STAT 425 - Homework #4

PROBLEM 1

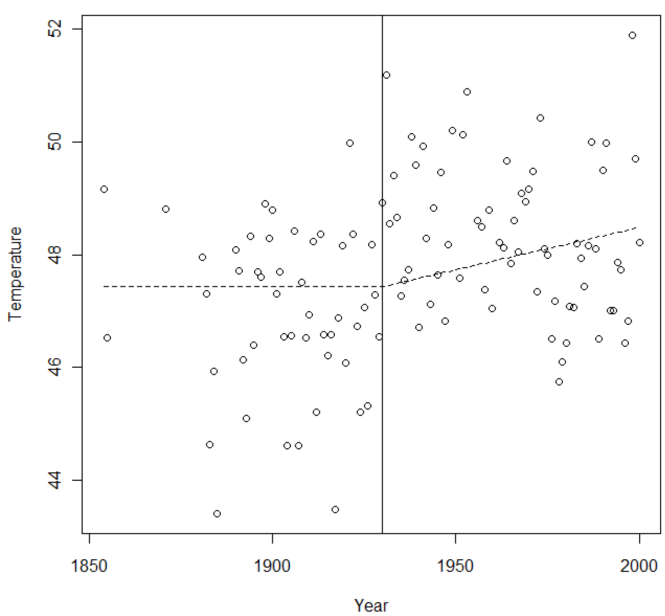
a) Fit a broken line regression, constant before 1930 and linear after it

- Code:

```
> #right hockey stick function
> rhs = function(x) ifelse(x < 1930,0,x-1930)
>
> #broken line regression, constant before 1930 and linear after
> gb = lm(temp ~ rhs(year), aatemp)
>
> #report coefficients
> gb$coeff
(Intercept)    rhs(year)
47.43215071   0.01496967
>
> #plot data
> plot(aatemp$year,aatemp$temp,xlab="Year",ylab="Temperature")
> abline(v=1930)
>
> #plot fitted model
> x = seq(1854,2000,by=1)
> py = gb$coef[1]+gb$coef[2]*rhs(x)
> lines(x,py,lty=2)
```

- Estimated coefficients:
  - Before 1930: Slope = 0, Intercept = 47.432
  - After 1930: Slope = 0.01497, Intercept (at axis through 1930) = 47.432

- Plot:



b) Fit ordinary linear model and compare

- Code

```
> ##Part b
>
> #ordinary linear model
> g = lm(temp ~ year, aatemp)
>
> #report r squared of both models
> summary(gb)$r.sq
[1] 0.05464502
> summary(g)$r.sq
[1] 0.08535947
```

- Since their dimensions are the same, make decision based on R^2
  - Broken line regression model : $R^2 = 0.0546$
  - Ordinary linear model: $R^2 = 0.0854$
  - Since R^2 of ordinary model is higher($0.0854 > 0.0546$), ordinary model is better

c) Backward elimination to reduce the degree of the polynomial (start with degree 10)

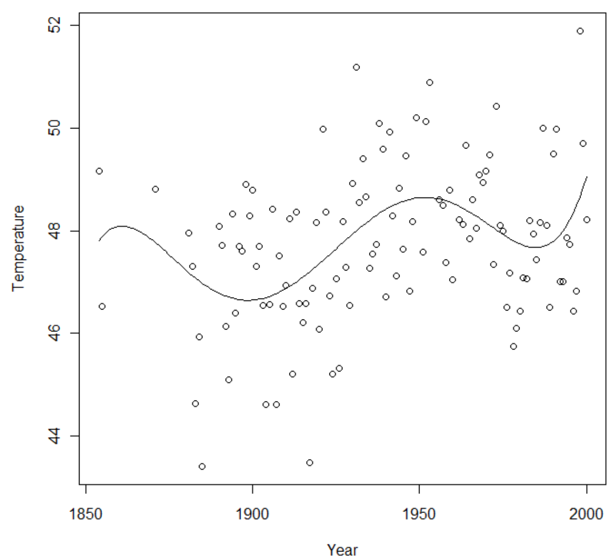
- Eliminate statistically insignificant terms starting with the highest order term
- *Note:* since sample large, no need to refit poly (year,9), poly (year,8), etc. Equivalence results are obtained.

```
> #Use poly 10
> summary(lm(temp~poly(year,10)))

Call:
lm(formula = temp ~ poly(year, 10))

Residuals:
    Min      1Q  Median      3Q     Max
-3.4987 -0.8641 -0.1745  1.1450  3.4255

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)        47.7426     0.1319 361.927  < 2e-16 ***
poly(year, 10)1     4.7616     1.4146   3.366  0.00107 **
poly(year, 10)2    -0.9071     1.4146  -0.641  0.52277
poly(year, 10)3    -3.3132     1.4146  -2.342  0.02108 *
poly(year, 10)4     2.4383     1.4146   1.724  0.08774 .
poly(year, 10)5     3.3824     1.4146   2.391  0.01860 *
poly(year, 10)6     1.2124     1.4146   0.857  0.39337
poly(year, 10)7    -0.9373     1.4146  -0.663  0.50908
poly(year, 10)8    -1.1011     1.4146  -0.778  0.43812
poly(year, 10)9     1.3994     1.4146   0.989  0.32483
poly(year, 10)10    0.3474     1.4146   0.246  0.80652
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.415 on 104 degrees of freedom
Multiple R-squared: 0.2165,     Adjusted R-squared: 0.1411
F-statistic: 2.873 on 10 and 104 DF,  p-value: 0.003335
```

- Decide to use polynomial with degree 5

```
> #Model with power 5 selected
> g1=lm(temp ~ poly(year,5),aatemp)
> summary(g1)

Call:
lm(formula = temp ~ poly(year, 5), data = aatemp)

Residuals:
    Min      1Q  Median      3Q     Max
-3.7142 -0.9198 -0.1420  0.9903  3.2364

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)     47.7426     0.1306 365.604  < 2e-16 ***
poly(year, 5)1   4.7616     1.4004   3.400 0.000942 ***
poly(year, 5)2  -0.9071     1.4004  -0.648 0.518500
poly(year, 5)3  -3.3132     1.4004  -2.366 0.019749 *
poly(year, 5)4   2.4383     1.4004   1.741 0.084470 .
poly(year, 5)5   3.3824     1.4004   2.415 0.017384 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.4 on 109 degrees of freedom
Multiple R-squared: 0.1952,     Adjusted R-squared: 0.1583
F-statistic: 5.289 on 5 and 109 DF,  p-value: 0.0002176

>
> #plot data
> plot(aatemp$year,aatemp$temp,xlab="Year",ylab="Temperature")
>
> #plot fitted model
> grid = seq(1854,2000,by=1)
> lines(grid,predict(g1,data.frame(year=grid)))
```

- Plot



- Predicted temperature for year 2020 = 60.078 ^{o}F

```
> #predict temperature
> new=data.frame(year=2020)
> pred.PI = predict(g1, new, level=0.95, interval="prediction")
> pred.PI
       fit      lwr      upr
1 60.07774 49.84092 70.31456
```

PROBLEM 2

a) Fit O3 as response and temp, humidity, and ibh as predictors

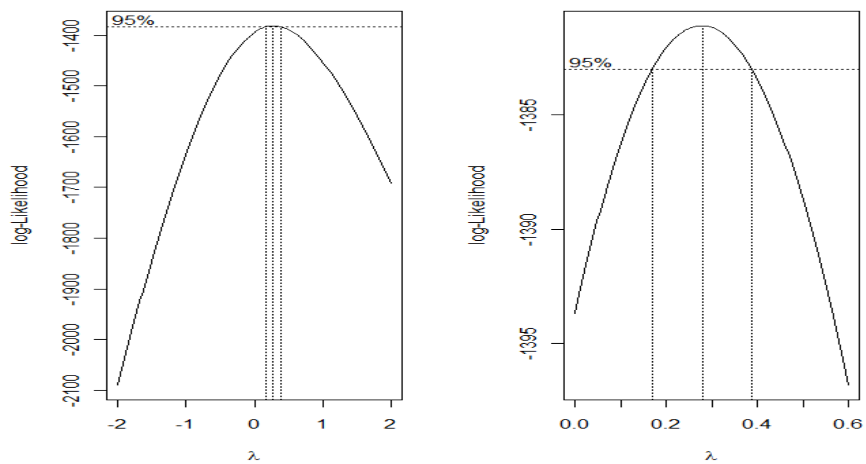
- $R^2 = 0.684$

```
> g = lm(O3 ~ temp + humidity + ibh, ozone) # fit model
> summary(g)$r.sq # r-squared
[1] 0.6839717
```

b) Box-Cox transformation

- Plot

```
> boxcox(g,plotit=T) # plotit=T is the default setting
> boxcox(g,plotit=T,lambda=seq(0,0.6,by=0.1)) # zoom-in
```




```
> O3.trans=boxcox(g, lambda=seq(-2, 2, length=400))
> lambda.hat = O3.trans$x[O3.trans$y == max(O3.trans$y)]
> lambda.hat # use 0.25, fourth root transformation
[1] 0.2756892
```

- Transformation
  - $\hat{\lambda} = 0.276$ (maximizes log likelihood)
  - Use $\hat{\lambda} = 0.25$ for easier interpretation (fourth root power)
  - Transformed response : $y^{1/4}$
- Check 95% CI
  - The likelihood ratio test supports a transformation since 1 does not belong to the confidence interval (see graph or confidence interval below)

```
> tmp=O3.trans$x[O3.trans$y > max(O3.trans$y) - qchisq(0.95, 1)/2];
> CI=range(tmp) # 95% CI.
> CI
[1] 0.1754386 0.3859649
> 1>CI[1] & 1<CI[2] # Check contains 1
[1] FALSE
```

c) $R^2 = 0.715$ from model with transformed response (note this is higher than a)

```
> O3.lambda.25 = O3^.25
> g.trans = lm(O3.lambda.25 ~ temp + humidity + ibh)
> r2.trans=summary(g.trans)$r.sq
> r2.trans
[1] 0.7152378
```

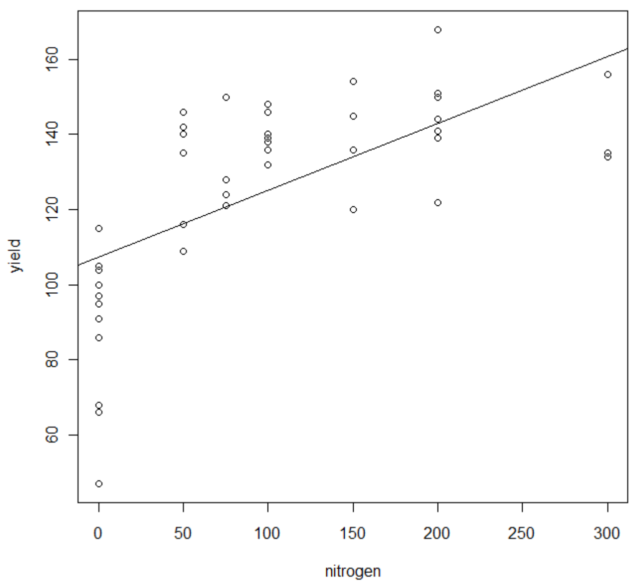
PROBLEM 3

a) Fit Yield ~ nitrogen

- Code:

```
> g.a = lm(yield ~ nitrogen, cornnit) # fit
>
> plot(yield ~ nitrogen,xlab="nitrogen",ylab="yield") # plot
> abline(g.a)
>
> g1=lm(yield~factor(nitrogen)) # lack of fit test
> anova(g.a, g1)
Analysis of Variance Table

Model 1: yield ~ nitrogen
Model 2: yield ~ factor(nitrogen)
  Res.Df     RSS Df Sum of Sq      F    Pr(>F)
1     42 17699.2
2     37  8186.8  5    9512.4 8.5982 1.774e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- Plot



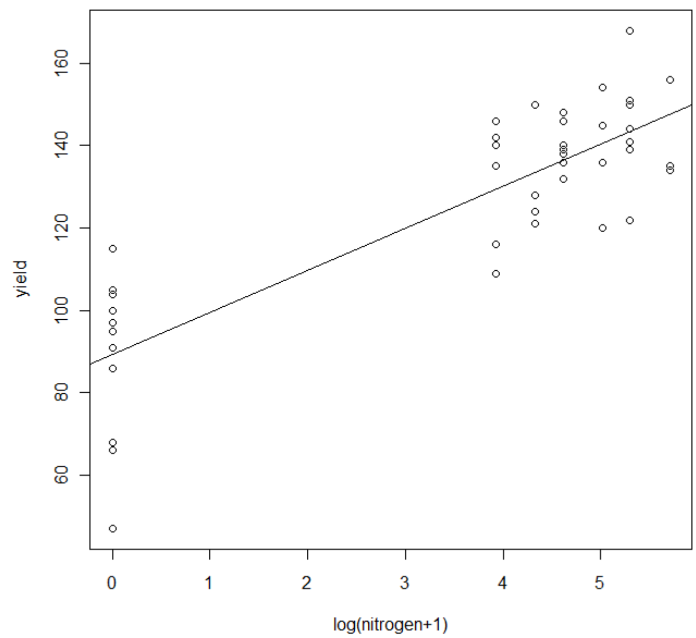
- Goodness of fit test
  - p-value < 0.05 , reject null (pure error sd is substantially less than the regression standard error)
  - Conclude that there is a lack of fit

b) Fit Yield ~ log(nitrogen+1)

- Code:

```
> g.b = lm(yield ~ log(nitrogen+1), cornnit)
>
> plot(yield ~ log(nitrogen+1),xlab="log(nitrogen+1)",ylab="yield")
> abline(g.b)
>
> g2=lm(yield~factor(log(nitrogen+1)))
> anova(g.b, g2)
Analysis of Variance Table

Model 1: yield ~ log(nitrogen + 1)
Model 2: yield ~ factor(log(nitrogen + 1))
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1     42 8633.5
2     37 8186.8  5    446.72 0.4038  0.843
```

- Plot



- Goodness of fit test
  - p-value >0.05 , no evidence to reject null (pure error sd is not substantially less than the regression standard error)
  - Conclude that there is no evidence of lack of fit

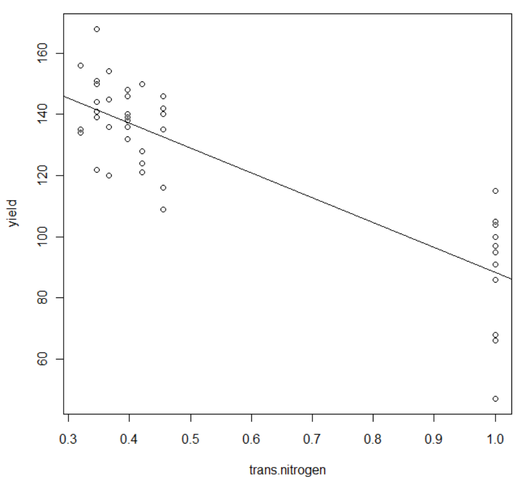
c) Box-Tidwell transformation for Yield ~ (nitrogen+1)

- Code:

```
> new.nitrogen=nitrogen+1;
> boxTidwell(yield ~ new.nitrogen)
 Score Statistic p-value MLE of lambda
       -6.381925       0    -0.1840455

iterations =  6
> trans.nitrogen = new.nitrogen^-.2
>
> g.c=lm(yield~trans.nitrogen,cornnit)
>
> plot(yield ~ trans.nitrogen,xlab="trans.nitrogen",ylab="yield")
> abline(g.c)
>
> g3=lm(yield~factor(trans.nitrogen))
> anova(g.c, g3)
Analysis of Variance Table

Model 1: yield ~ trans.nitrogen
Model 2: yield ~ factor(trans.nitrogen)
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1     42 8463.9
2     37 8186.8  5    277.14 0.2505 0.9369
```

  - Suggested transformation: predictor to the power -0.184
  - Round to -0.2 for better interpretation
  - Transformation of predictor: $(nitrogen + 1)^{-0.2}$

- Plot



- Goodness of fit test
  - p-value >0.05 , no evidence to reject null (pure error sd is not substantially less than the regression standard error)
  - Conclude that there is no evidence of lack of fit

d) Comparison

- Transformation (c) ($\mathrm{R}^2 = 0.711$) gives a higher R^2 than transformation (b) ($\mathrm{R}^2 = 0.705$)
- However, the difference is not very significant
- For the sake interpretability, transformation (b) is recommended

```
> summary(g.b)$r.sq
[1] 0.7054787
> summary(g.c)$r.sq
[1] 0.7112635
```

PROBLEM 4

- Fit model

```
> dim(prostate)   #dimension
[1] 97  9
> names(prostate) #variable names
[1] "lcavol"  "lweight" "age"     "lbph"    "svi"     "lcp"     "gleason" "pgg45"   "lpsa"
> prostate[1:5,]  #first 5 observations
      lcavol lweight age      lbph svi      lcp gleason pgg45     lpsa
1 -0.5798185  2.7695  50 -1.386294   0 -1.38629       6     0 -0.43078
2 -0.9942523  3.3196  58 -1.386294   0 -1.38629       6     0 -0.16252
3 -0.5108256  2.6912  74 -1.386294   0 -1.38629       7    20 -0.16252
4 -1.2039728  3.2828  58 -1.386294   0 -1.38629       6     0 -0.16252
5  0.7514161  3.4324  62 -1.386294   0 -1.38629       6     0  0.37156
> prostate$svi=as.factor(prostate$svi); #categorical
>
> n=dim(prostate)[1]; #number of observations
> p=dim(prostate)[2]; #number of parameters including intercept
> fullfit=lm(lpsa~., data=prostate) # full regression model
> summary(fullfit)

Call:
lm(formula = lpsa ~ ., 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 .
svi1         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
```

- Model selection: level-wise searching algorithm with AIC, BIC, and Cp

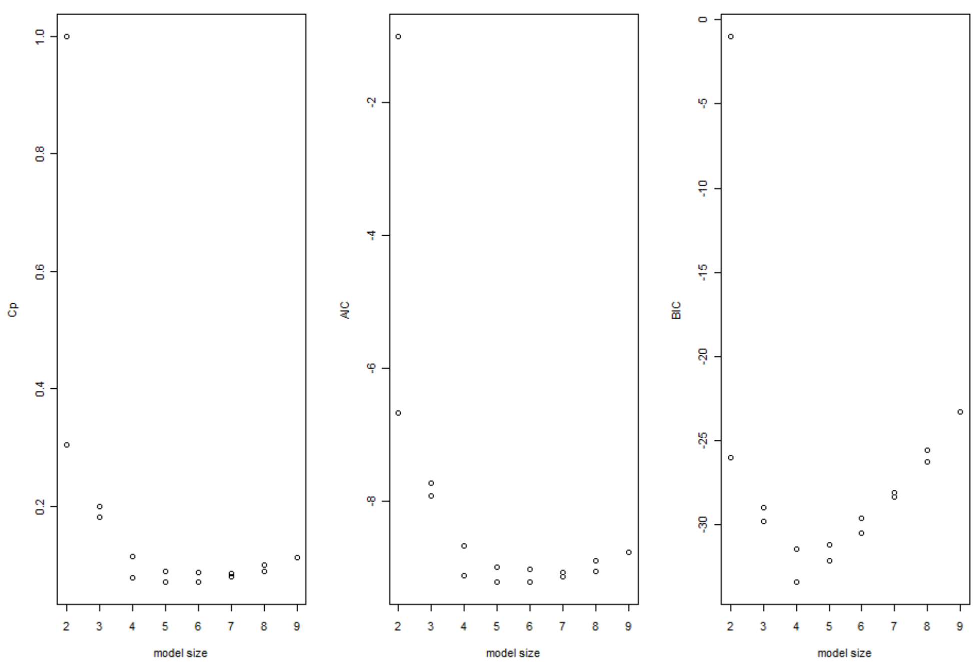
```
> # Compute RSS using regsubsets function with the following inputs
> #Model matrix (with no intercept column)
> #Data (response)
> #Include intercept
> #Number of subsets of each size to record = 2
> #Maximum size of subsets to examine = p
> #Exhaustive Search
> #Use exhaustive search
>
> library(leaps)
> RSSleaps=regsubsets(model.matrix(fullfit)[,-1],
+ prostate[,p],int=T,nbest=2,nvmax=p, really.big=T,method=c("ex"))
>
> sumleaps=summary(RSSleaps,matrix=T)
> # performs an exhaustive search over models, and gives back the best 2 models
> # (with low RSS) of each size.
>
> names(sumleaps) # components returned by summary(RSSleaps)
[1] "which"  "rsq"    "rss"    "adjr2"  "cp"     "bic"    "outmat" "obj"
>
> sumleaps$which # A logical matrix indicating which elements are in each model
  (Intercept) lcavol lweight   age  lbph  svi1   lcp gleason pgg45
1        TRUE   TRUE   FALSE FALSE FALSE FALSE FALSE   FALSE FALSE
1        TRUE  FALSE   FALSE FALSE FALSE  TRUE FALSE   FALSE FALSE
2        TRUE   TRUE    TRUE FALSE FALSE FALSE FALSE   FALSE FALSE
2        TRUE   TRUE   FALSE FALSE FALSE  TRUE FALSE   FALSE FALSE
3        TRUE   TRUE    TRUE FALSE FALSE  TRUE FALSE   FALSE FALSE
3        TRUE   TRUE   FALSE FALSE  TRUE  TRUE FALSE   FALSE FALSE
4        TRUE   TRUE    TRUE FALSE  TRUE  TRUE FALSE   FALSE FALSE
4        TRUE   TRUE    TRUE FALSE FALSE  TRUE FALSE   FALSE  TRUE
5        TRUE   TRUE    TRUE  TRUE  TRUE  TRUE FALSE   FALSE FALSE
5        TRUE   TRUE    TRUE FALSE  TRUE  TRUE FALSE   FALSE  TRUE
6        TRUE   TRUE    TRUE  TRUE  TRUE  TRUE FALSE   FALSE  TRUE
6        TRUE   TRUE    TRUE  TRUE  TRUE  TRUE FALSE    TRUE FALSE
7        TRUE   TRUE    TRUE  TRUE  TRUE  TRUE  TRUE   FALSE  TRUE
7        TRUE   TRUE    TRUE  TRUE  TRUE  TRUE  TRUE    TRUE FALSE
8        TRUE   TRUE    TRUE  TRUE  TRUE  TRUE  TRUE    TRUE  TRUE
```

- Calculate Cp, AIC and BIC for all models

```
> # Create vector of model sizes
> # (include the intercept, so model size = 2 is intercept + 1 predictor)
> msize=apply(sumleaps$which,1,sum);
> msize=as.numeric(msize)
>
> # Calculate Cp, AIC and BIC for all models
>
> Cp=sumleaps$rss/(summary(fullfit)$sigma^2) + 2*msize - n;
> AIC = n*log(sumleaps$rss/n) + 2*msize;
> BIC = n*log(sumleaps$rss/n) + msize*log(n);
```

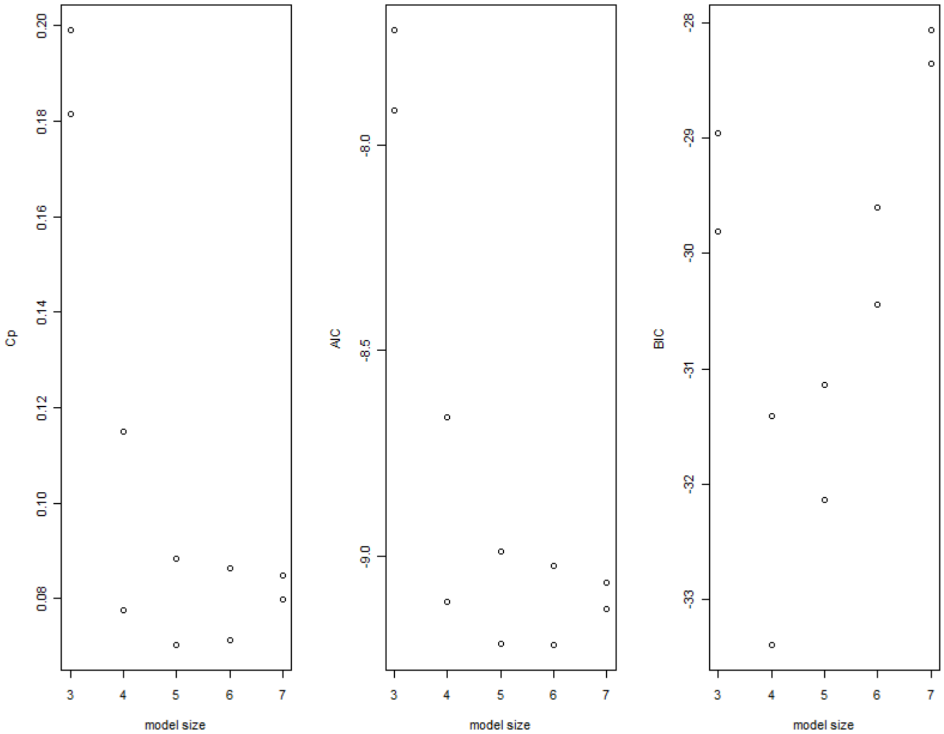
- Plots (note model size includes intercept)

```
> # Plot Criterias (scale by maximum value) vs model size
> par(mfrow=c(1,3));
>
> plot(msize, Cp/abs(max(Cp)), xlab="model size", ylab="Cp")
> plot(msize, AIC/abs(max(AIC)), xlab="model size", ylab="AIC")
> plot(msize, BIC/abs(max(BIC)), xlab="model size", ylab="BIC")
```



- Plots with zoom in

```
> # Plot Criterias vs model size with zoom
> par(mfrow=c(1,3));
> plot(msize[msize>=3 & msize<=7], Cp[msize>=3 & msize<=7]/abs(max(Cp)), xlab="model size", ylab="Cp")
> plot(msize[msize>=3 & msize<=7], AIC[msize>=3 & msize<=7]/abs(max(AIC)), xlab="model size", ylab="AIC")
> plot(msize[msize>=3 & msize<=7], BIC[msize>=3 & msize<=7]/abs(max(BIC)), xlab="model size", ylab="BIC")
```



- It looks like the sizes of the best models are:
  - Cp: model size = 5, with 4 predictors
  - AIC: model size = 6, with 5 predictors
  - BIC: model size = 4, with 3 predictors

- Find best model returned by Cp, AIC, and BIC

```
> # To find model for each criteria:
> #Find the location of the minimum value of the criteria
> #Find the model in the RSS matrix corresponding to this location
> #Find the variables names corresponding to this model
>
> # Cp
> varid.Cp =sumleaps$which[order(Cp)[1],]
> model.Cp = names(prostate)[1:p-1][varid.Cp[-1]]
>
> # AIC
> varid.AIC=sumleaps$which[order(AIC)[1],]
> model.AIC=names(prostate)[1:p-1][varid.AIC[-1]]
>
> #BIC
> varid.BIC=sumleaps$which[order(BIC)[1],]
> model.BIC=names(prostate)[1:p-1][varid.BIC[-1]]
>
> # Models
> model.Cp
[1] "lcavol"  "lweight" "lbph"    "svi"
> model.AIC
[1] "lcavol"  "lweight" "age"     "lbph"    "svi"
> model.BIC
[1] "lcavol"  "lweight" "svi"
```

  - Selected variables for best models:
    - Cp: lcavol, lweight, lbph, svi
    - AIC: lcavol, lweight, age, lbph, svi
    - BIC: lcavol, lweight, svi

PROBLEM 5

- Backward elimination (from Faraway)
  1. Start with all the predictors in the model
  2. Remove the predictor with highest p-value greater than α_{crit}
  3. Refit the model and go to 2
  4. Stop when all p-values are less than α_{crit} .
  - Fit linear, quadratic terms plus interaction term

```
> names(trees)
[1] "Girth"  "Height" "Volume"
>
> fullfit = lm( log(Volume) ~ Girth + Height +
+     I(Girth^2) + I(Height^2) +
+           Girth:Height , data=trees)
> summary(fullfit)

Call:
lm(formula = log(Volume) ~ Girth + Height + I(Girth^2) + I(Height^2) +
    Girth:Height, data = trees)

Residuals:
      Min        1Q    Median        3Q       Max
-0.159718 -0.041905 -0.003371  0.055167  0.133780

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)  -1.9660208  2.0066922  -0.980  0.33660
Girth         0.2808126  0.0786856   3.569  0.00149 **
Height        0.0484196  0.0567321   0.853  0.40150
I(Girth^2)   -0.0042410  0.0032183  -1.318  0.19953
I(Height^2)  -0.0002022  0.0004186  -0.483  0.63326
Girth:Height -0.0001975  0.0018089  -0.109  0.91395
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.08469 on 25 degrees of freedom
Multiple R-squared: 0.9784,     Adjusted R-squared: 0.9741
F-statistic: 226.7 on 5 and 25 DF,  p-value: < 2.2e-16
```

    - Eliminate interaction term
  - Model A: Fit linear and quadratic terms

```
> fit.A = lm( log(Volume) ~ Girth + Height + I(Girth^2) + I(Height^2), data=trees)
> summary(fit.A)

Call:
lm(formula = log(Volume) ~ Girth + Height + I(Girth^2) + I(Height^2),
    data = trees)

Residuals:
     Min       1Q   Median       3Q      Max
-0.16094 -0.04023 -0.00295  0.05474  0.13434

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.9434909  1.9577536  -0.993  0.33000
Girth        0.2738856  0.0456299   6.002 2.45e-06 ***
Height       0.0489426  0.0554449   0.883  0.38547
I(Girth^2)  -0.0045434  0.0016059  -2.829  0.00887 **
I(Height^2) -0.0002220  0.0003699  -0.600  0.55349
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.08307 on 26 degrees of freedom
Multiple R-squared: 0.9784,     Adjusted R-squared: 0.9751
F-statistic: 294.5 on 4 and 26 DF,  p-value: < 2.2e-16
```

    - Need to compare 2 more cases, when the quadratic term of Girth is removed, and when the quadratic term of Height is removed to draw conclusions of significance.

- Model B: Fit all linear and quadratic term of Girth only

```
> fit.B = lm( log(Volume) ~ Girth + Height + I(Girth^2) , data=trees)
> summary(fit.B)

Call:
lm(formula = log(Volume) ~ Girth + Height + I(Girth^2), data = trees)

Residuals:
      Min        1Q    Median        3Q       Max 
-0.174348 -0.043284 -0.000147  0.059198  0.138282 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -0.783931   0.315182  -2.487  0.01935 *  
Girth        0.285333   0.040960   6.966 1.74e-07 ***
Height       0.015701   0.002759   5.690 4.80e-06 ***
I(Girth^2)  -0.004954   0.001435  -3.451  0.00185 ** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.08208 on 27 degrees of freedom
Multiple R-squared: 0.9781,	Adjusted R-squared: 0.9757 
F-statistic: 402.1 on 3 and 27 DF,  p-value: < 2.2e-16
```

- Model C: Fit all linear and quadratic term of Height only

```
> fit.C = lm( log(Volume) ~ Girth + Height + I(Height^2) , data=trees)
> summary(fit.C)

Call:
lm(formula = log(Volume) ~ Girth + Height + I(Height^2), data = trees)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.15193 -0.05238 -0.01024  0.05430  0.19418 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -3.6069925  2.0956378  -1.721   0.0967 .  
Girth        0.1457870  0.0063525  22.950   <2e-16 ***
Height       0.1162199  0.0562091   2.068   0.0484 *  
I(Height^2) -0.0006680  0.0003755  -1.779   0.0865 .  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.09322 on 27 degrees of freedom
Multiple R-squared: 0.9718,	Adjusted R-squared: 0.9686 
F-statistic: 309.7 on 3 and 27 DF,  p-value: < 2.2e-16
```

- Summary (look at corresponding p-values to determine significance)

	Girth^2	Height^2
Model A (Girth^2 & Height^2)	✓	×
Model B (Girth^2)	✓	•
Model C (Height^2)	•	×

  - This shows that whenever Girth^2 is in the model, it is significant, regardless of the presence of Height^2. On the other hand, Height^2 is never significant.
  - Conclusion: stay with model B by dropping Height^2 from model A
  - Since all coefficients significant in model B, stop.
  - Simplified model (plus intercept) : log (Volume) ~ Girth + Height + Height^2

- Alternatively, AIC gives the same result

```
> n=length(trees[,1])
> step(fullfit, direction="both")
Start:  AIC=-147.73
log(Volume) ~ Girth + Height + I(Girth^2) + I(Height^2) + Girth:Height

               Df Sum of Sq     RSS     AIC
- Girth:Height  1 0.0000855 0.17941 -149.72
- I(Height^2)   1 0.0016737 0.18100 -149.44
<none>                      0.17932 -147.73
- I(Girth^2)    1 0.0124560 0.19178 -147.65

Step:  AIC=-149.71
log(Volume) ~ Girth + Height + I(Girth^2) + I(Height^2)

               Df Sum of Sq     RSS     AIC
- I(Height^2)   1  0.002487 0.18189 -151.29
- Height        1  0.005377 0.18478 -150.80
<none>                      0.17941 -149.72
+ Girth:Height  1  0.000085 0.17932 -147.73
- I(Girth^2)    1  0.055232 0.23464 -143.39
- Girth         1  0.248603 0.42801 -124.76

Step:  AIC=-151.29
log(Volume) ~ Girth + Height + I(Girth^2)

               Df Sum of Sq     RSS     AIC
<none>                      0.18189 -151.29
+ I(Height^2)   1   0.00249 0.17941 -149.72
+ Girth:Height  1   0.00090 0.18100 -149.44
- I(Girth^2)    1   0.08025 0.26214 -141.96
- Height        1   0.21815 0.40004 -128.85
- Girth         1   0.32692 0.50881 -121.40

Call:
lm(formula = log(Volume) ~ Girth + Height + I(Girth^2), data = trees)

Coefficients:
(Intercept)        Girth       Height   I(Girth^2)
  -0.783931     0.285333     0.015701    -0.004954
```