Stat_364_HW_Five

Chiayu Tu (Louis Tu)

2022-11-14

Question One

Researchers at National Institutes of Standards and Technology (NIST) collected pipeline data on ultrasonic measurements of the depth of defects in the Alaska pipeline in the field. The depth of the defects were then remeasured in the laboratory. These measurements were performed in six different batches. It turns out that this batch effect is not significant and so can be ignored in the analysis that follows. The laboratory measurements are more accurate than the in-field measurements, but more time consuming and expensive. We want to develop a regression equation for correcting the in-field measurements.

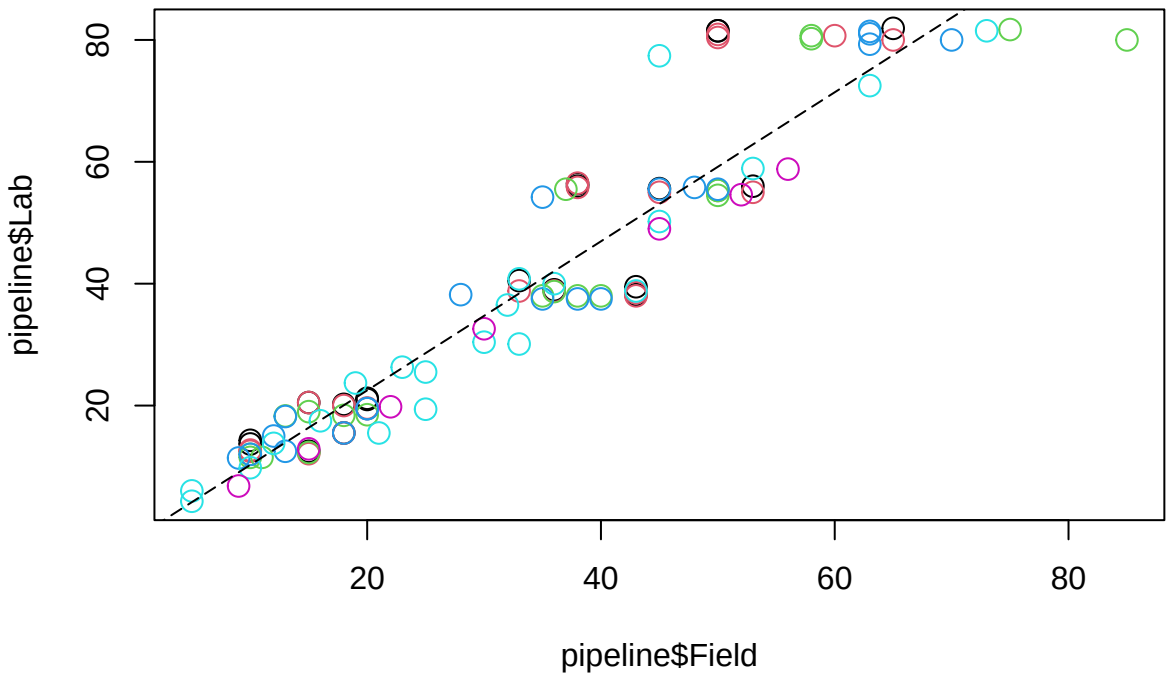
a

Fit a regression model Lab ~ Field. Check for non-constant variance.

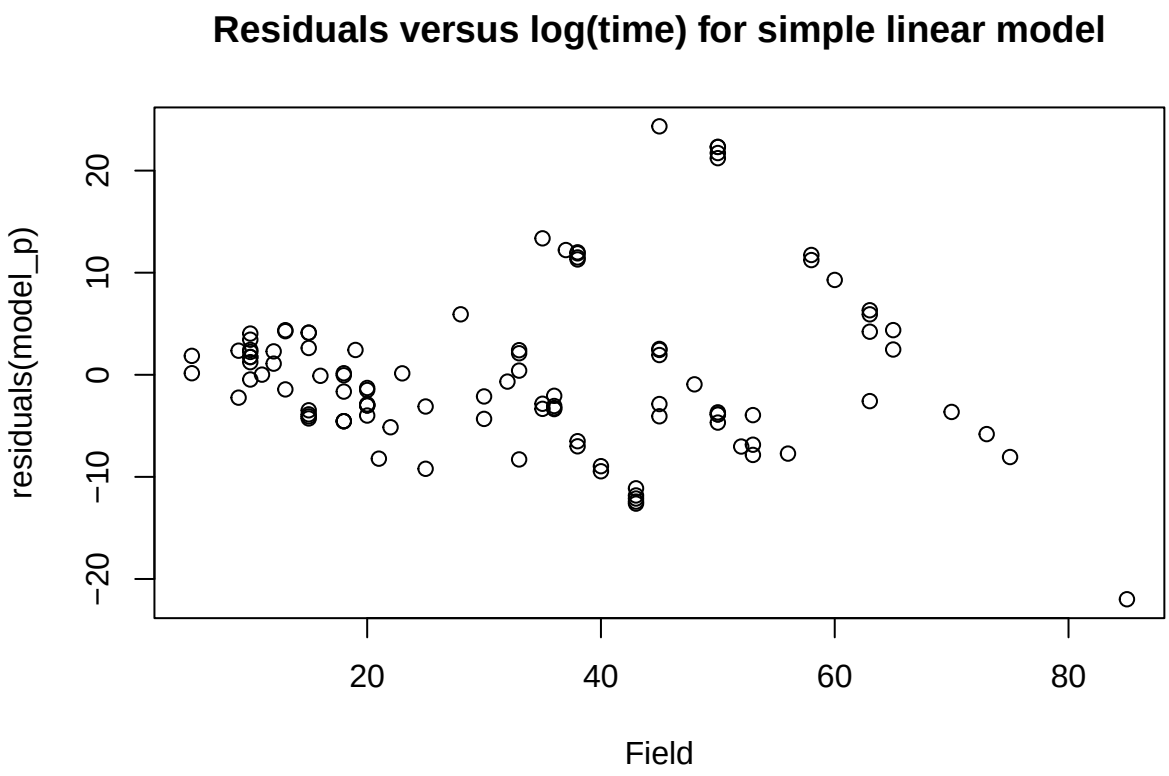
```
model_p <- lm(Lab ~Field, data = pipeline)
summary(model_p)
```

```
##
## Call:
## lm(formula = Lab ~ Field, data = pipeline)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -21.985  -4.072  -1.431   2.504  24.334
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.96750    1.57479  -1.249    0.214
## Field        1.22297    0.04107  29.778   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.865 on 105 degrees of freedom
## Multiple R-squared:  0.8941, Adjusted R-squared:  0.8931
## F-statistic: 886.7 on 1 and 105 DF,  p-value: < 2.2e-16
```

```
plot(pipeline$Field, pipeline$Lab, col = as.factor(pipeline$Batch), cex=1.5)
abline(coef(model_p),lty=5)
```




```
plot(residuals(model_p) ~ Field, pipeline, main ="Residuals versus log(time) for simple linear model")
```



b

We wish to use weights to account for the non-constant variance. Here we split the range of Field into 12 groups of size nine (except for the last group which has only eight values). Within each group, we compute the variance of Lab as varlab and the mean of Field as meanfield. Supposing pipeline is the name of your data frame, the following R code will make the needed computations:

c

An alternative to weighting is transformation. Find transformations on Lab and/or Field so that in the transformed scale the relationship is approximately linear with constant variance. You may restrict your choice of transformation to square root, log and inverse.

Question Two

Using the ozone data, fit a model with O3 as the response and temp, humidity and ibh as predictors. Use the Box–Cox method to determine the best transformation on the response.

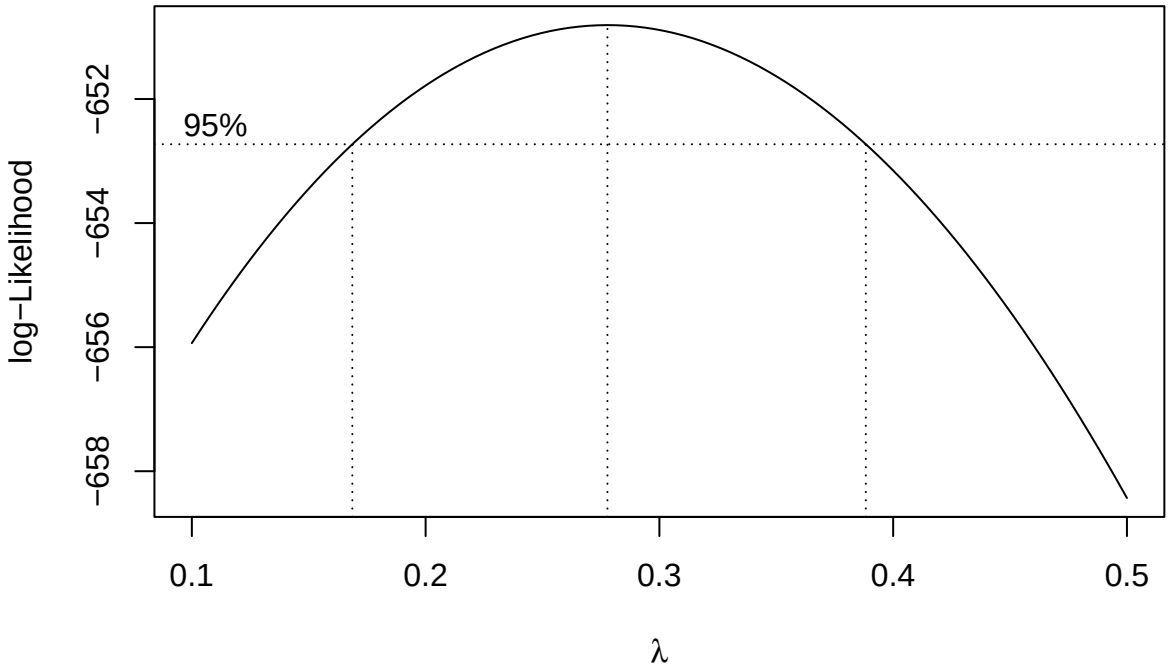
```
require(MASS)
```

```
## Loading required package: MASS
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
data(ozone)

model_o <- lm(O3 ~ temp + humidity + ibh, data = ozone)
boxcox(model_o, plotit=T, lambda=seq(0.1, 0.5, by=0.1))
```



Question Three

Use the prostate data with lpsa as the response and the other variables as predictors. Implement the following variable selection methods to determine the "best" model:

a

Backward elimination

```
model_l <- lm(lpsa ~ ., prostate)
summary(model_l)
```

```
##
## 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 .
## 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
```

```
model_l <- update(model_l, . ~ . - gleason)
summary(model_l)
```

```
##
## Call:
## lm(formula = lpsa ~ lcavol + lweight + age + lbph + svi + lcp +
##     pgg45, data = prostate)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.73117 -0.38137 -0.01728  0.43364  1.63513
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.953926   0.829439   1.150  0.25319
## lcavol       0.591615   0.086001   6.879 8.07e-10 ***
## lweight      0.448292   0.167771   2.672  0.00897 **
## age         -0.019336   0.011066  -1.747  0.08402 .
## lbph         0.107671   0.058108   1.853  0.06720 .
## svi          0.757734   0.241282   3.140  0.00229 **
## lcp         -0.104482   0.090478  -1.155  0.25127
## pgg45        0.005318   0.003433   1.549  0.12488
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7048 on 89 degrees of freedom
## Multiple R-squared:  0.6544, Adjusted R-squared:  0.6273
## F-statistic: 24.08 on 7 and 89 DF,  p-value: < 2.2e-16
```

```r
model_l <- update(model_l, . ~ . - lcp)
summary(model_l)
```

```
##
## Call:
## lm(formula = lpsa ~ lcavol + lweight + age + lbph + svi + pgg45,
##     data = prostate)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.77711 -0.41708  0.00002  0.40676  1.59681
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.980085   0.830665   1.180  0.24116
## lcavol       0.545770   0.076431   7.141 2.31e-10 ***
## lweight      0.449450   0.168078   2.674  0.00890 **
## age         -0.017470   0.010967  -1.593  0.11469
## lbph         0.105755   0.058191   1.817  0.07249 .
## svi          0.641666   0.219757   2.920  0.00442 **
## pgg45        0.003528   0.003068   1.150  0.25331
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7061 on 90 degrees of freedom
## Multiple R-squared:  0.6493, Adjusted R-squared:  0.6259
## F-statistic: 27.77 on 6 and 90 DF,  p-value: < 2.2e-16
```

```r
model_l <- update(model_l, . ~ . - pgg45)
summary(model_l)
```

```
##
## Call:
## lm(formula = lpsa ~ lcavol + lweight + age + lbph + svi, data = prostate)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.83505 -0.39396  0.00414  0.46336  1.57888
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.95100    0.83175   1.143 0.255882
## lcavol       0.56561    0.07459   7.583 2.77e-11 ***
## lweight      0.42369    0.16687   2.539 0.012814 *
## age         -0.01489    0.01075  -1.385 0.169528
## lbph         0.11184    0.05805   1.927 0.057160 .
## svi          0.72095    0.20902   3.449 0.000854 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7073 on 91 degrees of freedom
## Multiple R-squared:  0.6441, Adjusted R-squared:  0.6245
## F-statistic: 32.94 on 5 and 91 DF,  p-value: < 2.2e-16
```

```
model_1 <- update(model_1, . ~ . - age)
summary(model_1)
```

```
##
## Call:
## lm(formula = lpsa ~ lcavol + lweight + lbph + svi, data = prostate)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.82653 -0.42270  0.04362  0.47041  1.48530
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.14554    0.59747   0.244  0.80809
## lcavol       0.54960    0.07406   7.422 5.64e-11 ***
## lweight      0.39088    0.16600   2.355  0.02067 *
## lbph         0.09009    0.05617   1.604  0.11213
## svi          0.71174    0.20996   3.390  0.00103 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7108 on 92 degrees of freedom
## Multiple R-squared:  0.6366, Adjusted R-squared:  0.6208
## F-statistic: 40.29 on 4 and 92 DF,  p-value: < 2.2e-16
```

```
model_1 <- update(model_1, . ~ . - lbph)
summary(model_1)
```

```
##
## Call:
## lm(formula = lpsa ~ lcavol + lweight + svi, data = prostate)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.72964 -0.45764  0.02812  0.46403  1.57013
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.26809    0.54350  -0.493  0.62298
## lcavol       0.55164    0.07467   7.388  6.3e-11 ***
## lweight      0.50854    0.15017   3.386  0.00104 **
## svi          0.66616    0.20978   3.176  0.00203 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7168 on 93 degrees of freedom
## Multiple R-squared:  0.6264, Adjusted R-squared:  0.6144
## F-statistic: 51.99 on 3 and 93 DF,  p-value: < 2.2e-16
```

```
sumary(lm(lpsa ~ gleason + lcp + pgg45 + age + lbph, prostate))
```

```
##              Estimate Std. Error t value  Pr(>|t|)
```

```
## (Intercept) 1.6159145  1.5295150  1.0565   0.29354
## gleason     0.1026772  0.2076994  0.4944   0.62225
## lcp         0.3980561  0.0910969  4.3696 3.296e-05
## pgg45       0.0021052  0.0059042  0.3566   0.72224
## age         0.0027526  0.0146683  0.1877   0.85157
## lbph        0.1336152  0.0722865  1.8484   0.06779
##
## n = 97, p = 6, Residual SE = 0.96082, R-Squared = 0.34
```

b

AIC

```
require(leaps)
```

```
## Loading required package: leaps
```

```
## Warning: package 'leaps' was built under R version 4.1.3
```

```
model_l_1 <- regsubsets(lpsa ~., prostate)
rs <- summary(model_l_1)
rs$which
```

```
##   (Intercept) lcavol lweight   age  lbph   svi   lcp gleason pgg45
## 1        TRUE   TRUE   FALSE FALSE FALSE FALSE FALSE   FALSE FALSE
## 2        TRUE   TRUE    TRUE FALSE FALSE FALSE FALSE   FALSE FALSE
## 3        TRUE   TRUE    TRUE FALSE FALSE  TRUE FALSE   FALSE FALSE
## 4        TRUE   TRUE    TRUE FALSE  TRUE  TRUE FALSE   FALSE FALSE
## 5        TRUE   TRUE    TRUE  TRUE  TRUE  TRUE FALSE   FALSE FALSE
## 6        TRUE   TRUE    TRUE  TRUE  TRUE  TRUE FALSE   FALSE  TRUE
## 7        TRUE   TRUE    TRUE  TRUE  TRUE  TRUE  TRUE   FALSE  TRUE
## 8        TRUE   TRUE    TRUE  TRUE  TRUE  TRUE  TRUE    TRUE  TRUE
```

```
AIC <- 50 * log(rs$rss / 50) + (2:8) * 2
```

```
## Warning in 50 * log(rs$rss/50) + (2:8) * 2: longer object length is not a
## multiple of shorter object length
```

```
plot(AIC ~ I(1:8), ylabs = "AIC", xlab = "Number of Predictors")
```

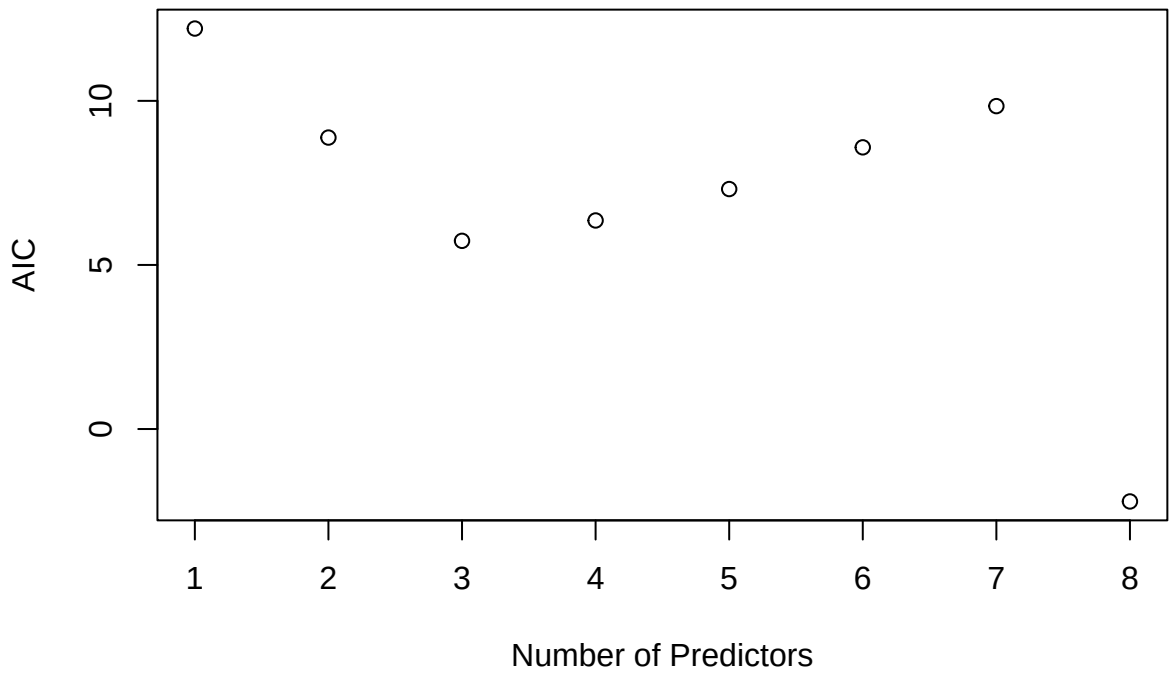
```
## Warning in plot.window(...): "ylabs" is not a graphical parameter
```

```
## Warning in plot.xy(xy, type, ...): "ylabs" is not a graphical parameter
```

```
## Warning in axis(side = side, at = at, labels = labels, ...): "ylabs" is not a
## graphical parameter
```

```
## Warning in axis(side = side, at = at, labels = labels, ...): "ylabs" is not a
## graphical parameter
```

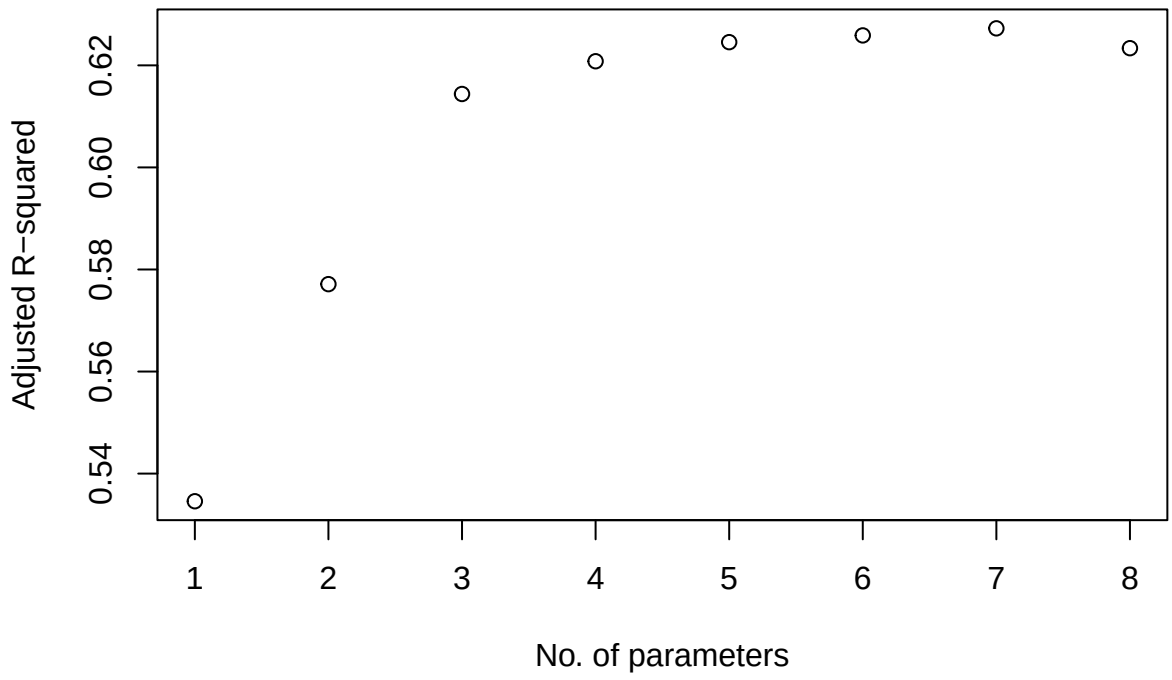
```
## Warning in box(...): "ylabs" is not a graphical parameter

## Warning in title(...): "ylabs" is not a graphical parameter
```



c

Adjusted R^2

```
plot(1:8, rs$adjr2, xlab = "No. of parameters", ylab = "Adjusted R-squared")
```



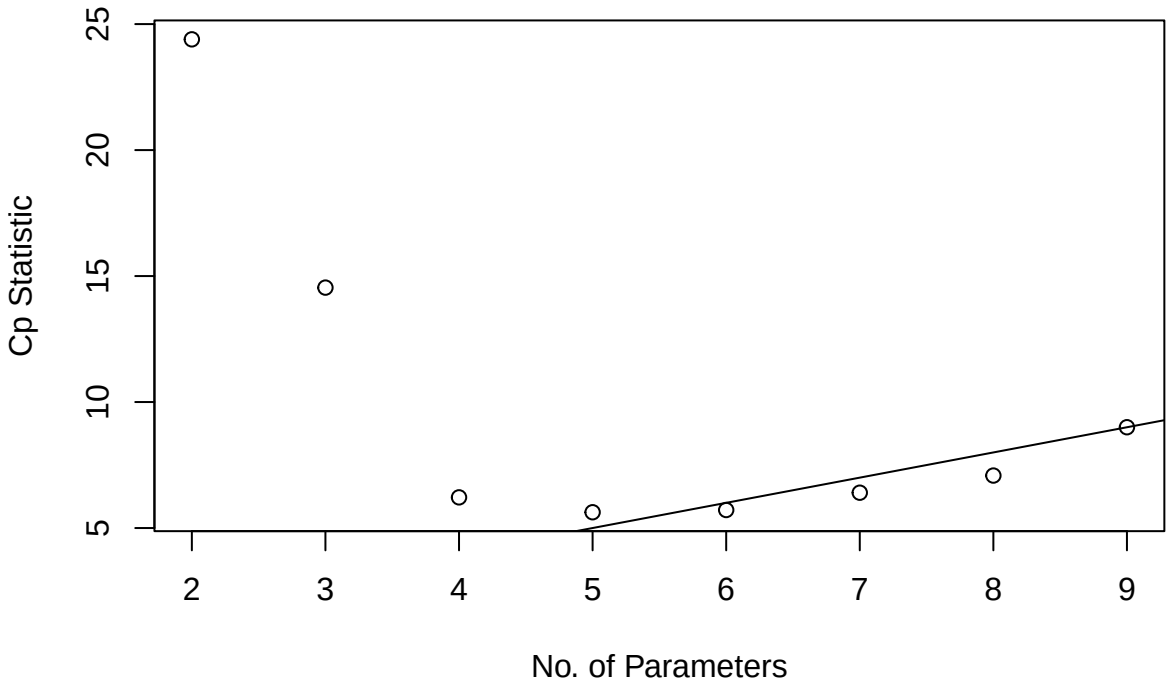

```
which.max(rs$adjr2)
```

```
## [1] 7
```

d

Mallows Cp

```
plot(2:9, rs$cp, xlab="No. of Parameters", ylab="Cp Statistic")
abline(0,1)
```



Question Four

Use the seatpos data with hipcenter as the response.

a

Fit a model with all eight predictors. Comment on the effect of leg length on the response.

```
model_s <- lm(hipcenter ~ ., data=seatpos)
summary(model_s)
```

```
##
## Call:
## lm(formula = hipcenter ~ ., data = seatpos)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -73.827 -22.833  -3.678  25.017  62.337
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 436.43213  166.57162   2.620   0.0138 *
## Age           0.77572    0.57033   1.360   0.1843
## Weight        0.02631    0.33097   0.080   0.9372
```

```
## HtShoes      -2.69241    9.75304  -0.276   0.7845
## Ht            0.60134   10.12987   0.059   0.9531
## Seated        0.53375    3.76189   0.142   0.8882
## Arm          -1.32807    3.90020  -0.341   0.7359
## Thigh        -1.14312    2.66002  -0.430   0.6706
## Leg          -6.43905    4.71386  -1.366   0.1824
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 37.72 on 29 degrees of freedom
## Multiple R-squared:  0.6866, Adjusted R-squared:  0.6001
## F-statistic:  7.94 on 8 and 29 DF,  p-value: 1.306e-05
```

b

Compute a 95% prediction interval for the mean value of the predictors.

```
predict(model_s, interval = "confidence")
```

```
##           fit       lwr        upr
## 1  -230.82470 -263.2205 -198.42890
## 2  -158.22231 -199.0855 -117.35914
## 3   -96.85463 -131.7243  -61.98494
## 4  -255.78273 -294.7597 -216.80574
## 5  -188.59572 -233.3630 -143.82845
## 6  -186.02614 -214.1290 -157.92325
## 7  -153.98285 -189.6664 -118.29935
## 8  -244.79086 -286.3104 -203.27128
## 9  -139.71030 -173.4665 -105.95404
## 10 -112.98566 -148.1345  -77.83680
## 11 -163.72509 -186.4322 -141.01801
## 12  -89.14799 -120.0194  -58.27658
## 13 -194.10261 -249.8912 -138.31401
## 14 -128.43355 -157.7773  -99.08975
## 15 -186.44972 -226.2569 -146.64258
## 16 -177.90902 -217.5495 -138.26853
## 17 -201.58090 -250.4299 -152.73188
## 18  -98.43069 -141.8463  -55.01511
## 19 -145.80244 -174.2207 -117.38415
## 20 -167.75364 -199.5743 -135.93300
## 21 -178.41491 -214.6476 -142.18225
## 22 -279.07627 -336.5054 -221.64716
## 23 -245.56763 -285.6346 -205.50071
## 24  -81.55529 -114.4871  -48.62343
## 25 -141.13605 -167.5849 -114.68722
## 26 -222.49965 -247.7190 -197.28026
## 27 -156.83929 -184.1675 -129.51112
## 28 -128.68145 -170.6894  -86.67351
## 29 -193.00256 -225.2335 -160.77163
## 30  -93.20235 -125.8015  -60.60319
## 31 -102.96051 -160.7042  -45.21677
## 32 -182.39983 -222.0483 -142.75134
## 33 -166.93549 -205.3431 -128.52790
```

```
## 34 -102.63962 -131.3436  -73.93562
## 35 -194.49288 -227.4769 -161.50888
## 36 -142.50545 -185.0056 -100.00534
## 37 -178.52201 -207.8089 -149.23515
## 38 -154.08219 -186.4553 -121.70905
```

c

Use AIC to select a model. Now interpret the effect of leg length and compute the prediction interval. Compare the conclusions from the two models.

```
model_s_1 <- regsubsets(hipcenter ~ ., data=seatpos)
rsc <- summary(model_s_1)
rsc$which
```

```
##   (Intercept)   Age Weight HtShoes    Ht Seated   Arm Thigh   Leg
## 1        TRUE FALSE  FALSE   FALSE  TRUE  FALSE FALSE FALSE FALSE
## 2        TRUE FALSE  FALSE   FALSE  TRUE  FALSE FALSE FALSE  TRUE
## 3        TRUE  TRUE  FALSE   FALSE  TRUE  FALSE FALSE FALSE  TRUE
## 4        TRUE  TRUE  FALSE    TRUE FALSE  FALSE FALSE  TRUE  TRUE
## 5        TRUE  TRUE  FALSE    TRUE FALSE  FALSE  TRUE  TRUE  TRUE
## 6        TRUE  TRUE  FALSE    TRUE FALSE   TRUE  TRUE  TRUE  TRUE
## 7        TRUE  TRUE   TRUE    TRUE FALSE   TRUE  TRUE  TRUE  TRUE
## 8        TRUE  TRUE   TRUE    TRUE  TRUE   TRUE  TRUE  TRUE  TRUE
```

```
AIC_hip <- nrow(seatpos)*log(rsc$rss/nrow(seatpos)) + (2:9)*2
AIC_hip
```

```
## [1] 275.0667 274.7798 274.2418 275.8291 277.6712 279.6389 281.6286 283.6240
```

```
model_s_2 <- lm(hipcenter ~ Age + Ht, data=seatpos)
summary(model_s_2)
```

```
##
## Call:
## lm(formula = hipcenter ~ Age + Ht, data = seatpos)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -91.534 -23.028   2.131  24.994  53.939
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 526.9589    92.2479   5.712 1.85e-06 ***
## Age           0.5211     0.3862   1.349    0.186
## Ht           -4.2004     0.5313  -7.906 2.69e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 35.96 on 35 degrees of freedom
## Multiple R-squared:  0.6562, Adjusted R-squared:  0.6365
## F-statistic:  33.4 on 2 and 35 DF,  p-value: 7.694e-09
```