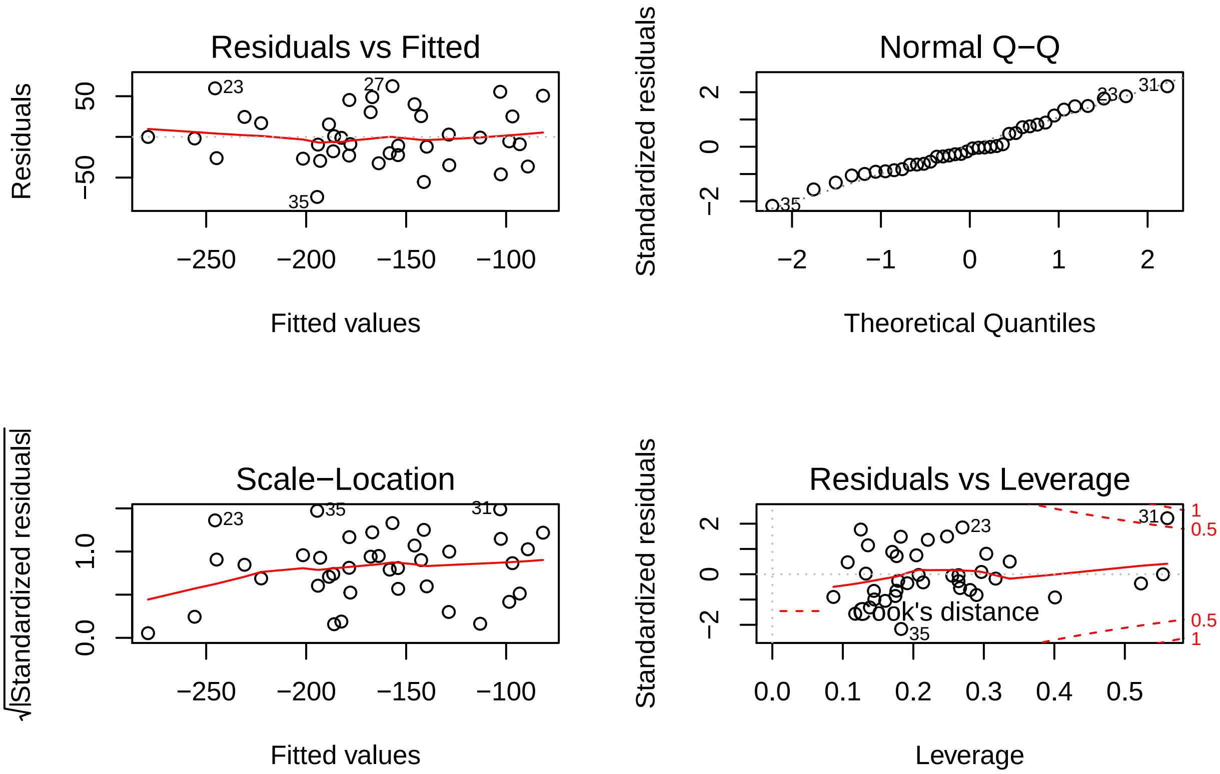
```r
mod_b<-step(lm(yield ~ rain+year+I(year^2)+ I(rain^2)+rain*year, data = misner),k=2,direction = 'backwa
summary(mod_b)
```

```
##
## Call:
## lm(formula = yield ~ rain + year + I(rain^2) + rain:year, data = misner)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.2969 -2.5471  0.6011  1.9923  5.0204
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.909e+03  4.862e+02  -3.927 0.000414 ***
## rain         1.588e+02  4.457e+01   3.564 0.001138 **
## year         1.001e+00  2.555e-01   3.919 0.000423 ***
## I(rain^2)   -1.862e-01  7.198e-02  -2.588 0.014257 *
## rain:year   -8.064e-02  2.345e-02  -3.439 0.001599 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.028 on 33 degrees of freedom
## Multiple R-squared:  0.5706, Adjusted R-squared:  0.5185
## F-statistic: 10.96 on 4 and 33 DF,  p-value: 9.127e-06
```

Therefore, best model select by backward selection based on AIC is `yield ~ rain + year + I(rain^2) + rain:year`, which is the same as the result of forward selection based on F-statistics.

2.

```r
library(faraway)
data("seatpos")
mod1=lm(hipcenter~.,data = seatpos)
par(mfrow=c(2,2))
plot(mod1)
```

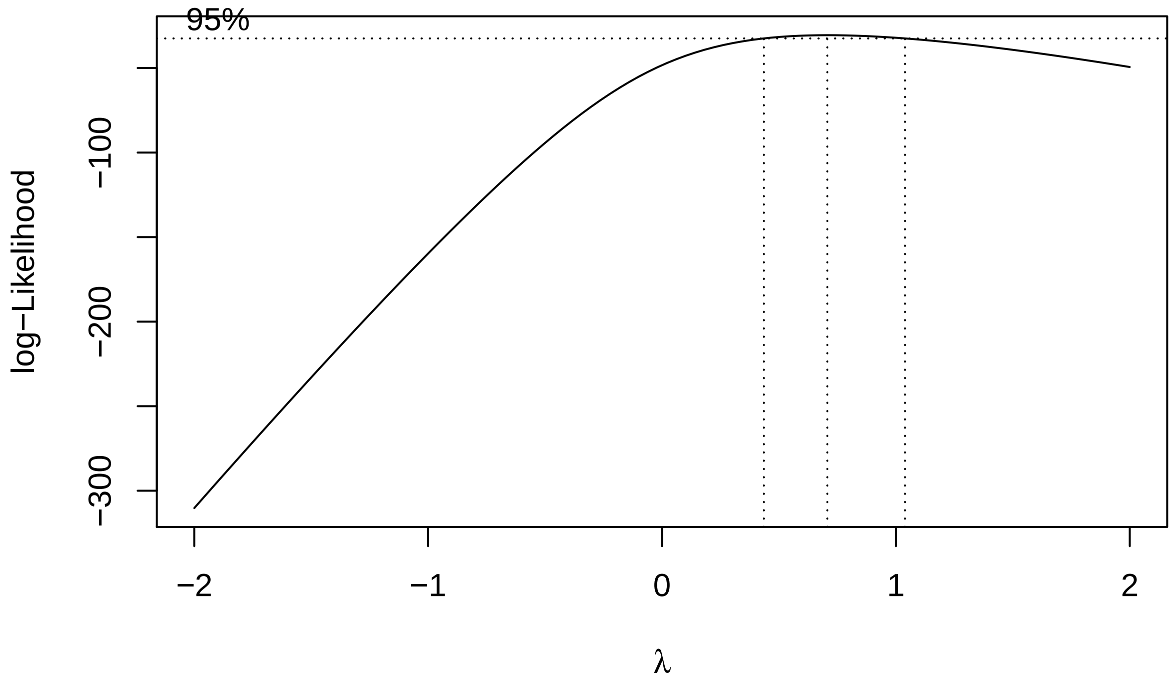


We can see a quadratic trend of residuals in the `scale-location` plot. We might need to trandform the response.

```
min(seatpos$hipcenter)
```

```
## [1] -279.15
```

```
seatpos$hipcenter<-seatpos$hipcenter+280
library(MASS)
mod1=lm(hipcenter~.,data = seatpos)
boxcox(mod1)
```



Since 1 is inside the 95% CI of lambda, it seems there is no need to trandform the response. Because there is quadtic trend in the `Residuals vs Fitted` plot, I add some quadtic terms and then perform model selection to find a small(sparse) model.

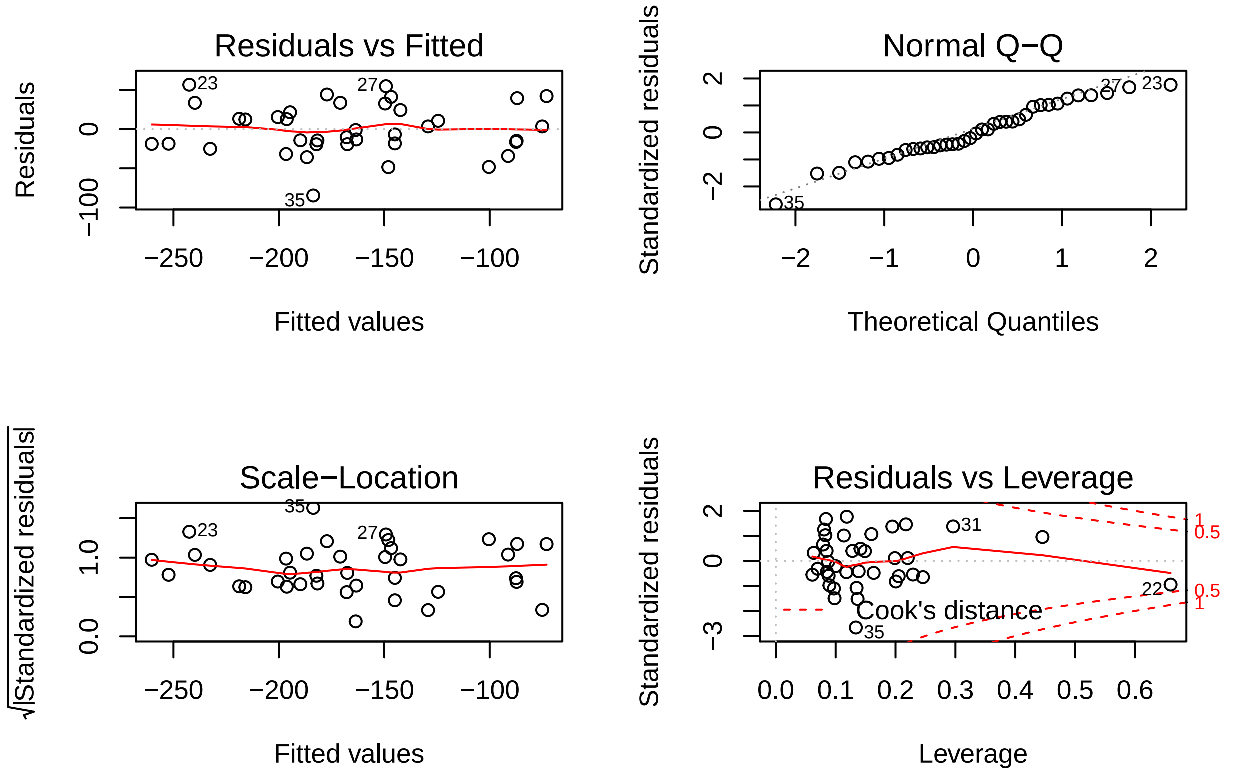
```
seatpos$hipcenter<-seatpos$hipcenter-280

mod2=lm(hipcenter~.+I(Age^2) + I(Weight^2) + I(HtShoes^2) +I(Ht^2) + I(Seated^2) + I(Arm^2)
        + I(Thigh^2) +I(Leg^2),data = seatpos)
stepmode<-step(mod2,k=2,direction = "both",trace = 0 )#stepwise selection based on AIC
summary(stepmode)
```

```
##
## Call:
## lm(formula = hipcenter ~ Age + HtShoes + Leg + I(Age^2) + I(HtShoes^2),
##     data = seatpos)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -84.64 -18.80  -3.99  23.58  56.60
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2157.85352 1079.36331   1.999   0.0541 .
## Age            -3.44084    2.38515  -1.443   0.1588
## HtShoes       -21.09396   12.34996  -1.708   0.0973 .
## Leg            -6.63403    3.96031  -1.675   0.1037
## I(Age^2)        0.04847    0.02827   1.715   0.0961 .
## I(HtShoes^2)    0.05367    0.03528   1.521   0.1380
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 34.15 on 32 degrees of freedom
## Multiple R-squared:  0.7164, Adjusted R-squared:  0.6721
```

```
## F-statistic: 16.17 on 5 and 32 DF,  p-value: 5.915e-08
```

```
par(mfrow=c(2,2))
plot(stepmode)
```



Therefore, the model selected by stepwise selection based on AIC is `hipcenter ~ Age + HtShoes + Leg + I(Age^2) + I(HtShoes^2)` which can explain 71.64% variation of `hipcenter`. According to the diagnostics plots, all plots looks good, and `Scale-Location` plot is more flatter than before.

```
nrow(seatpos)
```

```
## [1] 38
```

```
forwardmode=step(mod2,k=log(38),direction = "forward" )
```

```
## Start:  AIC=314.38
## hipcenter ~ Age + Weight + HtShoes + Ht + Seated + Arm + Thigh +
##     Leg + I(Age^2) + I(Weight^2) + I(HtShoes^2) + I(Ht^2) + I(Seated^2) +
##     I(Arm^2) + I(Thigh^2) + I(Leg^2)
```

```
summary(forwardmode)#too many vairables, not good
```

```
##
## Call:
## lm(formula = hipcenter ~ Age + Weight + HtShoes + Ht + Seated +
##     Arm + Thigh + Leg + I(Age^2) + I(Weight^2) + I(HtShoes^2) +
##     I(Ht^2) + I(Seated^2) + I(Arm^2) + I(Thigh^2) + I(Leg^2),
##     data = seatpos)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -60.128 -20.498  -3.378  16.156  67.924
##
```

```
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.646e+03  3.183e+03   1.460   0.1591
## Age          -5.208e+00  3.494e+00  -1.490   0.1510
## Weight        1.142e+00  1.583e+00   0.721   0.4788
## HtShoes      -5.267e+02  2.917e+02  -1.805   0.0854 .
## Ht            4.542e+02  2.915e+02   1.558   0.1341
## Seated        4.588e+01  1.002e+02   0.458   0.6517
## Arm           1.046e+02  7.335e+01   1.426   0.1686
## Thigh        -4.670e+01  4.645e+01  -1.005   0.3262
## Leg          -3.313e+01  6.316e+01  -0.525   0.6054
## I(Age^2)      8.356e-02  4.473e-02   1.868   0.0758 .
## I(Weight^2)  -3.671e-03  4.268e-03  -0.860   0.3994
## I(HtShoes^2)  1.501e+00  8.393e-01   1.788   0.0882 .
## I(Ht^2)      -1.305e+00  8.465e-01  -1.542   0.1381
## I(Seated^2)  -2.627e-01  5.647e-01  -0.465   0.6466
## I(Arm^2)     -1.703e+00  1.144e+00  -1.489   0.1513
## I(Thigh^2)    5.696e-01  5.810e-01   0.980   0.3381
## I(Leg^2)      2.984e-01  8.522e-01   0.350   0.7298
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 37.32 on 21 degrees of freedom
## Multiple R-squared:  0.7779, Adjusted R-squared:  0.6086
## F-statistic: 4.596 on 16 and 21 DF,  p-value: 0.000713
```

4.

```r
library(MASS)
beta=vector(length = 20)
beta[1:3]=c(1,-1,1)
beta[4:20]=0
I<-diag(x=1,nrow = 20,ncol = 20)
J<- matrix(1,20,20)
sigma<-0.7*I + 0.3*J
mean<-matrix(0,100,1)

set.seed(1)
X=mvrnorm(100, mu=rep(0,20), Sigma=sigma)
error<-rnorm(100, mean = 0, sd = 1)
Y=X%*%beta+error
##least square
ls<-lm(Y~X)
mse_ls<-sum((ls$coefficients[-1]-beta)^2)
mse_ls
```

```
## [1] 0.2884138
```

```r
#ridge
library(glmnet)
```

```
## Warning: package 'glmnet' was built under R version 3.4.4

## Loading required package: Matrix

## Loading required package: foreach
```

```
## Warning: package 'foreach' was built under R version 3.4.3
```

```
## Loaded glmnet 2.0-16
```

```
cv.ridge<- cv.glmnet(X,Y, alpha =0)
model.ridge<- glmnet(X,Y, lambda = cv.ridge$lambda.min,alpha = 0)
mse_ridge<-sum((model.ridge$beta-beta)^2)
mse_ridge
```

```
## [1] 0.2062827
```

```
#lasso
cv.lasso<- cv.glmnet(X,Y, alpha =1)
model.lasso<- glmnet(X,Y, lambda = cv.lasso$lambda.min,alpha = 1)
mse_lasso<-sum((model.lasso$beta-beta)^2)
mse_lasso
```

```
## [1] 0.09373847
```

```
#100 synthetic datasets and average the values of the MSEs
n=100
p=20
time=100
mse_ls=mse_ridge=mse_lasso=rep(0,time)
set.seed(1)

for(i in 1:time)
{
X=mvrnorm(n, mu=rep(0,p), Sigma=sigma)
error<-rnorm(100, mean = 0, sd = 1)
Y=X%*%beta+error
ls<-lm(Y~X)
mse_ls[i]<-sum((ls$coefficients[-1]-beta)^2)

cv.ridge<- cv.glmnet(X,Y, alpha =0)
model.ridge<- glmnet(X,Y, lambda = cv.ridge$lambda.min,alpha = 0)
mse_ridge[i]<-sum((model.ridge$beta-beta)^2)


cv.lasso<- cv.glmnet(X,Y, alpha =1)
model.lasso<- glmnet(X,Y, lambda = cv.lasso$lambda.min,alpha = 1)
mse_lasso[i]<-sum((model.lasso$beta-beta)^2)
}
mean(mse_ls)
```

```
## [1] 0.3589478
```

```
mean(mse_ridge)
```

```
## [1] 0.3275249
```

```
mean(mse_lasso)
```

```
## [1] 0.1648888
```

As we can see the performance of least squares is the worst, and the performance of lasso regression is the best. The performance of ridge is slightly better than least squares.