Statistics for Data Analysis - Coursework

Dhruv Gandotra
AB26054

December 15, 2020

Question 1

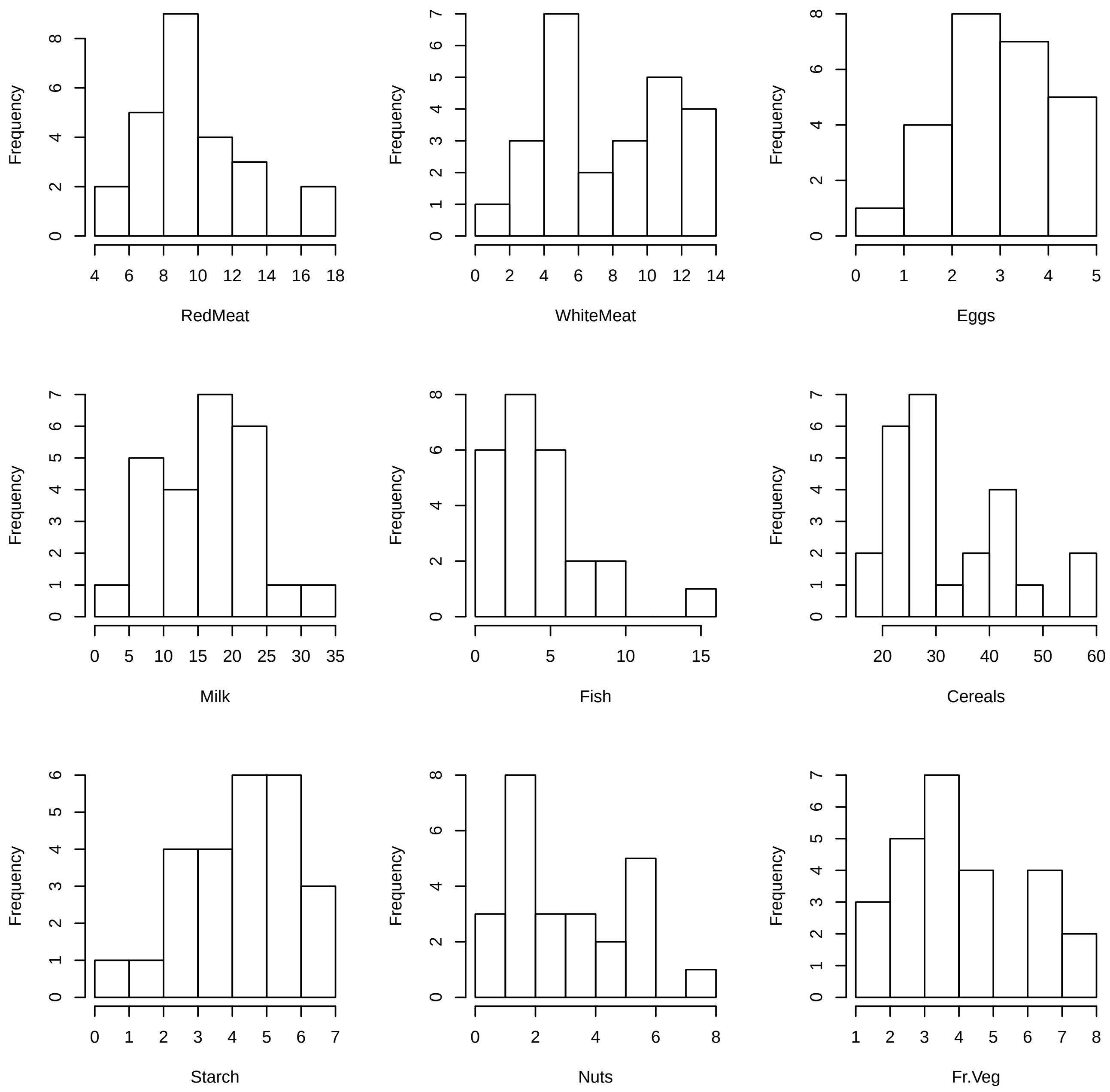
a) Generated the summary statistics of every food category in Protein

```
protein<-read.csv("protein.csv",header=TRUE)
attach(protein)
lapply(protein[2:10], summary)
```

```
## $RedMeat
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   4.400   7.800   9.500   9.828  10.600  18.000 
## 
## $WhiteMeat
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   1.400   4.900   7.800   7.896  10.800  14.000 
## 
## $Eggs
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   0.500   2.700   2.900   2.936   3.700   4.700 
## 
## $Milk
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    4.90   11.10   17.60   17.11   23.30   33.70 
## 
## $Fish
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   0.200   2.100   3.400   4.284   5.800  14.200 
## 
## $Cereals
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   18.60   24.30   28.00   32.25   40.10   56.70 
## 
## $Starch
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   0.600   3.100   4.700   4.276   5.700   6.500 
## 
## $Nuts
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   0.700   1.500   2.400   3.072   4.700   7.800 
## 
## $Fr.Veg
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   1.400   2.900   3.800   4.136   4.900   7.900 
```

b) Generated a histogram for each food category in Protein

```
par(mfrow=c(3,3))
par(mar = c(5, 5, 2, 1))
lapply(2:10,function(x)hist(protein[,x],main=NULL,xlab=colnames(protein[x])))
```



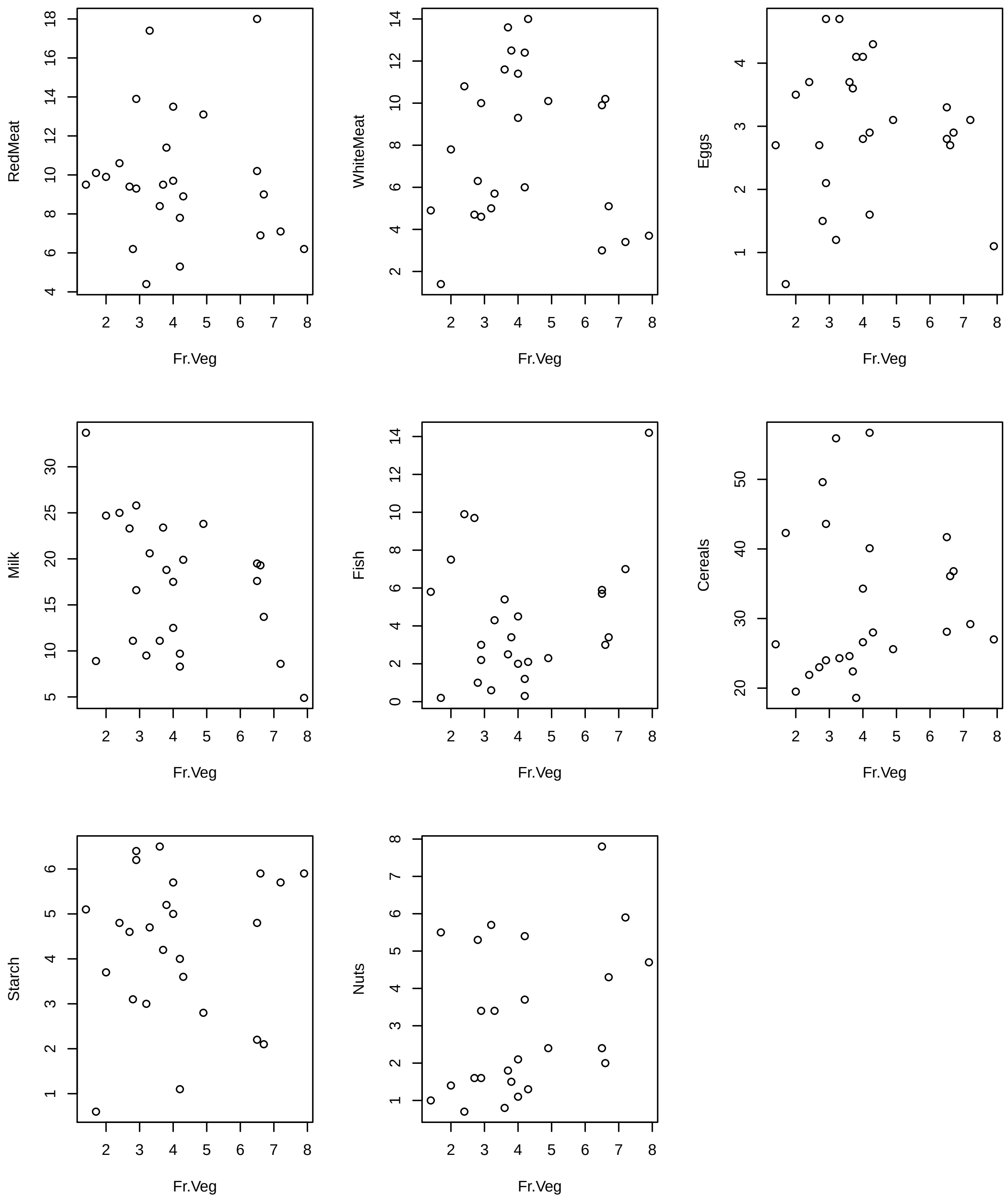
c) Generated the Pearson correlation coefficient of Fr.Veg with every other food category

```
cor(Fr.Veg,protein[,2:9])
```

```
##          RedMeat  WhiteMeat        Eggs       Milk      Fish    Cereals
## [1,] -0.07422123 -0.0613167 -0.04551755 -0.4083641 0.2661387 0.04654808
##          Starch      Nuts
## [1,] 0.08440956 0.3749697
```

d) Generated scatter plots of Fr.Veg vs every other food category

```
par(mfrow=c(3,3))
par(mar = c(5, 5, 2, 1))
lapply(2:9,function(x) plot(Fr.Veg,protein[,x],main=NULL,ylab=colnames(protein[x])))
```



e) Created a function that generates the 95% confidence intervals for the mean of the specified food category in Protein, then applied the function to each food category in Protein

```
ci<-function(x,data){
  t<-t.test(data[,x], alternative="t", mu=mean(data[,x]),conf.level=0.95,var.equal=TRUE)
  paste('95% Confidence Interval for',colnames(data[x]),'is',t$conf.int[1],',',t$conf.int[2])
  }
lapply(2:10, function(x) ci(x,protein))
```

```
## [[1]]
## [1] "95% Confidence Interval for RedMeat is 8.44639397066365 , 11.2096060293363"
##
## [[2]]
## [1] "95% Confidence Interval for WhiteMeat is 6.37115836903857 , 9.42084163096143"
##
## [[3]]
## [1] "95% Confidence Interval for Eggs is 2.47467057791314 , 3.39732942208686"
##
## [[4]]
## [1] "95% Confidence Interval for Milk is 14.1790285205081 , 20.0449714794919"
##
## [[5]]
## [1] "95% Confidence Interval for Fish is 2.87950325439136 , 5.68849674560864"
##
## [[6]]
## [1] "95% Confidence Interval for Cereals is 27.7178308870067 , 36.7781691129933"
##
## [[7]]
## [1] "95% Confidence Interval for Starch is 3.6014829211123 , 4.9505170788877"
##
## [[8]]
## [1] "95% Confidence Interval for Nuts is 2.25235072115611 , 3.89164927884389"
##
## [[9]]
## [1] "95% Confidence Interval for Fr.Veg is 3.39138536614383 , 4.88061463385617"
```

f) Performed a t-test with the alternative hypothesis that the average consumption of starch is greater than the average consumption of nuts, and the null hypothesis as the inverse

```
t.test(Starch,Nuts,alternative="g",mu=0,paired=TRUE,var.equal=TRUE)
```

```
##
##  Paired t-test
##
## data:  Starch and Nuts
## t = 1.9338, df = 24, p-value = 0.03251
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  0.1387793       Inf
## sample estimates:
## mean of the differences
##                   1.204
```

Generated p-value is 0.03251, which is less than the significance level 0.05. This indicates strong evidence against the null hypothesis, so we reject the null hypothesis. Therefore, the t-test suggests that the average consumption of starch is in fact less than the average consumption of nuts, as predicted.

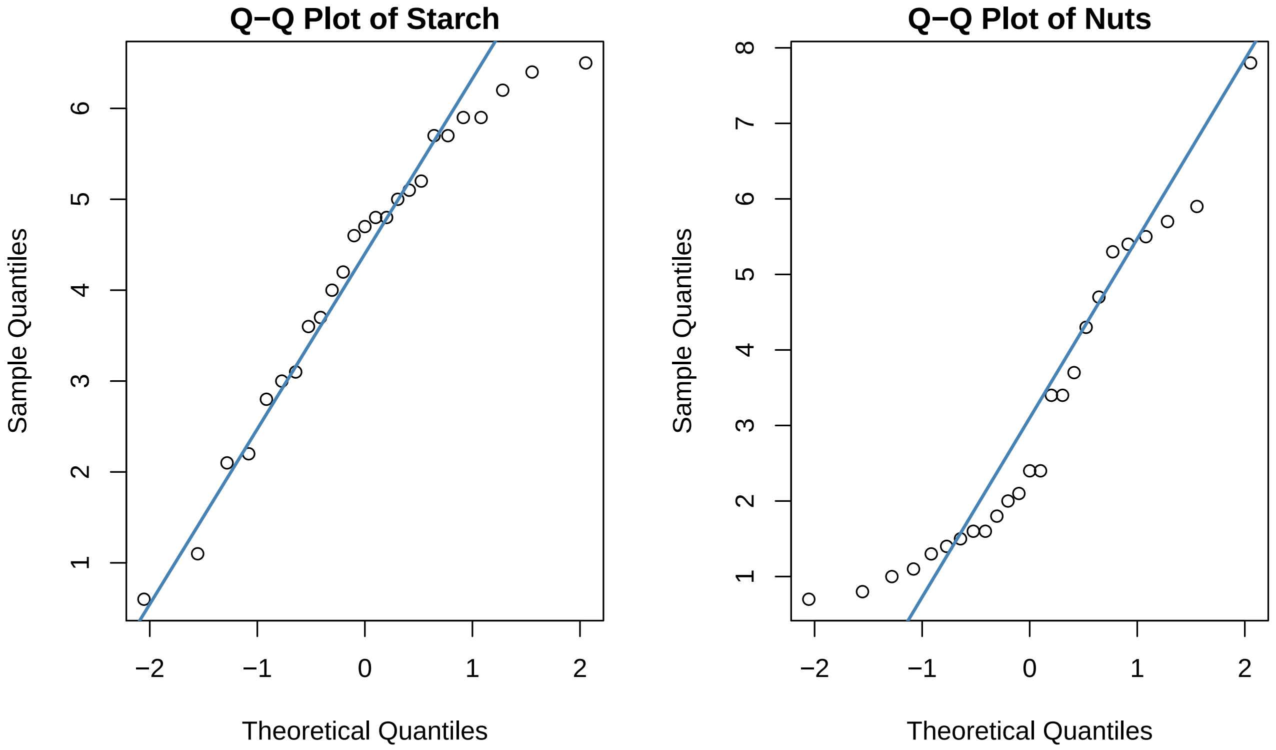
In this t-test, we assumed that the variances of starch and nuts are equal. To check if our assumption is reasonable, we first have to check if the data sets are normally distributed. This can be done in two ways.

Firstly, with a Shapiro-Wilk's test. It is a normality test based on the correlation between the data and the corresponding normal scores. [1]

```
shapiro.test(Starch)

##
##  Shapiro-Wilk normality test
##
## data:  Starch
## W = 0.94501, p-value = 0.193

shapiro.test(Nuts)

##
##  Shapiro-Wilk normality test
##
## data:  Nuts
## W = 0.90262, p-value = 0.02093
```

Secondly, with a Q-Q plot

```
par(mfrow=c(1,2))
par(mar = c(5, 4, 1.5, 2))
qqnorm(Starch, pch = 1, main='Q-Q Plot of Starch', frame = TRUE)
qqline(Starch, col = "steelblue", lwd = 2)
qqnorm(Nuts, pch = 1,main='Q-Q Plot of Nuts', frame = TRUE)
qqline(Nuts, col = "steelblue", lwd = 2)
```



Starch: p-value of 0.193 (>0.05) and the Q-Q plot suggests that Starch follows a normal distribution

Nuts: p-value of 0.02093 (<0.05) and the Q-Q plot suggests that Nuts doesn't follow a normal distribution

> Note: The reason we have to use both these tools is because normality tests are sensitive to sample size. Small samples most often pass normality tests. In this case, the sample size is 25. Therefore, it's important to combine visual inspection and significance tests in order to take the right decision. [2]

Now, if we assume that the data sets do not follow a normal distribution, we should perform a Fligner-Killeen Test for the homogeneity of the variances. This test is used when the data is non-normally distributed. [3]

```
fligner.test(Starch, Nuts)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Starch and Nuts
## Fligner-Killeen:med chi-squared = 23.927, df = 21, p-value = 0.2966
```

Generated p-value is 0.2966, which is much higher than 0.05. This indicates that the variances are homogeneous.

However, if we assume that the data sets do indeed follow a normal distribution, we should perform an F-test to check if our assumption that the variances are equal is reasonable. The null hypothesis will be that the variances are equal (ratio = 1), while the alternative hypothesis will be the inverse.

```
var.test(Starch,Nuts,ratio=1, alternative="t")
```

```
##
##  F test to compare two variances
##
## data:  Starch and Nuts
## F = 0.67722, num df = 24, denom df = 24, p-value = 0.3463
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.2984298 1.5368000
## sample estimates:
## ratio of variances
##            0.67722
```

Generated p-value is 0.3463, which is much higher than the significance level 0.05. This indicates very weak evidence against the null hypothesis, so we fail to reject the null hypothesis. Similarly to the previous method, this test also indicates that the variances are homogeneous.

Finally, both tests for the homogeneity of the variance generate high p-values (0.2966 & 0.3463), which clearly indicate that our assumption that the variances are equal is valid.

Question 2

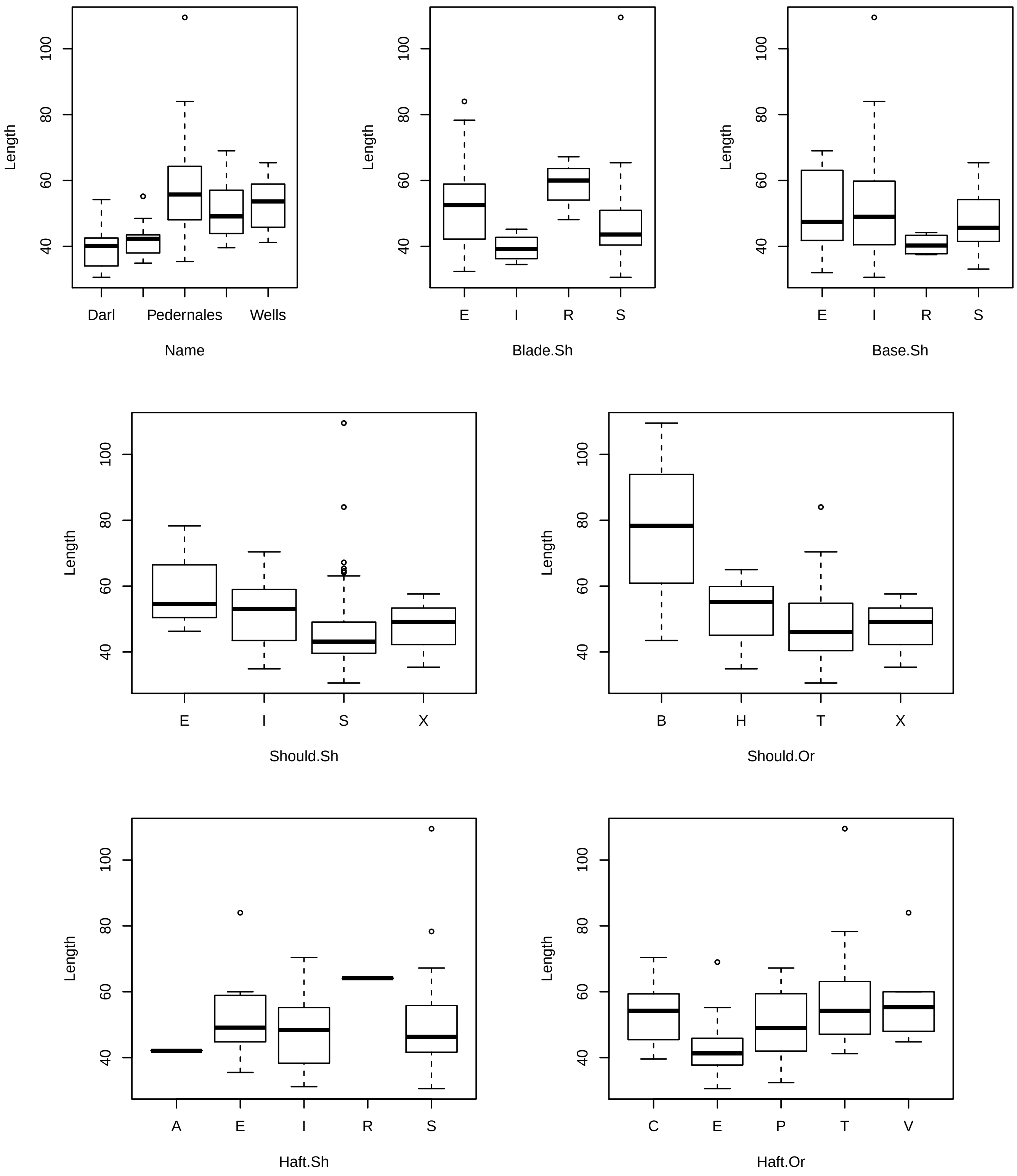
a) Scaling: Qualitative, Nominal - Name, Blade.Sh, Base.Sh, Should.Sh, Should.Or, Haft.Sh, Haft.Or
Numerical, Continuous - Length, Width, Thickness, B.Width, J.Width, H.Length, Weight

```
DartPoints<-read.csv("DartPoints.csv",header=TRUE)
attach(DartPoints)
head(DartPoints)
```

```
##   X Name Length Width Thickness B.Width J.Width H.Length Weight Blade.Sh
## 1 1 Darl   42.8  15.8       5.8    11.3    10.6     11.6    3.6        S
## 2 2 Darl   40.5  17.4       5.8      NA    13.7     12.9    4.5        S
## 3 3 Darl   37.5  16.3       6.1    12.1    11.3      8.2    3.6        S
## 4 4 Darl   40.3  16.1       6.3    13.5    11.7      8.3    4.0        S
## 5 5 Darl   30.6  17.1       4.0    12.6    11.2      8.9    2.3        S
## 6 6 Darl   41.8  16.8       4.1    12.7    11.5     11.0    3.0        S
##   Base.Sh Should.Sh Should.Or Haft.Sh Haft.Or
## 1       I         S         T       S       E
## 2       I         S         T       S       E
## 3       I         S         T       S       E
## 4       I         S         T       S       E
## 5       I         S         T       S       E
## 6       E         I         T       I       C
```

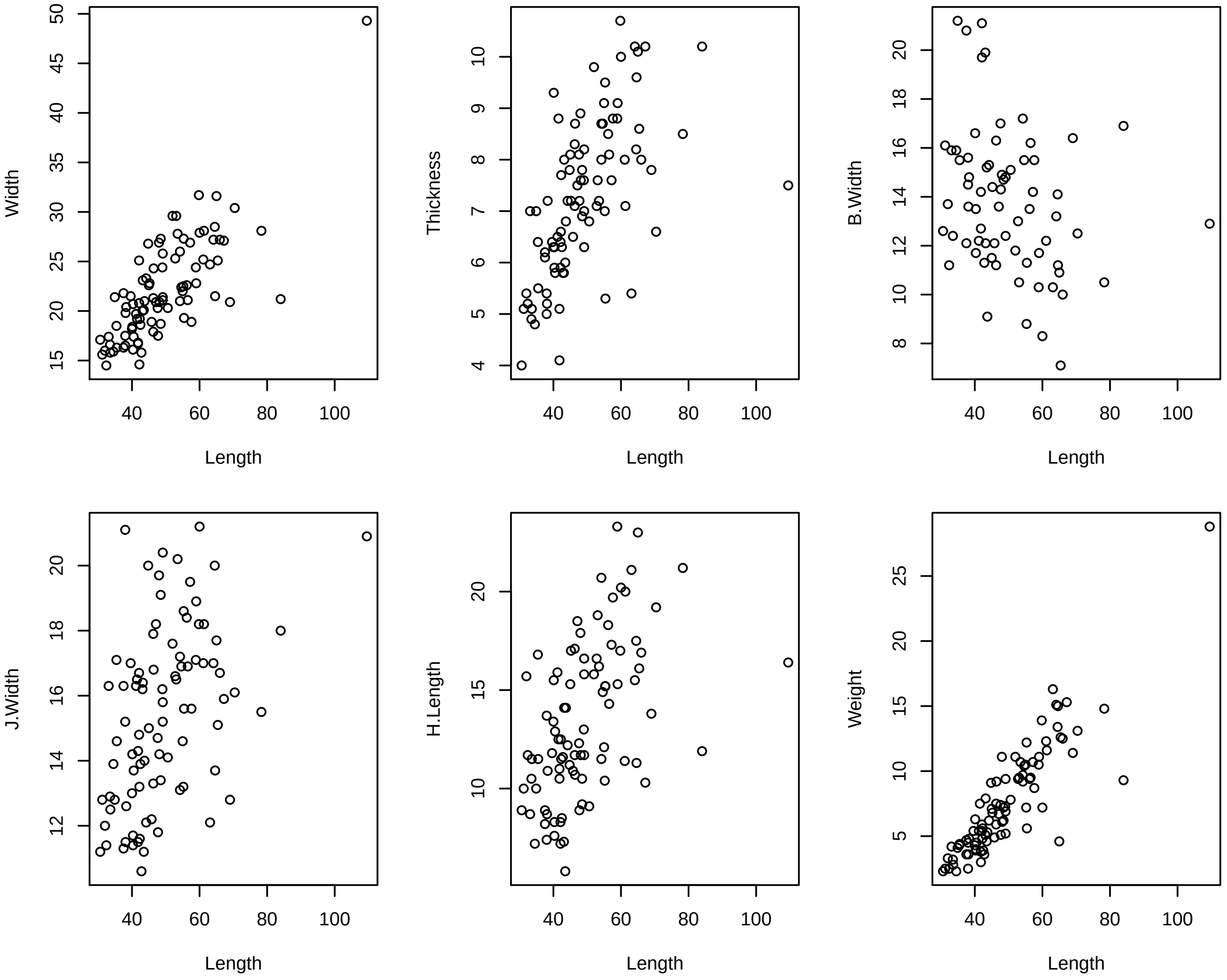
b) Firstly, generated boxplots of Length categorised by every nominal variable

```
par(mar = c(6,4,1,0.1))
layout(matrix(c(1,1,1,1,1,0,2,2,2,2,2,0,3,3,3,3,3,0,4,4,4,4,4,4,4,0,
5,5,5,5,5,5,5,0,0,6,6,6,6,6,6,6,0,7,7,7,7,7,7,7,0),3,17,byrow=TRUE))
lapply(c(2,10:15),function(x) boxplot(Length~DartPoints[,x],xlab=colnames(DartPoints[x])))
```



Next, generated scatter plots of Length vs every other continuous variable

```
par(mfrow=c(2,3))
par(mar = c(5, 5, 1, 1))
lapply(4:9,function(x) plot(Length,DartPoints[,x],ylab=colnames(DartPoints[x])))
```



Interestingly, from the plots we can infer that Length has a substantial correlation with most of the continuous variables, specially Width and Weight

c) Generated the Pearson correlation coefficient of Length with every other continuous variable

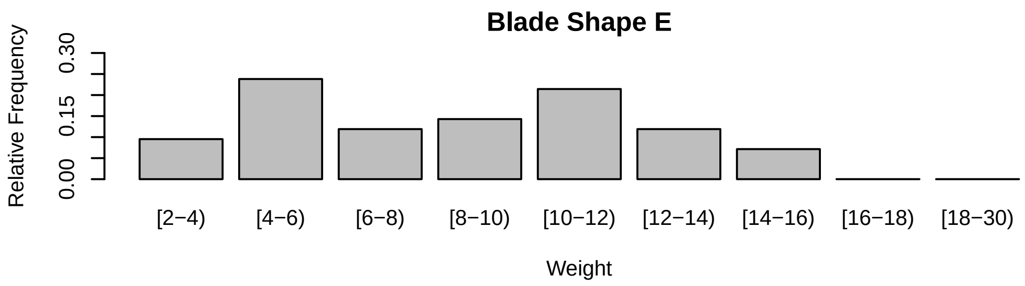
```
cor(Length,DartPoints[,4:9],use='complete.obs')

##          Width Thickness    B.Width   J.Width  H.Length    Weight
## [1,] 0.7845433 0.5914023 -0.2831846 0.5001206 0.5489668 0.8819988
```

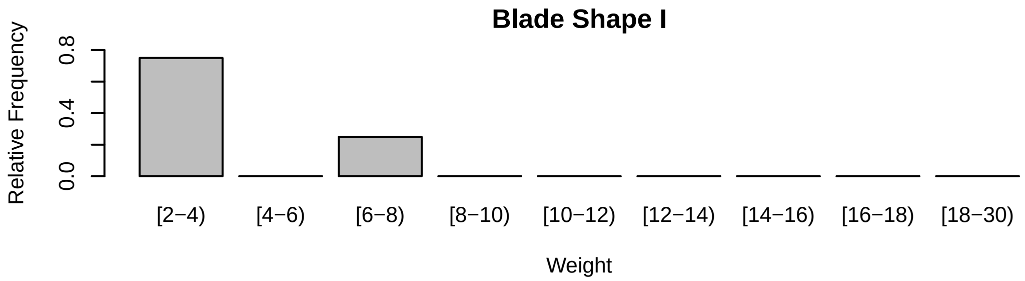
These values provide further evidence of the strong correlation between Length, and Width and Weight. Also, they provide some evidence for a reasonable correlation with Thickness, J.Width, and B.Length.

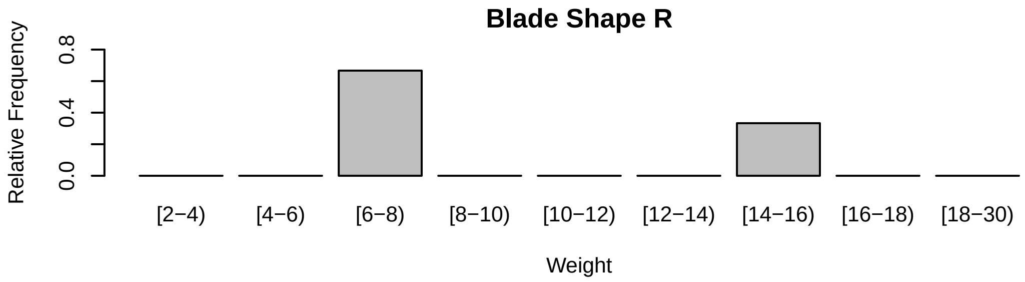
d) Firstly, created weight intervals for Weight. Then split the data by Blade Shape (E,I,R,S), then by the intervals created previously. Finally, generated relative frequency histograms for each Blade Shape.

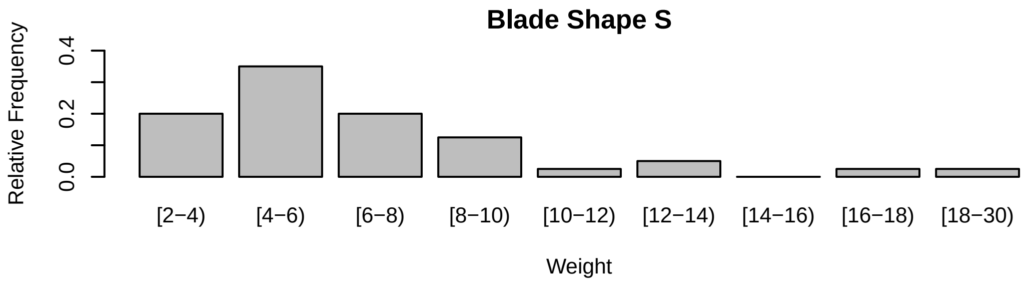
```
par(mfrow=c(4,1))
par(mar = c(5, 4, 2.5, 4), oma = c(0,0,0.5,0))
breaks <- c(2,4,6,8,10,12,14,16,18,30)
tags <- c("[2-4)","[4-6)","[6-8)","[8-10)","[10-12)","[12-14)","[14-16)","[16-18)","[18-30)")
```

```
E<-DartPoints[which(Blade.Sh == 'E'),]
grouptagsE <- cut(E$Weight, breaks=breaks,include.lowest=TRUE, right=FALSE, labels=tags)
barplot(table(grouptagsE)/length(E$Weight),main="Blade Shape E",
xlab='Weight',ylab="Relative Frequency",ylim=c(0,0.3))
I<-DartPoints[which(Blade.Sh == 'I'),]
grouptagsI <- cut(I$Weight, breaks=breaks,include.lowest=TRUE, right=FALSE, labels=tags)
barplot(table(grouptagsI)/length(I$Weight),main="Blade Shape I",
xlab='Weight',ylab="Relative Frequency",ylim=c(0,0.8))
R<-DartPoints[which(Blade.Sh == 'R'),]
grouptagsR <- cut(R$Weight, breaks=breaks,include.lowest=TRUE, right=FALSE, labels=tags)
barplot(table(grouptagsR)/length(R$Weight),main="Blade Shape R",
xlab='Weight',ylab="Relative Frequency",ylim=c(0,0.8))
S<-DartPoints[which(Blade.Sh == 'S'),]
grouptagsS <- cut(S$Weight, breaks=breaks,include.lowest=TRUE, right=FALSE, labels=tags)
barplot(table(grouptagsS)/length(S$Weight),main="Blade Shape S",
xlab='Weight',ylab="Relative Frequency",ylim=c(0,0.4))
```









e) Firstly, generated a summary of the data to see the missing values (NAs)

```
summary(DartPoints)
```

```
##       X                Name        Length           Width
##  Min.   : 1.0   Darl      :28   Min.   : 30.60   Min.   :14.50
##  1st Qu.:23.5   Ensor     :10   1st Qu.: 40.85   1st Qu.:18.55
##  Median :46.0   Pedernales:32   Median : 47.10   Median :21.10
##  Mean   :46.0   Travis    :11   Mean   : 49.33   Mean   :22.08
##  3rd Qu.:68.5   Wells     :10   3rd Qu.: 55.80   3rd Qu.:25.15
##  Max.   :91.0                   Max.   :109.50   Max.   :49.30
##
##    Thickness         B.Width          J.Width          H.Length
##  Min.   : 4.000   Min.   : 7.10   Min.   :10.60   Min.   : 5.80
##  1st Qu.: 6.250   1st Qu.:11.70   1st Qu.:13.12   1st Qu.:10.50
##  Median : 7.200   Median :13.60   Median :15.55   Median :12.50
##  Mean   : 7.271   Mean   :13.75   Mean   :15.40   Mean   :13.41
##  3rd Qu.: 8.250   3rd Qu.:15.50   3rd Qu.:17.07   3rd Qu.:16.30
##  Max.   :10.700   Max.   :21.20   Max.   :21.20   Max.   :23.30
##                   NA's   :22      NA's   :1
##      Weight        Blade.Sh  Base.Sh   Should.Sh Should.Or Haft.Sh   Haft.Or
##  Min.   : 2.300   E   :42   E   : 6   E   : 3   B   : 3   A   : 2   C   : 8
##  1st Qu.: 4.550   I   : 4   I   :53   I   :37   H   :11   E   : 9   E   :32
##  Median : 6.800   R   : 3   R   : 4   S   :46   T   :72   I   :22   P   :27
##  Mean   : 7.643   S   :40   S   :26   X   : 3   X   : 3   R   : 1   T   :17
##  3rd Qu.:10.050   NA's: 2   NA's: 2   NA's: 2   NA's: 2   S   :55   V   : 5
##  Max.   :28.800                                           NA's: 2   NA's: 2
##
```

Then, removed the columns: X (Just a count) and B.width (Too many missing values). Also, only selected rows without NAs, and named the resulting data-set, DP1.

```
DP1<-DartPoints[c(1:27,29:37,39:67,69:91),c(2:5,7:15)]
summary(DP1)
```

```
##           Name          Length           Width          Thickness
##  Darl      :27   Min.   : 30.60   Min.   :14.50   Min.   : 4.000
##  Ensor     : 9   1st Qu.: 41.02   1st Qu.:18.57   1st Qu.: 6.175
##  Pedernales:31   Median : 47.35   Median :21.10   Median : 7.200
##  Travis    :11   Mean   : 49.49   Mean   :22.05   Mean   : 7.277
##  Wells     :10   3rd Qu.: 56.27   3rd Qu.:25.12   3rd Qu.: 8.350
##                  Max.   :109.50   Max.   :49.30   Max.   :10.700
##     J.Width         H.Length          Weight        Blade.Sh Base.Sh Should.Sh
##  Min.   :10.60   Min.   : 5.80   Min.   : 2.300   E:42     E: 6    E: 3
##  1st Qu.:13.18   1st Qu.:10.50   1st Qu.: 4.575   I: 3     I:52    I:37
##  Median :15.55   Median :12.40   Median : 6.800   R: 3     R: 4    S:45
##  Mean   :15.39   Mean   :13.40   Mean   : 7.695   S:40     S:26    X: 3
##  3rd Qu.:17.02   3rd Qu.:16.25   3rd Qu.:10.425
##  Max.   :21.20   Max.   :23.30   Max.   :28.800
##  Should.Or Haft.Sh Haft.Or
##  B: 3      A: 2    C: 8
##  H:11      E: 9    E:32
##  T:71      I:22    P:27
##  X: 3      R: 1    T:16
##            S:54    V: 5
##
```

Then, created a linear model with Weight as the response variable, DP1 as the data-set, and all the columns of DP1 as the predictor variables.

After that, used the stepAIC function to perform step-wise selection on the linear model in both directions. AIC stands for Akaike Information Criteria, and it quantifies the amount of information loss due to simplification of the model. It modifies the linear model by removing or adding predictor variables based on its AIC value. Hence, we prefer the model with the lowest AIC value. It is similar to the adjusted R-squared in that, it also penalises for adding more variables to the model. It also removes the multicollinearity, if it exists. Therefore, it is a very useful tool to produce a model with the optimum selection of predictor variables. [4]

Note: One thing to remember is that there are problems with step-wise variable selection. For example, when you remove some of the non-significant predictor variables, other variables that are correlated with those may become significant. However, for the purpose of this exercise, we will ignore this. [5]

```
library(MASS)
fit <- lm(Weight ~.,DP1)
step <- stepAIC(fit, direction="both")
```

```
## Start:  AIC=108.92
## Weight ~ Name + Length + Width + Thickness + J.Width + H.Length +
##     Blade.Sh + Base.Sh + Should.Sh + Should.Or + Haft.Sh + Haft.Or
##
##             Df Sum of Sq    RSS    AIC
## - Haft.Sh    4     6.104 163.06 104.28
## - Name       4     6.203 163.16 104.33
## - Base.Sh    3     5.506 162.46 105.95
## - J.Width    1     0.161 157.12 107.01
## - Haft.Or    4    12.865 169.82 107.85
## - H.Length   1     2.236 159.19 108.17
## - Thickness  1     3.149 160.11 108.67
## <none>                   156.96 108.92
## - Should.Sh  2    10.131 167.09 110.42
## - Should.Or  2    13.762 170.72 112.32
## - Blade.Sh   3    17.765 174.72 112.36
## - Width      1    80.490 237.45 143.35
## - Length     1    81.164 238.12 143.60
##
## Step:  AIC=104.28
## Weight ~ Name + Length + Width + Thickness + J.Width + H.Length +
##     Blade.Sh + Base.Sh + Should.Sh + Should.Or + Haft.Or
##
##             Df Sum of Sq    RSS     AIC
## - Name       4     5.988 169.05  99.451
## - J.Width    1     0.444 163.50 102.517
## - Base.Sh    3     8.054 171.12 102.520
## - H.Length   1     2.617 165.68 103.679
## <none>                   163.06 104.278
## - Thickness  1     4.299 167.36 104.567
## - Should.Or  2     9.926 172.99 105.477
## - Haft.Or    4    23.077 186.14 107.925
## + Haft.Sh    4     6.104 156.96 108.920
## - Blade.Sh   3    23.640 186.70 110.191
## - Should.Sh  2    21.921 184.98 111.377
## - Length     1    84.675 247.74 139.083
## - Width      1    87.978 251.04 140.248
##
```

```
## Step:  AIC=99.45
## Weight ~ Length + Width + Thickness + J.Width + H.Length + Blade.Sh +
##     Base.Sh + Should.Sh + Should.Or + Haft.Or
##
##             Df Sum of Sq    RSS     AIC
## - Base.Sh    3     9.029 178.08  98.030
## - J.Width    1     1.185 170.23  98.066
## - H.Length   1     1.653 170.70  98.308
## <none>                   169.05  99.451
## - Should.Or  2    14.086 183.14 102.495
## - Haft.Or    4    25.030 194.08 103.602
## - Thickness  1    12.730 181.78 103.840
## + Name       4     5.988 163.06 104.278
## + Haft.Sh    4     5.889 163.16 104.331
## - Blade.Sh   3    25.379 194.43 105.760
## - Should.Sh  2    25.703 194.75 107.907
## - Width      1    88.850 257.90 134.621
## - Length     1    95.549 264.60 136.877
##
## Step:  AIC=98.03
## Weight ~ Length + Width + Thickness + J.Width + H.Length + Blade.Sh +
##     Should.Sh + Should.Or + Haft.Or
##
##             Df Sum of Sq    RSS     AIC
## - J.Width    1     0.416 178.50  96.236
## - H.Length   1     0.665 178.74  96.358
## <none>                   178.08  98.030
## + Base.Sh    3     9.029 169.05  99.451
## - Thickness  1     7.903 185.98  99.851
## - Haft.Or    4    21.610 199.69 100.109
## - Should.Or  2    13.832 191.91 100.613
## - Blade.Sh   3    19.115 197.19 101.003
## + Haft.Sh    4     7.025 171.05 102.488
## + Name       4     6.964 171.12 102.520
## - Should.Sh  2    21.721 199.80 104.158
## - Width      1    84.880 262.96 130.330
## - Length     1   119.152 297.23 141.111
##
## Step:  AIC=96.24
## Weight ~ Length + Width + Thickness + H.Length + Blade.Sh + Should.Sh +
##     Should.Or + Haft.Or
##
##             Df Sum of Sq    RSS     AIC
## - H.Length   1     0.687 179.18  94.574
## <none>                   178.50  96.236
## + J.Width    1     0.416 178.08  98.030
## + Base.Sh    3     8.261 170.23  98.066
## - Thickness  1     8.618 187.11  98.385
## - Haft.Or    4    22.138 200.63  98.524
## - Should.Or  2    14.062 192.56  98.909
## - Blade.Sh   3    18.838 197.33  99.065
## + Name       4     7.281 171.21 100.571
## + Haft.Sh    4     7.190 171.31 100.618
## - Should.Sh  2    21.489 199.98 102.239
## - Width      1   115.342 293.84 138.100
## - Length     1   120.251 298.75 139.559
##
```

```
## Step:  AIC=94.57
## Weight ~ Length + Width + Thickness + Blade.Sh + Should.Sh +
##     Should.Or + Haft.Or
##
##              Df Sum of Sq    RSS     AIC
## <none>                    179.18  94.574
## + H.Length   1     0.687 178.50  96.236
## + J.Width    1     0.437 178.74  96.358
## - Thickness  1     8.255 187.44  96.537
## + Base.Sh    3     7.457 171.72  96.833
## - Blade.Sh   3    18.661 197.84  97.292
## - Should.Or  2    14.636 193.82  97.483
## - Haft.Or    4    23.993 203.17  97.632
## + Haft.Sh    4     6.653 172.53  99.244
## + Name       4     6.632 172.55  99.254
## - Should.Sh  2    21.635 200.82 100.605
## - Width      1   114.966 294.15 136.194
## - Length     1   120.629 299.81 137.872
```

Finally, we can do an analysis of variance and generate a summary for the final linear model

```
anova(step)
```

```
## Analysis of Variance Table
##
## Response: Weight
##           Df  Sum Sq Mean Sq  F value    Pr(>F)
## Length     1 1221.70 1221.70 490.9140 < 2.2e-16 ***
## Width      1   90.55   90.55  36.3859  6.35e-08 ***
## Thickness  1    5.89    5.89   2.3653  0.128445
## Blade.Sh   3   13.37    4.46   1.7910  0.156536
## Should.Sh  3   30.91   10.30   4.1406  0.009148 **
## Should.Or  2   15.04    7.52   3.0211  0.054967 .
## Haft.Or    4   23.99    6.00   2.4103  0.056955 .
## Residuals 72  179.18    2.49
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(step)
```

```
##
## Call:
## lm(formula = Weight ~ Length + Width + Thickness + Blade.Sh +
##     Should.Sh + Should.Or + Haft.Or, data = DP1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.4915 -0.5139  0.0276  0.6229  5.9092
##
## Coefficients: (1 not defined because of singularities)
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -11.45876    2.08117  -5.506 5.36e-07 ***
## Length        0.16642    0.02390   6.962 1.30e-09 ***
## Width         0.40629    0.05978   6.797 2.61e-09 ***
## Thickness     0.28877    0.15855   1.821   0.0727 .
## Blade.ShI    -1.43197    1.03583  -1.382   0.1711
## Blade.ShR    -0.73815    0.99124  -0.745   0.4589
## Blade.ShS    -1.08909    0.41730  -2.610   0.0110 *
## Should.ShI   -0.54056    1.08301  -0.499   0.6192
## Should.ShS    0.81087    1.05512   0.769   0.4447
## Should.ShX    0.34707    1.68345   0.206   0.8372
## Should.OrH   -1.14175    1.30505  -0.875   0.3846
## Should.OrT    0.28296    1.20939   0.234   0.8157
## Should.OrX         NA         NA      NA       NA
## Haft.OrE      0.52678    0.73898   0.713   0.4782
## Haft.OrP      0.37495    0.71656   0.523   0.6024
## Haft.OrT     -0.25120    0.78423  -0.320   0.7497
## Haft.OrV     -2.12609    1.05096  -2.023   0.0468 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.578 on 72 degrees of freedom
## Multiple R-squared:  0.8866,Adjusted R-squared:  0.863
## F-statistic: 37.54 on 15 and 72 DF,  p-value: < 2.2e-16
```

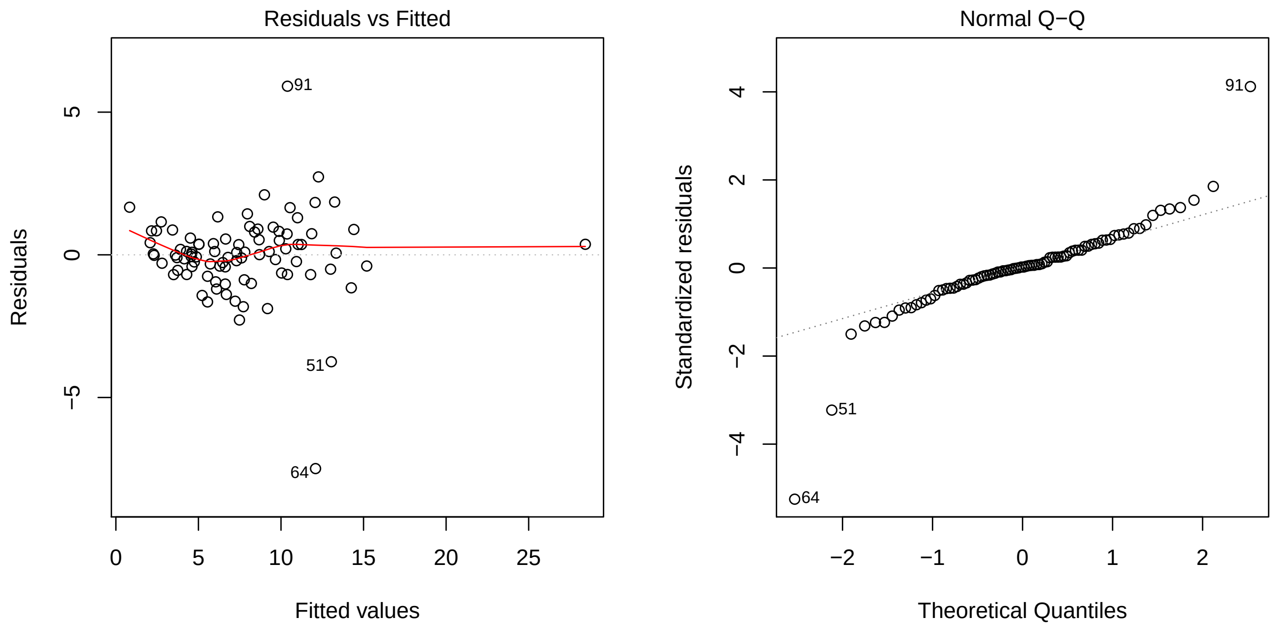
Here, we have a combination of various continuous and indicator variables with different levels. Therefore, our complicated multiple regression model is

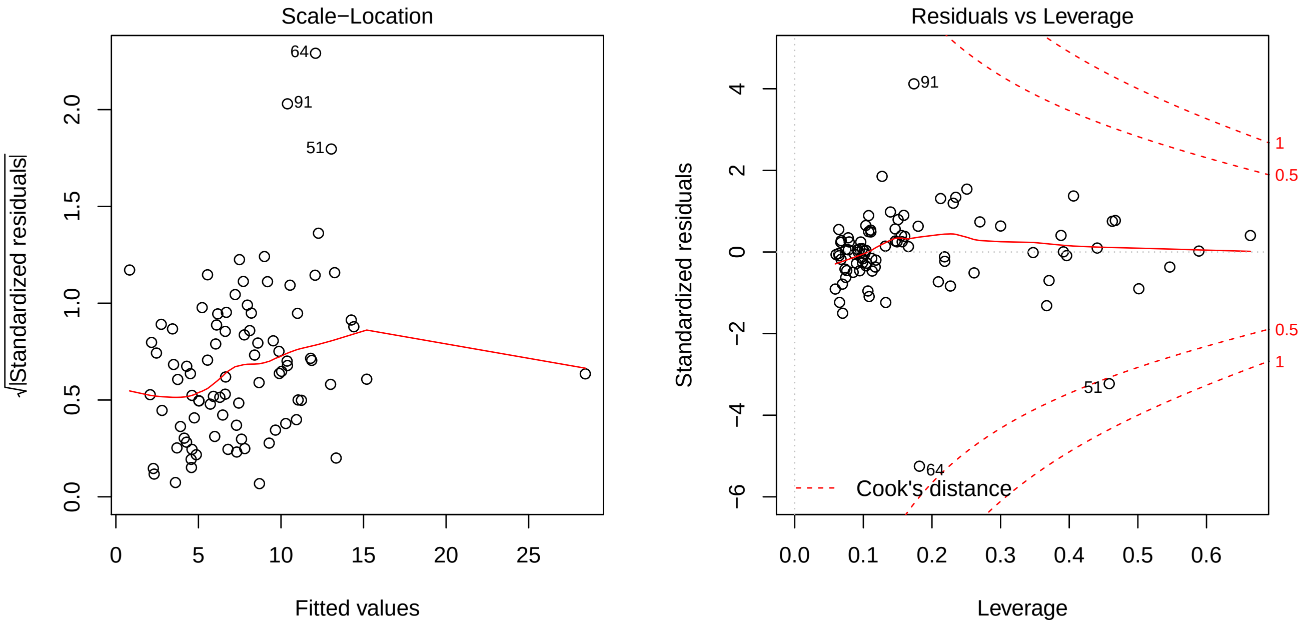
$Weight_i = -11.45876 + 0.16642 * Length_i + 0.40629 * Width_i + 0.28877 * Thickness_i - 1.43197 * Blade.ShI - 0.73815 * Blade.ShR - 1.08909 * Blade.ShS - 0.54056 * Should.ShI + 0.81087 * Should.ShS + 0.34707 * Should.ShX - 1.14175 * Should.OrH + 0.28296 * Should.ShT + 0.52678 * Haft.OrE + 0.37495 * Haft.OrP - 0.25120 * Haft.OrT - 2.12609 * Haft.OrV$

> Note: All the variables other than Length, Width, and Thickness are indicator variables. Which means that for each category, we use 1 if the value falls under that category, and 0 otherwise. For example, if a dart has a Straight Blade Shape, we use 1 for Blade.ShS and 0 for the other Blade.Sh variables. Also, if a dart has an Excurvate Blade Shape (not in this regression model), we use 0 for all the Blade.Sh variables. The same applies to all other indicator variables.

f) To check how good our fit is, we can generate diagnostic plots for our regression model.

```
par(mfrow=c(1,2))
par(mar = c(5, 4.5, 2, 2))
plot(step)
```





From the Residuals vs Fitted plot, we see that there are no non-linear patterns and the red line is close to the dashed line. Also, we have 3 outliers (51,64,91), but their residuals are still low (<±10). Therefore, we can assume a linear relationship between the predictors and the response variables. [6, 7]

From the Normal Q-Q plot, we see that the data follows a normal distribution and most of the residuals points lie on or near the straight line (other than the mentioned outliers). Therefore, our residuals and errors are Gaussian, and so our confidence intervals and significance tests are valid. [6, 8]

From the Scale-Location plot, we see a relatively horizontal line (factoring in the usual outliers) and that the residuals are spread equally randomly along the range of predictors. Therefore, we can assume constant variances in the residual errors, or homoscedasity. [6, 9]

From the Residuals vs Leverage plot, we see that almost all of the data points are close to the regression line. This mean that they have a low leverage value, or are within 2 standard errors away from the regression line (rule of thumb is 3 standard errors away), or both. Therefore, they have a low Cook's Distance, which takes into account the leverage and residual size. The leverage of a point tells us how influential the point is against the regression line. Usually, problematic points are found in the upper right or lower right corners, but we do not have any. Even the usual outliers are within the dashed lines. Therefore, the outliers are not influential to the regression results, and we do not need to remove them. [6, 10]

From these diagnostic plots, we have evidence of a linear relationship, normality of errors, homoscedasity, and non-influential outliers. Therefore, we can say that assumptions are valid and our regression model is a reasonably accurate one.

g) Firstly, let us generate the summary of our final model again.

```
summary(step)

##
## Call:
## lm(formula = Weight ~ Length + Width + Thickness + Blade.Sh +
##     Should.Sh + Should.Or + Haft.Or, data = DP1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.4915 -0.5139  0.0276  0.6229  5.9092
##
```

```
## Coefficients: (1 not defined because of singularities)
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -11.45876    2.08117  -5.506 5.36e-07 ***
## Length        0.16642    0.02390   6.962 1.30e-09 ***
## Width         0.40629    0.05978   6.797 2.61e-09 ***
## Thickness     0.28877    0.15855   1.821   0.0727 .
## Blade.ShI    -1.43197    1.03583  -1.382   0.1711
## Blade.ShR    -0.73815    0.99124  -0.745   0.4589
## Blade.ShS    -1.08909    0.41730  -2.610   0.0110 *
## Should.ShI   -0.54056    1.08301  -0.499   0.6192
## Should.ShS    0.81087    1.05512   0.769   0.4447
## Should.ShX    0.34707    1.68345   0.206   0.8372
## Should.OrH   -1.14175    1.30505  -0.875   0.3846
## Should.OrT    0.28296    1.20939   0.234   0.8157
## Should.OrX         NA         NA      NA       NA
## Haft.OrE      0.52678    0.73898   0.713   0.4782
## Haft.OrP      0.37495    0.71656   0.523   0.6024
## Haft.OrT     -0.25120    0.78423  -0.320   0.7497
## Haft.OrV     -2.12609    1.05096  -2.023   0.0468 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.578 on 72 degrees of freedom
## Multiple R-squared:  0.8866,Adjusted R-squared:  0.863
## F-statistic: 37.54 on 15 and 72 DF,  p-value: < 2.2e-16
```

From the coefficients produced, we can see that, on average, a 10 mm increase in length, width, and thickness each leads to a 8.6148 g increase in weight. The significance test of β_1, β_2, and β_3 is significant at 0.1% significance level, which means at all the usual levels, we would reject the hypothesis of absence of a linear relationship. The adjusted R-squared is 86.3%, which means that most of the variation in Weight is explained by the explanatory variables.

h) Firstly, we have to calculate the predicted value using our regression model. We substitute the given values for the continuous predictor variables, 1 for the applicable indicator variable, and 0 for the non-applicable indicator variables.

```
options(warn=-1)
predict(step,data.frame(Length=70,Width=60,Thickness=50,Blade.Sh='R',Should.Sh='S',
                        Should.Or='B',Haft.Or='P'),interval='prediction')
```

```
##       fit      lwr      upr
## 1 39.4546 25.52641 53.38279
```

We get 39.4546 as the predicted value for the weight, and our 95% prediction intervals are (25.52641, 53.38279). This means that for the given values of the predictor variables, based on our regression model, there is a 95% chance that the corresponding weight is between 25.52641 and 53.38279.

However, we should be cautious about our prediction. The given width and thickness for the predicted data point is outside the range of Width and Thickness.

```
range(Width)
```

```
## [1] 14.5 49.3
```

```
range(Thickness)
```

```
## [1]  4.0 10.7
```

Regression predictions are usually valid only for the range of data used to estimate the model. The relationship between the predictor variables and the response variable can change outside of that range. In other words, we don't know whether the shape of the regression line changes. If it does, our prediction will be invalid. [11]
Therefore, this prediction should be taken with a grain of salt.

References

[1] *Shapiro-Wilk Test for Normality in R.* URL: https://www.r-bloggers.com/2019/08/shapiro-wilk-test-for-normality-in-r/. (accessed: 07.12.2020).

[2] *Normality Test in R.* URL: http://www.sthda.com/english/wiki/normality-test-in-r. (accessed: 07.12.2020).

[3] *Fligner-Killeen Test in R Programming.* URL: https://www.geeksforgeeks.org/fligner-killeen-test-in-r-programming/. (accessed: 07.12.2020).

[4] *What is stepAIC in R?* URL: https://medium.com/@ashutosh.optimistic/what-is-stepaic-in-r-a65b71c9eeba. (accessed: 12.12.2020).

[5] *Choosing variables to include in a multiple linear regression model.* URL: https://stats.stackexchange.com/questions/21265/choosing-variables-to-include-in-a-multiple-linear-regression-model. (accessed: 12.12.2020).

[6] *Understanding Diagnostic Plots for Linear Regression Analysis.* URL: https://data.library.virginia.edu/diagnostic-plots/. (accessed: 12.12.2020).

[7] *Linear Regression Plots: Fitted vs Residuals.* URL: https://boostedml.com/2019/03/linear-regression-plots-fitted-vs-residuals.html. (accessed: 12.12.2020).

[8] *The QQ Plot in Linear Regression.* URL: https://boostedml.com/2019/03/linear-regression-plots-how-to-read-a-qq-plot.html. (accessed: 12.12.2020).

[9] *The Scale Location Plot: Interpretation in R.* URL: https://boostedml.com/2019/03/linear-regression-plots-scale-location-plot.html. (accessed: 12.12.2020).

[10] *Linear Regression Plots: Residuals vs Leverage.* URL: https://boostedml.com/2019/03/linear-regression-plots-residuals-vs-leverage.html. (accessed: 12.12.2020).

[11] *Making Predictions with Regression Analysis.* URL: https://statisticsbyjim.com/regression/predictions-regression/. (accessed: 13.12.2020).