

STAT 210
Applied Statistics and Data Analysis
Second Exam

November 29, 2025

You are not allowed to use AI tools to solve this exam

You are reminded to adhere to the academic integrity code established at KAUST.

This exam is open notes and open book but not open internet. You are not allowed to use the internet except for downloading the exam and uploading the solution.

Show complete solutions to get full credit. Writing code is not enough to answer a question. Your comments are more important than the code. Do not write comments inside chunks. Label your graphs appropriately. Please identify the files you submit with your surname

**For all tests in this exam use a significance level of $\alpha = 0.02$
unless otherwise specified**

Question 1 (30 points)

The file `XM225F_q1.csv` has data on a sample of women that belong to the Pima indian tribe and has 768 observations of 6 variables. For this question we will consider only the variables `age`, `mass` (body mass index), and `diabetes`, a binary variable with values `pos` and `neg`.

- (a) Read the data. Create a new data frame called `Q1data` with the variables `age`, `mass`, and `diabetes`. Check whether the new data frame has entries with `age` or `mass` equal to zero, and delete those entries from your file. How many observations are left?

```
data_1 <- read.csv("XM225F_q1.csv")
head(data_1)
```

```
##   pregnant glucose pressure triceps insulin mass pedigree age
## 1       6    148       72      35        0 33.6  0.627  50
## 2       1     85       66      29        0 26.6  0.351  31
## 3       8    183       64       0        0 23.3  0.672  32
## 4       1     89       66      23       94 28.1  0.167  21
## 5       0    137       40      35      168 43.1  2.288  33
## 6       5    116       74       0        0 25.6  0.201  30
##   diabetes
## 1     pos
```

```

## 2      neg
## 3      pos
## 4      neg
## 5      pos
## 6      neg

Q1data <- data_1[, c("mass", "age", "diabetes")] #extract the data frame of wanted values
str(Q1data)

## 'data.frame':    768 obs. of  3 variables:
##   $ mass     : num  33.6 26.6 23.3 28.1 43.1 25.6 31 35.3 30.5 0 ...
##   $ age      : int  50 31 32 21 33 30 26 29 53 54 ...
##   $ diabetes: chr  "pos" "neg" "pos" "neg" ...

sum(is.na(Q1data[, "mass"])) # check if mass has any empty values

## [1] 0

sum(is.na(Q1data[, "age"])) # check if age has any empty values.

## [1] 0

```

No empty values, No observations deleted all 768 observations remain.

- (b) Create a new ordered factor in `Q1data` with name `bmi` according to the following rules: If `mass` is less than or equal to 25 the value of `bmi` is `normal`; if `mass` is over 25 and up to 30 the value of `bmi` is `overweight`, and if `mass` is above 30 the value is `obese`.

```

bmi_cutoff_names <- c("Normal", "Overweight", "Obese") # the three categories labels
bmi_cutoff_values <- c(0, 25, 30, 1000) # from 0 to 25 is normal, 25 to 30 is overweight, 30 and above

Q1data$bmi <- cut(Q1data$mass,
                     breaks = bmi_cutoff_values,
                     labels = bmi_cutoff_names,
                     ordered = TRUE) #cuts of the mass using our breaks and naming them using labels.

```

- (c) Create a contingency table for `bmi` and `diabetes`. Graph a mosaic plot of this table using different colors for the rectangles. Comment on what you observe. Using an appropriate statistical test, determine whether these two variables are independent. What are the underlying assumptions for the test? Discuss whether they are satisfied in this case.

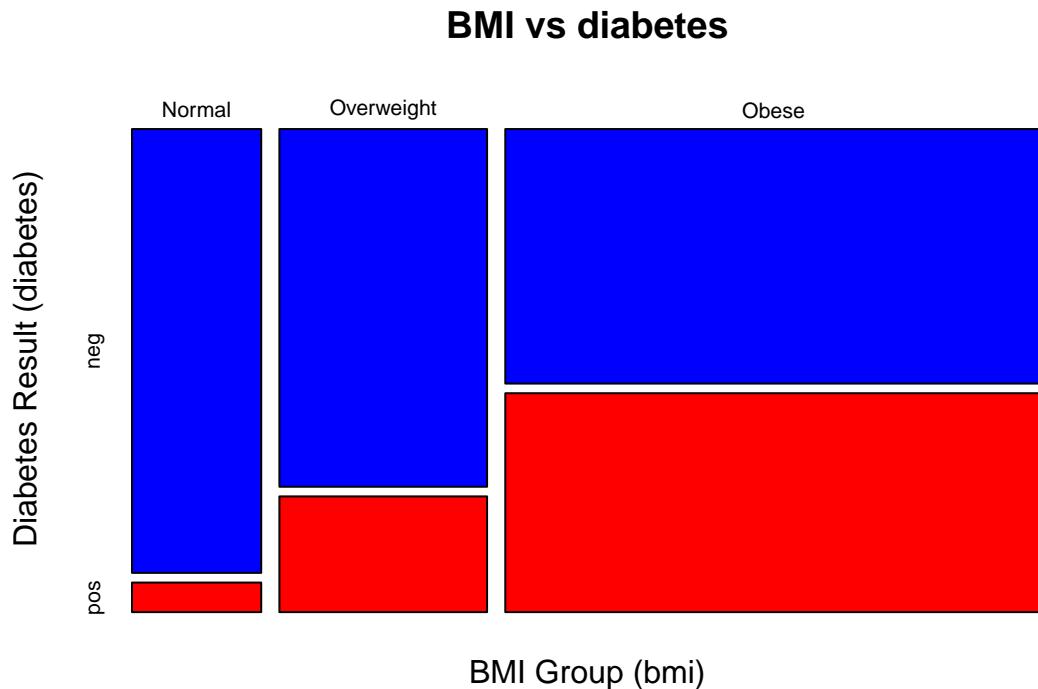
```

(table1 <- table(Q1data$bmi, Q1data$diabetes)) # create a contingency table for bmi and diabetes

##
##          neg  pos
## Normal    105   7
## Overweight 136  44
## Obese     250 215

```

```
#draw the mosaic plot.
mosaicplot(table1,
           main = "BMI vs diabetes",
           xlab = "BMI Group (bmi)",
           ylab = "Diabetes Result (diabetes)",
           color = c("blue", "red"))
```



From the mosaic plot we can notice two things,

- 1: the amount of people considered obese is more than half the population of the dataset #we can see that through the width of each rectangle.
- 2: diabetes seems to be linked to BMI in a positive correlation, with higher BMI #suggesting a higher probability of diabetes.

```
(chi_bmi_diabetes <- chisq.test(table1))
```

```
##
##  Pearson's Chi-squared test
##
##  data:  table1
##  X-squared = 75.2, df = 2, p-value <2e-16
```

with a p-value of $2.2e-16 \ll 0.02$, we reject the null hypothesis of independency. and we say that BMI and diabetes have an dependant relationship

we then check the expected cell count to check the reliability of the test

```

chi_bmi_diabetes$expected

##          neg      pos
##  Normal    72.645  39.355
##  Overweight 116.750  63.250
##  Obese     301.605 163.395

sum(chi_bmi_diabetes$expected < 5)

```

```
## [1] 0
```

the test requires that the expected cell values must not be less than 5 a maximum of about 20% of the values being less than 5 is acceptable to suggest reliability. since we have 0 cells with expected value < 5, we can say that the test is reliable and that our deduction of dependency is valid.

- (d) Create a new ordered factor in Q1data named **fage** by dividing age into four groups having approximately the same number of subjects. Name the levels **a1**, **a2**, **a3**, and **a4**. Produce a contingency table for **fage** and **bmi** (**bmi** should correspond to the rows of the table). Graph a mosaic plot of this table using different colors for the rectangles. Comment on what you observe. Using an appropriate statistical test, determine whether these two variables are independent. What are the underlying assumptions for the test? Discuss whether they are satisfied in this case.

```

(q <- quantile(Q1data$age, probs = seq(0, 1, length.out = 5))) #split into approximately equal size chuncks

##    0%   25%   50%   75% 100%
##  21    24    29    41    81

#make the new factor "fage" using the equal split sized of ages
Q1data$fage <- cut(Q1data$age,
                      breaks = q,
                      labels = c("a1", "a2", "a3", 'a4'),
                      ordered = TRUE)

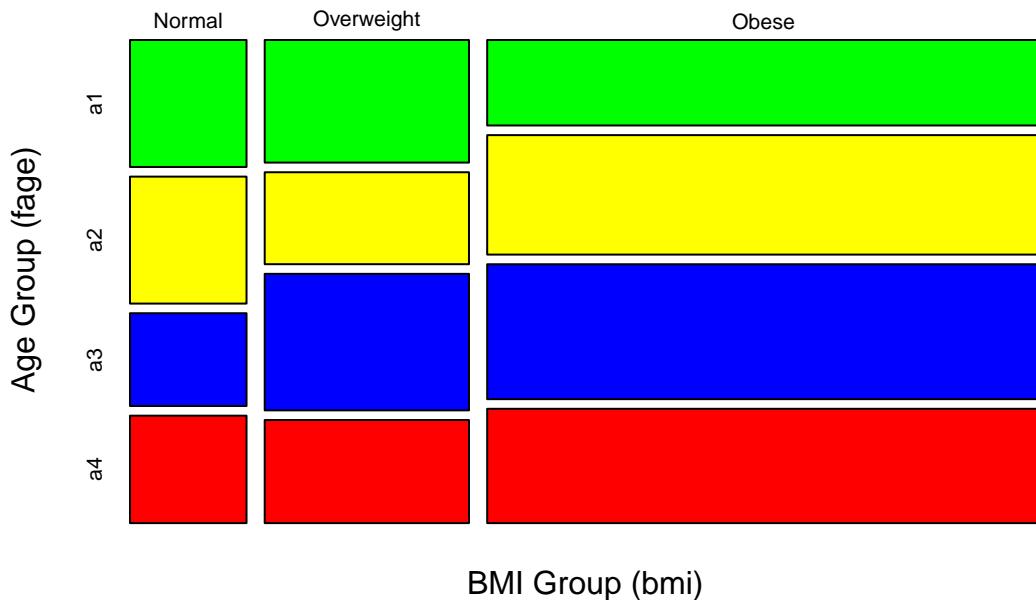
(table2 <- table(Q1data$bmi, Q1data$fage)) # create a contingency table for bmi and diabetes

##          a1   a2   a3   a4
##  Normal    26   26   19   22
##  Overweight 44   33   49   37
##  Obese     83  116  131  111

mosaicplot(table2,
           main = "Fage vs BMI",
           xlab = "BMI Group (bmi)",
           ylab = "Age Group (fage)",
           color = c("green", "yellow", "blue", "red")) #draw the mosaic plot of the table.

```

Fage vs BMI



From the mosaic plot, we can see that for a1 group, they are mostly normal, and then overweight and have the lowest percentage of obesity. a2 has a high percentage of normal and obese, with a lowly populated portion for overweight. we also see a3 having the highest percentage of overweight people, followed by approximately the highest percentage of obese people. and we see that a4 has about an equal spread of different bmi levels.

```
(chi_fage_bmi <- chisq.test(table2))
```

```
##  
## Pearson's Chi-squared test  
##  
## data: table2  
## X-squared = 10.3, df = 6, p-value = 0.11
```

with a p-value of $0.1134 > 0.02$, we fail to reject the null hypothesis and we say that age group and BMI are independent.

we then check the expected cell count to check the reliability of the test

```
chi_fage_bmi$expected
```

```
##  
##          a1      a2      a3      a4  
## Normal   20.415  23.350  26.552  22.683  
## Overweight 35.780  40.925  46.538  39.756  
## Obese    96.805 110.725 125.910 107.561
```

```
sum(chi_bmi_diabetes$expected < 5)
```

```
## [1] 0
```

The test requires that the expected cell values must not be less than 5 since we have 0 cells with expected value < 5, we can say that the test is reliable and that out deduction of independency is valid.

Question 2 (30 points)

The file XM225F_q2.csv contains the results of an experiment to study the effect of fertilizer type and rainfall on crop yield. There are two types of fertilizer, `Organic` and `Chemical`. The other two variables are `rainfall` and `fertilizer`. Read the data into a new data frame named `Q2data`. Transform `fertilizer` into a factor.

```
Q2data <- read.csv("XM225F_q2.csv")  
head(Q2data)
```

```
##   fertilizer rainfall yield  
## 1    Organic     583   3.9  
## 2    Organic     620   5.3  
## 3    Organic     576   4.5  
## 4  Chemical      565   7.1  
## 5    Organic     392   3.7  
## 6    Organic     567   4.6
```

```
str(Q2data)
```

```
## 'data.frame': 120 obs. of 3 variables:  
## $ fertilizer: chr "Organic" "Organic" "Organic" "Chemical" ...  
## $ rainfall  : int 583 620 576 565 392 567 384 738 480 618 ...  
## $ yield    : num 3.9 5.3 4.5 7.1 3.7 4.6 4.4 8 6.1 6.4 ...
```

```
Q2data$fertilizer <- as.factor(Q2data$fertilizer) #change fertilizer to type factor.  
str(Q2data)
```

```
## 'data.frame': 120 obs. of 3 variables:  
## $ fertilizer: Factor w/ 2 levels "Chemical","Organic": 2 2 2 1 2 2 2 1 1 1 ...  
## $ rainfall  : int 583 620 576 565 392 567 384 738 480 618 ...  
## $ yield    : num 3.9 5.3 4.5 7.1 3.7 4.6 4.4 8 6.1 6.4 ...
```

- (a) Do a scatterplot of `yield` as a function of `rainfall`, including the regression line. Fit a regression and print the summary table. Interpret the output in the table. State explicitly the assumptions that underlie this model and use diagnostic plots and tests to verify whether they are satisfied. Include your comments on every step that you take.

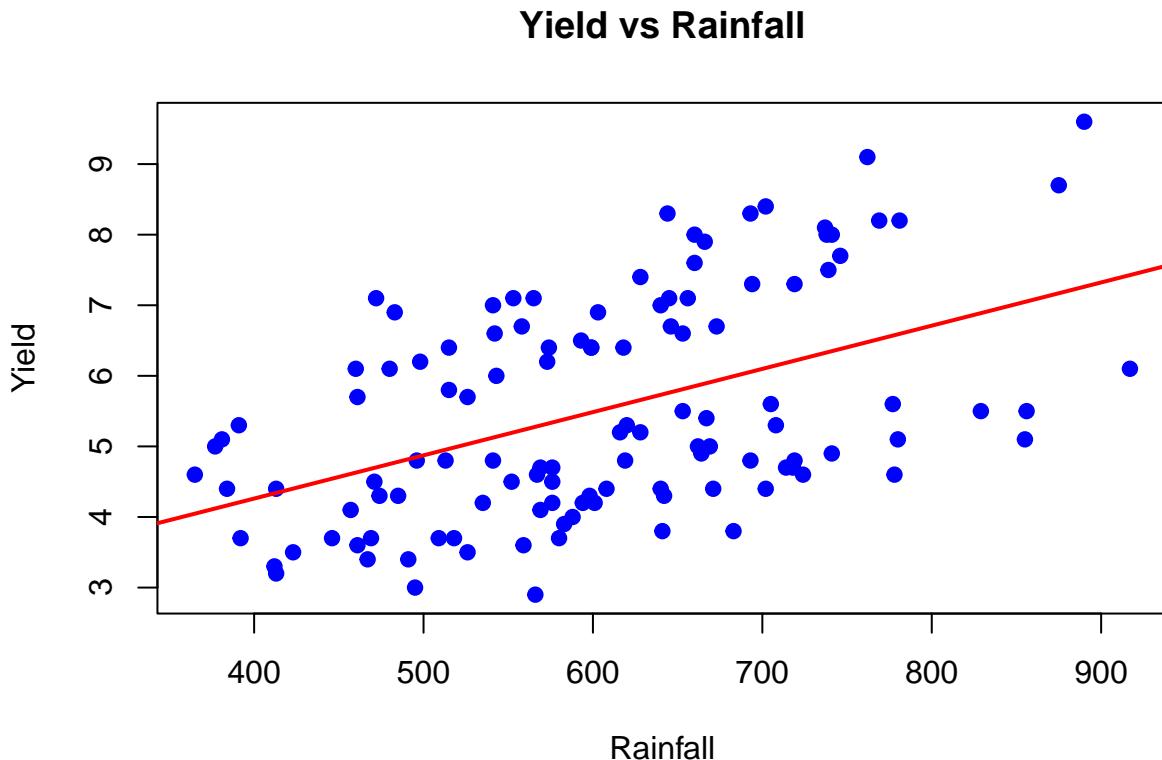
```

model_1 <- lm(yield ~ rainfall, data = Q2data) # fit a linear regression model of yield in terms of rainfall

plot(Q2data$rainfall, Q2data$yield,
     main = "Yield vs Rainfall",
     xlab = "Rainfall",
     ylab = "Yield",
     pch = 19, col = "blue") # do a scatterplot of rainfall vs yield.

abline(model_1, col = "red", lwd = 2) #draw the regression model on the plot

```



```
summary(model_1) #print summary of model.
```

```

##
## Call:
## lm(formula = yield ~ rainfall, data = Q2data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -2.378 -1.234 -0.399  1.270  2.622 
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 1.81169   0.63738   2.84   0.0053 **  
## rainfall    0.00612   0.00103   5.92  3.3e-08 *** 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
```

```

## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.36 on 118 degrees of freedom
## Multiple R-squared:  0.229, Adjusted R-squared:  0.222
## F-statistic:   35 on 1 and 118 DF,  p-value: 3.28e-08

```

From the table we can see the information of the residuals, the coefficient estimates etc.

we focus on the p-value of each predictor in the model, and we see that both the intercept and rainfall have p-values < 0.02 (0.0528 and 3.28e-8 respectively).

For both cases we reject the null hypothesis of these coefficients actually being 0 and we say that our model suggests that they are not 0 and are useful in predicting yield.

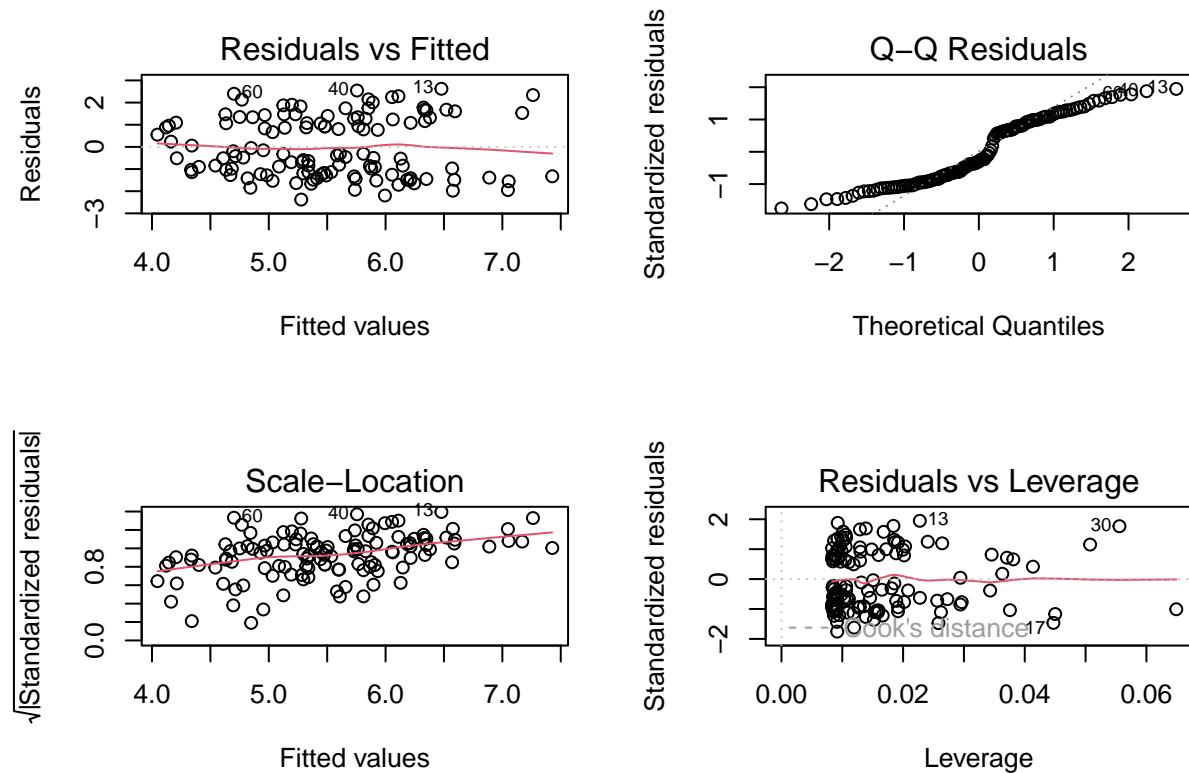
we also see the R^2 and adjusted R^2 of 0.2288 and 0.2223, indicating that the model somewhat captures the plot but isn't a great fit at all.

Regarding assumptions, A simple linear regression model has 4 assumptions. Linearity, Independence, Constant variance, normality of residuals. And we test these next:

```

par(mfrow = c(2, 2))
plot(model_1)

```



```

par(mfrow = c(1, 1))

```

From the residuals vs fitted, we see that the line is roughly horizontal and close to 0 with an even spread of points, indicating Linearity.

From QQ residuals we see the values fitting somewhat well in the middle but having strong tails, we also notice a weird curve in the middle. all of this indicates non-normality and further testing is required.

From Scale-location we see the line having a somewhat steady positive slope going away from the horizontal 0 which may indicate non-constant variance, with the data points having some kind of even spread, but we would require further testing.

From residuals vs leverage we determine there being no strongly influential points, but we do point out some points with high leverage.

we then do the shapiro-wilk test of normality and the ncv Test for constant variance.

```
shapiro.test(residuals(model_1))
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: residuals(model_1)  
## W = 0.926, p-value = 5.4e-06
```

```
library(car)
```

```
## Loading required package: carData
```

```
ncvTest(model_1)
```

```
## Non-constant Variance Score Test  
## Variance formula: ~ fitted.values  
## Chisquare = 5.2376, Df = 1, p = 0.0221
```

The shapiro-wilk test gave a p-value of $5.378E-6 < 0.02$, which means we reject the null hypothesis of normality, and we say that the spread of residuals is non-normal.

the ncv test gave us a p-value of $0.022 > 0.02$, which means we fail to reject the null hypothesis, and we say that the residuals have constant variance

- (b) Fit a model that includes `rainfall`, `fertilizer`, and the interaction between the two. Using a critical value for α of 0.05 and starting with the complete model, select a minimal adequate model. Compare the adjusted R^2 and the standard deviation for the errors (residual standard error) with the previous model. Check the assumptions for the final model. Write down the equation for this regression model and predict the value of the `yield` for a rainfall of 770 for both types of fertilizer, including confidence intervals at the 98% confidence level. Include your comments on every step that you take.

```
model_2 <- lm(yield ~ rainfall * fertilizer, data = Q2data) #rainfall * fertilizer is basically both +  
summary(model_2)
```

```
##  
## Call:  
## lm(formula = yield ~ rainfall * fertilizer, data = Q2data)  
##  
## Residuals:  
##      Min       1Q   Median       3Q      Max  
## -1.5743 -0.3496 -0.0427  0.3437  1.2873
```

```

## 
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)           1.970871  0.363936   5.42  3.4e-07 ***
## rainfall              0.008139  0.000589  13.81 < 2e-16 ***
## fertilizerOrganic     -0.118637  0.490685  -0.24    0.81    
## rainfall:fertilizerOrganic -0.003893  0.000796  -4.89  3.3e-06 ***
## ---                
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.521 on 116 degrees of freedom
## Multiple R-squared:  0.889, Adjusted R-squared:  0.886 
## F-statistic: 309 on 3 and 116 DF,  p-value: <2e-16

```

with a p-value of 0.809, the T test shows a very high probability of fertilizer being 0, thus having no significant effect on predicting yield. therefore we delete it from our model since it's larger than our a_critical of 0.05

```

model_3 <- update(model_2, .~. - fertilizer)
summary(model_3)

```

```

## 
## Call:
## lm(formula = yield ~ rainfall + rainfall:fertilizer, data = Q2data)
## 
## Residuals:
##      Min       1Q       Median      3Q      Max  
## -1.5651 -0.3529 -0.0478  0.3444  1.3037 
## 
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)           1.905608  0.243123   7.84  2.3e-12 ***
## rainfall              0.008243  0.000403  20.46 < 2e-16 ***
## rainfall:fertilizerOrganic -0.004082  0.000155 -26.35 < 2e-16 ***
## ---                
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.519 on 117 degrees of freedom
## Multiple R-squared:  0.889, Adjusted R-squared:  0.887 
## F-statistic: 467 on 2 and 117 DF,  p-value: <2e-16

```

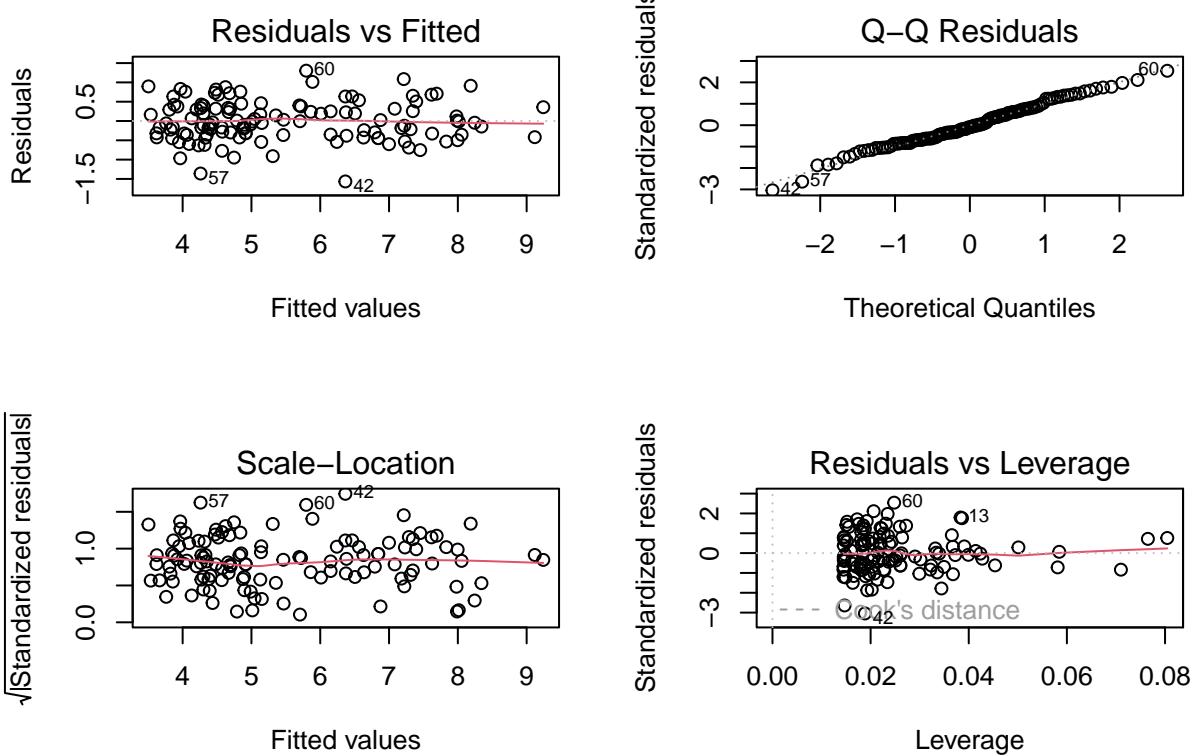
We can see that our new model has the same R^2 values, but it's expected since the predictor we got rid off was not affecting the prediction by much. we can also see that our adjusted R^2 increased, which is also expected since we got one less predictor for our model.

we also see a lower value of the residual standard error, meaning the residuals are more tightly packed.

```

par(mfrow = c(2, 2))
plot(model_3)

```



```
par(mfrow = c(1, 1))
```

From the residuals vs fitted plot, we see the line being horizontal with somewhat an even spread of points, indicating linearity.

From Q-Q residuals we see the points lying neatly on the line, indicating normality

From the scale-location plot, we see the line being mostly horizontal with an even spready of points, indicating constant variance.

from residuals vs leverage: we see that there are no influential points, but some points with moderate leverage.

we then do a shapiro-wilk and an ncv Test for further evidence.

```
shapiro.test(residuals(model_3))
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: residuals(model_3)  
## W = 0.989, p-value = 0.48
```

```
## Non-constant Variance Score Test  
## Variance formula: ~ fitted.values  
## Chisquare = 0.0095182, Df = 1, p = 0.922
```

With a shapiro-wilk test p-value of 0.48 » 0.02, we fail to reject the null hypothesis and say that the residuals are normally distributed. with the ncv test p-value of 0.92228 » 0.02, we fail to reject the null hypothesis and say that the residuals have constant variance.

From the summary table of our minimal model, we extract two functions for the lines, one if the fertilizer is organic and one otherwise.

```
Yield_organic = 1.9056084 + 0.0082430 * Rainfall - 0.0040822 Rainfall
```

```
Yield_organic = 1.9056084 + 0.0041608 * Rainfall
```

```
Yield_chemical = 1.9056084 + 0.0082430 * Rainfall
```

We now do the prediction using the two different types of fertilizers.

```
predict(model_3, newdata = data.frame(rainfall = 770, fertilizer = "Organic"), interval = "c", level = 0.98)

##      fit     lwr     upr
## 1 5.1095 4.8804 5.3386

predict(model_3, newdata = data.frame(rainfall = 770, fertilizer = "Chemical"), interval = "c", level = 0.98)

##      fit     lwr     upr
## 1 8.2528 8.0073 8.4982
```

We predict a rainfall of 770 with use of Organic fertilizer would give a yield of 5.109491 with 98% confidence interval [4.880409, 5.338573]

We also predict a rainfall of 770 with use of Chemical fertilizer would give a yield of 8.252753 with 98% confidence interval [8.007318, 8.498189]

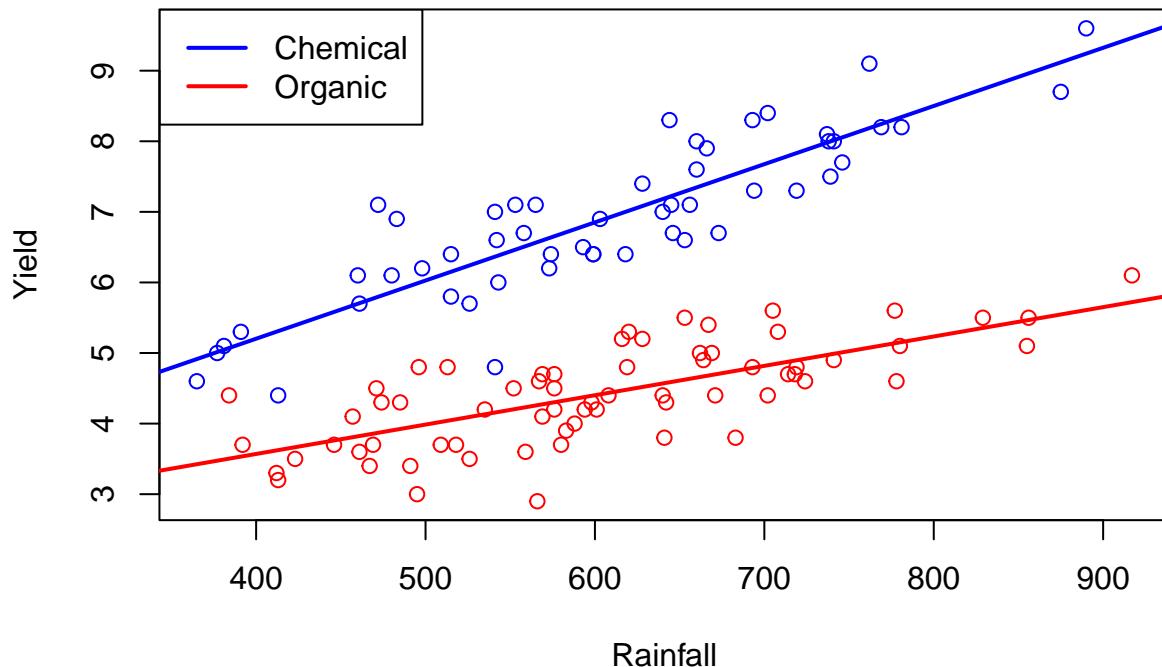
```
#plot of yield ~ rainfall with fertilizer types
plot(Q2data$rainfall, Q2data$yield,
      col = c("blue", "red")[Q2data$fertilizer],
      xlab = "Rainfall",
      ylab = "Yield",
      main = "Scatterplot of Yield vs Rainfall")

abline(a = coef(model_3)[1], b = coef(model_3)[2], col = "blue", lwd = 2)

abline(a = coef(model_3)[1],
       b = coef(model_3)[2] + coef(model_3)[3],
       col = "red", lwd = 2)

legend("topleft",
       legend = c("Chemical", "Organic"),
       col = c("blue", "red"),
       lwd = 2)
```

Scatterplot of Yield vs Rainfall



Question 3 (40 points)

For this question use the data in the file XM225F_q3.csv. The data set has information on 150 patients of a hospital and contains the following variables:

- `los`: length of stay
- `age`: patient's age,
- `bmi`: body mass index,
- `sbp`: systolic blood pressure,
- `dbp`: diastolic blood pressure,
- `hr`: heart rate,
- `cmb`: comorbidity, an index related to the presence of two or more conditions,
- `p.adm`: number of prior admissions,
- `severity`: an index of the severity of the patient's condition

Read the data and create a new data frame named `Q3data`. You have to fit a multiple regression model for length of stay (`los`) as a function of the rest of the variables.

```
library(corrplot)

## Warning: package 'corrplot' was built under R version 4.5.2

## corrplot 0.95 loaded
```

```

library(MASS)

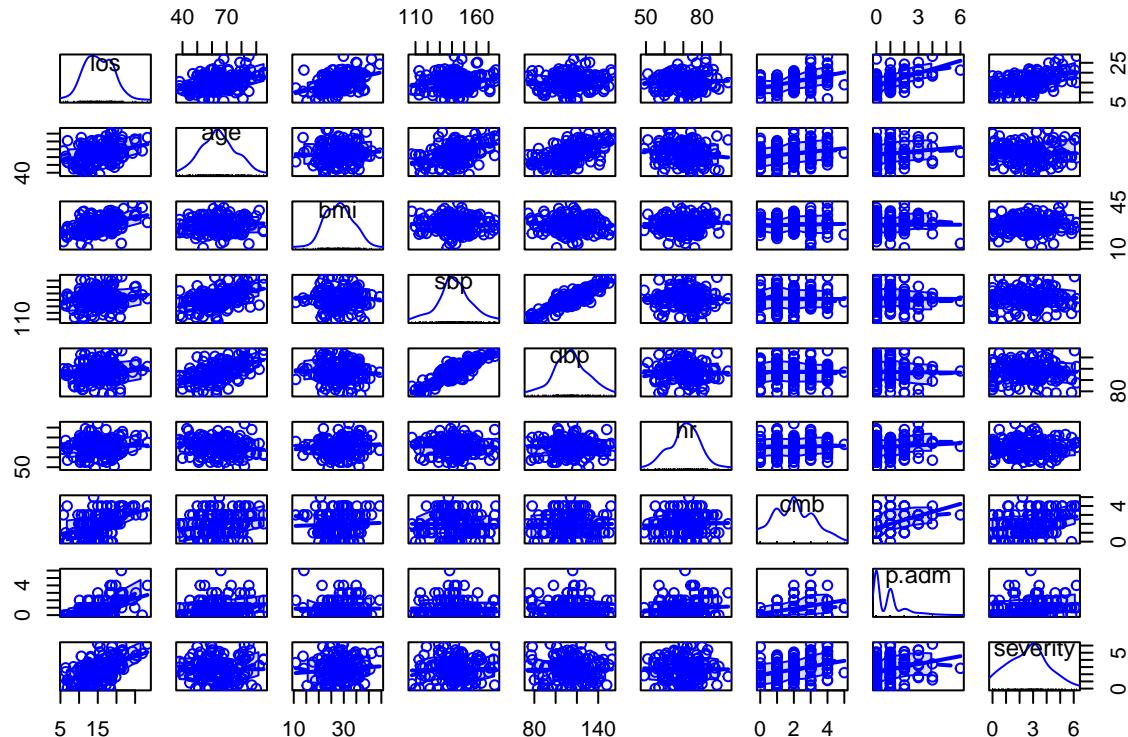
Q3data <- read.csv("XM225F_q3.csv")
str(Q3data)

## 'data.frame':   150 obs. of  9 variables:
## $ los      : num  14 14.1 10.5 13.2 12.3 ...
## $ age      : int  63 54 56 45 60 52 45 95 82 ...
## $ bmi      : num  29.5 28.2 32.8 26.6 30.8 ...
## $ sbp      : int  131 114 141 146 147 150 143 164 160 ...
## $ dbp      : int  103 80 115 121 102 117 113 145 142 135 ...
## $ hr       : int  80 59 80 73 75 74 67 63 69 67 ...
## $ cmb      : int  1 0 2 0 1 3 0 3 2 0 ...
## $ p.adm    : int  0 0 0 0 4 1 1 1 0 ...
## $ severity: num  2.11 1.21 1.71 4.32 3.26 1.21 0.8 4.06 2.11 0.96 ...

```

- (a) Do a scatterplot matrix for the variables in the data set. Calculate and graph the correlation matrix for these variables. Comment on the results.

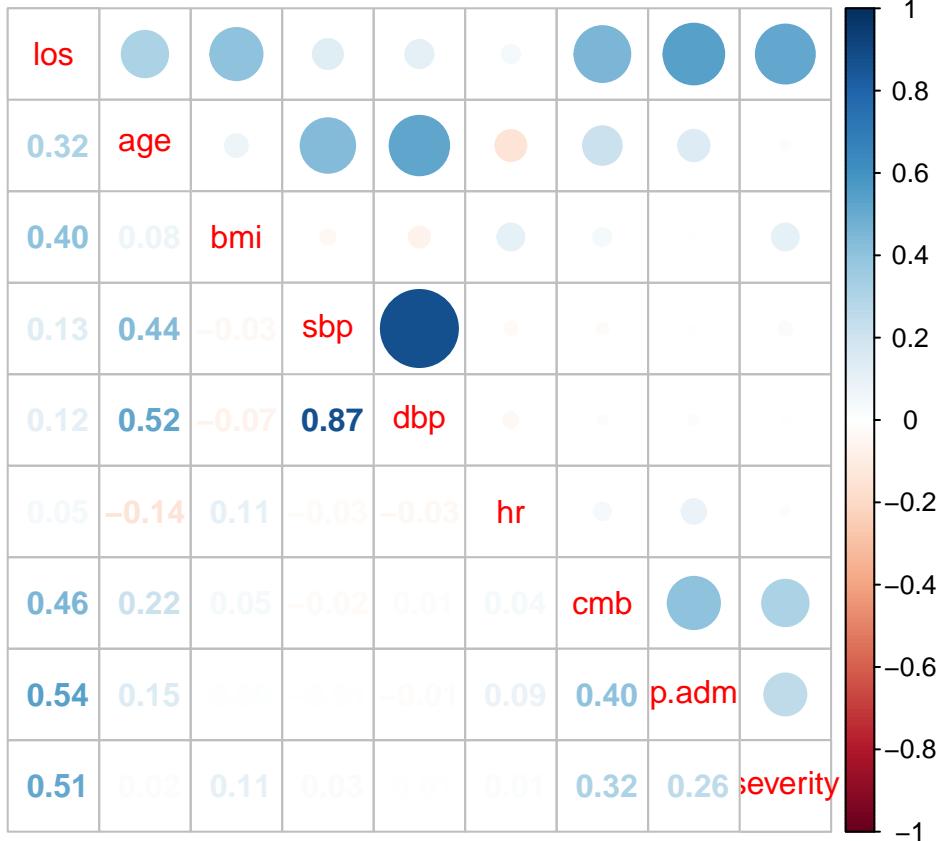
```
scatterplotMatrix(Q3data)
```



```

cor_1 <- cor(Q3data)
corrplot.mixed(cor_1)

```



From the scatterplot matrix, we can see that los has a generally positive correlation with severity, bmi, p.adm, and age we can also see positive correlation with steady values for hr and cmb

The correlation matrix graph shows the trends more clearly. with spb and dpb having showing the strongest positive correlation, seconded by los and p.adm, and then age and dbp.

we also see that LOS seems to be postively correlated mostly with p.adm, severity, cmb, bmi, and age in this order.

- (b) Fit a regression model for length of stay (`los`) as a function of the rest of the variables. With a threshold for the variance inflation factor of 2, use a sequential procedure to eliminate variables that may cause multicollinearity problems.

```
#make the full model and print out its summary
full_model <- lm(los ~ age + bmi + sbp + dbp + hr + cmb + p.adm + severity, data = Q3data)
summary(full_model)
```

```
##
## Call:
## lm(formula = los ~ age + bmi + sbp + dbp + hr + cmb + p.adm +
##     severity, data = Q3data)
##
## Residuals:
##      Min      1Q Median      3Q     Max 
## -5.865 -1.844  0.249  1.417  7.463 
##
```

```

## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.98464   3.18143  -0.94  0.3498
## age          0.05826   0.02171   2.68  0.0082 **
## bmi          0.24812   0.03715   6.68  5.2e-10 ***
## sbp          0.02763   0.03157   0.88  0.3829
## dbp         -0.00718   0.02824  -0.25  0.7995
## hr           -0.00264   0.02652  -0.10  0.9209
## cmb          0.54028   0.20243   2.67  0.0085 **
## p.adm        1.43237   0.21755   6.58  8.4e-10 ***
## severity     0.88519   0.14924   5.93  2.2e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.6 on 141 degrees of freedom
## Multiple R-squared:  0.641, Adjusted R-squared:  0.62
## F-statistic: 31.4 on 8 and 141 DF,  p-value: <2e-16

```

```
vif(full_model) # print out the vif of the model
```

```

##      age      bmi      sbp      dbp      hr      cmb      p.adm
## 1.5694  1.0578  4.3280  4.8459  1.0592  1.3389  1.2527
## severity
## 1.1681

```

#we see that the highest value is dbp and is higher than the threshold of 2, so we get rid of it and change the model

```
mod1 <- update(full_model, .~. - dbp)
vif(mod1)
```

```

##      age      bmi      sbp      hr      cmb      p.adm severity
## 1.4032  1.0431  1.2818  1.0565  1.3385  1.2456  1.1678

```

We now see that this model has all values under the threshold of 2.

- (c) Using a backward selection procedure with a critical α of 0.10 and starting with the variables you selected in (b), obtain a minimal adequate model. Comment on the steps that you take.

The general procedure is to use drop1 to print the single term deletions table and drop the predictor with the highest p-value larger than a_critical of 0.2.

```
drop1(mod1, test = "F")
```

```

## Single term deletions
##
## Model:
## los ~ age + bmi + sbp + hr + cmb + p.adm + severity
##             Df Sum of Sq  RSS AIC F value    Pr(>F)
## <none>            952 293
## age      1      51.0 1003 299    7.61  0.0066 **
## bmi      1     308.0 1260 333   45.94 3.0e-10 ***
## sbp      1      10.0  962 293    1.49  0.2245

```

```

## hr      1     0.1  952 291    0.01  0.9103
## cmb     1     47.9 1000 299    7.15  0.0084 **
## p.adm   1    295.9 1248 332   44.14 6.1e-10 ***
## severity 1    237.8 1190 325   35.47 1.9e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

mod2 <- update(mod1, .~. -hr) #here we drop hr which has a value of 0.91 > 0.1

drop1(mod2, test = "F")

## Single term deletions
##
## Model:
## los ~ age + bmi + sbp + cmb + p.adm + severity
##          Df Sum of Sq RSS AIC F value Pr(>F)
## <none>        952 291
## age      1     53.6 1006 297    8.05  0.0052 **
## bmi      1    311.8 1264 332   46.84 2.1e-10 ***
## sbp      1     9.9  962 291    1.49  0.2247
## cmb      1     47.8 1000 297    7.19  0.0082 **
## p.adm    1    297.9 1250 330   44.76 4.7e-10 ***
## severity 1    238.7 1191 323   35.86 1.6e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

mod3 <- update(mod2, .~. -sbp) #here we drop sbp, which has a value of 0.22 > 0.1

drop1(mod3, test = "F")

## Single term deletions
##
## Model:
## los ~ age + bmi + cmb + p.adm + severity
##          Df Sum of Sq RSS AIC F value Pr(>F)
## <none>        962 291
## age      1     98.0 1060 303   14.67 0.00019 ***
## bmi      1    304.9 1267 330   45.64 3.3e-10 ***
## cmb      1     43.2 1005 295    6.47  0.01205 *
## p.adm    1    293.4 1255 329   43.93 6.3e-10 ***
## severity 1    249.0 1211 323   37.28 9.0e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

#Since all values are over a_crit, we got our minimally adequate model.
summary(mod3)

```

```

##
## Call:
## lm(formula = los ~ age + bmi + cmb + p.adm + severity, data = Q3data)
##
## Residuals:
```

```

##      Min     1Q Median     3Q    Max
## -6.153 -1.713  0.277  1.598  7.348
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.5939    1.4835   -0.40  0.68950
## age          0.0682    0.0178    3.83  0.00019 ***
## bmi          0.2451    0.0363    6.76  3.3e-10 ***
## cmb          0.5074    0.1995    2.54  0.01205 *
## p.adm        1.4215    0.2145    6.63  6.3e-10 ***
## severity     0.9022    0.1477    6.11  9.0e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.58 on 144 degrees of freedom
## Multiple R-squared:  0.637, Adjusted R-squared:  0.624
## F-statistic: 50.5 on 5 and 144 DF,  p-value: <2e-16

```

- (d) Starting with the complete model, fit a model using a stepwise procedure for the AIC criterion. Compare your final model with the result of (c).

```
aic_model <- stepAIC(full_model) #use AIC to get a minimal model.
```

```

## Start:  AIC=295.09
## los ~ age + bmi + sbp + dbp + hr + cmb + p.adm + severity
##
##              Df Sum of Sq  RSS AIC
## - hr         1     0.1  951 293
## - dbp        1     0.4  952 293
## - sbp        1     5.2  957 294
## <none>
## - cmb         1    48.1  999 300
## - age         1    48.6 1000 301
## - severity   1   237.4 1189 327
## - p.adm       1   292.5 1244 333
## - bmi         1   300.9 1252 334
##
## Step:  AIC=293.1
## los ~ age + bmi + sbp + dbp + cmb + p.adm + severity
##
##              Df Sum of Sq  RSS AIC
## - dbp        1     0.5  952 291
## - sbp        1     5.2  957 292
## <none>
## - cmb         1    48.0  999 298
## - age         1    51.1 1003 299
## - severity   1   238.2 1190 325
## - p.adm       1   294.8 1246 332
## - bmi         1   305.2 1257 333
##
## Step:  AIC=291.17
## los ~ age + bmi + sbp + cmb + p.adm + severity
##
```

```

##          Df Sum of Sq  RSS AIC
## - sbp      1     9.9  962 291
## <none>           952 291
## - cmb      1    47.8 1000 297
## - age      1    53.6 1006 297
## - severity 1   238.7 1191 323
## - p.adm    1   297.9 1250 330
## - bmi      1   311.8 1264 332
##
## Step:  AIC=290.73
## los ~ age + bmi + cmb + p.adm + severity
##
##          Df Sum of Sq  RSS AIC
## <none>           962 291
## - cmb      1    43.2 1005 295
## - age      1    98.0 1060 303
## - severity 1   249.0 1211 323
## - p.adm    1   293.4 1255 329
## - bmi      1   304.9 1267 330

summary(aic_model) # print out the summary

##
## Call:
## lm(formula = los ~ age + bmi + cmb + p.adm + severity, data = Q3data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.153 -1.713  0.277  1.598  7.348
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.5939     1.4835   -0.40  0.68950
## age          0.0682     0.0178    3.83  0.00019 ***
## bmi          0.2451     0.0363    6.76  3.3e-10 ***
## cmb          0.5074     0.1995    2.54  0.01205 *
## p.adm        1.4215     0.2145    6.63  6.3e-10 ***
## severity     0.9022     0.1477    6.11  9.0e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.58 on 144 degrees of freedom
## Multiple R-squared:  0.637, Adjusted R-squared:  0.624
## F-statistic: 50.5 on 5 and 144 DF,  p-value: <2e-16

```

the two models are exactly the same using the same predictors and using the same predictors gives us the same values of R^2 , adj R^2 , and Residual standard error.

- (e) Write an equation for the final model in (c) and interpret the coefficients. Predict the length of stay (`los`) for a patient with the following covariates. Include a confidence interval at the 98% level.

Table 1: Covariates for prediction

age	bmi	sbp	dbp	hr	cmb	p.adm	severity
46	30.1	132	93	77	3	1	2.1

```

summary(mod3)

##
## Call:
## lm(formula = los ~ age + bmi + cmb + p.adm + severity, data = Q3data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.153 -1.713  0.277  1.598  7.348
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.5939    1.4835   -0.40  0.68950
## age         0.0682    0.0178    3.83  0.00019 ***
## bmi         0.2451    0.0363    6.76  3.3e-10 ***
## cmb         0.5074    0.1995    2.54  0.01205 *
## p.adm       1.4215    0.2145    6.63  6.3e-10 ***
## severity    0.9022    0.1477    6.11  9.0e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.58 on 144 degrees of freedom
## Multiple R-squared:  0.637, Adjusted R-squared:  0.624
## F-statistic: 50.5 on 5 and 144 DF,  p-value: <2e-16

```

The equation for the model is:

$\text{los} = 0.59390 + 0.06819 \text{age} + 0.24510 \text{bmi} + 0.50735 \text{cmb} + 1.42154 \text{p.adm} + 0.90215 \text{severity}$

```

predict_data <- data.frame(age = 46, bmi = 30.1, sbp = 132, dbp = 93,
                            hr = 77, cmb = 3, p.adm = 1, severity = 2.1)

```

```

predict(mod3, newdata = predict_data, interval = "c", level = 0.98)

```

```

##      fit     lwr     upr
## 1 14.759 13.579 15.938

```

with the given inputs, we predict that length of stay (los) would be 14.7587 with a 98% confidence interval of [13.57924, 15.93816]

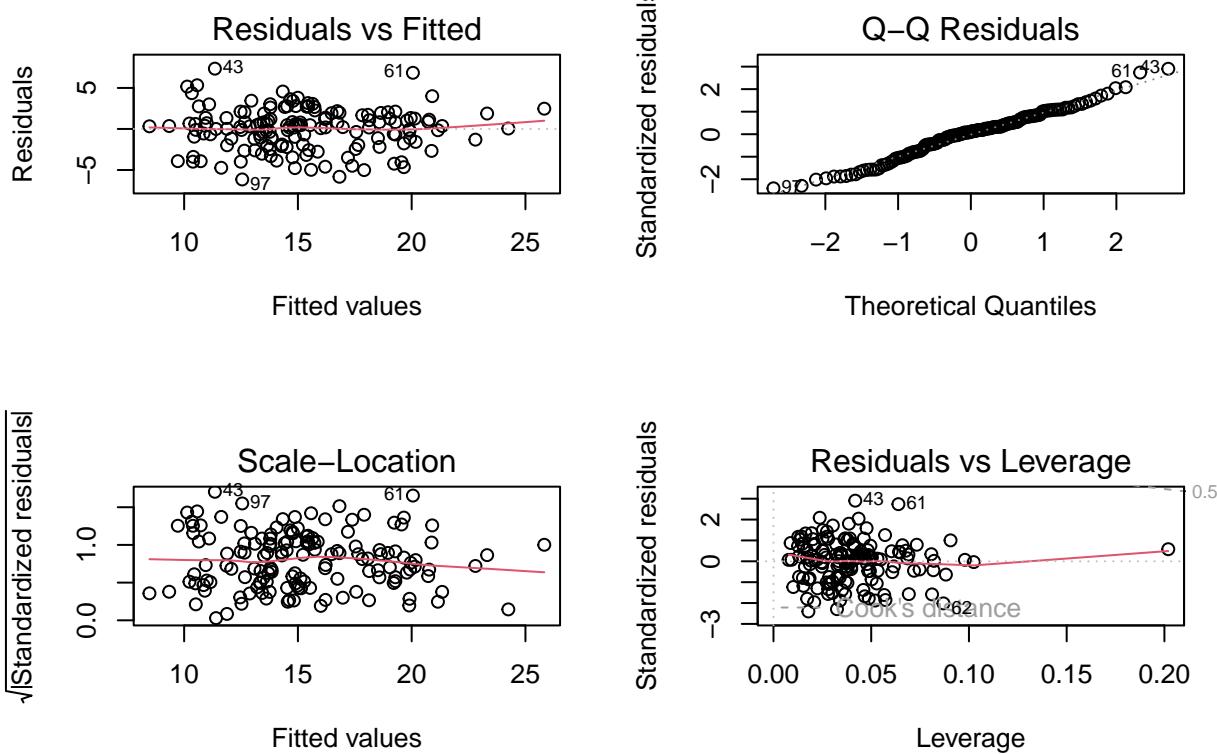
- (f) State explicitly the assumptions required for the multiple regression model. Using graphs and tests verify whether these assumptions are satisfied.

our assumptions are Linearity, Independence, Constant variance, normality of residuals. we test these out using the plots and further testing with shapiro-wilk and ncv tests.

```

par(mfrow = c(2, 2))
plot(mod3)

```



```
par(mfrow = c(1, 1))
```

From the residuals vs fitted we can see that the line is mostly horizontal based on 0, which indicates linearity from the qq-resudals we can see the line almost perfectly fitting the points, indicating normality.

From the scale location, we can see the line being mostly horizontal with a roughly even spread of points, indicating constant variance but may need further testing.

From residuals vs leverage we see that there are no influential points, with most having weak influence.

We can then do the shapiro-wilk and ncv tests.

```
shapiro.test(residuals(mod3))
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: residuals(mod3)  
## W = 0.986, p-value = 0.14
```

```
ncvTest(mod3)
```

```
## Non-constant Variance Score Test  
## Variance formula: ~ fitted.values  
## Chisquare = 0.99079, Df = 1, p = 0.32
```

with a p-value of $0.1384 > 0.02$ for the shapiro test, we fail to reject the null hypothesis and say the residuals are normally distributed.

with a p-value of $0.31955 > 0.02$ for the ncv test, we fail to reject the null hypothesis and say that the residuals have constant variance.