Homework 4

Rolando Santos

2023-10-30

Question 1: 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:

```
prostate <- read.csv("prostate.csv")
head(prostate)
```

```
##       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
## 6 -1.0498221  3.2288  50 -1.386294   0 -1.38629       6     0  0.76547
```

a. Backward elimination (0.05 cutoff)

```
lm.model <- lm(lpsa ~ ., data = prostate)
summary(lm.model)
```

```
##
## 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
```

We can see that the following parameters are not significant at the 5% level: lcp, gleason, pgg45, age and lbph (age and lbph are significant at the 10% level).

```
#Removing gleason
lm.model <- update(lm.model, .~. -gleason)
summary(lm.model)
```

```
##
## 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
```

```
#Removing lcp
lm.model <- update(lm.model, .~. -lcp)
summary(lm.model)
```

```
##
## 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
```

```
#Removing pgg45
lm.model <- update(lm.model, .~. -pgg45)
summary(lm.model)
```

```
##
## 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
```

```
#Removing age
lm.model <- update(lm.model, .~. -age)
summary(lm.model)
```

```
##
## 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
```

```
#Removing lbph
lm.model <- update(lm.model, .~. -lbph)
summary(lm.model)
```

```
##
## 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
```

After removing all insignifcant predictors, we are left with lcavol, lweight and svi as the remaining predictors.

b. AIC

```
lm.model <- lm(lpsa ~ ., data = prostate)
step(lm.model)
```

```
## Start:  AIC=-58.32
## lpsa ~ lcavol + lweight + age + lbph + svi + lcp + gleason +
##     pgg45
```

```
##
##           Df Sum of Sq    RSS     AIC
## - gleason  1    0.0412 44.204 -60.231
## - pgg45    1    0.5258 44.689 -59.174
## - lcp      1    0.6740 44.837 -58.853
## <none>                 44.163 -58.322
## - age      1    1.5503 45.713 -56.975
## - lbph     1    1.6835 45.847 -56.693
## - lweight  1    3.5861 47.749 -52.749
## - svi      1    4.9355 49.099 -50.046
## - lcavol   1   22.3721 66.535 -20.567
##
## Step:  AIC=-60.23
## lpsa ~ lcavol + lweight + age + lbph + svi + lcp + pgg45
##
##           Df Sum of Sq    RSS     AIC
## - lcp      1    0.6623 44.867 -60.789
## <none>                 44.204 -60.231
## - pgg45    1    1.1920 45.396 -59.650
## - age      1    1.5166 45.721 -58.959
## - lbph     1    1.7053 45.910 -58.560
## - lweight  1    3.5462 47.750 -54.746
## - svi      1    4.8984 49.103 -52.037
## - lcavol   1   23.5039 67.708 -20.872
##
## Step:  AIC=-60.79
## lpsa ~ lcavol + lweight + age + lbph + svi + pgg45
##
##           Df Sum of Sq    RSS     AIC
## - pgg45    1    0.6590 45.526 -61.374
## <none>                 44.867 -60.789
## - age      1    1.2649 46.131 -60.092
## - lbph     1    1.6465 46.513 -59.293
## - lweight  1    3.5647 48.431 -55.373
## - svi      1    4.2503 49.117 -54.009
## - lcavol   1   25.4189 70.285 -19.248
##
## Step:  AIC=-61.37
## lpsa ~ lcavol + lweight + age + lbph + svi
##
##           Df Sum of Sq    RSS     AIC
## <none>                 45.526 -61.374
## - age      1    0.9592 46.485 -61.352
## - lbph     1    1.8568 47.382 -59.497
## - lweight  1    3.2251 48.751 -56.735
## - svi      1    5.9517 51.477 -51.456
## - lcavol   1   28.7665 74.292 -15.871

##
## Call:
## lm(formula = lpsa ~ lcavol + lweight + age + lbph + svi, data = prostate)
##
## Coefficients:
## (Intercept)       lcavol      lweight          age         lbph          svi
```

```
##     0.95100      0.56561      0.42369     -0.01489      0.11184      0.72095
```

The AIC method eliminates gleason, lcp and pgg45 in that order, however keeps age and svi, so the model ends with the following parameters: age, lbph, lweight, svi and lcavol.

```
lm.model <- update(lm.model, .~. -gleason)
lm.model <- update(lm.model, .~. -lcp)
lm.model <- update(lm.model, .~. -pgg45)
summary(lm.model)
```

```
##
## 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
```

c. Do these model selection methods give you the same result? If not, do you think it is an issue that they are different? Give your insight.

The AIC model has more predictors included compared to the Backwards elimination model. The AIC model chose to keep a few of the predictors that are not significant at the 5% level. There is no issue if the models vary, if we do end up with different models then we need to start looking at other aspects to start comparing which of the two models can be considered better i.e. looking at the Adjusted R^2. We can see that the Adjusted R^2 of the AIC model is slightly higher with a value of 0.6245 compared the the Adjusted R^2 of the backwards elimination model with a value of 0.6144.

Question 2: The aatemp data come from the U.S. Historical Climatology Network. They are the annual mean temperatures (in degrees F) in Ann Arbor, Michigan going back about 150 years. Download this dataset from Sakai and answer the following questions.

```
aatemp <- read.csv("aatemp.csv")
head(aatemp)
```

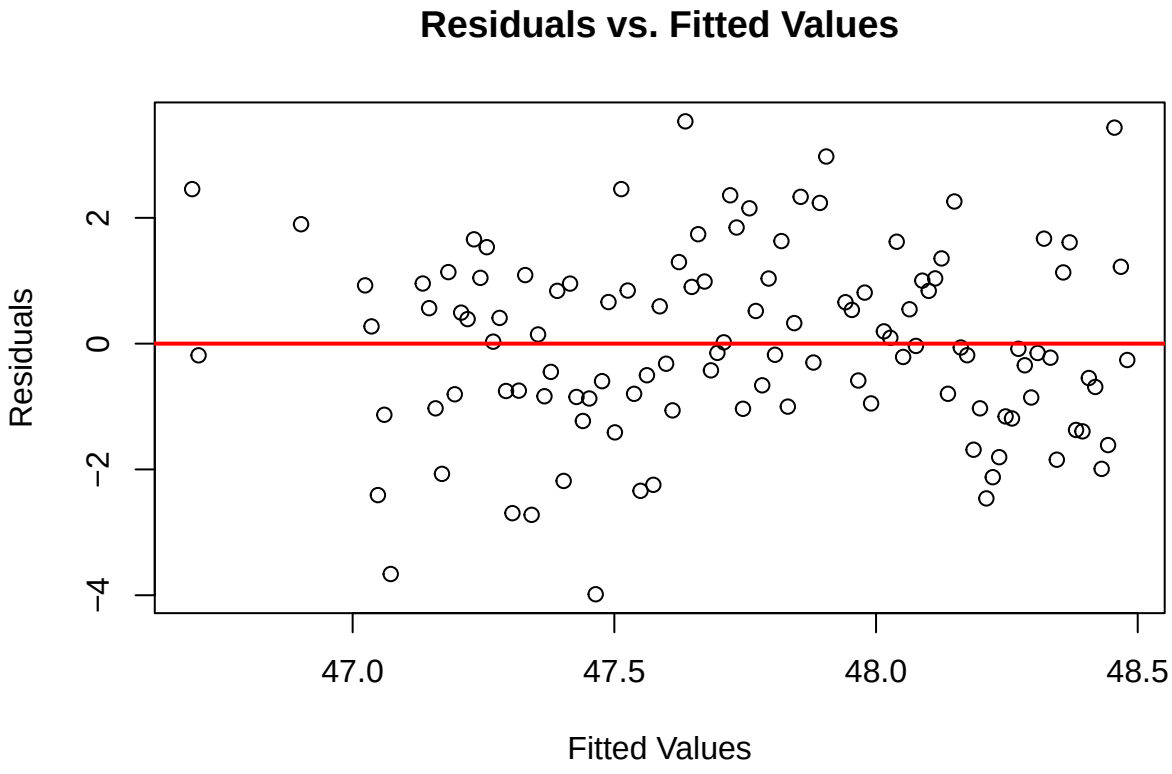
```
##   year  temp
## 1 1854 49.15
## 2 1855 46.52
## 3 1871 48.80
## 4 1881 47.95
## 5 1882 47.31
## 6 1883 44.64
```

a. Fit a linear model of temp~year. Do you think there is a linear trend? (Hint: check plot, parameters, and model goodness of fit)

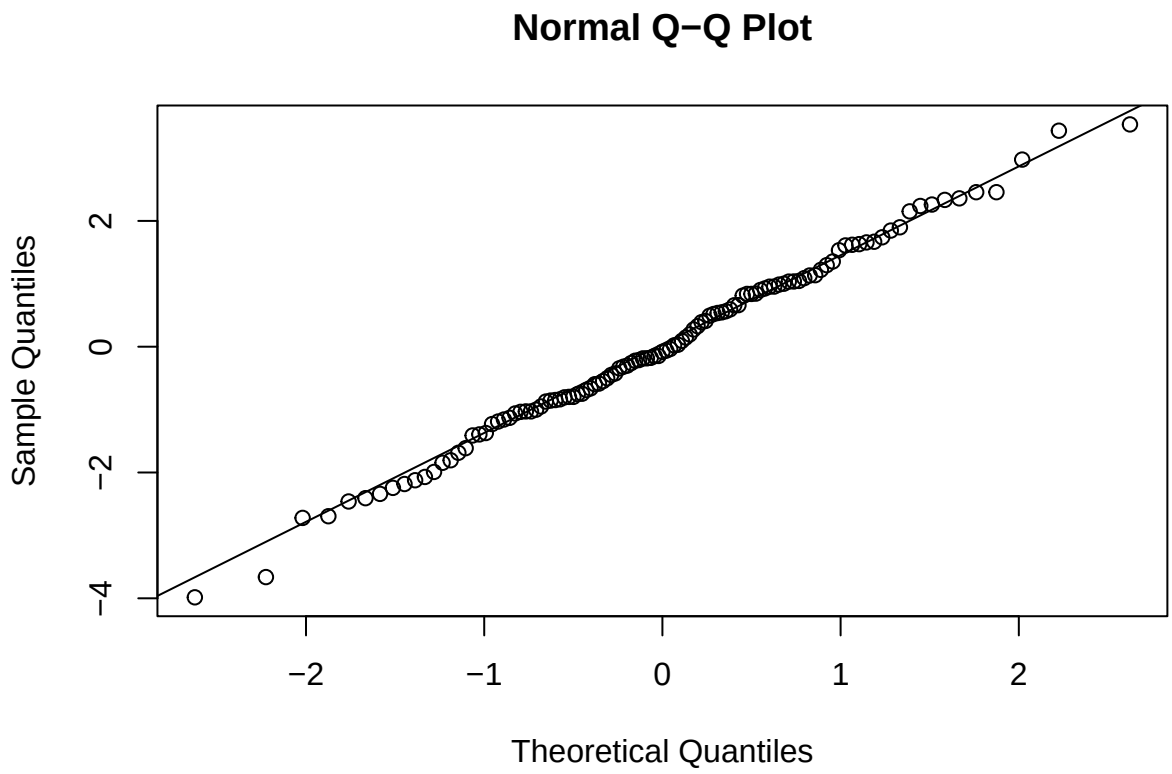
```
lm.model <- lm(temp ~ year, data = aatemp)
summary(lm.model)
```

```
##
## Call:
## lm(formula = temp ~ year, data = aatemp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.9843 -0.9113 -0.0820  0.9946  3.5343
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 24.005510   7.310781   3.284  0.00136 **
## year         0.012237   0.003768   3.247  0.00153 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.466 on 113 degrees of freedom
## Multiple R-squared:  0.08536,    Adjusted R-squared:  0.07727
## F-statistic: 10.55 on 1 and 113 DF,  p-value: 0.001533
```

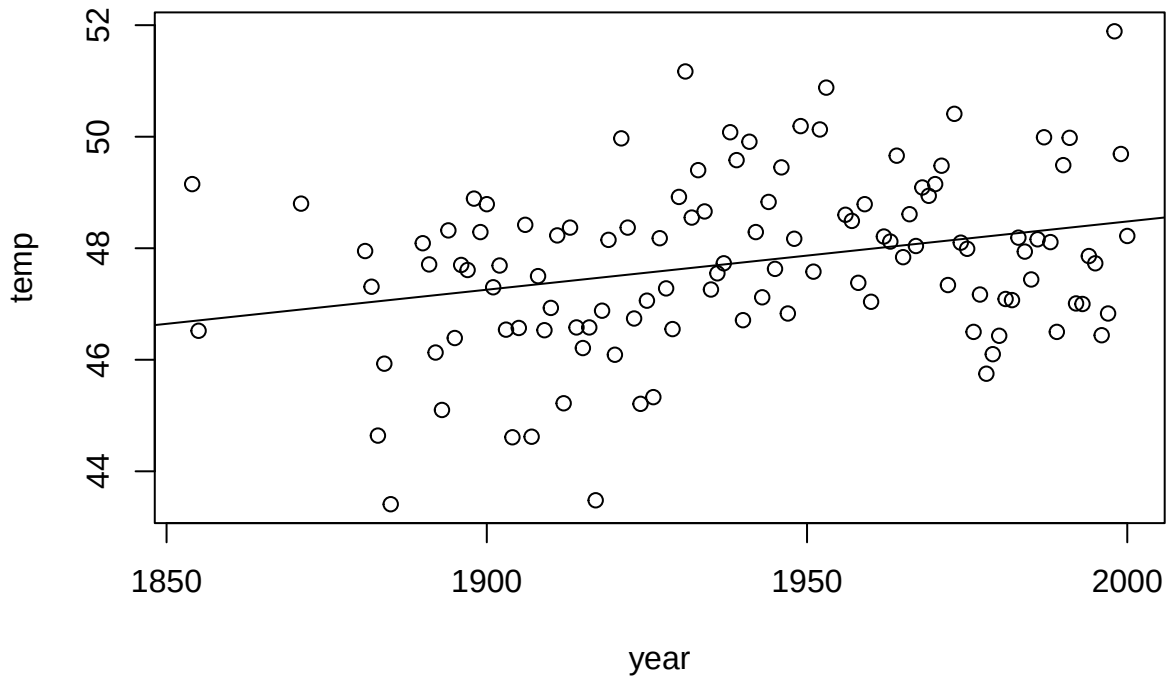
```
plot(lm.model$residuals ~ lm.model$fitted.values, xlab = "Fitted Values",
     ylab = "Residuals", main = "Residuals vs. Fitted Values")
abline(h=0 ,col='red', lwd=2)
```




```
qqnorm(lm.model$residuals)
qqline(lm.model$residuals)
```



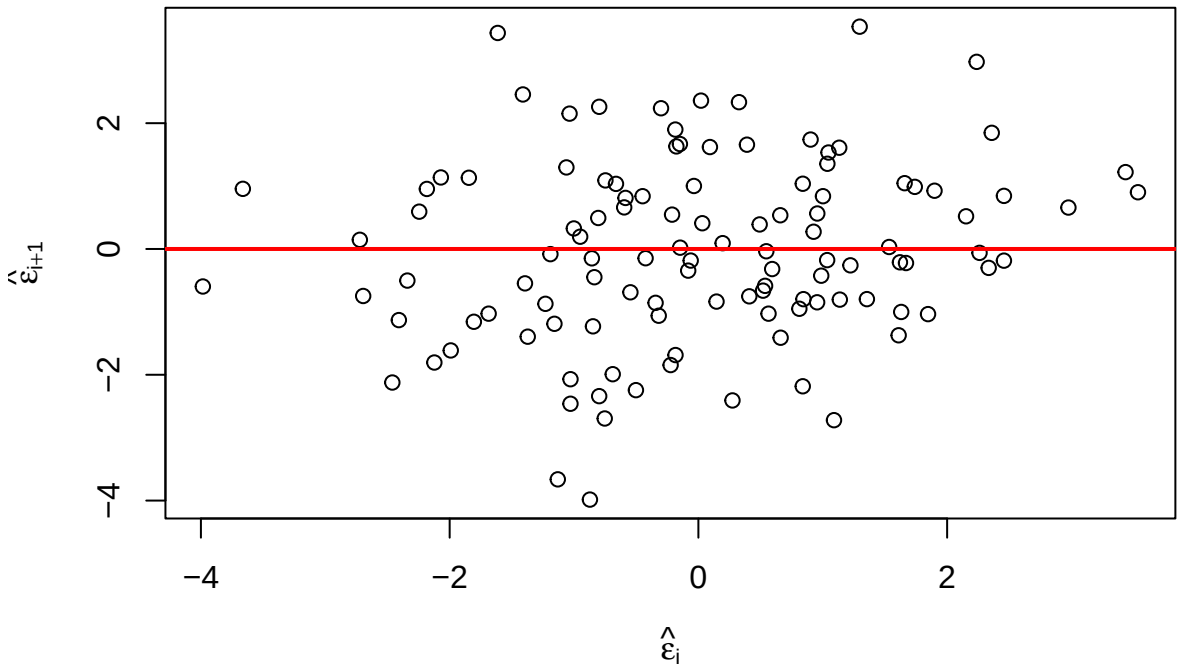
```
plot(temp ~ year, data = aatemp)
abline(coefficients(lm.model))
```



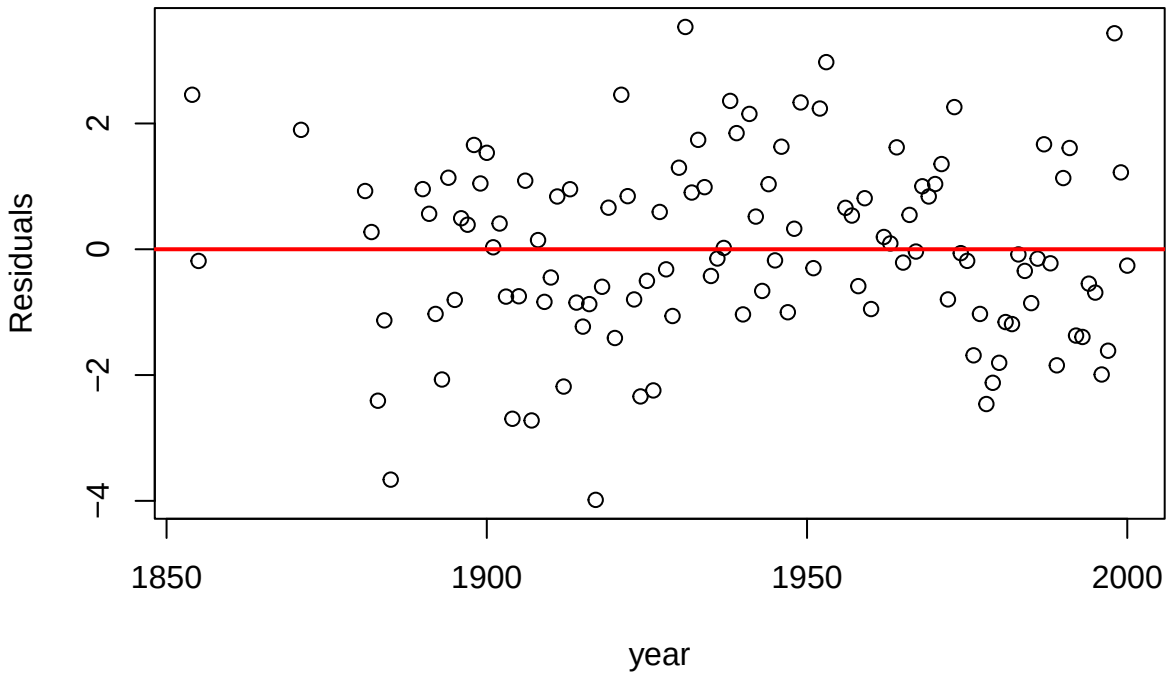
From the Residuals vs. Fitted plot we can see that the trend is linear (points are scattered randomly around the center line), and the q-q plot shows that the residuals do follow the line indicating a linear trend as well. We can also see that data on temp vs. year also follows a linear trend.

b. Observations in successive years may be correlated. Fit a model that estimates this correlation. Does this change your opinion about the trend?

```
n <- dim(aatemp)[1]
plot(tail(lm.model$residuals, n-1) ~ head(lm.model$residuals, n-1), xlab=
expression(hat(epsilon)[i]),ylab=expression(hat(epsilon)[i+1]))
abline(h = 0, col = 'red', lwd = 2)
```




```
plot(lm.model$residuals ~ year, na.omit(aatemp), ylab = "Residuals")
abline(h = 0, col = 'red', lwd = 2)
```




```
cor(tail(lm.model$residuals, n-1), head(lm.model$residuals, n-1))
```

```
## [1] 0.1809103
```

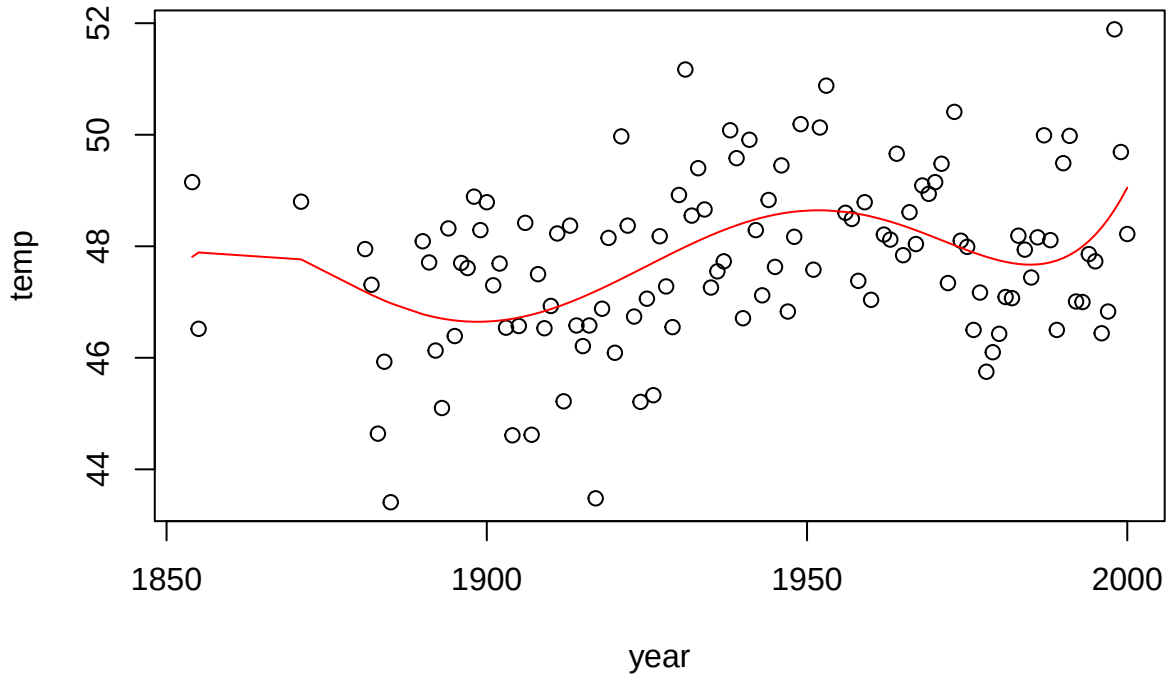
From the correlation value show us that there is a positive serial correlation, however it is small and difficult to point out from viewing the plots.

c. Fit a polynomial model with degree 5. Plot your fitted model on top of the data.

```r
#Using poly() here because using year + I(year^2) + ... + I(year^5) causes
#the I(year^5) to return as NA due to perfect collinearity.
lm.model.poly5 <- lm(temp ~ poly(year, 5), data = aatemp)
summary(lm.model.poly5)
```

```
##
## 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
```

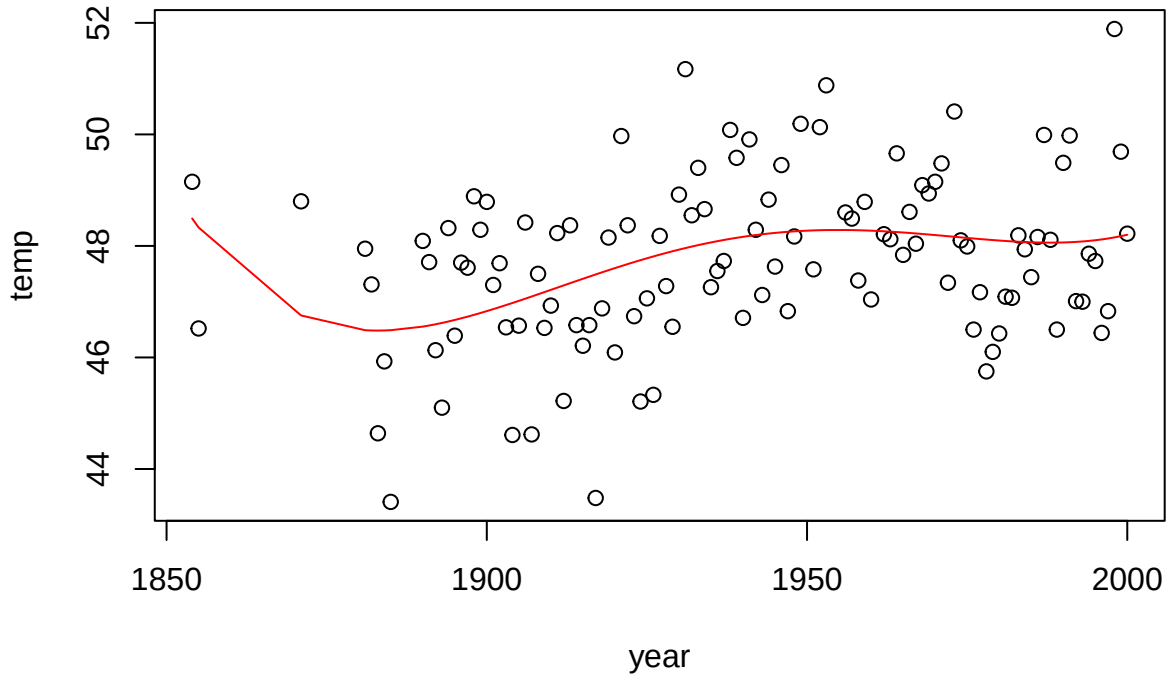
```r
plot(temp ~ year, data = aatemp)
lines(aatemp$year, fitted(lm.model.poly5), col = "red")
```




```
#Here we see that I(year^5) does not have any coefficients due to lm() preventing
#perfectly collinear predictors from having coefficients
lm.model.poly5 <- lm(temp ~ year + I(year^2) + I(year^3) + I(year^4) + I(year^5),
                     data = aatemp)
summary(lm.model.poly5)
```

```
##
## Call:
## lm(formula = temp ~ year + I(year^2) + I(year^3) + I(year^4) +
##     I(year^5), data = aatemp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.0085 -0.9618 -0.0913  0.9926  3.7370
##
## Coefficients: (1 not defined because of singularities)
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.497e+06  8.553e+05   1.750   0.0829 .
## year        -3.086e+03  1.775e+03  -1.739   0.0849 .
## I(year^2)    2.385e+00  1.381e+00   1.727   0.0869 .
## I(year^3)   -8.189e-04  4.773e-04  -1.716   0.0890 .
## I(year^4)    1.054e-07  6.186e-08   1.704   0.0912 .
## I(year^5)           NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.431 on 110 degrees of freedom
## Multiple R-squared:  0.1522, Adjusted R-squared:  0.1213
## F-statistic: 4.936 on 4 and 110 DF,  p-value: 0.001068
```

```r
plot(temp ~ year, data = aatemp)
lines(aatemp$year, fitted(lm.model.poly5), col = "red")
```

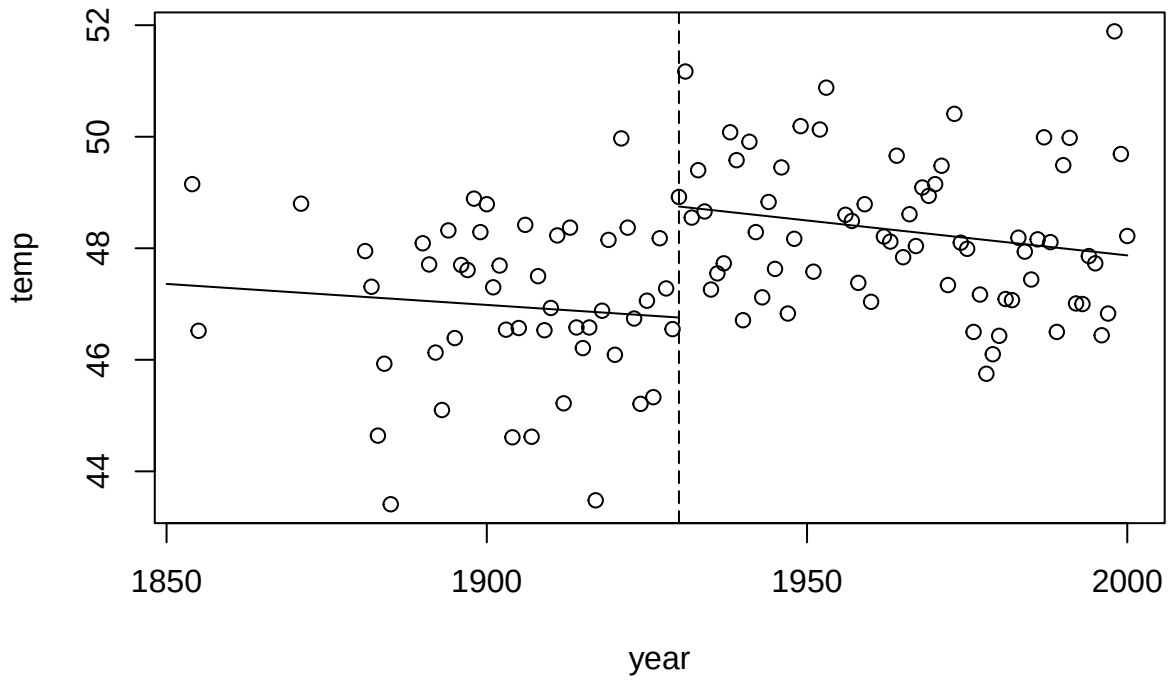


d. Suppose someone claims that the temperature trend was different before and after 1930. Fit a segmented regression model to check this claim.

```r
plot(temp ~ year, data = aatemp)
abline(v=1930, lty=5)

lm.model1 <- lm(temp ~ year, data = aatemp, subset = (year < 1930))
lm.model2 <- lm(temp ~ year, data = aatemp, subset = (year > 1930))

segments(x0 = 1850, y0 = lm.model1$coefficients[1]+lm.model1$coefficients[2]*1850,
         x1 = 1930, y1 = lm.model1$coefficients[1]+lm.model1$coefficients[2]*1930)
segments(x0 = 1930, y0 = lm.model2$coefficients[1]+lm.model2$coefficients[2]*1930,
         x1 = 2000, y1 = lm.model2$coefficients[1]+lm.model2$coefficients[2]*2000)
```



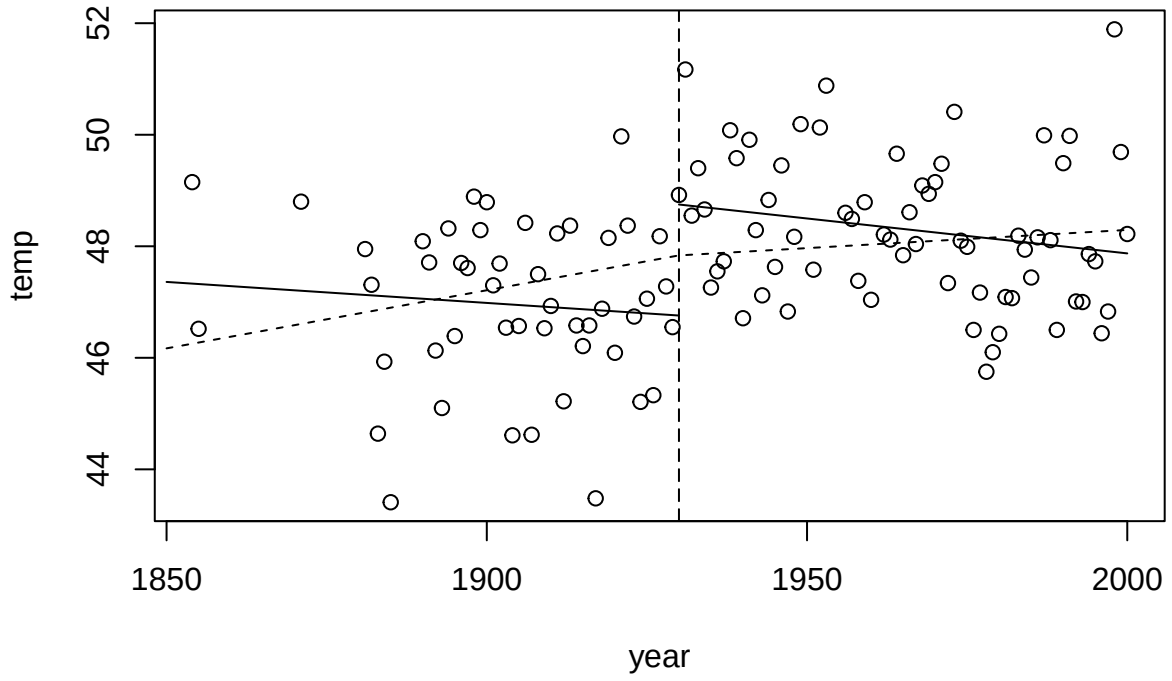

```
bl <- function(x){
  ifelse(x<1930, 1930-x, 0)
}

br <- function(x){
  ifelse(x>1930, x-1930, 0)
}
# fit a segmented model
lm.seg <- lm(temp ~ bl(year) +br (year), data = aatemp)
```

```
plot(temp ~ year, data = aatemp)
abline(v=1930, lty=5)

segments(x0 = 1850, y0 = lm.model1$coefficients[1]+lm.model1$coefficients[2]*1850,
         x1 = 1930, y1 = lm.model1$coefficients[1]+lm.model1$coefficients[2]*1930)
segments(x0 = 1930, y0 = lm.model2$coefficients[1]+lm.model2$coefficients[2]*1930,
         x1 = 2000, y1 = lm.model2$coefficients[1]+lm.model2$coefficients[2]*2000)

x <- seq(1850, 2000, by=1)
y <- lm.seg$coefficients[1]+lm.seg$coefficients[2]*bl(x)+lm.seg$coefficients[3]*br(x)
lines(x,y,lty=2)
```




```
summary(lm.seg)
```

```
##
## Call:
## lm(formula = temp ~ bl(year) + br(year), data = aatemp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.0855 -0.9492 -0.0380  1.0289  3.6096
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 47.836412   0.259571  184.29   <2e-16 ***
## bl(year)    -0.020838   0.009559   -2.18   0.0314 *
## br(year)     0.006529   0.006942    0.94   0.3490
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.467 on 112 degrees of freedom
## Multiple R-squared:  0.09312,    Adjusted R-squared:  0.07693
## F-statistic:  5.75 on 2 and 112 DF,  p-value: 0.004195
```

```
summary(lm.model1)
```

```
##
## Call:
## lm(formula = temp ~ year, data = aatemp, subset = (year < 1930))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6872 -0.7584  0.1109  1.2926  3.1441
```

```
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 61.300439  23.189506   2.643   0.0112 *
## year        -0.007535   0.012181  -0.619   0.5392
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.479 on 46 degrees of freedom
## Multiple R-squared:  0.008249,   Adjusted R-squared:  -0.01331
## F-statistic: 0.3826 on 1 and 46 DF,  p-value: 0.5392
```

```
summary(lm.model2)
```

```
##
## Call:
## lm(formula = temp ~ year, data = aatemp, subset = (year > 1930))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.3978 -0.9834 -0.1354  0.8804  3.9925
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 72.909528  15.287079   4.769 1.11e-05 ***
## year        -0.012519   0.007775  -1.610    0.112
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.302 on 64 degrees of freedom
## Multiple R-squared:  0.03893,    Adjusted R-squared:  0.02392
## F-statistic: 2.593 on 1 and 64 DF,  p-value: 0.1123
```

Question 3: The "longley" dataset includes seven social-economic variables from 1947-1962 in the US. Our goal is to explore the relationship between Employed and other variables. Download this dataset from Sakai and answer the following questions.

```
longley <- read.csv("longley.csv")
head(longley)
```

```
##   GNP.deflator     GNP Unemployed Armed.Forces Population Year Employed
## 1         83.0 234.289      235.6        159.0    107.608 1947   60.323
## 2         88.5 259.426      232.5        145.6    108.632 1948   61.122
## 3         88.2 258.054      368.2        161.6    109.773 1949   60.171
## 4         89.5 284.599      335.1        165.0    110.929 1950   61.187
## 5         96.2 328.975      209.9        309.9    112.075 1951   63.221
## 6         98.1 346.999      193.2        359.4    113.270 1952   63.639
```

```
lm.model <- lm(Employed ~ ., data = longley)
summary(lm.model)
```

```
##
## Call:
## lm(formula = Employed ~ ., data = longley)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.41011 -0.15767 -0.02816  0.10155  0.45539
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -3.482e+03  8.904e+02  -3.911 0.003560 **
## GNP.deflator  1.506e-02  8.492e-02   0.177 0.863141
## GNP          -3.582e-02  3.349e-02  -1.070 0.312681
## Unemployed   -2.020e-02  4.884e-03  -4.136 0.002535 **
## Armed.Forces -1.033e-02  2.143e-03  -4.822 0.000944 ***
## Population   -5.110e-02  2.261e-01  -0.226 0.826212
## Year          1.829e+00  4.555e-01   4.016 0.003037 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3049 on 9 degrees of freedom
## Multiple R-squared:  0.9955, Adjusted R-squared:  0.9925
## F-statistic: 330.3 on 6 and 9 DF,  p-value: 4.984e-10
```

a. Construct a correlation matrix of six predictors in this dataset. Which predictors do you think are highly correlated? What are the potential reasons for those high correlations?

```
round(cor(longley[,-7]), 2)
```

```
##              GNP.deflator  GNP Unemployed Armed.Forces Population Year
## GNP.deflator         1.00 0.99       0.62         0.46       0.98 0.99
## GNP                  0.99 1.00       0.60         0.45       0.99 1.00
## Unemployed           0.62 0.60       1.00        -0.18       0.69 0.67
## Armed.Forces         0.46 0.45      -0.18         1.00       0.36 0.42
## Population           0.98 0.99       0.69         0.36       1.00 0.99
## Year                 0.99 1.00       0.67         0.42       0.99 1.00
```

We can see between both GNPs, Population and Year there are extremely high correlations.

b. Regress each predictor on others to examine the collinearity. Do you have same conclusion as in (a)?

```
X <- model.matrix(lm.model)[,-1]
for(i in 1:dim(X)[2]){
  r2 <- summary(lm(X[,i]~X[,-i]))$r.squared
  cat(colnames(X)[i], '\t', r2, '\n')
}
```

```
## GNP.deflator      0.9926217
```

```
## GNP   0.9994409
## Unemployed    0.9702548
## Armed.Forces      0.7213654
## Population    0.9974947
## Year      0.9986824
```

We can see that both GNPs, population, year and now even Unemployed have very high R^2 values indicating collinearity.

c. Try to remove some highly correlated predictors. Compare the full model and the smaller model. Do you think the smaller model is better? Give you reason.

```
lm.model.small <- lm(Employed ~ GNP + Armed.Forces + Unemployed + Year, data = longley)
summary(lm.model.small)
```

```
##
## Call:
## lm(formula = Employed ~ GNP + Armed.Forces + Unemployed + Year,
##     data = longley)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.42165 -0.12457 -0.02416  0.08369  0.45268
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -3.599e+03  7.406e+02  -4.859 0.000503 ***
## GNP          -4.019e-02  1.647e-02  -2.440 0.032833 *
## Armed.Forces -1.015e-02  1.837e-03  -5.522 0.000180 ***
## Unemployed   -2.088e-02  2.900e-03  -7.202 1.75e-05 ***
## Year          1.887e+00  3.828e-01   4.931 0.000449 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2794 on 11 degrees of freedom
## Multiple R-squared:  0.9954, Adjusted R-squared:  0.9937
## F-statistic: 589.8 on 4 and 11 DF,  p-value: 9.5e-13
```

In the final model I've decided to remove GNP.deflator and Population. I decided to keep Year because it was significant in the original model. The smaller model appears to be the better choice. We can see that GNP is now significant, Year is now significant at the 0.1% level and the adjusted R^2 has improved slightly. The standard deviations were not high in the original model and are not high in the smaller model either.

Question 4: The gala dataset contains 30 Galapagos islands and 7 variables. The relationship between the number of plant species and several geographic variables is of interest.

```
gala <- read.csv("gala.csv")
head(gala)
```

```
##   Species Endemics  Area Elevation Nearest Scruz Adjacent
## 1      58       23 25.09       346     0.6   0.6     1.84
## 2      31       21  1.24       109     0.6  26.3   572.33
## 3       3        3  0.21       114     2.8  58.7     0.78
## 4      25        9  0.10        46     1.9  47.4     0.18
## 5       2        1  0.05        77     1.9   1.9   903.82
## 6      18       11  0.34       119     8.0   8.0     1.84
```

The dataset galamiss contains the Galapagos data with missing values left in. Use two datasets to answer the following questions.

```
galamiss <- read.csv("galamiss.csv")
head(galamiss)
```

```
##   Species Endemics  Area Elevation Nearest Scruz Adjacent
## 1      58       23 25.09        NA     0.6   0.6     1.84
## 2      31       21  1.24       109     0.6  26.3   572.33
## 3       3        3  0.21       114     2.8  58.7     0.78
## 4      25        9  0.10        46     1.9  47.4     0.18
## 5       2        1  0.05        NA     1.9   1.9   903.82
## 6      18       11  0.34        NA     8.0   8.0     1.84
```

a. Fit a linear model using gala (the data without missing) with the number of species as the response and the five geographic predictors (without Endemics).

```
lm.model <- lm(Species ~ Area + Elevation + Nearest + Scruz + Adjacent, data = gala)
summary(lm.model)
```

```
##
## Call:
## lm(formula = Species ~ Area + Elevation + Nearest + Scruz + Adjacent,
##     data = gala)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -111.679  -34.898   -7.862   33.460  182.584
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.068221  19.154198   0.369 0.715351
## Area        -0.023938   0.022422  -1.068 0.296318
## Elevation    0.319465   0.053663   5.953 3.82e-06 ***
## Nearest      0.009144   1.054136   0.009 0.993151
## Scruz       -0.240524   0.215402  -1.117 0.275208
## Adjacent    -0.074805   0.017700  -4.226 0.000297 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 60.98 on 24 degrees of freedom
## Multiple R-squared:  0.7658, Adjusted R-squared:  0.7171
## F-statistic:  15.7 on 5 and 24 DF,  p-value: 6.838e-07
```

b. In galamiss, which variable(s) includes missing value? How many missing values do we have?

```
summary(galamiss)
```

```
##     Species          Endemics          Area            Elevation
##  Min.   :  2.00   Min.   : 0.00   Min.   :   0.010   Min.   :  25.0
##  1st Qu.: 13.00   1st Qu.: 7.25   1st Qu.:   0.258   1st Qu.: 111.2
##  Median : 42.00   Median :18.00   Median :   2.590   Median : 240.0
##  Mean   : 85.23   Mean   :26.10   Mean   : 261.709   Mean   : 424.5
##  3rd Qu.: 96.00   3rd Qu.:32.25   3rd Qu.:  59.237   3rd Qu.: 659.0
##  Max.   :444.00   Max.   :95.00   Max.   :4669.320   Max.   :1707.0
##                                                      NA's   :6
##     Nearest           Scruz          Adjacent
##  Min.   : 0.20   Min.   :  0.00   Min.   :   0.03
##  1st Qu.: 0.80   1st Qu.: 11.03   1st Qu.:   0.52
##  Median : 3.05   Median : 46.65   Median :   2.59
##  Mean   :10.06   Mean   : 56.98   Mean   : 261.10
##  3rd Qu.:10.03   3rd Qu.: 81.08   3rd Qu.:  59.24
##  Max.   :47.40   Max.   :290.20   Max.   :4669.32
##
```

```
mean(is.na(galamiss))
```

```
## [1] 0.02857143
```

```
1 - mean(complete.cases(galamiss))
```

```
## [1] 0.2
```

In galamiss the Elevation variable is the only one with missing values (with a total of 6 missing values). We can see that ~2.9% of all values are missing and 20% of observations having missing values.

c. Fit the same linear model to galamiss using the deletion strategy for missing values. Compare the fit to that in (a).

```
lm.model.deletion <- lm(Species ~ Area + Elevation + Nearest + Scruz + Adjacent,
                        data = galamiss)
summary(lm.model.deletion)
```

```
##
## Call:
## lm(formula = Species ~ Area + Elevation + Nearest + Scruz + Adjacent,
##     data = galamiss)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -115.17  -37.60  -10.08   35.17  172.54
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 15.32286   27.47417   0.558  0.58391
## Area        -0.02765    0.02557  -1.081  0.29388
## Elevation    0.32550    0.06476   5.026 8.78e-05 ***
## Nearest     -0.11042    1.17784  -0.094  0.92635
## Scruz       -0.28427    0.25422  -1.118  0.27818
## Adjacent    -0.07880    0.02092  -3.766  0.00141 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 67.36 on 18 degrees of freedom
##   (6 observations deleted due to missingness)
## Multiple R-squared:  0.7668, Adjusted R-squared:  0.702
## F-statistic: 11.83 on 5 and 18 DF,  p-value: 3.54e-05
```

```
#Standard error for original model
sqrt(diag(vcov(lm.model)))
```

```
## (Intercept)        Area   Elevation     Nearest       Scruz    Adjacent
## 19.15419782  0.02242235  0.05366280  1.05413595  0.21540225  0.01770019
```

```
#Standard error for model with deletions
sqrt(diag(vcov(lm.model.deletion)))
```

```
## (Intercept)        Area   Elevation     Nearest       Scruz    Adjacent
## 27.47417243  0.02557323  0.06476468  1.17783953  0.25422056  0.02092401
```

We can see that the deletion model is worse due to a lower adjusted R^2, higher p-values and higher standard errors for each predictor.

d. Use mean value imputation on galamiss and again fit the model. Compare to previous fits.

```
means <- colMeans(galamiss, na.rm = TRUE)
galamiss.impute <- galamiss
for (i in 1:7){
  galamiss.impute[is.na(galamiss.impute[, i]), i] <- means[i]
}
lm.model.mean_impute <- lm(Species ~ Area + Elevation + Nearest + Scruz + Adjacent,
                           data = galamiss.impute)
summary(lm.model.mean_impute)
```

```
##
## Call:
## lm(formula = Species ~ Area + Elevation + Nearest + Scruz + Adjacent,
##     data = galamiss.impute)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -94.710 -42.598  -9.742  26.146 220.893
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -12.48266   28.62644  -0.436 0.666695
## Area         -0.00137    0.02683  -0.051 0.959697
## Elevation     0.27388    0.06891   3.975 0.000562 ***
## Nearest       0.37776    1.28270   0.295 0.770905
## Scruz        -0.08544    0.27140  -0.315 0.755629
## Adjacent     -0.06553    0.02215  -2.958 0.006856 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 74.52 on 24 degrees of freedom
## Multiple R-squared:  0.6503, Adjusted R-squared:  0.5774
## F-statistic: 8.925 on 5 and 24 DF,  p-value: 6.77e-05
```

```
#Standard error for original model
sqrt(diag(vcov(lm.model)))
```

```
## (Intercept)        Area   Elevation     Nearest       Scruz    Adjacent
## 19.15419782  0.02242235  0.05366280  1.05413595  0.21540225  0.01770019
```

```
#Standard error for model with deletions
sqrt(diag(vcov(lm.model.deletion)))
```

```
## (Intercept)        Area   Elevation     Nearest       Scruz    Adjacent
## 27.47417243  0.02557323  0.06476468  1.17783953  0.25422056  0.02092401
```

```
#Standard error with mean impute
sqrt(diag(vcov(lm.model.mean_impute)))
```

```
## (Intercept)        Area   Elevation     Nearest       Scruz    Adjacent
## 28.62643748  0.02682636  0.06890968  1.28270017  0.27140277  0.02215368
```

We can see with the mean imputes the standard error is even higher and the p-values are still high causing adjacent to be significant at the 1% level compared to the 0.1% level (similar to the simple deletion model). The adjusted R^2 also appears to be way worse compared to the other 2 models.

e. Use a regression-based imputation based on the other four geographic predictors to fill in the missing values in galamiss. Fit the same model and compare to previous fits.

```
galamiss.impute <- galamiss
lm.model.impute <- lm(Elevation ~ Area + Nearest + Scruz + Adjacent,
                      data = galamiss)
elevation.impute <- predict(lm.model.impute,
                        galamiss[is.na(galamiss$Elevation),])
galamiss.impute[is.na(galamiss.impute$Elevation),] <- elevation.impute
lm.model.reg_impute <- lm(Species ~ Area + Elevation + Nearest + Scruz + Adjacent,
                      data = galamiss.impute)
summary(lm.model.reg_impute)
```

```
##
## Call:
## lm(formula = Species ~ Area + Elevation + Nearest + Scruz + Adjacent,
##     data = galamiss.impute)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -122.376  -33.096   -1.496   14.058  183.169
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 12.33709   23.84477   0.517 0.609619
## Area        -0.02119    0.02211  -0.958 0.347532
## Elevation    0.30902    0.05546   5.572 9.85e-06 ***
## Nearest      1.14590    0.18952   6.046 3.04e-06 ***
## Scruz       -0.40790    0.20023  -2.037 0.052804 .
## Adjacent    -0.07235    0.01783  -4.058 0.000455 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 60.24 on 24 degrees of freedom
## Multiple R-squared:  0.8498, Adjusted R-squared:  0.8185
## F-statistic: 27.15 on 5 and 24 DF,  p-value: 3.826e-09
```

```
#Ground Truth vs. Imputation
gala$Elevation[is.na(galamiss$Elevation)]
```

```
## [1] 346  77 119  71  94 147
```

```
galamiss.impute$Elevation[is.na(galamiss$Elevation)]
```

```
## [1] 270.2647 466.5880 289.3535 265.1305 267.0589 260.8922
```

```
median(galamiss$Elevation, na.rm = TRUE)
```

```
## [1] 240
```

```
mean(galamiss$Elevation, na.rm = TRUE)
```

```
## [1] 424.4583
```

```
#Standard error for original model
sqrt(diag(vcov(lm.model)))
```

```
## (Intercept)        Area   Elevation     Nearest       Scruz    Adjacent
## 19.15419782  0.02242235  0.05366280  1.05413595  0.21540225  0.01770019
```

```
#Standard error for model with deletions
sqrt(diag(vcov(lm.model.deletion)))
```

```
## (Intercept)        Area   Elevation     Nearest       Scruz    Adjacent
## 27.47417243  0.02557323  0.06476468  1.17783953  0.25422056  0.02092401
```

```
#Standard error with mean imputes
sqrt(diag(vcov(lm.model.mean_impute)))
```

```
## (Intercept)        Area   Elevation     Nearest       Scruz    Adjacent
## 28.62643748  0.02682636  0.06890968  1.28270017  0.27140277  0.02215368
```

```
#Standard error with regression imputes
sqrt(diag(vcov(lm.model.reg_impute)))
```

```
## (Intercept)        Area   Elevation     Nearest       Scruz    Adjacent
## 23.84476850  0.02211350  0.05546058  0.18952468  0.20022888  0.01782930
```

We can see that the model with regression imputation has a much better standard error compared to the simple deletion and mean imputation models. Something to note is that the adjusted R^2 is higher in this final model and the predictor Nearest is significant in this model. Something else to take note of is that in the ground truth vs. imputed values we see that the imputed values are way off compared to the ground truth. after taking a look at the dataset there are a few data points that are way higher than the median and mean Elevation values which could be causing this skew.