hw3 stat425

Aldo Sanjoto

October 13, 2017

1a)

```
library("alr4")
```

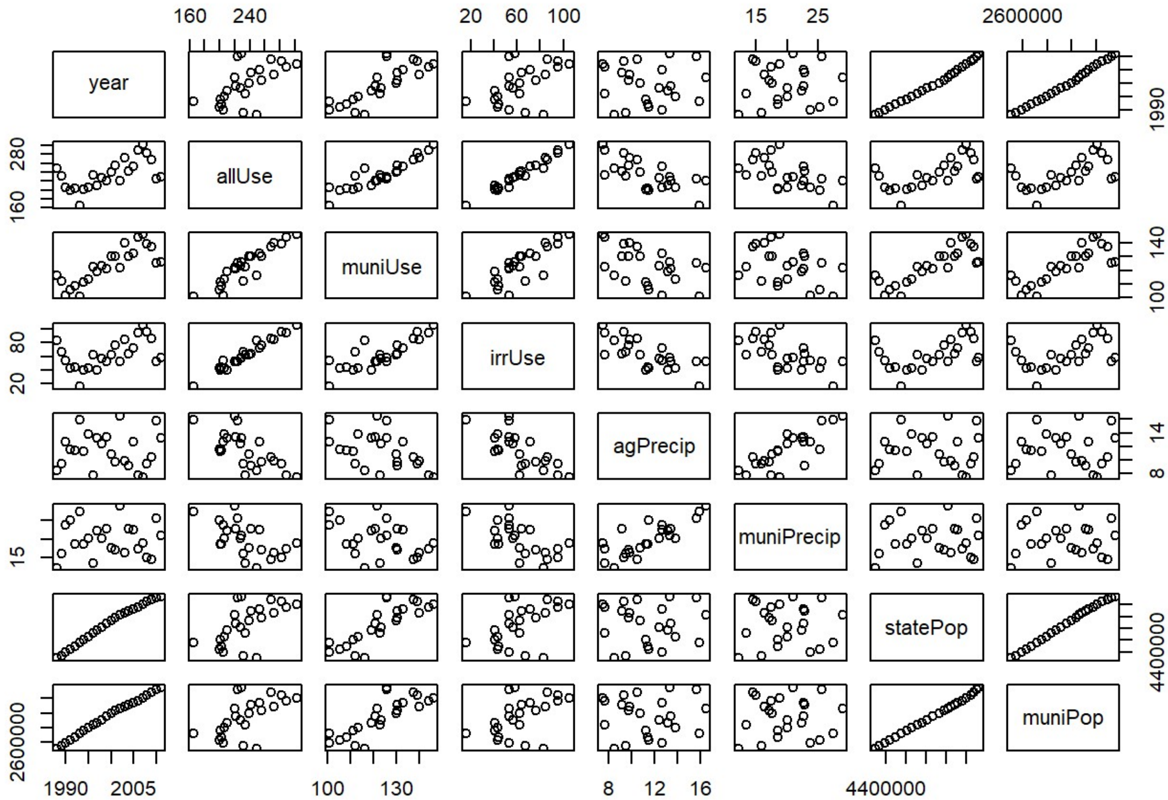
```
## Loading required package: car
```

```
## Loading required package: effects
```

```
##
## Attaching package: 'effects'
```

```
## The following object is masked from 'package:car':
##
##     Prestige
```

```
data("MinnWater")
pairs(MinnWater)
```




```
#??Minnwater
```

1b) Year, statePop, muniPop

1c)

```
fit_mw = lm(formula = muniUse ~ ., data = MinnWater)
summary(fit_mw)
```

```
##
## Call:
## lm(formula = muniUse ~ ., data = MinnWater)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.38834 -0.44331  0.02071  0.47878  1.26227
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.163e+02  1.170e+03   0.527  0.60567
## year        -3.763e-01  6.034e-01  -0.624  0.54167
## allUse       6.909e-01  7.140e-02   9.677 4.33e-08 ***
## irrUse      -6.535e-01  8.681e-02  -7.528 1.21e-06 ***
## agPrecip     1.623e-01  1.562e-01   1.040  0.31398
## muniPrecip  -2.491e-01  7.980e-02  -3.122  0.00658 **
## statePop     6.082e-05  2.237e-05   2.719  0.01517 *
## muniPop     -5.228e-05  3.876e-05  -1.349  0.19621
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8131 on 16 degrees of freedom
## Multiple R-squared:  0.9973, Adjusted R-squared:  0.9962
## F-statistic: 855.2 on 7 and 16 DF,  p-value: < 2.2e-16
```

```
vif(fit_mw)
```

```
##       year     allUse     irrUse   agPrecip muniPrecip   statePop
##  633.34563  190.15277  118.11767    5.72228    4.28763 1904.44626
##    muniPop
## 3441.37710
```

1d) the variable year, allUse, irrUse, statePop, muniPop have a VIF indicating a possible problem.

```
fit_mw2 = lm(formula = muniUse ~ allUse + irrUse + muniPrecip + statePop, data = MinnW
ater)
summary(fit_mw2)
```

```
##
## Call:
## lm(formula = muniUse ~ allUse + irrUse + muniPrecip + statePop,
##     data = MinnWater)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.3681 -0.6835 -0.2439  0.7012  2.1665
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6.256e+01  4.904e+00 -12.758 9.15e-11 ***
## allUse       8.219e-01  5.251e-02  15.650 2.60e-12 ***
## irrUse      -8.227e-01  7.270e-02 -11.317 6.92e-10 ***
## muniPrecip  -1.332e-01  7.084e-02  -1.881   0.0754 .
## statePop     9.731e-06  1.244e-06   7.819 2.35e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.008 on 19 degrees of freedom
## Multiple R-squared:  0.9951, Adjusted R-squared:  0.9941
## F-statistic: 972.5 on 4 and 19 DF,  p-value: < 2.2e-16
```

1e) Only the variable muniPrecip is not significant at level 5%

```
vif(fit_mw2)
```

```
##     allUse     irrUse muniPrecip   statePop
##  66.985801  53.944057   2.200120   3.838104
```

1f) compared to the VIF in 1e, the VIF decreased. The variables muniPrecip and statePop have a VIF indicating a problem.

2a)

```
age = c(1, 2, 3, 4, 5)
number = c(123, 78, 32, 17, 24)
average_weight = c(7.9725, 7.9503, 7.9276, 7.8962, 7.8730)
std = c(0.01409, 0.02272, 0.03426, 0.04057, 0.05353)
money = data.frame(age, number, average_weight, std)
#View(money)
fit_money = lm(formula = average_weight ~ age, data = money)
summary(fit_money)
```

```
##
## Call:
## lm(formula = average_weight ~ age, data = money)
##
## Residuals:
##        1        2        3        4        5
## -0.00204  0.00107  0.00368 -0.00241 -0.00030
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.9998500  0.0030123 2655.74 1.18e-10 ***
## age         -0.0253100  0.0009082  -27.87 0.000101 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.002872 on 3 degrees of freedom
## Multiple R-squared:  0.9962, Adjusted R-squared:  0.9949
## F-statistic: 776.6 on 1 and 3 DF,  p-value: 0.0001014
```

```
confint(fit_money, "age")
```

```
##           2.5 %      97.5 %
## age -0.02820043 -0.02241957
```

2b) the 95% confidence interval for the regression slope is -0.02820043 to -0.02241957

```
fit_money2 = lm(formula = average_weight ~ age, data = money, weights = money$number)
summary(fit_money2)
```

```
##
## Call:
## lm(formula = average_weight ~ age, data = money, weights = money$number)
##
## Weighted Residuals:
##         1         2         3         4         5
## -0.011442  0.012903  0.019537 -0.013415 -0.008627
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.9982244  0.0020332  3933.9 3.62e-11 ***
## age         -0.0246927  0.0008427   -29.3 8.73e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01764 on 3 degrees of freedom
## Multiple R-squared:  0.9965, Adjusted R-squared:  0.9954
## F-statistic: 858.6 on 1 and 3 DF,  p-value: 8.729e-05
```

```
weight =  c(123, 78, 32, 17, 24)
w = diag(weight, ncol = 5, nrow = 5)

print("Weight matrix:")
```

```
## [1] "Weight matrix:"
```

```
print(w)
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]  123    0    0    0    0
## [2,]    0   78    0    0    0
## [3,]    0    0   32    0    0
## [4,]    0    0    0   17    0
## [5,]    0    0    0    0   24
```

```
s = solve(w)
print("Sigma matrix:")
```

```
## [1] "Sigma matrix:"
```

```
print(s)
```

```
##             [,1]       [,2]    [,3]       [,4]       [,5]
## [1,] 0.008130081 0.00000000 0.00000 0.00000000 0.00000000
## [2,] 0.000000000 0.01282051 0.00000 0.00000000 0.00000000
## [3,] 0.000000000 0.00000000 0.03125 0.00000000 0.00000000
## [4,] 0.000000000 0.00000000 0.00000 0.05882353 0.00000000
## [5,] 0.000000000 0.00000000 0.00000 0.00000000 0.04166667
```

2d) The weight is a diagonal matrix with its element is equal to numbers respectively, the sigma matrix is the inverse of weight matrix

```
confint(fit_money2, "age")
```

```
##          2.5 %      97.5 %
## age -0.0273745 -0.02201086
```

2e) the 95% confidence interval is from -0.0273745 to -0.02201086

2f)

```
fit_money3 = lm(formula = average_weight ~ age, data = money, weights = (money$numbe
r / (money$std^2)))
summary(fit_money3)
```

```
## 
## Call:
## lm(formula = average_weight ~ age, data = money, weights = (money$number/(money$std
^2)))
## 
## Weighted Residuals:
##       1       2       3       4       5 
## -0.2091  0.5017  0.3875 -0.5383 -0.4339 
## 
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  7.9965218  0.0013220    6049 9.96e-12 ***
## age         -0.0237562  0.0008797     -27 0.000111 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.5549 on 3 degrees of freedom
## Multiple R-squared:  0.9959, Adjusted R-squared:  0.9945 
## F-statistic: 729.2 on 1 and 3 DF,  p-value: 0.0001114
```

```
weight2 = c(619559.6942, 151104.6915, 27263.10154, 10328.53929, 8375.615944)
w2 = diag(weight2, ncol = 5, nrow = 5)

print("Weight matrix:")
```

```
## [1] "Weight matrix:"
```

```
print(w2)
```

```
##          [,1]     [,2]    [,3]     [,4]     [,5]
## [1,] 619559.7      0.0     0.0     0.00    0.000
## [2,]      0.0 151104.7     0.0     0.00    0.000
## [3,]      0.0      0.0 27263.1     0.00    0.000
## [4,]      0.0      0.0     0.0 10328.54    0.000
## [5,]      0.0      0.0     0.0     0.00 8375.616
```

```
s2 = solve(w2)
print("Sigma matrix:")
```

```
## [1] "Sigma matrix:"
```

```
print(s2)
```

```
##              [,1]         [,2]         [,3]         [,4]         [,5]
## [1,] 1.614049e-06 0.000000e+00 0.000000e+00 0.000000e+00 0.0000000000
## [2,] 0.000000e+00 6.617928e-06 0.000000e+00 0.000000e+00 0.0000000000
## [3,] 0.000000e+00 0.000000e+00 3.667961e-05 0.000000e+00 0.0000000000
## [4,] 0.000000e+00 0.000000e+00 0.000000e+00 9.681911e-05 0.0000000000
## [5,] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.0001193942
```

2g) The weight is a diagonal matrix with its element is equal to numbers/(std^2) respectively, the sigma matrix is the inverse of weight matrix

```
confint(fit_money3, "age")
```

```
##           2.5 %      97.5 %
## age -0.02655593 -0.02095642
```

2h) the 95% confidence interval is from -0.02655593 to -0.02095642

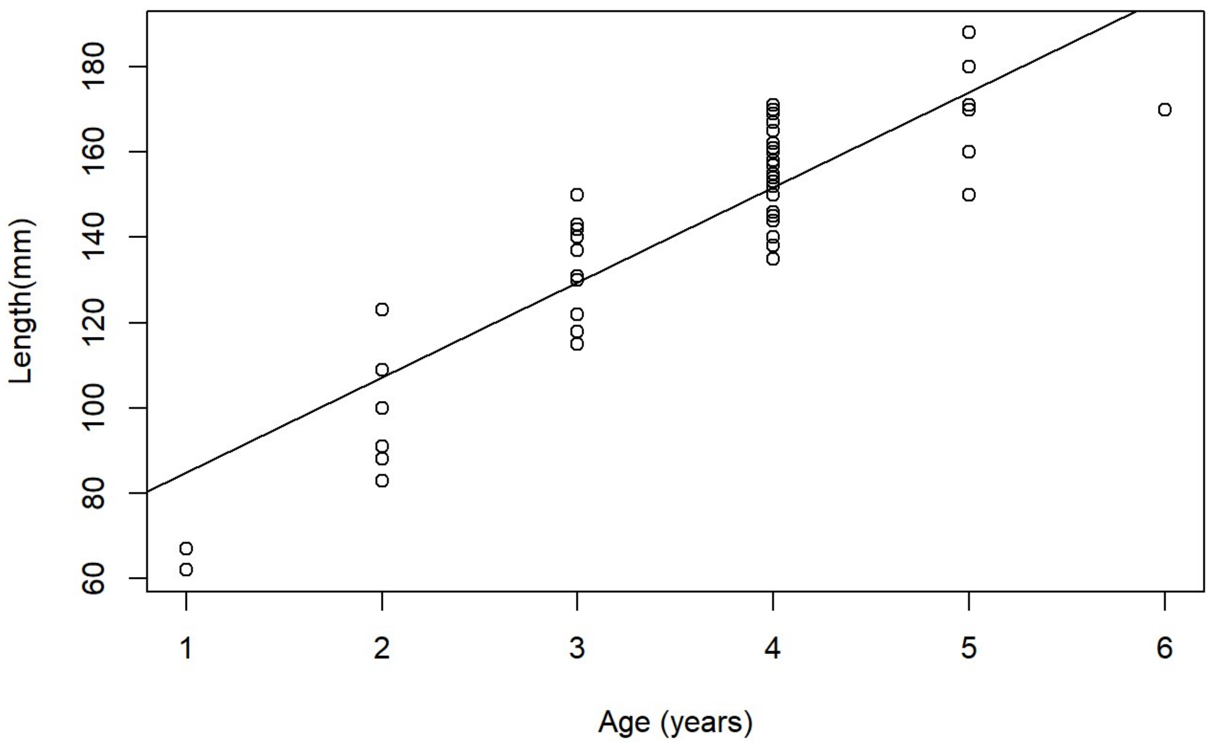
3a)

```
data("lakemary")
#head(lakemary)
fit_lm = lm(formula = Length ~ Age, data = lakemary)
summary(fit_lm)
```

```
##
## Call:
## lm(formula = Length ~ Age, data = lakemary)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -26.523  -7.586   0.258  10.102  20.414
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   62.649      5.755   10.89   <2e-16 ***
## Age           22.312      1.537   14.51   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.51 on 76 degrees of freedom
## Multiple R-squared:  0.7349, Adjusted R-squared:  0.7314
## F-statistic: 210.7 on 1 and 76 DF,  p-value: < 2.2e-16
```

3b)

```
plot(lakemary$Age, lakemary$Length, xlab="Age (years)", ylab ="Length(mm)")
abline(fit_lm)
```



3c) Because there are repeated data observation of the same number of Age.

```
fit_lm_lof = lm(formula = Length ~ factor(Age), data = lakemary)
anova(fit_lm, fit_lm_lof)
```

```
## Analysis of Variance Table
##
## Model 1: Length ~ Age
## Model 2: Length ~ factor(Age)
##   Res.Df     RSS Df Sum of Sq      F    Pr(>F)
## 1     76 11892.8
## 2     72  8812.7  4    3080.2 6.2912 0.0002125 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

3d) from the lack-of-fit anova model, we can see that the pvalue 0.0002125 is really small so we decide to reject the null. Thus, the model is insufficient.

3e) the estimate pure error variance is 8812.7/72 = 122.4

3f)

```
fit_lm2 = lm(formula = Length ~ Age + I(Age^2), data = lakemary)
summary(fit_lm2)
```

```
##
## Call:
## lm(formula = Length ~ Age + I(Age^2), data = lakemary)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -19.846  -8.321  -1.137   6.698  22.098
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   13.622     11.016   1.237     0.22
## Age           54.049      6.489   8.330 2.81e-12 ***
## I(Age^2)      -4.719      0.944  -4.999 3.67e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.91 on 75 degrees of freedom
## Multiple R-squared:  0.8011, Adjusted R-squared:  0.7958
## F-statistic: 151.1 on 2 and 75 DF,  p-value: < 2.2e-16
```

```
anova(fit_lm2, fit_lm_lof)
```

```
## Analysis of Variance Table
##
## Model 1: Length ~ Age + I(Age^2)
## Model 2: Length ~ factor(Age)
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     75 8920.7
## 2     72 8812.7  3    108.01 0.2942 0.8295
```

3g) from the lack-of-fit anova model, we can see that the pvalue 0.8295 is quite large so we fail to reject the null. Thus, the quadratic model is a sufficient model.