

7CCMMS61T Statistics for Data Analysis Coursework

The deadline to submit your assignment on the KEATS page of the module is on **16th December 2019 at 5pm.**

Please submit a report which answers the questions below, including an appropriate description of the methodology you used (with code snippets when appropriate) and the graphics you produced.

Question 1

The file protein.csv contains data from several European countries in the 1980s on consumption of different categories of food.

First of all, read "protein.csv" file, set protein as table name

```
> protein <- read.csv('/Users/B/Desktop/textbook/2 Stat data/Coursework/protein.csv',header=TRUE)
> head(protein)
  Country RedMeat WhiteMeat Eggs Milk Fish Cereals Starch Nuts Fr.Veg
1  Albania   10.1     1.4  0.5  8.9  0.2   42.3    0.6  5.5   1.7
2  Austria    8.9    14.0  4.3 19.9  2.1   28.0    3.6  1.3   4.3
3  Belgium   13.5     9.3  4.1 17.5  4.5   26.6    5.7  2.1   4.0
4  Bulgaria    7.8     6.0  1.6  8.3  1.2   56.7    1.1  3.7   4.2
5 Czechoslovakia 9.7    11.4  2.8 12.5  2.0   34.3    5.0  1.1   4.0
6  Denmark   10.6    10.8  3.7 25.0  9.9   21.9    4.8  0.7   2.4
> attach(protein)
```

Exploratory data Analysis:

- (a) For each variable, calculate appropriate summary statistics to show the level and spread of the data (one statistic for each is enough).

By using levels() function, to see different kinds of Country

```
> #(a) level
> levels(Country)
 [1] "Albania"      "Austria"      "Belgium"      "Bulgaria"
 [5] "Czechoslovakia" "Denmark"      "E Germany"    "Finland"
 [9] "France"       "Greece"       "Hungary"      "Ireland"
[13] "Italy"        "Netherlands"  "Norway"       "Poland"
[17] "Portugal"     "Romania"      "Spain"        "Sweden"
[21] "Switzerland"  "UK"           "USSR"         "W Germany"
[25] "Yugoslavia"
```

By using summary() function, show statistical data of each column which only have numeric data

```
> #a)summary
> summary(protein[2:10])
```

RedMeat	WhiteMeat	Eggs	Milk
Min. : 4.400	Min. : 1.400	Min. :0.500	Min. : 4.90
1st Qu.: 7.800	1st Qu.: 4.900	1st Qu.:2.700	1st Qu.:11.10
Median : 9.500	Median : 7.800	Median :2.900	Median :17.60
Mean : 9.828	Mean : 7.896	Mean :2.936	Mean :17.11
3rd Qu.:10.600	3rd Qu.:10.800	3rd Qu.:3.700	3rd Qu.:23.30
Max. :18.000	Max. :14.000	Max. :4.700	Max. :33.70

Fish	Cereals	Starch	Nuts
Min. : 0.200	Min. :18.60	Min. :0.600	Min. :0.700
1st Qu.: 2.100	1st Qu.:24.30	1st Qu.:3.100	1st Qu.:1.500
Median : 3.400	Median :28.00	Median :4.700	Median :2.400
Mean : 4.284	Mean :32.25	Mean :4.276	Mean :3.072
3rd Qu.: 5.800	3rd Qu.:40.10	3rd Qu.:5.700	3rd Qu.:4.700
Max. :14.200	Max. :56.70	Max. :6.500	Max. :7.800

Fr.Veg
Min. :1.400
1st Qu.:2.900
Median :3.800
Mean :4.136
3rd Qu.:4.900
Max. :7.900

Choose one statistic for each variable

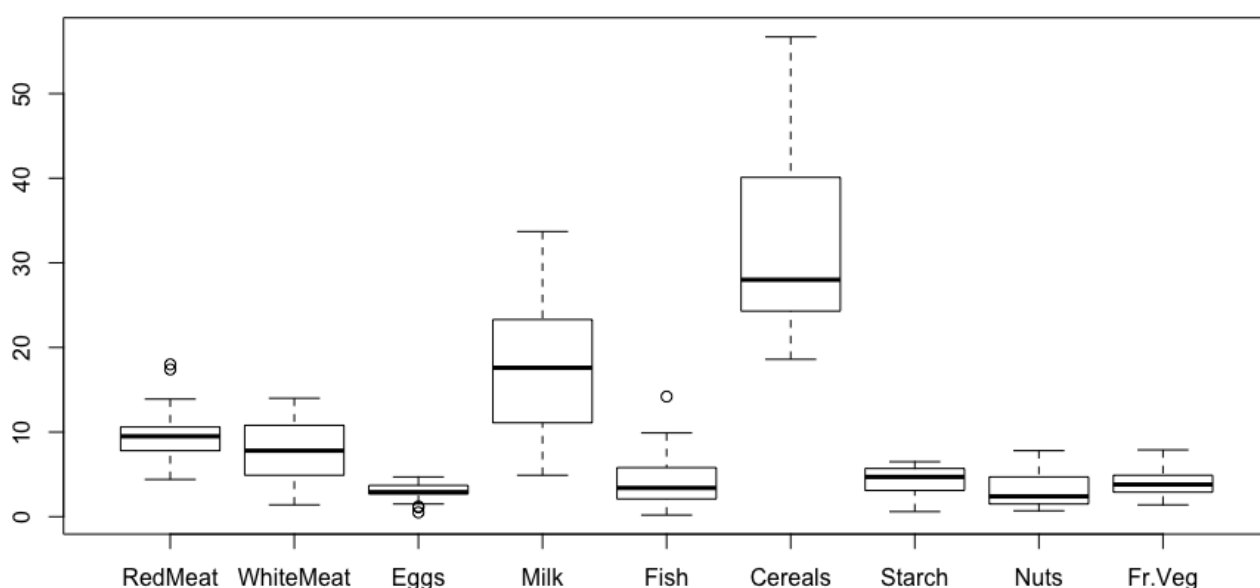
```
> max(RedMeat)
[1] 18
> min(WhiteMeat)
[1] 1.4
> median(Eggs)
[1] 2.9
> quantile(Milk, 0.25)
25%
11.1
> quantile(Fish, 0.75)
75%
5.8
> mean(Cereals)
[1] 32.248
> max(Starch)
[1] 6.5
> min(Nuts)
[1] 0.7
> median(Fr.Veg)
[1] 3.8
```

(b) For each variable, plot the data in a suitable way to illustrate the level and the spread.

The reason that I choose boxplot rather than others is because boxplot can show much more information at the same time, also it can illustrate all variables in one figure.

If I choose histogram, then I will generate nine plots here, and I will not receive quartile data.

```
> #(b)plot  
> boxplot(protein[2:10])
```



(c) Calculate a summary statistic to show the association of the consumption of fruit and vegetables with each of the other food categories.

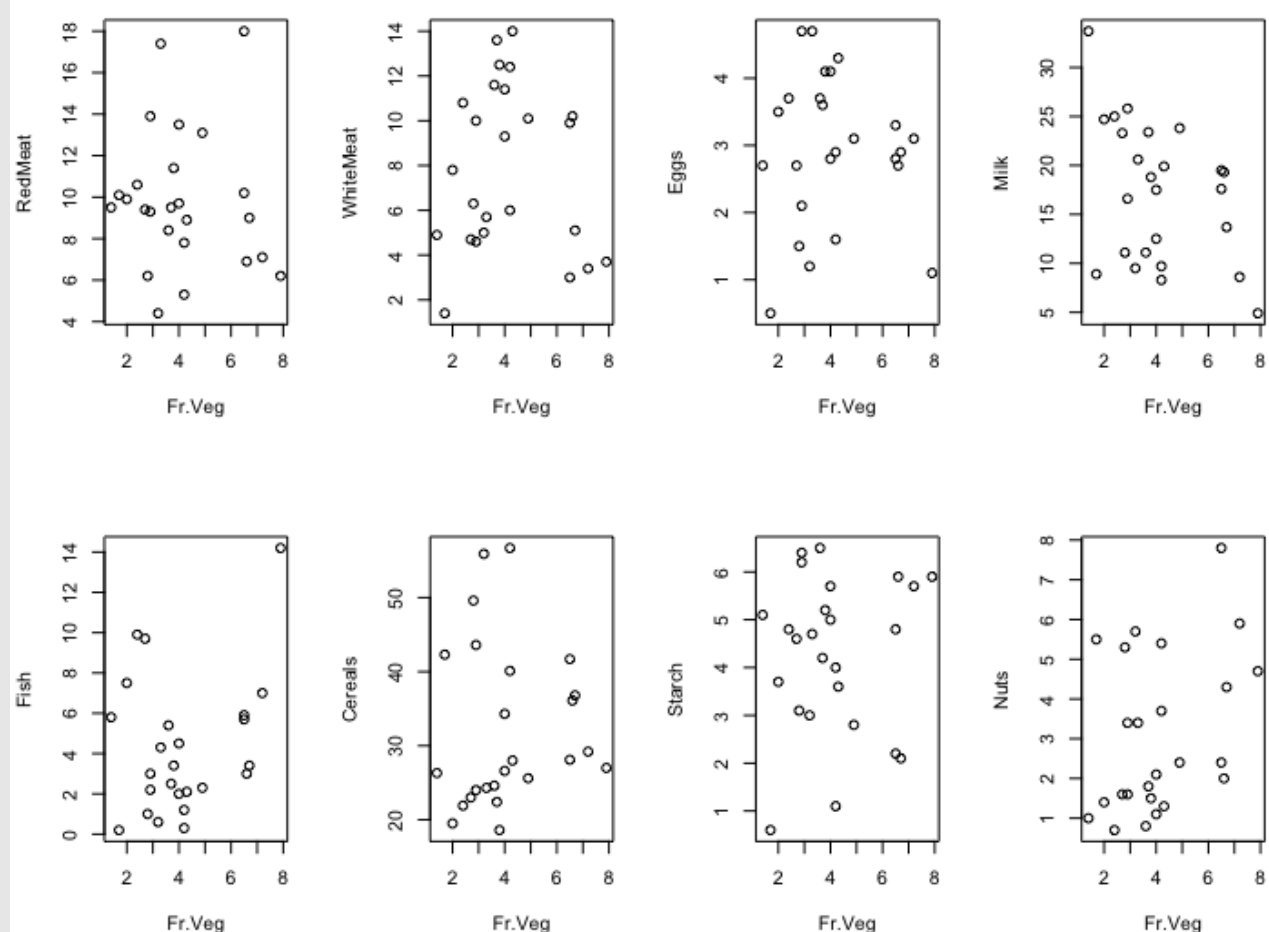
Use `cor()`, to show the correlation coefficient between Fr.Veg (`protein[,10]`) & each other variables (`protein[2:9]`)

```
> #(c)association  
> cor(protein[,10], y= protein[,2:9])  
      RedMeat WhiteMeat Eggs Milk Fish Cereals Starch Nuts  
[1,] -0.07422123 -0.0613167 -0.04551755 -0.4083641 0.2661387 0.04654808 0.08440956 0.3749697
```

- (d) Show a plot illustrating the association of the consumption of fruit and vegetables with each of the other food categories.

Use scatter plot to demonstrate relationship between Fr.Veg & each variable. By these plots, there are less apparent correlation between them.

```
> #(d)plot of association  
> par(mfrow=c(2,4))  
> for (i in 2:9){  
+   plot(Fr.Veg,protein[,i],xlab = 'Fr.Veg',ylab = names(protein)[i])  
+ }
```



Inference:

- (e) Provide confidence intervals at level 95% for the mean consumption of each category of food.

Count confidence interval for each column (numbers of column: dim(protein)[2])

Because first column is level, could not count confidence level

```
> #(e)confidenence level
> for (i in 2:dim(protein)[2]){
+   print(paste("95% Confidence interval for", names(protein)[i]))
+   print(t.test(protein[,i],conf.level = 0.95)$conf.int)
+ }
[1] "95% Confidence interval for RedMeat"
[1] 8.446394 11.209606
attr("conf.level")
[1] 0.95
[1] "95% Confidence interval for WhiteMeat"
[1] 6.371158 9.420842
attr("conf.level")
[1] 0.95
[1] "95% Confidence interval for Eggs"
[1] 2.474671 3.397329
attr("conf.level")
[1] 0.95
[1] "95% Confidence interval for Milk"
[1] 14.17903 20.04497
attr("conf.level")
[1] 0.95
[1] "95% Confidence interval for Fish"
[1] 2.879503 5.688497
attr("conf.level")
[1] 0.95
[1] "95% Confidence interval for Cereals"
[1] 27.71783 36.77817
attr("conf.level")
[1] 0.95
[1] "95% Confidence interval for Starch"
[1] 3.601483 4.950517
attr("conf.level")
[1] 0.95
[1] "95% Confidence interval for Nuts"
[1] 2.252351 3.891649
attr("conf.level")
[1] 0.95
[1] "95% Confidence interval for Fr.Veg"
[1] 3.391385 4.880615
attr("conf.level")
[1] 0.95
```

- (f) Carry out the appropriate test of hypothesis to check if the average consumption of starch is larger than the average consumption of nuts. Also check if the assumptions behind this test are reasonable in this case.

Because length of protein data is 25, $25 < 30$ is small sample, then use t test.

```
> #(f)t test
> length(Country) #25<30, use t test
[1] 25
> t.test(Starch, alternative="greater",mu = mean(Nuts), conf.level=0.95,var.equal=TRUE)

      One Sample t-test

data:  Starch
t = 3.684, df = 24, p-value = 0.000583
alternative hypothesis: true mean is greater than 3.072
95 percent confidence interval:
 3.716855      Inf
sample estimates:
mean of x
 4.276
```

Question 2

The file DartPoints.csv contains data on 91 Archaic dart points recovered during surface surveys at Fort Hood, Texas. These data have been extracted from the R package archdata. The dataset contains the following variables:

- **Name.** Dart point type: Darl, Ensor, Pedernales, Travis, Wells
- **Length.** Maximum Length (mm)
- **Width.** Maximum Width (mm)
- **Thickness.** Maximum Thickness (mm)
- **B.Width.** Basal width (mm)
- **J.Width.** Juncture width (mm)
- **H.Length.** Haft element length (mm)
- **Weight.** Weight (gm)
- **Blade.Sh** Blade shape: E - Excurvate, I - Incurvate, R - Recurvate, S - Straight.
- **Base.Sh** Base shape: E - Excurvate, I - Incurvate, R - Recurvate, S - Straight.
- **Should.Sh** Shoulder shape: E - Excurvate, I - Incurvate, S - Straight, X - None.
- **Should.Or** Shoulder orientation: B - Barbed, H - Horizontal, T - Tapered, X - None.
- **Haft.Sh** Shape lateral haft element A - Angular, E - Excurvate, I - Incurvate, R - Recurvate, S - Straight.
- **Haft.Or** Orientation lateral haft element: C - Concave, E - Expanding, P - Parallel, T - Contracting, V - Convex.

First of all, read "DartPoints.csv" file, set DP as table name

```
> DP <- read.csv('/Users/B/Desktop/textbook/2 Stat data/Coursework/DartPoints.csv', header = TRUE)
> head(DP)
  X Name Length Width Thickness B.Width J.Width H.Length Weight Blade.Sh Base.Sh Should.Sh Should.Or Haft.Sh Haft.Or
1 1 Darl  42.8  15.8    5.8    11.3   10.6    11.6    3.6      S      I      S      T      S      E
2 2 Darl  40.5  17.4    5.8     NA    13.7    12.9    4.5      S      I      S      T      S      E
3 3 Darl  37.5  16.3    6.1    12.1   11.3     8.2    3.6      S      I      S      T      S      E
4 4 Darl  40.3  16.1    6.3    13.5   11.7     8.3    4.0      S      I      S      T      S      E
5 5 Darl  30.6  17.1    4.0    12.6   11.2     8.9    2.3      S      I      S      T      S      E
6 6 Darl  41.8  16.8    4.1    12.7   11.5    11.0    3.0      S      E      I      T      I      C
> attach(DP)
```

Exploratory data Analysis:

(a) State the scaling of each of the above variables.

By using `scale()` to scale DP, this function will calculate mean and standard deviation of each column, then return each data by $(x - \text{mean}) / \text{sd}$

This function just can use on double type data (DP[3:9])

```
> #(a)scaling
> DP_scale <- scale(DP[3:9])
```

DP_scale looks like this picture

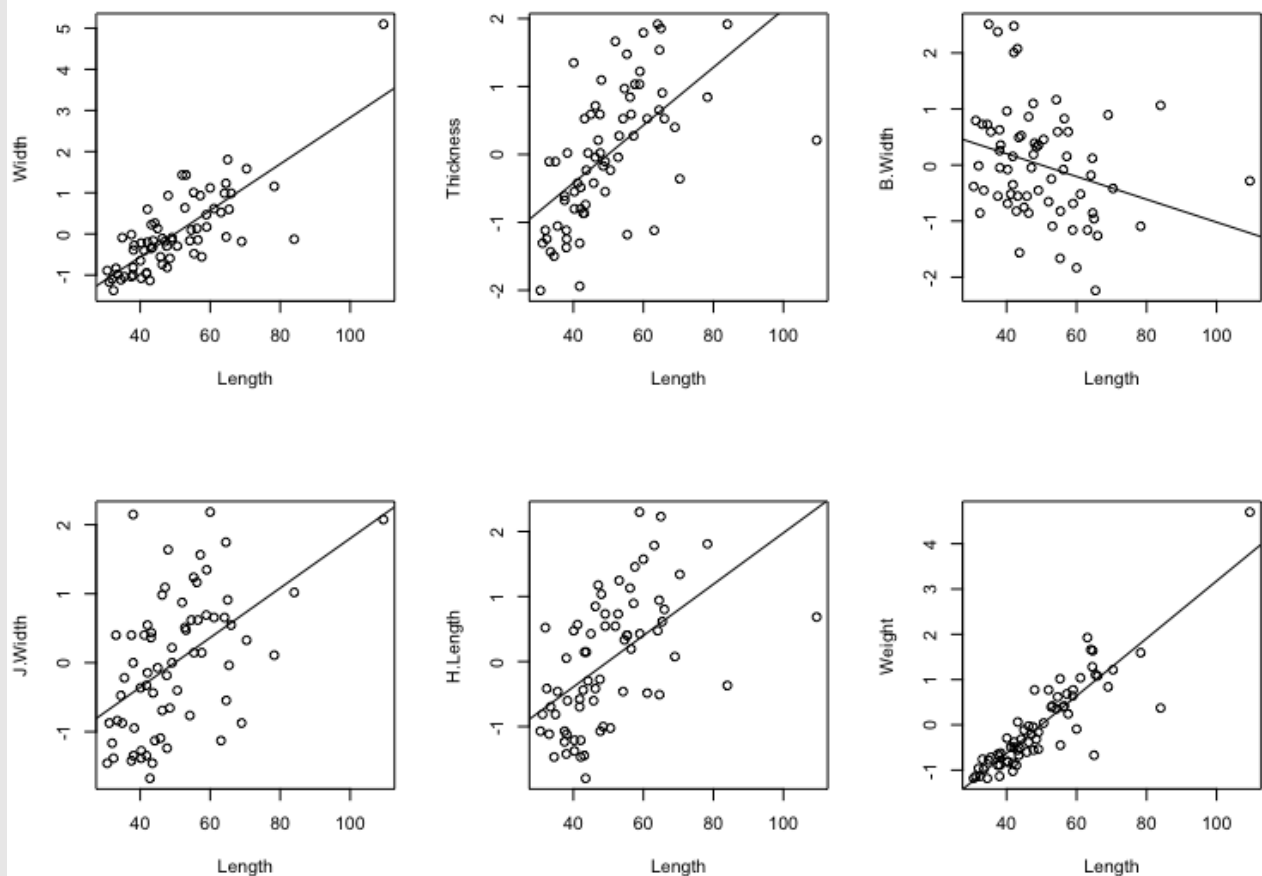
```
> head(DP_scale)
      Length      Width  Thickness      B.Width      J.Width      H.Length      Weight
[1,] -0.5127725 -1.2173225 -0.9608867 -0.83146927 -1.7581608 -0.4512425 -0.9609634
[2,] -0.6933602 -0.9070246 -0.9608867          NA -0.6237323 -0.1271260 -0.7470386
[3,] -0.9289094 -1.1203544 -0.7649777 -0.56004938 -1.5019995 -1.2989319 -0.9609634
[4,] -0.7090635 -1.1591416 -0.6343718 -0.08506457 -1.3556217 -1.2739998 -0.8658857
[5,] -1.4706726 -0.9652055 -2.1363403 -0.39041195 -1.5385940 -1.1244076 -1.2699657
[6,] -0.5912889 -1.0233863 -2.0710373 -0.35648446 -1.4288106 -0.6008348 -1.1035798
```

- (b) First consider the variable Length. Represent graphically the relationship between Length and the other variables and describe any interesting patterns.

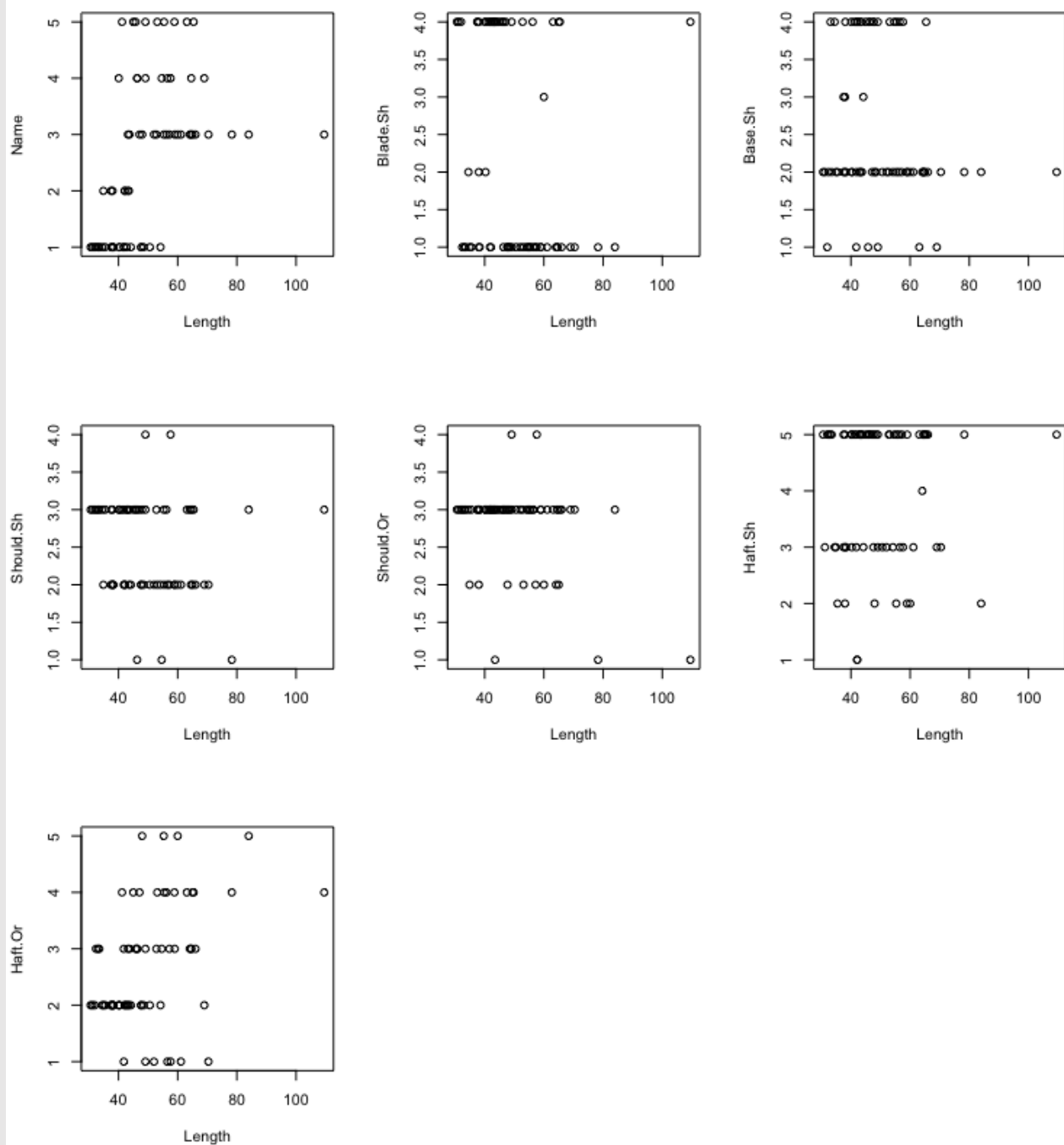
For 4 to 9 column, data type is double, so can using the data after scaling

For other columns, because do not have scaling data (cannot scaling), use original data

```
> #(b)plot relationship between Length & each other variables
> par(mfrow=c(2,3))
> for (i in 4:9){
+   plot(Length, DP_scale[,i-2],xlab = 'Length', ylab = names(DP)[i])
+   abline(lm(DP_scale[,i-2]~Length))
+ }
>
> plot(Length, DP[,2],xlab = 'Length', ylab = 'Name')
> for (i in 10:15){
+   plot(Length, DP[,i],xlab = 'Length', ylab = names(DP)[i])
+ }
```



Apparently, Length has a strong relationship with Width and Weight, also it looks like there is a slightly strong relationship between Length and Thickness



Because these variables are class variables, they only have 4 to 5 levels. The plot of them would display in several line

- (c) For the variables which seems to be associated with Length calculate a summary statistic which will describe the strength of the association, if possible.

According to previous plots, I choose Weight, Width and Thickness which have stronger relationship with Length, so I choose them and calculate the correlation (Width=DP[4], Thickness=DP[5], Weight=DP[9])

```
> #(c)variables have strong relationship with Length
> asso_item <- c(4,5,9)
> for (i in asso_item){
+   cor1 <- cor(Length, DP_scale[,i-2],use = "complete.obs")
+   print(paste("Correlation between Length &", names(DP)[i]))
+   print(cor1)
+ }
[1] "Correlation between Length & Width"
[1] 0.7845433
[1] "Correlation between Length & Thickness"
[1] 0.5914023
[1] "Correlation between Length & Weight"
[1] 0.8819988
```

- (d) Compute and represent graphically the relative frequency distribution of Weight conditionally to the various types of blade shape.

Classify Weight into four levels by 1st quartile, mean and 3rd quartile, therefore it would become four levels – “0-25%”, “25-50%”, “50-75%”, “75-100%” (defined as weight_class)

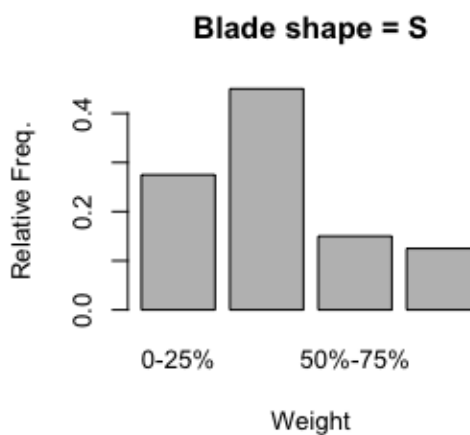
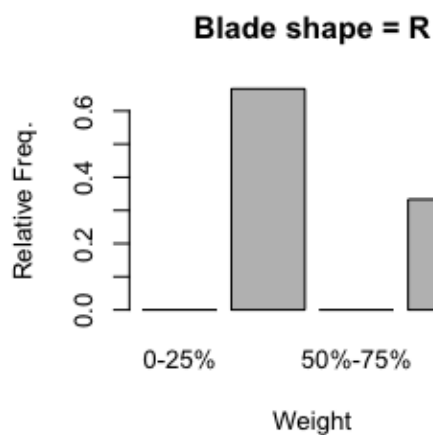
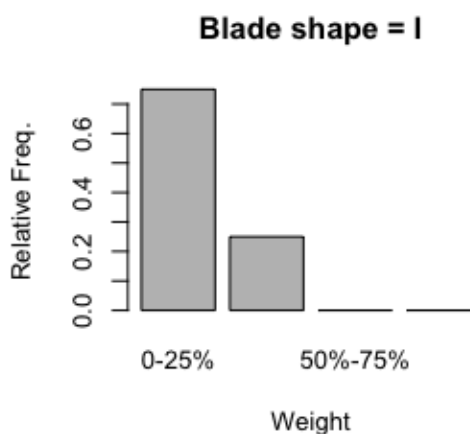
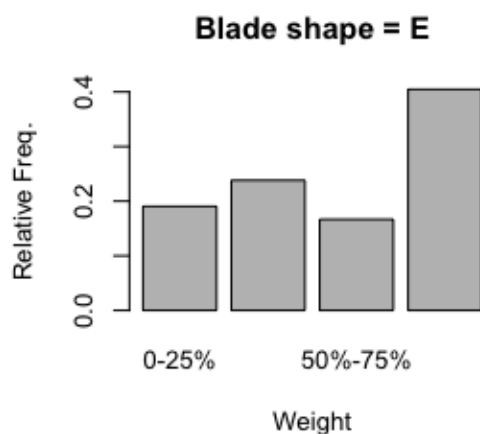
Then, make weight_class & blade_sh (is factor of Blade.Sh) as a table, calculate conditionally relative frequency under four condition (blade.sh = “E”, “I”, “R”, “S”)

```
> #(d)frequency of distribution of Weight
> weight_class<-rep(NA,dim(DP)[1])
> weight_class[which(Weight<=quantile(Weight,0.25))]<-"0-25%"
> weight_class[which(Weight>quantile(Weight,0.25) & Weight<=mean(Weight))]<-"25%-50%"
> weight_class[which(Weight>mean(Weight) & Weight<=quantile(Weight,0.75))]<-"50%-75%"
> weight_class[which(Weight>quantile(Weight,0.75))]<-"75%+"
> weight_class<-factor(weight_class,levels=c("0-25%", "25%-50%", "50%-75%", "75%+"))
> blade_sh <-factor(Blade.Sh,levels=c("E", "I", "R", "S"))
>
> par(mfrow=c(2,2))
>
> #for blade.Sh=E
> table(blade_sh,weight_class)[1,]/length(which(blade_sh=='E'))
   0-25%   25%-50%   50%-75%   75%+
0.1904762 0.2380952 0.1666667 0.4047619
> barplot(table(blade_sh,weight_class)[1,]/length(which(blade_sh=='E')),main="Blade shape = E",
,xlab = "Weight", ylab = "Relative Freq.")
```

```

> #for blade.Sh=I
> table(blade_sh,weight_class)[2,]/length(which(blade_sh=='I'))
  0-25% 25%-50% 50%-75% 75%+
    0.75    0.25    0.00    0.00
> barplot(table(blade_sh,weight_class)[2,]/length(which(blade_sh=='I')),main="Blade shape = I",
,xlab = "Weight", ylab = "Relative Freq.")
>
> #for blade.Sh=R
> table(blade_sh,weight_class)[3,]/length(which(blade_sh=='R'))
  0-25% 25%-50% 50%-75% 75%+
0.0000000 0.6666667 0.0000000 0.3333333
> barplot(table(blade_sh,weight_class)[3,]/length(which(blade_sh=='R')),main="Blade shape = R",
,xlab = "Weight", ylab = "Relative Freq.")
>
> #for blade.Sh=S
> table(blade_sh,weight_class)[4,]/length(which(blade_sh=='S'))
  0-25% 25%-50% 50%-75% 75%+
  0.275  0.450  0.150  0.125
> barplot(table(blade_sh,weight_class)[4,]/length(which(blade_sh=='S')),main="Blade shape = S",
,xlab = "Weight", ylab = "Relative Freq.")

```



Multiple linear regression:

- (e) Select an appropriate multiple regression model, which can be used to predict the weight of the dart, using some or all (after appropriate selection) of the variables listed above as explanatory variables (with the exclusion of the weight itself, of course).

(Click [here](#) to Skip model finding process, to final model build process)

First of all, make class variables become factor

```
> #(e)find multiple regression model
> Name<- as.factor(Name)
> Blade.Sh<- as.factor(Blade.Sh)
> Base.Sh<-as.factor(Base.Sh)
> Should.Sh<- as.factor(Should.Sh)
> Should.Or<- as.factor(Should.Or)
> Haft.Sh<-as.factor(Haft.Sh)
> Haft.Or<-as.factor(Haft.Or)
```

Then, make each level in factors become double variables, in order to make after work easier

```
> Name.D<-as.double(Name=="Darl")
> Name.E<-as.double(Name=="Ensor")
> Name.P<-as.double(Name=="Pedernales")
> Name.T<-as.double(Name=="Travis")
> Name.W<-as.double(Name=="Wells")
> Blade.Sh.E<-as.double(Blade.Sh=="E")
> Blade.Sh.I<-as.double(Blade.Sh=="I")
> Blade.Sh.R<-as.double(Blade.Sh=="R")
> Blade.Sh.S<-as.double(Blade.Sh=="S")
> Base.Sh.E<-as.double(Base.Sh=="E")
> Base.Sh.I<-as.double(Base.Sh=="I")
> Base.Sh.R<-as.double(Base.Sh=="R")
> Base.Sh.S<-as.double(Base.Sh=="S")
> Should.Sh.E<-as.double(Should.Sh=="E")
> Should.Sh.I<-as.double(Should.Sh=="I")
> Should.Sh.S<-as.double(Should.Sh=="S")
> Should.Sh.X<-as.double(Should.Sh=="X")
> Should.Or.B<-as.double(Should.Or=="B")
> Should.Or.H<-as.double(Should.Or=="H")
> Should.Or.T<-as.double(Should.Or=="T")
> Should.Or.X<-as.double(Should.Or=="X")
> Haft.Sh.A<-as.double(Haft.Sh=="A")
> Haft.Sh.E<-as.double(Haft.Sh=="E")
> Haft.Sh.I<-as.double(Haft.Sh=="I")
> Haft.Sh.R<-as.double(Haft.Sh=="R")
> Haft.Sh.S<-as.double(Haft.Sh=="S")
> Haft.Or.C<-as.double(Haft.Or=="C")
> Haft.Or.E<-as.double(Haft.Or=="E")
> Haft.Or.P<-as.double(Haft.Or=="P")
> Haft.Or.T<-as.double(Haft.Or=="T")
> Haft.Or.V<-as.double(Haft.Or=="V")
```

lm0 Next, put all variables in model, find the criteria has high correlation

```
> lm0<-lm(Weight~Name+Length+Width+Thickness+B.Width+J.Width+H.Length+Blade.Sh+Base.Sh+Should.
Sh+Should.Or+Haft.Sh+Haft.Or)
> summary(lm0)
```

Call:

```
lm(formula = Weight ~ Name + Length + Width + Thickness + B.Width +
    J.Width + H.Length + Blade.Sh + Base.Sh + Should.Sh + Should.Or +
    Haft.Sh + Haft.Or)
```

Residuals:

Min	1Q	Median	3Q	Max
-5.1354	-0.6533	0.0000	0.6880	4.8577

Coefficients: (1 not defined because of singularities)

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-8.59096	4.51732	-1.902	0.064801 .
NameEnsor	1.10030	1.40203	0.785	0.437441
NamePedernales	-0.01081	1.65548	-0.007	0.994823
NameTravis	1.60085	1.40155	1.142	0.260518
NameWells	0.62895	1.48708	0.423	0.674719
Length	0.15286	0.03702	4.130	0.000192 ***
Width	0.39290	0.10589	3.711	0.000660 ***
Thickness	0.05258	0.26938	0.195	0.846273
B.Width	-0.22788	0.17921	-1.272	0.211254
J.Width	0.21336	0.23678	0.901	0.373201
H.Length	0.03786	0.13207	0.287	0.775955
Blade.ShI	-3.59438	1.71322	-2.098	0.042606 *
Blade.ShR	-1.98878	2.67788	-0.743	0.462248
Blade.ShS	-2.37649	0.85218	-2.789	0.008223 **
Base.ShI	-1.11966	1.24500	-0.899	0.374145
Base.ShR	0.01721	1.82594	0.009	0.992528
Base.ShS	-1.05221	1.15291	-0.913	0.367178
Should.ShI	-0.27445	1.48119	-0.185	0.853985
Should.ShS	1.99638	1.43643	1.390	0.172673
Should.ShX	-2.26323	2.82070	-0.802	0.427332
Should.OrH	-2.73218	1.84938	-1.477	0.147823
Should.OrT	-0.67693	1.85121	-0.366	0.716640
Should.OrX	NA	NA	NA	NA
Haft.ShE	-0.94543	2.54450	-0.372	0.712285
Haft.ShI	0.77516	1.82496	0.425	0.673410
Haft.ShR	3.44126	3.32530	1.035	0.307270
Haft.ShS	0.25304	2.01726	0.125	0.900838
Haft.OrE	1.55146	1.37741	1.126	0.267077
Haft.OrP	-0.11168	1.30371	-0.086	0.932184
Haft.OrT	-0.45918	1.69309	-0.271	0.787696
Haft.OrV	-1.19381	2.14778	-0.556	0.581582

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.769 on 38 degrees of freedom

(23 observations deleted due to missingness)

Multiple R-squared: 0.9125, Adjusted R-squared: 0.8457

F-statistic: 13.67 on 29 and 38 DF, p-value: 1.313e-12

lm0 Select Width & Length, and put them into remaining models

lm1 Build model which compare 2 factors (Name & Blade.Sh)

```
> lm1<-lm(Weight~Length+Width+Name*Blade.Sh,data=DP)
> summary(lm1)

Call:
lm(formula = Weight ~ Length + Width + Name * Blade.Sh, data = DP)

Residuals:
    Min       1Q   Median       3Q      Max
-8.6400 -0.5572  0.0000  0.7000  5.0011

Coefficients: (6 not defined because of singularities)
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   -8.14964    1.05637   -7.715 4.83e-11 ***
Length         0.15747    0.02637    5.972 7.84e-08 ***
Width         0.38006    0.06989    5.438 6.83e-07 ***
NameEnsor     -1.74960    1.38235   -1.266  0.210
NamePedernales 0.28080    0.79010    0.355  0.723
NameTravis     1.10867    0.85403    1.298  0.198
NameWells     -1.39030    1.15713   -1.202  0.233
Blade.ShI     -1.59506    1.31750   -1.211  0.230
Blade.ShR     -1.41517    1.10312   -1.283  0.204
Blade.ShS     -0.77934    0.71500   -1.090  0.279
NameEnsor:Blade.ShI 1.35917    2.55393    0.532  0.596
NamePedernales:Blade.ShI 0.48062    2.23922    0.215  0.831
NameTravis:Blade.ShI      NA         NA         NA      NA
NameWells:Blade.ShI      NA         NA         NA      NA
NameEnsor:Blade.ShR      NA         NA         NA      NA
NamePedernales:Blade.ShR      NA         NA         NA      NA
NameTravis:Blade.ShR      NA         NA         NA      NA
NameWells:Blade.ShR      NA         NA         NA      NA
NameEnsor:Blade.ShS  1.06036    1.63041    0.650  0.517
NamePedernales:Blade.ShS -0.35761    0.97336   -0.367  0.714
NameTravis:Blade.ShS  0.18485    1.32558    0.139  0.889
NameWells:Blade.ShS  2.29397    1.39973    1.639  0.106
---
```

lm1 Unfortunately, there are 0 criteria have strong relationship with Weigh

lm2 Build model which compare 2 factors (Name & Base.Sh)

```
> lm2<-lm(Weight~Length+Width+Name*Base.Sh,data=DP)
> summary(lm2)

Call:
lm(formula = Weight ~ Length + Width + Name * Base.Sh, data = DP)

Residuals:
    Min       1Q   Median       3Q      Max
-9.1720 -0.4966  0.0596  0.6916  4.6332

Coefficients: (6 not defined because of singularities)
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   -9.01369    1.51661   -5.943 8.81e-08 ***
Length         0.16044    0.02585    6.207 2.96e-08 ***
Width         0.38069    0.07018    5.425 7.19e-07 ***
NameEnsor     -1.08138    1.07402   -1.007  0.317
NamePedernales -0.13273    0.70334   -0.189  0.851
NameTravis     1.38670    2.24204    0.618  0.538
NameWells     1.15345    1.62769    0.709  0.481
Base.ShI       0.45980    1.30461    0.352  0.726
Base.ShR      -0.74802    2.17627   -0.344  0.732
Base.ShS       0.22203    1.43160    0.155  0.877
NameEnsor:Base.ShI -0.01101    2.10452   -0.005  0.996
NamePedernales:Base.ShI      NA         NA         NA      NA
NameTravis:Base.ShI  3.61773    2.80101    1.292  0.201
NameWells:Base.ShI -0.83852    2.40586   -0.349  0.728
NameEnsor:Base.ShR  0.88498    2.31452    0.382  0.703
NamePedernales:Base.ShR      NA         NA         NA      NA
NameTravis:Base.ShR      NA         NA         NA      NA
NameWells:Base.ShR      NA         NA         NA      NA
NameEnsor:Base.ShS      NA         NA         NA      NA
NamePedernales:Base.ShS      NA         NA         NA      NA
NameTravis:Base.ShS -0.42517    2.39215   -0.178  0.859
NameWells:Base.ShS -1.47840    1.89230   -0.781  0.437
---
```

lm2 Unfortunately, there are 0 criteria have strong relationship with Weight

lm3 Build model which compare 2 factors (Name & Should.Sh)

```
> lm3<-lm(Weight~Length+Width+Name*Should.Sh,data = DP)
> summary(lm3)

Call:
lm(formula = Weight ~ Length + Width + Name * Should.Sh, data = DP)

Residuals:
    Min       1Q   Median       3Q      Max
-8.7154 -0.4957  0.0000  0.7259  4.7651

Coefficients: (6 not defined because of singularities)
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   -9.655743   1.860873  -5.189 1.83e-06 ***
Length         0.181647   0.026347   6.894 1.63e-09 ***
Width         0.364467   0.069585   5.238 1.51e-06 ***
NameEnsor     -0.948221   1.816435  -0.522  0.603
NamePernales  -0.008737   2.456585  -0.004  0.997
NameTravis     1.478051   0.913133   1.619  0.110
NameWells      0.140761   0.861366   0.163  0.871
Should.ShI     0.070358   1.640151   0.043  0.966
Should.ShS     0.585641   1.476016   0.397  0.693
Should.ShX    -1.057183   1.767261  -0.598  0.552
NameEnsor:Should.ShI  0.051781   2.009296   0.026  0.980
NamePernales:Should.ShI -0.414678   2.509387  -0.165  0.869
NameTravis:Should.ShI -0.494820   1.663760  -0.297  0.767
NameWells:Should.ShI -1.063085   1.413862  -0.752  0.455
NameEnsor:Should.ShS      NA         NA      NA      NA
NamePernales:Should.ShS  0.085797   2.381190   0.036  0.971
NameTravis:Should.ShS     NA         NA      NA      NA
NameWells:Should.ShS     NA         NA      NA      NA
NameEnsor:Should.ShX     NA         NA      NA      NA
NamePernales:Should.ShX  1.848722   3.196364   0.578  0.565
NameTravis:Should.ShX     NA         NA      NA      NA
NameWells:Should.ShX     NA         NA      NA      NA
-----
```

lm3 Unfortunately, there are 0 criteria have strong relationship with Weigh

lm4 Build model lm4 which compare 2 factors (Name & Should.Or)

```
> lm4<-lm(Weight~Length+Width+Name*Should.Or,data=DP)
> summary(lm4)

Call:
lm(formula = Weight ~ Length + Width + Name * Should.Or, data = DP)

Residuals:
    Min       1Q   Median       3Q      Max
-7.3661 -0.6140  0.0372  0.5869  5.0900

Coefficients: (7 not defined because of singularities)
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  -11.31562   3.72958  -3.034  0.00333 **
Length        0.17106   0.02592   6.600 5.43e-09 ***
Width        0.40138   0.07412   5.415 7.28e-07 ***
NameEnsor     0.40655   3.35074   0.121  0.90376
NamePernales  1.51920   2.20552   0.689  0.49309
NameTravis    1.28632   0.71411   1.801  0.07573 .
NameWells     -0.04893   0.75379  -0.065  0.94842
Should.OrH    1.23167   3.36717   0.366  0.71557
Should.OrT    1.86625   2.93556   0.636  0.52691
Should.OrX    0.61518   2.60073   0.237  0.81367
NameEnsor:Should.OrH -1.22113   3.86598  -0.316  0.75299
NamePernales:Should.OrH -3.27197   2.96677  -1.103  0.27366
NameTravis:Should.OrH      NA         NA      NA      NA
NameWells:Should.OrH     -1.33154   2.59377  -0.513  0.60923
NameEnsor:Should.OrT     -1.53032   3.51954  -0.435  0.66497
NamePernales:Should.OrT -1.53370   2.26196  -0.678  0.49986
NameTravis:Should.OrT     NA         NA      NA      NA
NameWells:Should.OrT     NA         NA      NA      NA
NameEnsor:Should.OrX     NA         NA      NA      NA
NamePernales:Should.OrX     NA         NA      NA      NA
NameTravis:Should.OrX     NA         NA      NA      NA
NameWells:Should.OrX     NA         NA      NA      NA
-----
```

lm4 Find a connection of Weight & Name.T

lm4 Build a new model, contain Length, Width and Name.T

```
> model4<-lm(Weight~Length+Width+Name.T)
> summary(model4)
```

Call:

```
lm(formula = Weight ~ Length + Width + Name.T)
```

Residuals:

Min	1Q	Median	3Q	Max
-8.9545	-0.7568	0.1516	0.8525	5.3886

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-8.98390	0.82363	-10.908	< 2e-16 ***
Length	0.18533	0.02295	8.076	3.48e-12 ***
Width	0.33203	0.05700	5.825	9.41e-08 ***
Name.T	1.27593	0.57421	2.222	0.0289 *

lm4 With summary(model4), we can add Name.T become a new criterion

lm5 Build model which compare 2 factors (Name & Haft.Sh)

```
> lm5<-lm(Weight~Length+Width+Name*Haft.Sh,data=DP)
> summary(lm5)
```

Call:

```
lm(formula = Weight ~ Length + Width + Name * Haft.Sh, data = DP)
```

Residuals:

Min	1Q	Median	3Q	Max
-9.0468	-0.5428	0.0303	0.6797	5.3730

Coefficients: (11 not defined because of singularities)

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-8.50956	1.90102	-4.476	2.74e-05 ***
Length	0.17325	0.02439	7.103	6.72e-10 ***
Width	0.35819	0.06874	5.211	1.68e-06 ***
NameEnsor	-1.35491	0.98515	-1.375	0.1732
NamePedernales	-0.25220	0.77534	-0.325	0.7459
NameTravis	1.77025	0.86317	2.051	0.0439 *
NameWells	-0.17128	0.80547	-0.213	0.8322
Haft.ShE	0.71815	2.00763	0.358	0.7216
Haft.ShI	-0.66345	1.66767	-0.398	0.6919
Haft.ShR	3.01340	2.34997	1.282	0.2038
Haft.ShS	-0.17175	1.52588	-0.113	0.9107
NameEnsor:Haft.ShE	NA	NA	NA	NA
NamePedernales:Haft.ShE	-1.39908	1.53411	-0.912	0.3648
NameTravis:Haft.ShE	NA	NA	NA	NA
NameWells:Haft.ShE	-0.48182	2.25335	-0.214	0.8313
NameEnsor:Haft.ShI	0.64781	1.50471	0.431	0.6681
NamePedernales:Haft.ShI	1.24543	1.19058	1.046	0.2990
NameTravis:Haft.ShI	-1.07137	1.28029	-0.837	0.4054
NameWells:Haft.ShI	NA	NA	NA	NA
NameEnsor:Haft.ShR	NA	NA	NA	NA
NamePedernales:Haft.ShR	NA	NA	NA	NA
NameTravis:Haft.ShR	NA	NA	NA	NA
NameWells:Haft.ShR	NA	NA	NA	NA

lm5 Find a connection of Weight & Name.T, but this criterion has already been added in model

lm6 Build model which compare 2 factors (Name & Haft.Or)

```
> lm6<-lm(Weight~Length+Width+Name*Haft.Or,data=DP)
> summary(lm6)
```

Call:
lm(formula = Weight ~ Length + Width + Name * Haft.Or, data = DP)

Residuals:

Min	1Q	Median	3Q	Max
-8.3376	-0.5331	0.0000	0.6603	4.9205

Coefficients: (11 not defined because of singularities)

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-10.37267	1.92189	-5.397	8.04e-07 ***
Length	0.17713	0.02369	7.476	1.35e-10 ***
Width	0.35527	0.06687	5.313	1.12e-06 ***
NameEnsor	-1.10339	0.69849	-1.580	0.1185
NamePedernales	1.61508	2.04605	0.789	0.4325
NameTravis	1.65409	1.90578	0.868	0.3883
NameWells	2.84488	2.16423	1.315	0.1928
Haft.OrE	1.42772	1.73999	0.821	0.4146
Haft.OrP	1.86756	1.83347	1.019	0.3118
Haft.OrT	-1.04490	1.13132	-0.924	0.3587
Haft.OrV	-1.05460	1.24410	-0.848	0.3994
NameEnsor:Haft.OrE	NA	NA	NA	NA
NamePedernales:Haft.OrE	0.27165	2.65525	0.102	0.9188
NameTravis:Haft.OrE	-0.50277	2.28790	-0.220	0.8267
NameWells:Haft.OrE	NA	NA	NA	NA
NameEnsor:Haft.OrP	NA	NA	NA	NA
NamePedernales:Haft.OrP	-1.45523	2.12781	-0.684	0.4962
NameTravis:Haft.OrP	-0.13133	2.15752	-0.061	0.9516
NameWells:Haft.OrP	-4.80010	2.54585	-1.885	0.0633 .
NameEnsor:Haft.OrT	NA	NA	NA	NA
NamePedernales:Haft.OrT	NA	NA	NA	NA
NameTravis:Haft.OrT	NA	NA	NA	NA
NameWells:Haft.OrT	NA	NA	NA	NA

lm6 Find a connection of Weight & Name.W:Haft.Or.P

lm6 Build a new model, contain Length, Width and Name.W*Haft.Or.P

```
> model6<-lm(Weight~Length+Width+Name.W*Haft.Or.P)
> summary(model6)
```

Call:
lm(formula = Weight ~ Length + Width + Name.W * Haft.Or.P)

Residuals:

Min	1Q	Median	3Q	Max
-8.5966	-0.7137	0.0601	0.7859	4.8222

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-8.87970	0.81707	-10.868	< 2e-16 ***
Length	0.19357	0.02225	8.699	2.62e-13 ***
Width	0.30045	0.05529	5.434	5.41e-07 ***
Name.W	0.72212	0.64856	1.113	0.2687
Haft.Or.P	1.17149	0.41091	2.851	0.0055 **
Name.W:Haft.Or.P	-3.15780	1.41154	-2.237	0.0280 *

lm6 There is no strong connection between Weight & Name.W

lm6 Remove Name.W

```
> model6<-lm(Weight~Length+Width+Name.W:Haft.Or.P+Haft.Or.P)
> summary(model6)

Call:
lm(formula = Weight ~ Length + Width + Name.W:Haft.Or.P + Haft.Or.P)

Residuals:
    Min       1Q   Median       3Q      Max
-8.7302 -0.7067  0.0629  0.8018  5.4254

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   -8.90017    0.81803  -10.880 < 2e-16 ***
Length         0.19511    0.02224   8.772 1.70e-13 ***
Width          0.30216    0.05535   5.459 4.76e-07 ***
Haft.Or.P      1.07705    0.40263   2.675 0.00898 **
Name.W:Haft.Or.P -2.42803    1.25192  -1.939 0.05580 .
---

```

lm6 With `summary(model6)`, we can add Name.W & Name.W:Haft.Or.P become new criteria

lm7 Build model which compare 2 factors (Blade.Sh&Base.Sh)

```
> lm7<-lm(Weight~Length+Width+Blade.Sh*Base.Sh,data=DP)
> summary(lm7)

Call:
lm(formula = Weight ~ Length + Width + Blade.Sh * Base.Sh, data = DP)

Residuals:
    Min       1Q   Median       3Q      Max
-8.7240 -0.6269  0.0684  0.6929  5.0268

Coefficients: (5 not defined because of singularities)
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   -7.92822    2.11230  -3.753 0.000339 ***
Length         0.17722    0.02589   6.846 1.71e-09 ***
Width          0.33970    0.06299   5.393 7.56e-07 ***
Blade.ShI      -0.58538    1.88872  -0.310 0.757460
Blade.ShR      -1.42160    1.09740  -1.295 0.199094
Blade.ShS      -0.37206    2.04015  -0.182 0.855778
Base.ShI       -0.06709    1.90419  -0.035 0.971988
Base.ShR      -1.31645    1.33434  -0.987 0.326973
Base.ShS      -0.70188    1.94930  -0.360 0.719793
Blade.ShI:Base.ShI -1.03415    2.29893  -0.450 0.654108
Blade.ShR:Base.ShI      NA         NA      NA      NA
Blade.ShS:Base.ShI -0.56273    2.08450  -0.270 0.787925
Blade.ShI:Base.ShR -0.34924    3.43900  -0.102 0.919379
Blade.ShR:Base.ShR      NA         NA      NA      NA
Blade.ShS:Base.ShR      NA         NA      NA      NA
Blade.ShI:Base.ShS      NA         NA      NA      NA
Blade.ShR:Base.ShS      NA         NA      NA      NA
Blade.ShS:Base.ShS   0.33726    2.14737   0.157 0.875615

```

lm7 Unfortunately, there are 0 criteria have strong relationship with Weigh

lm8 Build model which compare 2 factors (Blade.Sh&Should.Sh)

```
> lm8<-lm(Weight~Length+Width+Blade.Sh*Should.Sh,data=DP)
> summary(lm8)
```

Call:
lm(formula = Weight ~ Length + Width + Blade.Sh * Should.Sh,
data = DP)

Residuals:

Min	1Q	Median	3Q	Max
-7.7775	-0.5917	0.0105	0.6484	5.0264

Coefficients: (5 not defined because of singularities)

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-8.15385	1.32404	-6.158	3.22e-08 ***
Length	0.17228	0.02295	7.506	9.65e-11 ***
Width	0.36613	0.05586	6.554	5.99e-09 ***
Blade.ShI	-1.73293	1.04644	-1.656	0.1018
Blade.ShR	1.87485	1.72215	1.089	0.2797
Blade.ShS	0.19661	2.03829	0.096	0.9234
Should.ShI	-0.66890	1.01560	-0.659	0.5121
Should.ShS	0.07934	1.05629	0.075	0.9403
Should.ShX	-0.61506	1.50336	-0.409	0.6836
Blade.ShI:Should.ShI	0.10158	1.98388	0.051	0.9593
Blade.ShR:Should.ShI	-4.64767	2.10587	-2.207	0.0303 *
Blade.ShS:Should.ShI	-1.76456	2.10462	-0.838	0.4044
Blade.ShI:Should.ShS	NA	NA	NA	NA
Blade.ShR:Should.ShS	NA	NA	NA	NA
Blade.ShS:Should.ShS	-0.76312	2.10142	-0.363	0.7175
Blade.ShI:Should.ShX	NA	NA	NA	NA
Blade.ShR:Should.ShX	NA	NA	NA	NA
Blade.ShS:Should.ShX	NA	NA	NA	NA

lm8 Find a connection of Weight & Blade.Sh.R:Should.Sh.I

lm8 Build a new model, contain Length, Width and Blade.Sh.R*Should.Sh.I

```
> model8<-lm(Weight~Length+Width+Blade.Sh.R*Should.Sh.I)
> summary(model8)
```

Call:
lm(formula = Weight ~ Length + Width + Blade.Sh.R * Should.Sh.I)

Residuals:

Min	1Q	Median	3Q	Max
-8.7506	-0.7968	0.0000	0.8848	4.7330

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-8.73431	0.80045	-10.912	< 2e-16 ***
Length	0.19018	0.02198	8.652	3.24e-13 ***
Width	0.33608	0.05501	6.109	3.11e-08 ***
Blade.Sh.R	2.14671	1.71058	1.255	0.2130
Should.Sh.I	-0.89671	0.37687	-2.379	0.0196 *
Blade.Sh.R:Should.Sh.I	-4.34506	2.09386	-2.075	0.0411 *

lm8 There is no strong connection between Weight & Blade.Sh.R

lm8 Remove Blade.Sh.R

```
> model8<-lm(Weight~Length+Width+Should.Sh.I+Blade.Sh.R:Should.Sh.I)
> summary(model8)
```

Call:
lm(formula = Weight ~ Length + Width + Should.Sh.I + Blade.Sh.R:Should.Sh.I)

Residuals:

Min	1Q	Median	3Q	Max
-8.7906	-0.7720	-0.0147	0.8921	4.6461

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-8.83601	0.79906	-11.058	< 2e-16 ***
Length	0.19309	0.02193	8.804	1.47e-13 ***
Width	0.33628	0.05520	6.092	3.24e-08 ***
Should.Sh.I	-0.95052	0.37570	-2.530	0.0133 *
Should.Sh.I:Blade.Sh.R	-2.20505	1.21922	-1.809	0.0741 .

lm8 With summary(model8), we can add Should.Sh.I & Should.Sh.I:Blade.Sh.R become new criteria

lm9 Build model which compare 2 factors (Blade.Sh&Shoule.Or)

```
> lm9<-lm(Weight~Length+Width+Blade.Sh*Should.Or,data=DP)
> summary(lm9)
```

Call:
lm(formula = Weight ~ Length + Width + Blade.Sh * Should.Or,
data = DP)

Residuals:

Min	1Q	Median	3Q	Max
-6.5364	-0.5893	0.0000	0.6904	5.0452

Coefficients: (4 not defined because of singularities)

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-9.87692	2.09353	-4.718	1.08e-05 ***
Length	0.18378	0.02365	7.772	3.24e-11 ***
Width	0.36609	0.05770	6.345	1.53e-08 ***
Blade.ShI	-1.26872	1.01055	-1.255	0.2132
Blade.ShR	0.31670	1.20340	0.263	0.7931
Blade.ShS	-0.18542	2.05915	-0.090	0.9285
Should.OrH	0.35420	1.87374	0.189	0.8506
Should.OrT	0.87927	1.73178	0.508	0.6131
Should.OrX	0.49567	2.05732	0.241	0.8103
Blade.ShI:Should.OrH	-0.09870	2.07451	-0.048	0.9622
Blade.ShR:Should.OrH	-4.83457	2.16657	-2.231	0.0286 *
Blade.ShS:Should.OrH	-2.66947	2.33558	-1.143	0.2567
Blade.ShI:Should.OrT	NA	NA	NA	NA
Blade.ShR:Should.OrT	NA	NA	NA	NA
Blade.ShS:Should.OrT	-0.20095	2.08968	-0.096	0.9236
Blade.ShI:Should.OrX	NA	NA	NA	NA
Blade.ShR:Should.OrX	NA	NA	NA	NA
Blade.ShS:Should.OrX	0.58826	2.87157	0.205	0.8382

lm9 Find a connection of Weight & Blade.Sh.R:Should.Or.H

lm9 Build a new model, contain Length, Width and Blade.Sh.R*Should.Or.H

```
> model9<-lm(Weight~Length+Width+Blade.Sh.S*Should.Or.H)
> summary(model9)
```

Call:
lm(formula = Weight ~ Length + Width + Blade.Sh.S * Should.Or.H)

Residuals:

	Min	1Q	Median	3Q	Max
	-6.4808	-0.6934	0.0564	0.7339	5.1111

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-8.88256	0.82765	-10.732	< 2e-16 ***
Length	0.18311	0.02248	8.146	3.33e-12 ***
Width	0.35711	0.05586	6.393	9.04e-09 ***
Blade.Sh.S	-0.30368	0.38835	-0.782	0.4365
Should.Or.H	-1.23172	0.68827	-1.790	0.0772 .
Blade.Sh.S:Should.Or.H	-1.68833	1.11214	-1.518	0.1328

lm9 There is no strong connection between Weight, Blade.Sh.S & Blade.Sh.S:Should.Or.H

lm9 Remove Blade.Sh.S & Blade.Sh.S:Should.Or.H

```
> model9<-lm(Weight~Length+Width+Should.Or.H)
> summary(model9)
```

Call:
lm(formula = Weight ~ Length + Width + Should.Or.H)

Residuals:

	Min	1Q	Median	3Q	Max
	-7.7179	-0.6840	0.0447	0.7485	4.9294

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-8.96049	0.80172	-11.177	< 2e-16 ***
Length	0.18682	0.02205	8.474	6.26e-13 ***
Width	0.34586	0.05565	6.215	1.83e-08 ***
Should.Or.H	-1.79417	0.55366	-3.241	0.0017 **

lm9 With summary(model9), we can add Should.Or.H

lm10 Build model which compare 2 factors (Blade.Sh&Haft.Sh)

```
> lm10<-lm(Weight~Length+Width+Blade.Sh*Haft.Sh,data=DP)
> summary(lm10)
```

Call:
lm(formula = Weight ~ Length + Width + Blade.Sh * Haft.Sh, data = DP)

Residuals:

	Min	1Q	Median	3Q	Max
	-8.9416	-0.5354	0.0440	0.5355	5.4924

Coefficients: (7 not defined because of singularities)

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-10.01405	1.96381	-5.099	2.54e-06 ***
Length	0.16962	0.02443	6.942	1.26e-09 ***
Width	0.34953	0.05910	5.915	9.56e-08 ***
Blade.ShI	-1.42182	1.29068	-1.102	0.2742
Blade.ShR	0.56015	1.28489	0.436	0.6641
Blade.ShS	1.00298	2.45391	0.409	0.6839
Haft.ShE	1.95117	1.88007	1.038	0.3027
Haft.ShI	2.03858	1.83725	1.110	0.2708
Haft.ShR	4.73444	2.48141	1.908	0.0603 .
Haft.ShS	1.98681	1.81351	1.096	0.2768
Blade.ShI:Haft.ShE	NA	NA	NA	NA
Blade.ShR:Haft.ShE	-5.22613	2.24940	-2.323	0.0229 *
Blade.ShS:Haft.ShE	-0.88613	3.09888	-0.286	0.7757
Blade.ShI:Haft.ShI	-0.24276	1.83601	-0.132	0.8952
Blade.ShR:Haft.ShI	NA	NA	NA	NA
Blade.ShS:Haft.ShI	-2.39469	2.60249	-0.920	0.3605
Blade.ShI:Haft.ShR	NA	NA	NA	NA
Blade.ShR:Haft.ShR	NA	NA	NA	NA
Blade.ShS:Haft.ShR	NA	NA	NA	NA
Blade.ShI:Haft.ShS	NA	NA	NA	NA
Blade.ShR:Haft.ShS	NA	NA	NA	NA
Blade.ShS:Haft.ShS	-1.50435	2.51394	-0.598	0.5514

lm10 Find a connection of Weight, Haft.Sh.R & Blade.Sh.R:Haft.Sh.E

lm10 Build a new model, contain Length, Width, Haft.Sh.R & Blade.Sh.R*Haft.Sh.E

```
> model10<-lm(Weight~Length+Width+Haft.Sh.R+Blade.Sh.R*Haft.Sh.E)
> summary(model10)
```

Call:
lm(formula = Weight ~ Length + Width + Haft.Sh.R + Blade.Sh.R * Haft.Sh.E)

Residuals:

	Min	1Q	Median	3Q	Max
	-9.0363	-0.6838	0.0722	0.8246	5.2249

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-8.76776	0.81934	-10.701	< 2e-16 ***
Length	0.18961	0.02242	8.456	8.70e-13 ***
Width	0.31897	0.05566	5.730	1.61e-07 ***
Haft.Sh.R	3.03783	1.73003	1.756	0.0828 .
Blade.Sh.R	0.88152	1.22781	0.718	0.4748
Haft.Sh.E	0.39332	0.63508	0.619	0.5374
Blade.Sh.R:Haft.Sh.E	-5.58288	2.18636	-2.553	0.0125 *

lm10 There is no strong connection between Weight, Blade.Sh.R & Haft.Sh.E

lm10 Remove Blade.Sh.R & Haft.Sh.E

```
> model10<-lm(Weight~Length+Width+Haft.Sh.R+Blade.Sh.R:Haft.Sh.E)
> summary(model10)
```

Call:
lm(formula = Weight ~ Length + Width + Haft.Sh.R + Blade.Sh.R:Haft.Sh.E)

Residuals:

Min	1Q	Median	3Q	Max
-9.1203	-0.7280	0.0232	0.7653	5.1464

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-8.79369	0.81301	-10.816	< 2e-16 ***
Length	0.19111	0.02219	8.614	3.55e-13 ***
Width	0.31936	0.05522	5.784	1.22e-07 ***
Haft.Sh.R	2.95691	1.71582	1.723	0.0885 .
Blade.Sh.R:Haft.Sh.E	-4.38309	1.71530	-2.555	0.0124 *

lm10 With summary(model10), we can add Haft.Sh.R & Blade.Sh.R:Haft.Sh.E become new criteria

lm11 Build model which compare 2 factors (Blade.Sh&Haft.Or)

```
> lm11<-lm(Weight~Length+Width+Blade.Sh*Haft.Or,data=DP)
> summary(lm11)
```

Call:
lm(formula = Weight ~ Length + Width + Blade.Sh * Haft.Or, data = DP)

Residuals:

Min	1Q	Median	3Q	Max
-9.0749	-0.6192	0.0000	0.6274	5.2482

Coefficients: (6 not defined because of singularities)

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-8.15593	1.30848	-6.233	2.65e-08 ***
Length	0.17891	0.02393	7.475	1.36e-10 ***
Width	0.33089	0.05869	5.638	3.06e-07 ***
Blade.ShI	0.44652	1.93496	0.231	0.8181
Blade.ShR	-4.35594	1.91403	-2.276	0.0258 *
Blade.ShS	-1.26214	1.42663	-0.885	0.3792
Haft.OrE	-0.07525	0.86919	-0.087	0.9312
Haft.OrP	0.68139	0.83832	0.813	0.4190
Haft.OrT	-1.12150	1.10503	-1.015	0.3135
Haft.OrV	-0.25440	1.10494	-0.230	0.8186
Blade.ShI:Haft.OrE	-2.00931	2.20758	-0.910	0.3657
Blade.ShR:Haft.OrE	NA	NA	NA	NA
Blade.ShS:Haft.OrE	0.55269	1.54677	0.357	0.7219
Blade.ShI:Haft.OrP	NA	NA	NA	NA
Blade.ShR:Haft.OrP	6.14081	2.61575	2.348	0.0216 *
Blade.ShS:Haft.OrP	0.21823	1.57938	0.138	0.8905
Blade.ShI:Haft.OrT	NA	NA	NA	NA
Blade.ShR:Haft.OrT	4.21233	2.71915	1.549	0.1257
Blade.ShS:Haft.OrT	2.12933	1.73921	1.224	0.2248
Blade.ShI:Haft.OrV	NA	NA	NA	NA
Blade.ShR:Haft.OrV	NA	NA	NA	NA
Blade.ShS:Haft.OrV	NA	NA	NA	NA

lm11 Find a connection of Weight, Blade.Sh.R & Blade.Sh.R:Haft.Or.P

lm11 Build a new model, contain Length, Width, Blade.Sh.R & Blade.Sh.R*Haft.Or.P

```
> model11<-lm(Weight~Length+Width+Blade.Sh.R*Haft.Or.P)
> summary(model11)
```

Call:

```
lm(formula = Weight ~ Length + Width + Blade.Sh.R * Haft.Or.P)
```

Residuals:

Min	1Q	Median	3Q	Max
-8.8683	-0.6213	0.0000	0.7949	5.3697

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-8.89963	0.81924	-10.863	< 2e-16 ***
Length	0.19085	0.02226	8.575	4.62e-13 ***
Width	0.31526	0.05503	5.729	1.58e-07 ***
Blade.Sh.R	-2.45851	1.21889	-2.017	0.0469 *
Haft.Or.P	0.74068	0.39717	1.865	0.0657 .
Blade.Sh.R:Haft.Or.P	4.54837	2.11933	2.146	0.0348 *

lm11 With summary(model11), we can add Blade.Sh.R*Haft.Or.P become new criterion

lm12 Build model which compare 2 factors (Base.Sh&Should.Sh)

```
> lm12<-lm(Weight~Length+Width+Base.Sh*Should.Sh,data=DP)
> summary(lm12)
```

Call:

```
lm(formula = Weight ~ Length + Width + Base.Sh * Should.Sh, data = DP)
```

Residuals:

Min	1Q	Median	3Q	Max
-8.8226	-0.4821	0.0368	0.7905	4.3705

Coefficients: (5 not defined because of singularities)

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-7.654996	1.873237	-4.087	0.000108 ***
Length	0.194203	0.025420	7.640	5.37e-11 ***
Width	0.325948	0.062331	5.229	1.45e-06 ***
Base.ShI	-1.910235	2.440148	-0.783	0.436156
Base.ShR	-0.463711	1.603785	-0.289	0.773265
Base.ShS	-0.314508	1.030618	-0.305	0.761075
Should.ShI	-2.047968	2.062967	-0.993	0.323992
Should.ShS	-0.720646	1.362042	-0.529	0.598284
Should.ShX	-1.159077	1.785388	-0.649	0.518165
Base.ShI:Should.ShI	2.112606	2.811703	0.751	0.454755
Base.ShR:Should.ShI	NA	NA	NA	NA
Base.ShS:Should.ShI	0.006746	1.734584	0.004	0.996907
Base.ShI:Should.ShS	1.593691	2.297221	0.694	0.489956
Base.ShR:Should.ShS	NA	NA	NA	NA
Base.ShS:Should.ShS	NA	NA	NA	NA
Base.ShI:Should.ShX	2.119484	3.193330	0.664	0.508877
Base.ShR:Should.ShX	NA	NA	NA	NA
Base.ShS:Should.ShX	NA	NA	NA	NA

lm12 Unfortunately, there are 0 criteria have strong relationship with Weigh

lm13 Build model which compare 2 factors (Base.Sh&Should.Or)

```
> lm13<-lm(Weight~Length+Width+Base.Sh*Should.Or,data=DP)
> summary(lm13)

Call:
lm(formula = Weight ~ Length + Width + Base.Sh * Should.Or, data = DP)

Residuals:
    Min       1Q   Median       3Q      Max
-7.5390 -0.5739 -0.0352  0.6832  4.7856

Coefficients: (5 not defined because of singularities)
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   -10.408376    2.153791  -4.833 6.85e-06 ***
Length          0.182346    0.025131   7.256 2.88e-10 ***
Width          0.362415    0.063341   5.722 1.98e-07 ***
Base.ShI        1.060606    2.479528   0.428  0.670
Base.ShR       -1.228023    1.479195  -0.830  0.409
Base.ShS       -0.208226    0.831288  -0.250  0.803
Should.OrH     -0.006763    2.028447  -0.003  0.997
Should.OrT      1.465097    1.794681   0.816  0.417
Should.OrX      1.385770    2.158626   0.642  0.523
Base.ShI:Should.OrH -1.811242    2.612723  -0.693  0.490
Base.ShR:Should.OrH  0.962135    2.088653   0.461  0.646
Base.ShS:Should.OrH      NA         NA      NA      NA
Base.ShI:Should.OrT -1.120552    2.372887  -0.472  0.638
Base.ShR:Should.OrT      NA         NA      NA      NA
Base.ShS:Should.OrT      NA         NA      NA      NA
Base.ShI:Should.OrX -0.897735    3.278417  -0.274  0.785
Base.ShR:Should.OrX      NA         NA      NA      NA
Base.ShS:Should.OrX      NA         NA      NA      NA
```

lm13 Unfortunately, there are 0 criteria have strong relationship with Weigh

lm14 Build model which compare 2 factors (Base.Sh&Haft.Sh)

```
> lm14<-lm(Weight~Length+Width+Base.Sh*Haft.Sh,data=DP)
> summary(lm14)

Call:
lm(formula = Weight ~ Length + Width + Base.Sh * Haft.Sh, data = DP)

Residuals:
    Min       1Q   Median       3Q      Max
-9.0240 -0.4133  0.0375  0.8338  4.5674

Coefficients: (9 not defined because of singularities)
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   -8.77302    1.84093  -4.766 8.86e-06 ***
Length          0.18825    0.02527   7.449 1.24e-10 ***
Width          0.30852    0.06302   4.896 5.37e-06 ***
Base.ShI       -0.62018    0.96211  -0.645  0.5211
Base.ShR      -2.20460    1.56771  -1.406  0.1637
Base.ShS      -0.58279    1.01109  -0.576  0.5661
Haft.ShE       0.81542    1.53022   0.533  0.5957
Haft.ShI      -0.27151    2.12913  -0.128  0.8989
Haft.ShR       4.03475    2.30391   1.751  0.0839 .
Haft.ShS       1.03184    1.37445   0.751  0.4551
Base.ShI:Haft.ShE      NA         NA      NA      NA
Base.ShR:Haft.ShE      NA         NA      NA      NA
Base.ShS:Haft.ShE      NA         NA      NA      NA
Base.ShI:Haft.ShI  1.41713    1.72492   0.822  0.4139
Base.ShR:Haft.ShI  2.05204    2.44542   0.839  0.4040
Base.ShS:Haft.ShI  0.87849    1.76203   0.499  0.6195
Base.ShI:Haft.ShR      NA         NA      NA      NA
Base.ShR:Haft.ShR      NA         NA      NA      NA
Base.ShS:Haft.ShR      NA         NA      NA      NA
Base.ShI:Haft.ShS      NA         NA      NA      NA
Base.ShR:Haft.ShS      NA         NA      NA      NA
Base.ShS:Haft.ShS      NA         NA      NA      NA
```

lm14 Find a connection of Weight & Haft.Sh.R

lm14 Build a new model, contain Length, Width & Haft.Sh.R

```
> model14<-lm(Weight~Length+Width+Haft.Sh.R,data=DP)
> summary(model14)
```

Call:

```
lm(formula = Weight ~ Length + Width + Haft.Sh.R, data = DP)
```

Residuals:

Min	1Q	Median	3Q	Max
-8.9627	-0.7383	0.0987	0.8308	5.2227

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-8.60173	0.83545	-10.296	< 2e-16 ***
Length	0.19151	0.02290	8.365	1.04e-12 ***
Width	0.30747	0.05678	5.415	5.60e-07 ***
Haft.Sh.R	3.06253	1.77023	1.730	0.0873 .

lm14 With summary(model14), we can add Haft.Sh.R become new criterion

lm15 Build model which compare 2 factors (Base.Sh & Haft.Or)

```
> lm15<-lm(Weight~Length+Width+Base.Sh*Haft.Or,data=DP)
> summary(lm15)
```

Call:

```
lm(formula = Weight ~ Length + Width + Base.Sh * Haft.Or, data = DP)
```

Residuals:

Min	1Q	Median	3Q	Max
-8.4200	-0.5623	0.0000	0.6663	4.0294

Coefficients: (6 not defined because of singularities)

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-10.36945	1.81065	-5.727	2.13e-07 ***
Length	0.18268	0.02252	8.113	8.62e-12 ***
Width	0.34128	0.05889	5.795	1.61e-07 ***
Base.ShI	1.66993	1.96182	0.851	0.39743
Base.ShR	-1.83875	1.45718	-1.262	0.21102
Base.ShS	1.65964	1.84624	0.899	0.37165
Haft.OrE	2.19760	2.01903	1.088	0.27998
Haft.OrP	-0.07423	2.01829	-0.037	0.97076
Haft.OrT	6.71285	2.35460	2.851	0.00566 **
Haft.OrV	-1.07266	1.20551	-0.890	0.37650
Base.ShI:Haft.OrE	-2.23731	2.32617	-0.962	0.33932
Base.ShR:Haft.OrE	NA	NA	NA	NA
Base.ShS:Haft.OrE	-2.89277	2.24875	-1.286	0.20237
Base.ShI:Haft.OrP	0.60581	2.28602	0.265	0.79175
Base.ShR:Haft.OrP	NA	NA	NA	NA
Base.ShS:Haft.OrP	1.07769	2.30515	0.468	0.64152
Base.ShI:Haft.OrT	-7.65190	2.59741	-2.946	0.00432 **
Base.ShR:Haft.OrT	NA	NA	NA	NA
Base.ShS:Haft.OrT	-7.34969	2.57583	-2.853	0.00563 **
Base.ShI:Haft.OrV	NA	NA	NA	NA
Base.ShR:Haft.OrV	NA	NA	NA	NA
Base.ShS:Haft.OrV	NA	NA	NA	NA

lm15 Find a connection of Weight, Haft.Or.T, Base.Sh.I:Haft.Or.T & Base.Sh.S:Haft.Or.T

lm15 Build a new model, contain Length, Width, Haft.Or.T, Base.Sh.I*Haft.Or.T & Base.Sh.S*Haft.Or.T

```
> model15<-lm(Weight~Length+Width+Haft.Or.T*Base.Sh.I+Base.Sh.S*Haft.Or.T,data=DP)
> summary(model15)
```

Call:

```
lm(formula = Weight ~ Length + Width + Haft.Or.T * Base.Sh.I +
    Base.Sh.S * Haft.Or.T, data = DP)
```

Residuals:

Min	1Q	Median	3Q	Max
-8.3642	-0.5581	0.1260	0.7712	4.3514

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-9.64337	0.93932	-10.266	2.55e-16 ***
Length	0.18855	0.02198	8.578	5.42e-13 ***
Width	0.32298	0.05582	5.786	1.31e-07 ***
Haft.Or.T	6.06827	1.77708	3.415	0.001000 ***
Base.Sh.I	1.16761	0.61864	1.887	0.062690 .
Base.Sh.S	0.90423	0.66881	1.352	0.180140
Haft.Or.T:Base.Sh.I	-7.09023	1.85634	-3.819	0.000261 ***
Haft.Or.T:Base.Sh.S	-6.54935	1.92837	-3.396	0.001061 **

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.665 on 81 degrees of freedom
(2 observations deleted due to missingness)

Multiple R-squared: 0.858, Adjusted R-squared: 0.8457
F-statistic: 69.92 on 7 and 81 DF, p-value: < 2.2e-16

lm15 There is no strong connection between Weight & Base.Sh.S

lm15 Remove Base.Sh.S

```
> model15<-lm(Weight~Length+Width+Haft.Or.T+Haft.Or.T:Base.Sh.I+Haft.Or.T:Base.Sh.S,data=DP)
> summary(model15)
```

Call:

```
lm(formula = Weight ~ Length + Width + Haft.Or.T + Haft.Or.T:Base.Sh.I +
    Haft.Or.T:Base.Sh.S, data = DP)
```

Residuals:

Min	1Q	Median	3Q	Max
-8.4199	-0.7258	0.2473	0.8963	4.5693

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-8.95118	0.83506	-10.719	< 2e-16 ***
Length	0.18839	0.02218	8.493	6.75e-13 ***
Width	0.33542	0.05582	6.009	4.77e-08 ***
Haft.Or.T	5.07853	1.70928	2.971	0.00388 **
Haft.Or.T:Base.Sh.I	-5.95252	1.77211	-3.359	0.00118 **
Haft.Or.T:Base.Sh.S	-5.63454	1.82643	-3.085	0.00277 **

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.681 on 83 degrees of freedom
(2 observations deleted due to missingness)

Multiple R-squared: 0.8517, Adjusted R-squared: 0.8428
F-statistic: 95.36 on 5 and 83 DF, p-value: < 2.2e-16

lm15 With summary(model15), we can add Haft.Or.T, Haft.Or.T:Base.Sh.I & Haft.Or.T:Base.Sh.S become new criteria

lm16 Build model which compare 2 factors (Should.Sh&Should.Or)

```
> lm16<-lm(Weight~Length+Width+Should.Sh*Should.Or,data=DP)
> summary(lm16)
```

Call:

```
lm(formula = Weight ~ Length + Width + Should.Sh * Should.Or,
    data = DP)
```

Residuals:

Min	1Q	Median	3Q	Max
-7.2843	-0.6443	0.0000	0.6968	4.7845

Coefficients: (7 not defined because of singularities)

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-9.51772	2.20474	-4.317	4.60e-05 ***
Length	0.18337	0.02268	8.084	6.48e-12 ***
Width	0.35445	0.05914	5.993	6.00e-08 ***
Should.ShI	-0.98321	2.39511	-0.411	0.683
Should.ShS	0.76460	2.51235	0.304	0.762
Should.ShX	0.51833	1.96811	0.263	0.793
Should.OrH	2.45823	2.53520	0.970	0.335
Should.OrT	1.47210	2.06444	0.713	0.478
Should.OrX	NA	NA	NA	NA
Should.ShI:Should.OrH	-3.19250	3.13563	-1.018	0.312
Should.ShS:Should.OrH	NA	NA	NA	NA
Should.ShX:Should.OrH	NA	NA	NA	NA
Should.ShI:Should.OrT	-0.23060	2.68797	-0.086	0.932
Should.ShS:Should.OrT	-1.52889	2.80341	-0.545	0.587
Should.ShX:Should.OrT	NA	NA	NA	NA
Should.ShI:Should.OrX	NA	NA	NA	NA
Should.ShS:Should.OrX	NA	NA	NA	NA
Should.ShX:Should.OrX	NA	NA	NA	NA

lm16 Unfortunately, there are 0 criteria have strong relationship with Weigh

lm17 Build model which compare 2 factors (Should.Sh&Haft.Sh)

```
> lm17<-lm(Weight~Length+Width+Should.Sh*Haft.Sh,data=DP)
> summary(lm17)
```

Call:

```
lm(formula = Weight ~ Length + Width + Should.Sh * Haft.Sh, data = DP)
```

Residuals:

Min	1Q	Median	3Q	Max
-8.2680	-0.6003	0.0000	0.8521	4.6680

Coefficients: (9 not defined because of singularities)

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-8.43758	1.85450	-4.550	2.00e-05 ***
Length	0.18572	0.02400	7.740	3.46e-11 ***
Width	0.34541	0.05845	5.910	9.11e-08 ***
Should.ShI	-1.65824	1.09733	-1.511	0.135
Should.ShS	-0.15811	1.05977	-0.149	0.882
Should.ShX	-0.20414	2.03479	-0.100	0.920
Haft.ShE	-0.20188	1.52896	-0.132	0.895
Haft.ShI	-0.42631	2.55222	-0.167	0.868
Haft.ShR	2.39608	2.22759	1.076	0.285
Haft.ShS	-0.02273	1.31688	-0.017	0.986
Should.ShI:Haft.ShE	-0.06128	1.32953	-0.046	0.963
Should.ShS:Haft.ShE	NA	NA	NA	NA
Should.ShX:Haft.ShE	NA	NA	NA	NA
Should.ShI:Haft.ShI	1.32810	2.23562	0.594	0.554
Should.ShS:Haft.ShI	-0.57726	2.36348	-0.244	0.808
Should.ShX:Haft.ShI	NA	NA	NA	NA
Should.ShI:Haft.ShR	NA	NA	NA	NA
Should.ShS:Haft.ShR	NA	NA	NA	NA
Should.ShX:Haft.ShR	NA	NA	NA	NA
Should.ShI:Haft.ShS	NA	NA	NA	NA
Should.ShS:Haft.ShS	NA	NA	NA	NA
Should.ShX:Haft.ShS	NA	NA	NA	NA

lm17 Unfortunately, there are 0 criteria have strong relationship with Weigh

lm18 Build model which compare 2 factors (Should.Sh&Haft.Or)

```
> lm18<-lm(Weight~Length+Width+Should.Sh*Haft.Or,data=DP)
> summary(lm18)
```

Call:
lm(formula = Weight ~ Length + Width + Should.Sh * Haft.Or, data = DP)

Residuals:

	Min	1Q	Median	3Q	Max
	-7.0862	-0.6917	-0.0319	0.7712	4.3651

Coefficients: (6 not defined because of singularities)

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-10.65367	2.67135	-3.988	0.000156 ***
Length	0.19176	0.02321	8.260	4.57e-12 ***
Width	0.33777	0.05945	5.681	2.56e-07 ***
Should.ShI	1.33032	2.51173	0.530	0.597968
Should.ShS	1.19799	1.78261	0.672	0.503677
Should.ShX	1.41735	2.67191	0.530	0.597400
Haft.OrE	0.38523	1.72699	0.223	0.824107
Haft.OrP	2.54923	2.67291	0.954	0.343369
Haft.OrT	0.94774	1.76254	0.538	0.592409
Haft.OrV	0.12392	1.87773	0.066	0.947564
Should.ShI:Haft.OrE	-0.73507	1.91654	-0.384	0.702432
Should.ShS:Haft.OrE	NA	NA	NA	NA
Should.ShX:Haft.OrE	NA	NA	NA	NA
Should.ShI:Haft.OrP	-2.09403	2.84886	-0.735	0.464669
Should.ShS:Haft.OrP	-1.67222	2.16098	-0.774	0.441532
Should.ShX:Haft.OrP	-2.04989	3.35431	-0.611	0.543018
Should.ShI:Haft.OrT	-3.07593	2.03244	-1.513	0.134491
Should.ShS:Haft.OrT	NA	NA	NA	NA
Should.ShX:Haft.OrT	NA	NA	NA	NA
Should.ShI:Haft.OrV	-4.52981	2.60776	-1.737	0.086596 .
Should.ShS:Haft.OrV	NA	NA	NA	NA
Should.ShX:Haft.OrV	NA	NA	NA	NA

lm18 Find a connection of Weight & Should.Sh.I:Haft.Or.V

lm18 Build a new model, contain Length, Width & Should.Sh.I*Haft.Or.V

```
> model18<-lm(Weight~Length+Width+Should.Sh.I*Haft.Or.V)
> summary(model18)
```

Call:
lm(formula = Weight ~ Length + Width + Should.Sh.I * Haft.Or.V)

Residuals:

	Min	1Q	Median	3Q	Max
	-8.8712	-0.7234	-0.0121	0.8215	4.5550

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-9.02544	0.79863	-11.301	< 2e-16 ***
Length	0.19327	0.02173	8.895	1.06e-13 ***
Width	0.34716	0.05498	6.314	1.28e-08 ***
Should.Sh.I	-1.03647	0.37774	-2.744	0.00744 **
Haft.Or.V	-0.63423	0.88176	-0.719	0.47399
Should.Sh.I:Haft.Or.V	-3.38608	1.89107	-1.791	0.07701 .

lm18 There is no strong connection between Weight & Haft.Or.V

lm18 Remove Haft.Or.V

```
> model18<-lm(Weight~Length+Width+Should.Sh.I:Haft.Or.V)
> summary(model18)
```

Call:
lm(formula = Weight ~ Length + Width + Should.Sh.I:Haft.Or.V)

Residuals:

Min	1Q	Median	3Q	Max
-9.2110	-0.7329	0.0278	0.7408	5.0741

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-8.92328	0.81885	-10.897	< 2e-16 ***
Length	0.19352	0.02240	8.641	2.88e-13 ***
Width	0.32137	0.05584	5.755	1.34e-07 ***
Should.Sh.I:Haft.Or.V	-4.45430	1.73456	-2.568	0.012 *

lm18 With summary(model18), we can add Should.Sh.I:Haft.Or.V become a new criterion

lm19 Build model which compare 2 factors (Should.Or&Haft.Sh)

```
> lm19<-lm(Weight~Length+Width+Should.Or*Haft.Sh,data=DP)
> summary(lm19)
```

Call:
lm(formula = Weight ~ Length + Width + Should.Or * Haft.Sh, data = DP)

Residuals:

Min	1Q	Median	3Q	Max
-7.4838	-0.5609	0.0000	0.6515	4.7351

Coefficients: (9 not defined because of singularities)

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-11.44722	2.01683	-5.676	2.39e-07 ***
Length	0.18061	0.02259	7.994	1.13e-11 ***
Width	0.37290	0.05627	6.627	4.39e-09 ***
Should.OrH	-1.46180	1.18945	-1.229	0.22287
Should.OrT	0.93539	1.08635	0.861	0.39192
Should.OrX	0.98550	1.99017	0.495	0.62190
Haft.ShE	1.35402	1.30495	1.038	0.30274
Haft.ShI	1.11210	2.38096	0.467	0.64178
Haft.ShR	6.28887	2.13772	2.942	0.00432 **
Haft.ShS	1.46940	1.19913	1.225	0.22421
Should.OrH:Haft.ShE	-2.48566	1.85210	-1.342	0.18357
Should.OrT:Haft.ShE	NA	NA	NA	NA
Should.OrX:Haft.ShE	NA	NA	NA	NA
Should.OrH:Haft.ShI	1.12518	2.42070	0.465	0.64339
Should.OrT:Haft.ShI	-0.10766	2.07791	-0.052	0.95881
Should.OrX:Haft.ShI	NA	NA	NA	NA
Should.OrH:Haft.ShR	NA	NA	NA	NA
Should.OrT:Haft.ShR	NA	NA	NA	NA
Should.OrX:Haft.ShR	NA	NA	NA	NA
Should.OrH:Haft.ShS	NA	NA	NA	NA
Should.OrT:Haft.ShS	NA	NA	NA	NA
Should.OrX:Haft.ShS	NA	NA	NA	NA

lm19 Find a connection of Weight & Haft.Sh.R

lm19 Build a new model, contain Length, Width & Haft.Sh.R

```
> model19<-lm(Weight~Length+Width+Haft.Sh.R,data=DP)
> summary(model19)
```

Call:

```
lm(formula = Weight ~ Length + Width + Haft.Sh.R, data = DP)
```

Residuals:

Min	1Q	Median	3Q	Max
-8.9627	-0.7383	0.0987	0.8308	5.2227

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-8.60173	0.83545	-10.296	< 2e-16 ***
Length	0.19151	0.02290	8.365	1.04e-12 ***
Width	0.30747	0.05678	5.415	5.60e-07 ***
Haft.Sh.R	3.06253	1.77023	1.730	0.0873 .

lm19 With summary(model19), we can add Haft.Sh.R become a new criterion

lm20 Build model which compare 2 factors (Should.Or&Haft.Or)

```
> lm20<-lm(Weight~Length+Width+Should.Or*Haft.Or,data=DP)
> summary(lm20)
```

Call:

```
lm(formula = Weight ~ Length + Width + Should.Or * Haft.Or, data = DP)
```

Residuals:

Min	1Q	Median	3Q	Max
-4.1467	-0.6908	-0.0580	0.5928	4.8005

Coefficients: (7 not defined because of singularities)

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-10.56220	2.34659	-4.501	2.47e-05 ***
Length	0.18083	0.02136	8.465	1.71e-12 ***
Width	0.38513	0.05605	6.871	1.71e-09 ***
Should.OrH	-3.82638	2.16995	-1.763	0.0820 .
Should.OrT	0.66082	1.40361	0.471	0.6392
Should.OrX	0.95460	1.92342	0.496	0.6211
Haft.OrE	-0.44499	2.23492	-0.199	0.8427
Haft.OrP	0.38133	1.88043	0.203	0.8399
Haft.OrT	0.47782	0.74711	0.640	0.5244
Haft.OrV	-0.00630	0.97835	-0.006	0.9949
Should.OrH:Haft.OrE	3.98853	2.72648	1.463	0.1477
Should.OrT:Haft.OrE	0.72792	2.06839	0.352	0.7259
Should.OrX:Haft.OrE	NA	NA	NA	NA
Should.OrH:Haft.OrP	5.30981	2.75227	1.929	0.0575 .
Should.OrT:Haft.OrP	0.33836	2.00100	0.169	0.8662
Should.OrX:Haft.OrP	NA	NA	NA	NA
Should.OrH:Haft.OrT	-1.50219	2.05632	-0.731	0.4674
Should.OrT:Haft.OrT	NA	NA	NA	NA
Should.OrX:Haft.OrT	NA	NA	NA	NA
Should.OrH:Haft.OrV	NA	NA	NA	NA
Should.OrT:Haft.OrV	NA	NA	NA	NA
Should.OrX:Haft.OrV	NA	NA	NA	NA

lm20 Find a connection of Weight, Should.Or.H & Should.Or.H:Haft.Or.P

lm20 Build a new model, contain Length, Width & Should.Or.H*Haft.Or.P

```
> model20<-lm(Weight~Length+Width+Should.Or.H*Haft.Or.P,data=DP)
> summary(model20)
```

Call:

```
lm(formula = Weight ~ Length + Width + Should.Or.H * Haft.Or.P,
    data = DP)
```

Residuals:

Min	1Q	Median	3Q	Max
-6.5085	-0.6523	-0.0487	0.8619	5.1334

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-8.69522	0.76983	-11.295	< 2e-16 ***
Length	0.18597	0.02050	9.071	4.71e-14 ***
Width	0.32903	0.05202	6.325	1.21e-08 ***
Should.Or.H	-2.68175	0.59522	-4.505	2.15e-05 ***
Haft.Or.P	0.45918	0.38290	1.199	0.23385
Should.Or.H:Haft.Or.P	3.51443	1.13922	3.085	0.00277 **

lm20 There is no strong connection between Weight & Haft.Or.P

lm20 Remove Haft.Or.P

```
> model20<-lm(Weight~Length+Width+Should.Or.H+Should.Or.H:Haft.Or.P,data=DP)
> summary(model20)
```

Call:

```
lm(formula = Weight ~ Length + Width + Should.Or.H + Should.Or.H:Haft.Or.P,
    data = DP)
```

Residuals:

Min	1Q	Median	3Q	Max
-6.4935	-0.7205	0.0532	0.8183	4.9970

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-8.51530	0.75704	-11.248	< 2e-16 ***
Length	0.18604	0.02056	9.051	4.69e-14 ***
Width	0.32709	0.05213	6.275	1.46e-08 ***
Should.Or.H	-2.81988	0.58549	-4.816	6.40e-06 ***
Should.Or.H:Haft.Or.P	3.98310	1.07288	3.713	0.000368 ***

lm20 With summary(model20), we can add Should.Or.H & Should.Or.H:Haft.Or.P become new criteria

lm21 Build model which compare 2 factors (Haft.Sh&Haft.Or)

```
> lm21<-lm(Weight~Length+Width+Haft.Sh*Haft.Or,data=DP)
> summary(lm21)

Call:
lm(formula = Weight ~ Length + Width + Haft.Sh * Haft.Or, data = DP)

Residuals:
    Min       1Q   Median       3Q      Max
-8.7222 -0.5385  0.0720  0.6051  5.5541

Coefficients: (15 not defined because of singularities)
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   -9.62131    1.85249   -5.194 1.64e-06 ***
Length         0.19154    0.02378    8.055 7.95e-12 ***
Width         0.32062    0.06092    5.263 1.24e-06 ***
Haft.ShE      1.43365    1.90504    0.753  0.454
Haft.ShI      0.71135    1.35111    0.526  0.600
Haft.ShR      3.04028    2.26306    1.343  0.183
Haft.ShS      0.61486    1.36666    0.450  0.654
Haft.OrE     -0.15066    0.79610   -0.189  0.850
Haft.OrP      0.68263    0.98148    0.696  0.489
Haft.OrT     -0.25292    1.04794   -0.241  0.810
Haft.OrV     -1.56448    1.81766   -0.861  0.392
Haft.ShE:Haft.OrE  0.49115    1.88433    0.261  0.795
Haft.ShI:Haft.OrE      NA         NA      NA      NA
Haft.ShR:Haft.OrE      NA         NA      NA      NA
Haft.ShS:Haft.OrE      NA         NA      NA      NA
Haft.ShE:Haft.OrP      NA         NA      NA      NA
Haft.ShI:Haft.OrP      NA         NA      NA      NA
Haft.ShR:Haft.OrP      NA         NA      NA      NA
Haft.ShS:Haft.OrP      NA         NA      NA      NA
Haft.ShE:Haft.OrT      NA         NA      NA      NA
Haft.ShI:Haft.OrT      NA         NA      NA      NA
Haft.ShR:Haft.OrT      NA         NA      NA      NA
Haft.ShS:Haft.OrT      NA         NA      NA      NA
Haft.ShE:Haft.OrV      NA         NA      NA      NA
Haft.ShI:Haft.OrV      NA         NA      NA      NA
```

lm21 Unfortunately, there are 0 criteria have strong relationship with Weigh

Finally, use the criteria above, to create model

model.1

```
> #build final model
> model.1<-lm(Weight~Length+Width
+             +Name.T+Should.Sh.I+Should.Or.H+Haft.Sh.R+Haft.Or.P+Haft.Or.T
+
+Name.W:Haft.Or.P+Blade.Sh.R:Should.Sh.I+Blade.Sh.R:Haft.Sh.E+Should.Or.H:Haft.Or.P
+Haft.Or.T:Base.Sh.I+Haft.Or.T:Base.Sh.S
+             +Blade.Sh.R*Haft.Or.P)
> summary(model.1)

Call:
lm(formula = Weight ~ Length + Width + Name.T + Should.Sh.I +
    Should.Or.H + Haft.Sh.R + Haft.Or.P + Haft.Or.T + Name.W:Haft.Or.P +
    Blade.Sh.R:Should.Sh.I + Blade.Sh.R:Haft.Sh.E + Should.Or.H:Haft.Or.P +
    Haft.Or.T:Base.Sh.I + Haft.Or.T:Base.Sh.S + Blade.Sh.R *
    Haft.Or.P)

Residuals:
    Min       1Q   Median       3Q      Max
-6.4616 -0.5425  0.0348  0.7564  3.5410
```

```

Coefficients: (1 not defined because of singularities)
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    -8.86077    0.73690  -12.024  < 2e-16 ***
Length          0.16642    0.01944   8.561 1.24e-12 ***
Width           0.37992    0.04964   7.654 6.29e-11 ***
Name.T          1.01014    0.48943   2.064 0.042581 *
Should.Sh.I     -0.22490    0.37295  -0.603 0.548363
Should.Or.H     -2.16468    0.60120  -3.601 0.000576 ***
Haft.Sh.R       2.44096    1.74720   1.397 0.166625
Haft.Or.P       0.39026    0.38458   1.015 0.313567
Haft.Or.T       5.27533    1.44546   3.650 0.000490 ***
Blade.Sh.R      2.29092    1.46077   1.568 0.121136
Haft.Or.P:Name.W -1.97475    1.03854  -1.901 0.061188 .
Should.Sh.I:Blade.Sh.R -2.53950    2.16155  -1.175 0.243872
Blade.Sh.R:Haft.Sh.E -1.88635    2.19364  -0.860 0.392651

Should.Or.H:Haft.Or.P 2.29256    1.19339   1.921 0.058630 .
Haft.Or.T:Base.Sh.I  -5.78645    1.47201  -3.931 0.000191 ***
Haft.Or.T:Base.Sh.S  -5.39048    1.51220  -3.565 0.000647 ***
Haft.Or.P:Blade.Sh.R      NA         NA         NA         NA

```

Because some criteria do not have *, delete them

model.2

```

> model.2<-lm(Weight~Length+Width
+             +Name.T+Should.Or.H+Haft.Or.T
+             +Name.W:Haft.Or.P+Should.Or.H:Haft.Or.P+Haft.Or.T:Base.Sh.I+Haft.Or.T:Base.Sh.S)
> summary(model.2)

```

Call:

```
lm(formula = Weight ~ Length + Width + Name.T + Should.Or.H +
    Haft.Or.T + Name.W:Haft.Or.P + Should.Or.H:Haft.Or.P + Haft.Or.T:Base.Sh.I +
    Haft.Or.T:Base.Sh.S)

```

Residuals:

```

      Min       1Q   Median       3Q      Max
-6.0688 -0.6904  0.0557  0.8960  3.7421

```

Coefficients:

```

              Estimate Std. Error t value Pr(>|t|)
(Intercept)    -8.78688    0.72833  -12.064  < 2e-16 ***
Length          0.17131    0.01943   8.816 2.23e-13 ***
Width           0.36899    0.04976   7.415 1.19e-10 ***
Name.T          1.04469    0.48703   2.145 0.035023 *
Should.Or.H     -2.69350    0.53560  -5.029 3.02e-06 ***
Haft.Or.T       5.16301    1.45045   3.560 0.000632 ***
Name.W:Haft.Or.P -1.67170    1.02453  -1.632 0.106726
Should.Or.H:Haft.Or.P 3.81902    0.98889   3.862 0.000229 ***

Haft.Or.T:Base.Sh.I  -5.80920    1.49681  -3.881 0.000214 ***
Haft.Or.T:Base.Sh.S  -5.33569    1.54420  -3.455 0.000887 ***

```

Because some criteria do not have *, delete them

model.3

```
> model.3<-lm(Weight~Length+Width  
+             +Name.T+Should.Or.H+Haft.Or.T  
+             +Should.Or.H:Haft.Or.P+Haft.Or.T:Base.Sh.I+Haft.Or.T:Base.Sh.S)  
> summary(model.3)
```

Call:

```
lm(formula = Weight ~ Length + Width + Name.T + Should.Or.H +  
    Haft.Or.T + Should.Or.H:Haft.Or.P + Haft.Or.T:Base.Sh.I +  
    Haft.Or.T:Base.Sh.S)
```

Residuals:

Min	1Q	Median	3Q	Max
-6.1221	-0.6605	0.0655	0.8912	3.7615

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-8.86847	0.73412	-12.080	< 2e-16 ***
Length	0.16943	0.01960	8.645	4.38e-13 ***
Width	0.37406	0.05018	7.455	9.39e-11 ***
Name.T	1.11956	0.48987	2.285	0.024936 *
Should.Or.H	-2.65017	0.54047	-4.903	4.85e-06 ***
Haft.Or.T	5.23819	1.46470	3.576	0.000595 ***
Should.Or.H:Haft.Or.P	3.82595	0.99910	3.829	0.000254 ***
Haft.Or.T:Base.Sh.I	-5.83067	1.51222	-3.856	0.000232 ***
Haft.Or.T:Base.Sh.S	-5.35811	1.56010	-3.434	0.000944 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.433 on 80 degrees of freedom

(2 observations deleted due to missingness)

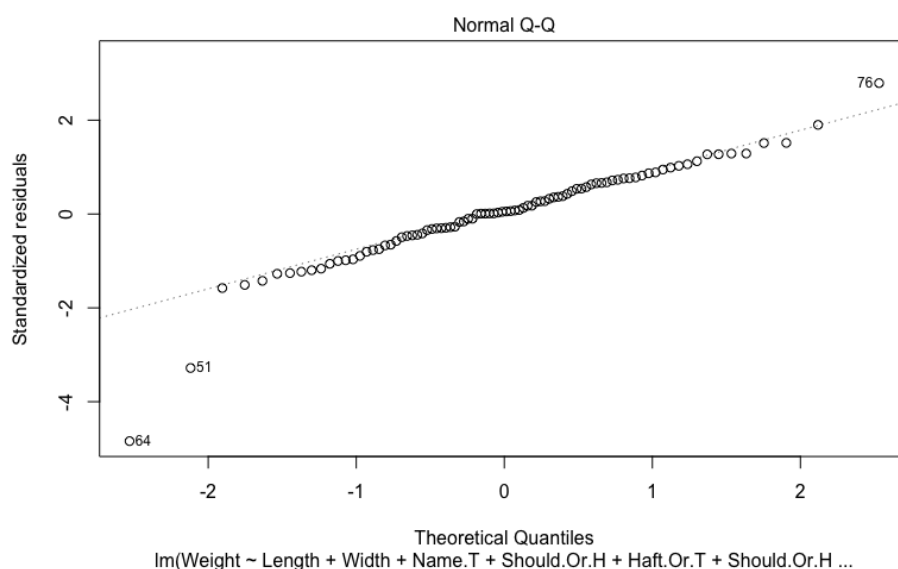
Multiple R-squared: 0.8961, Adjusted R-squared: 0.8857

F-statistic: 86.25 on 8 and 80 DF, p-value: < 2.2e-16

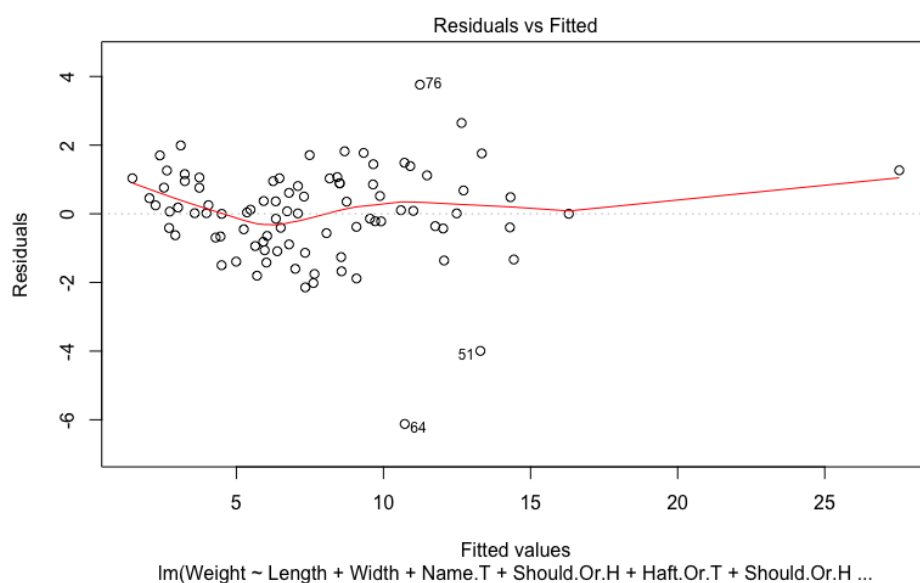
(f) Check and describe the fit of your model using whatever graphical or numerical methods seem appropriate.

```
> #(f)
> plot(model.3)
```

According to Q-Q plot, this model is normal distribution



This is residual plot could display x-axis (model.3) and y-axis (residuals) variables, it looks like data concentrate on left-hand side. It means this model has some errors should be corrected.



(g) Interpret the fitted model in practical terms. What does it tell you about predicting the dart weight?

At first, we should use Length and Width variables, multiple their own β (0.15687 & 0.41348)

Then, find the Name of new data, if Name is Trivas, then Name.T is 1, it times 1.39885

Check other criteria, if they match the category we choose, then times their β and sum them.

After them we can get Weight.

Function:

```
Weight= -8.8684728 + 0.1694280 * Length + 0.3740638 * Width + 1.1195626 * Name.T - 2.6501723  
        * Should.Or.H + 5.2381919 * Haft.Or.T + 3.8259520 * Should.Or.H:Haft.Or.P -5.8306698 *  
        Haft.Or.T:Base.Sh.I - 5.3581054 * Haft.Or.T:Base.Sh.S
```

(h) Predict the expected dart weight for a dart point of type Travis, with maximum length 70 mm, H.Length 60mm, Thickness 50 mm, B.Width 50 mm, J.Width 50 mm, Width 60 mm and with both blade shape and base shape recurvate, straight shoulder shape, barbed shoulder orientation, excurve shape for the lateral haft element and parallel orientation of the lateral haft element. Give a 95% confidence interval for this expected weight. Is there any reason to be cautious about your estimate?

First, put the given data into our variables, especially class variables, if the variable in model is given then that class variable is 1

```
> #(h) predict weight within 0.95 confidence interval  
> newdata<-data.frame(Length=70,Width=60,Name.T=1,Should.Or.H=0,Haft.Or.T=0,  
Haft.Or.P=1,Haft.Or.T=0,Base.Sh.I=0,Base.Sh.S=0)  
> predict(model.3,newdata,df=80,interval = "confidence",level=0.95)  
      fit      lwr      upr  
1 26.55488 23.00244 30.10731
```

Using predict() function, we can get the prediction of Weight, and the confidence interval is (23.00244, 30.10731)