STAT343 - Project

Hye Woong Jeon, Rohan Kapoor

In this project, we will analyze a dataset on Obsidian rocks, and try to build a working linear model for predicting the mass of a rock made of obsidian.

Step 0: Importing the data and looking at it, trying to get a feel for it.

```
data <- read.table("data/obsidian_data.txt", header = TRUE, sep = ",")
```

```
head(data, n=10)
```

```
##              ID  mass           type     site element_Rb element_Sr element_Y
## 1   288275.002a 0.502          Blade Ali Kosh        238         45        29
## 2  288275.002aa 0.227          Flake Ali Kosh        234         44        28
## 3  288275.002ab 0.188          Flake Ali Kosh        255         50        32
## 4  288275.002ac 0.153          Flake Ali Kosh        231         46        28
## 5  288275.002ad 0.102          Blade Ali Kosh        252         49        31
## 6  288275.002ae 0.440          Flake Ali Kosh        234         44        28
## 7  288275.002af 0.656          Blade Ali Kosh        226         44        28
## 8  288275.002ag 0.484          Flake Ali Kosh        230         45        29
## 9  288275.002ah 0.579          Blade Ali Kosh        230         44        28
## 10 288275.002ai 0.713 Core fragment? Ali Kosh        236         45        28
##    element_Zr
## 1         334
## 2         325
## 3         337
## 4         327
## 5         331
## 6         327
## 7         323
## 8         330
## 9         328
## 10        331
```

Data looks like it made it into R okay, so we can start analyzing it.

Step 1: Data Exploration, cleaning, dealing with missing data.

```
summary(data)
```

```
##       ID                 mass              type               site
##  Length:652         Min.   :  0.0320   Length:652         Length:652
##  Class :character   1st Qu.:  0.2125   Class :character   Class :character
```

```
##  Mode  :character   Median :  0.4190   Mode  :character   Mode  :character
##                     Mean   :  0.8777
##                     3rd Qu.:  0.6925
##                     Max.   :160.0000
##                     NA's   :1
##    element_Rb      element_Sr      element_Y       element_Zr
##  Min.   :206.0   Min.   :10.00   Min.   :22.00   Min.   : 65.0
##  1st Qu.:231.0   1st Qu.:45.00   1st Qu.:28.00   1st Qu.:326.0
##  Median :240.0   Median :47.00   Median :29.00   Median :332.0
##  Mean   :241.2   Mean   :46.95   Mean   :29.45   Mean   :331.9
##  3rd Qu.:250.0   3rd Qu.:49.00   3rd Qu.:30.00   3rd Qu.:338.2
##  Max.   :291.0   Max.   :65.00   Max.   :62.00   Max.   :365.0
##
```

Already, we spot some interesting features: we see a repeated ID, making me suspect an object has been logged twice. There seems to be a missing mass value, as well a terribly wrong outlier on the high side. A few missing and a few uncertain types. An ambigious site which we should probably predict. Element Rb and Element Sr look fine, but Element Y seems to have an outlier on the high side, and Element Zr has a low side outlier. Let's look at these one by one.

```
data[which(data$ID == "288275.002bh"), ]
```

```
##              ID  mass  type     site element_Rb element_Sr element_Y element_Zr
## 32 288275.002bh 0.215 Blade Ali Kosh        252         49        32        339
## 33 288275.002bh 0.215 Blade Ali Kosh        254         48        31        339
```

This just looks like a double-logged entry, so I will simply delete it.

```
data <- data[-33,]
#commenting out so I do not run it again, but I ran it once.
```

Now let us look at mass. I spot a few ourliers, so I will try to look at those. The 160 value is an order of magntude above anything else, so I just get rid of it, since I cannot fill in the value in any way.

```
data[which(data$mass >= 10), ]
```

```
##          ID mass  type         site element_Rb element_Sr element_Y element_Zr
## 465 297032q  160 Flake Chagha Sefid        214         41        27        312
```

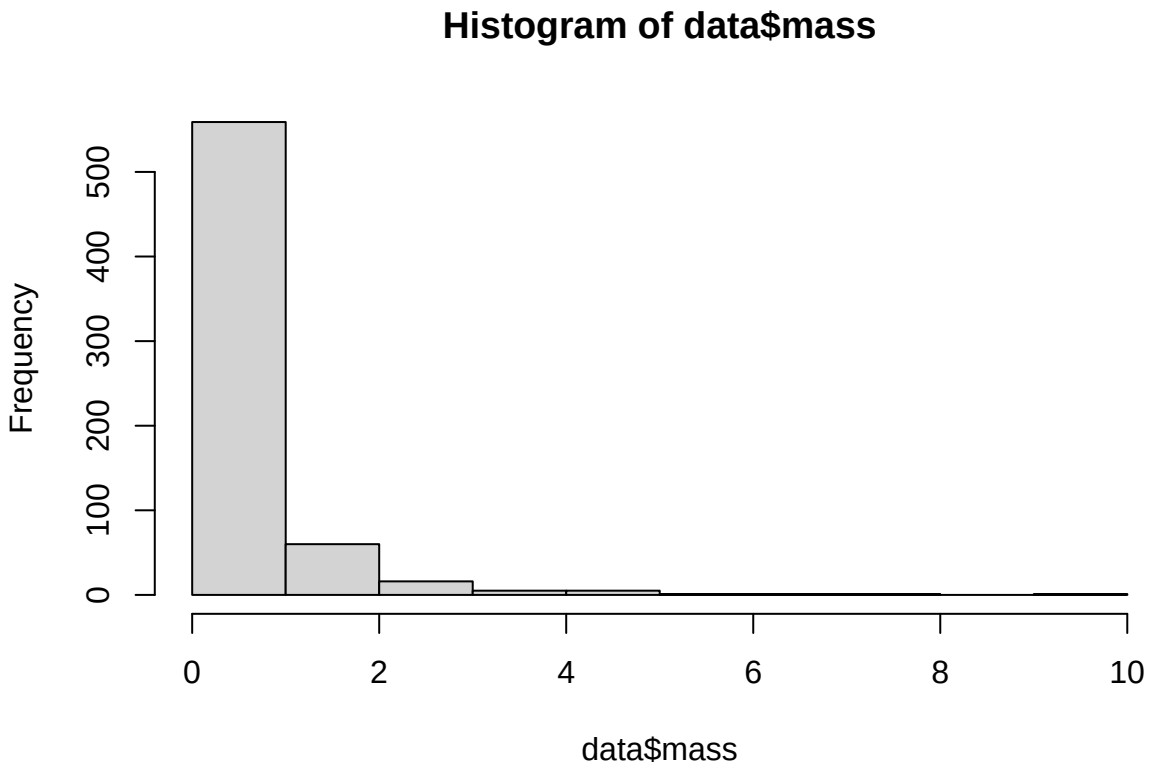
```
#data[which(data$mass == NA), ] #no null values returned.
```

```
data <- data[-464,]
#commenting out so I do not run it again, but I ran it once.
```

I also get rid of the NA value for mass, since I cannot impute for the regression output anyway

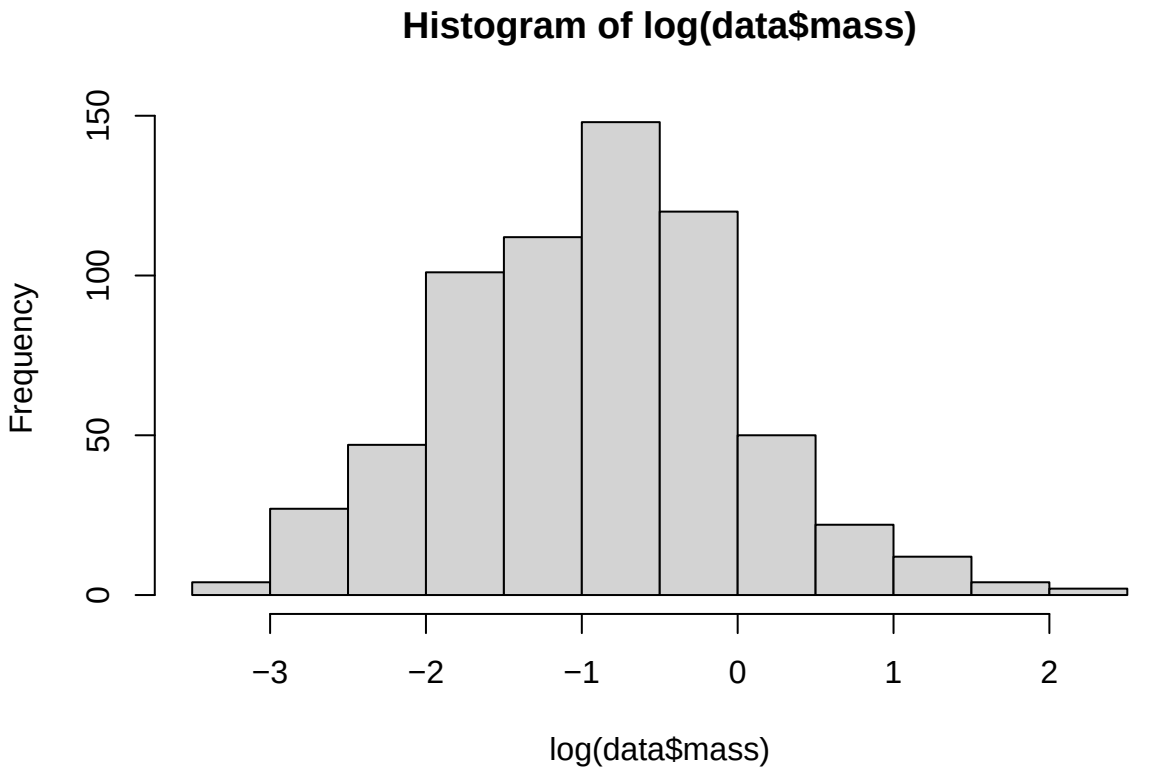
Now I plot the histogram of masses to see what kind of distribution it follows.

```
hist(data$mass)
```



Clearly, this does not seem normal It might be worth putting some sort of transformation onto it: probably transforming it on a log scale, or other variable. We will see about this later, but take a note of this.

```
hist(log(data$mass))
```



This looks pretty good so let's do it

```
data$mass <- log(data$mass)
```

We should combine some of the type variables: blade and blades, etc. I feel pretty comfortable doing this, since all the errors seem to be for similar objects not and just logged differently by one person. Even if it is not perfect, it seems necessary to do since we cannot deal with that large a number of different types and simplifying to 2-3 kinds of terms helps us save degrees of freedom for other considerations later. I first considered Retouched Blades being a different category to blades, but there are only 3 data points, which means even if they are differnet, they won't contribute much to a differnt effect, so I should just combine with Blade. Same with Used Flake to Flake.

```
levels(data$type)
```

```
## NULL
```

```
data$type[data$type == "Blades"] <- "Blade"
data$type[data$type == "blade"] <- "Blade"
data$type[data$type == "Distal end of prismatic blade?"] <- "Blade"
data$type[data$type ==  "Blade (Flake?)"] <- "Blade"


data$type[data$type == "flake"] <- "Flake"
data$type[data$type == "Flakes"] <- "Flake"
data$type[data$type == "Flake (listed as)"] <- "Flake"

data$type[data$type == "core"] <- "Core"
data$type[data$type == "Cores and frags"] <- "Core"
data$type[data$type == "Core/Fragment"] <- "Core"
data$type[data$type == "Core fragment"] <- "Core"
data$type[data$type == "Core fragment?"] <- "Core"
data$type[data$type == "Cores and fragments"] <- "Core"
data$type[data$type ==  "Fragment (from core?)"] <- "Core"

data$type[data$type == "Retouched blades"] <- "Retouched Blade"
data$type[data$type == "Retouched Blades"] <- "Retouched Blade"

data$type[data$type == "Retouched Blade"] <- "Blade"
data$type[data$type == "Used flake"] <- "Flake"

data$type[data$type == "Core fragment? Flake?"] <- "Flake/Core"
```

```
summary(data$type)
```

```
##    Length     Class      Mode
##       650 character character
```

Also, we drop the NA entry in mass or type

```
data <- data[complete.cases(data[, c('mass', 'type')]), ]
```

Now for the two site outliers.

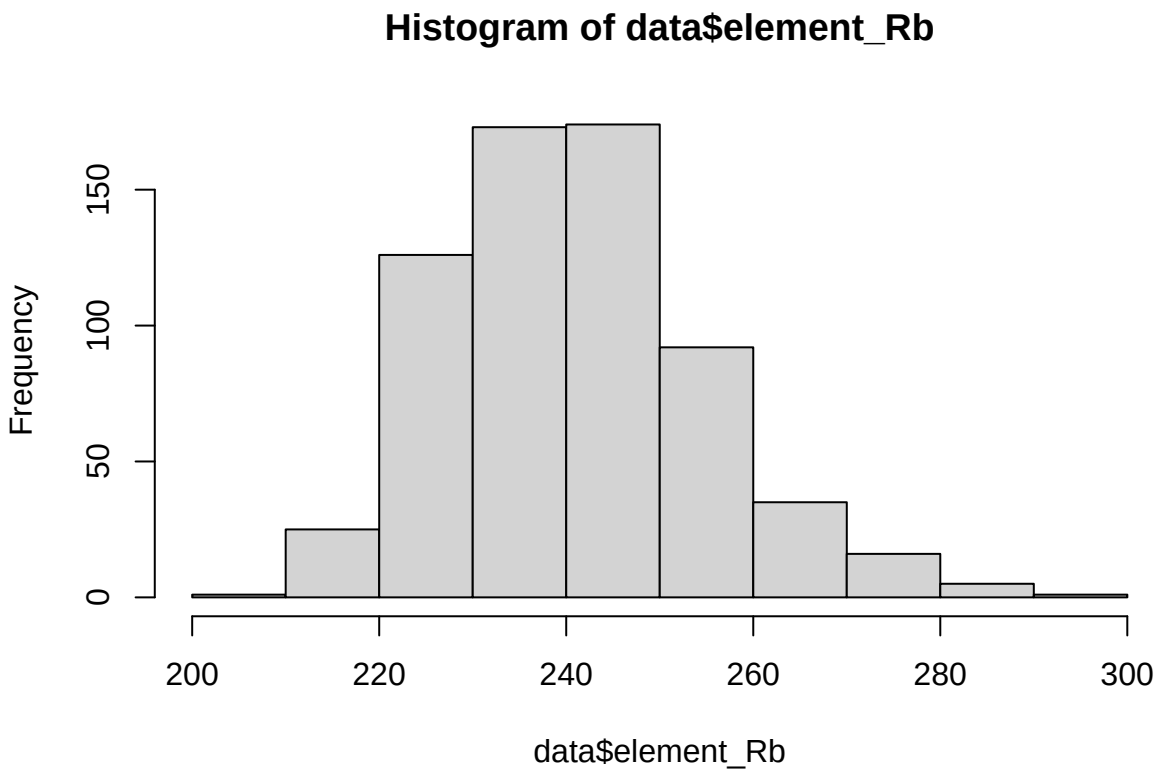
```
data[(data$site == "Ali Kosh/Chaga Sefid" | data$site == "Hulailan Tepe Guran"), ]
```

```
##          ID      mass  type                 site element_Rb element_Sr
## 215 288285h -2.292635 Blade Ali Kosh/Chaga Sefid        283         56
## 229 293319a -1.258781 Blade  Hulailan Tepe Guran        255         50
##     element_Y element_Zr
## 215        31        365
## 229        32        343
```

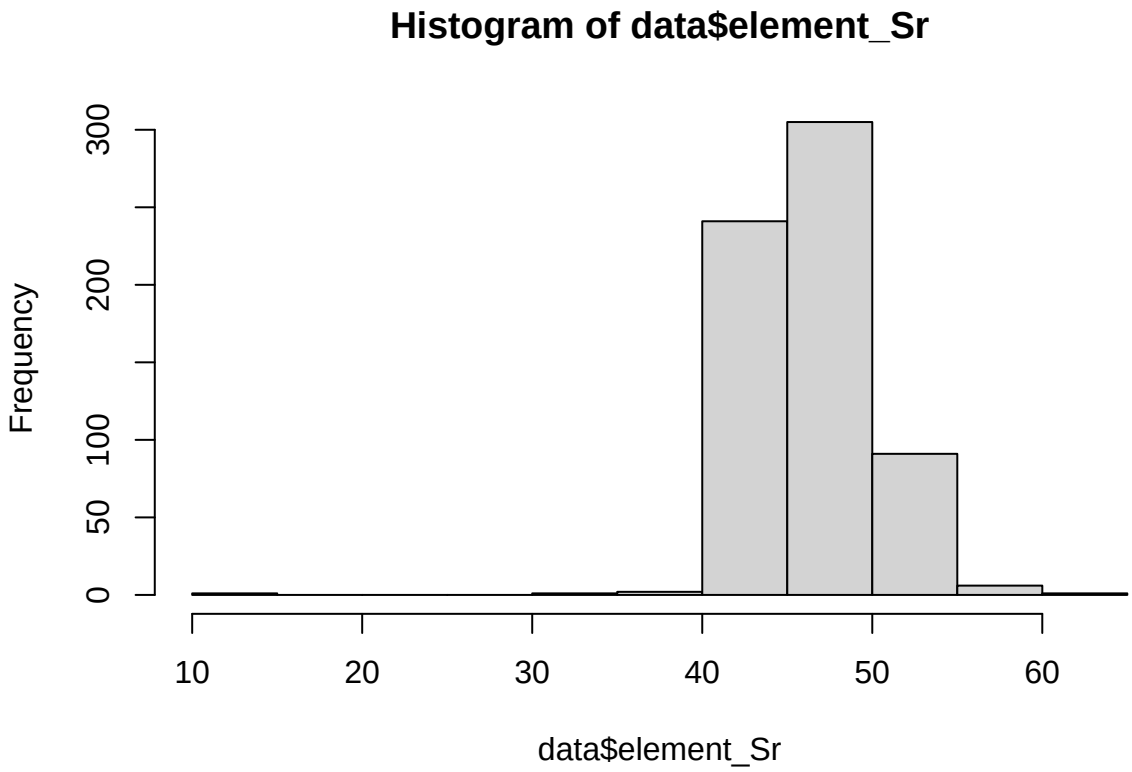
For the first one, we know that we just need to pick Ali Kosh/Chaga Sefid as its location, which we will do by imputing by mean. For the latter, we can either get rid of it and restrict our model to two sites, or try to learn which site looks more like Hulailan Tepe Guran. I will opt to do the latter.

Now I am just going to plot the histograms of the 4 elements and see what the distribution looks like.

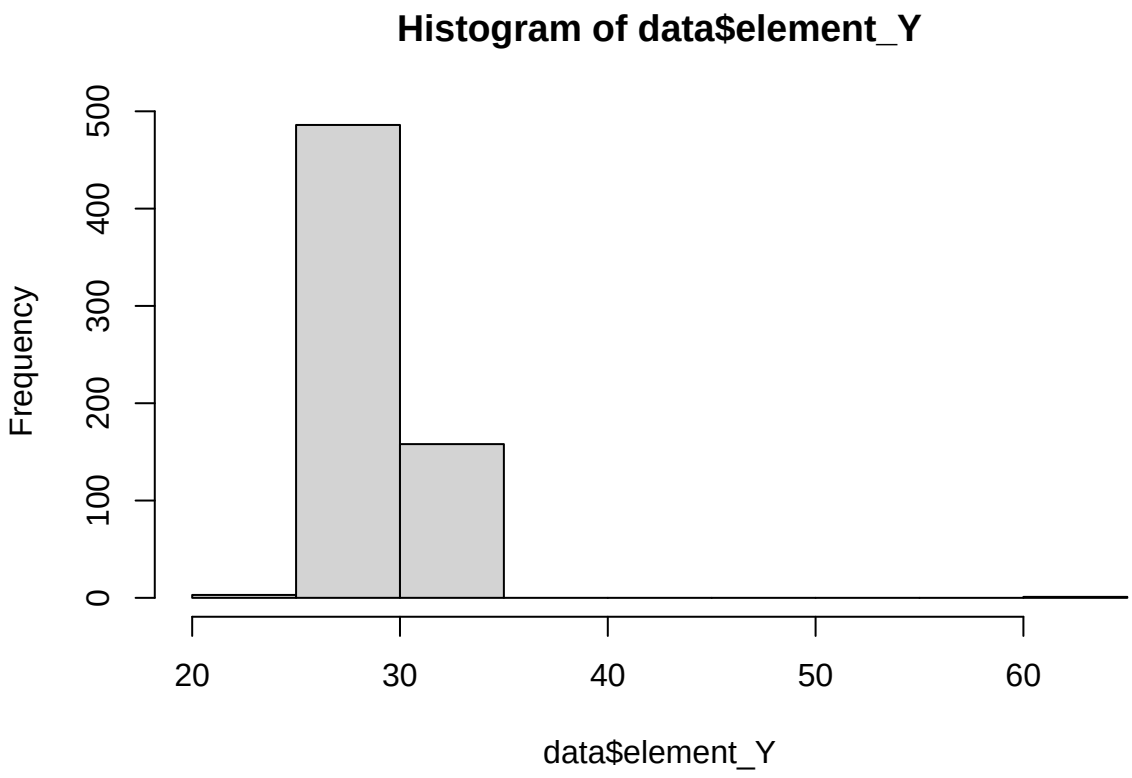
```
hist(data$element_Rb)
```



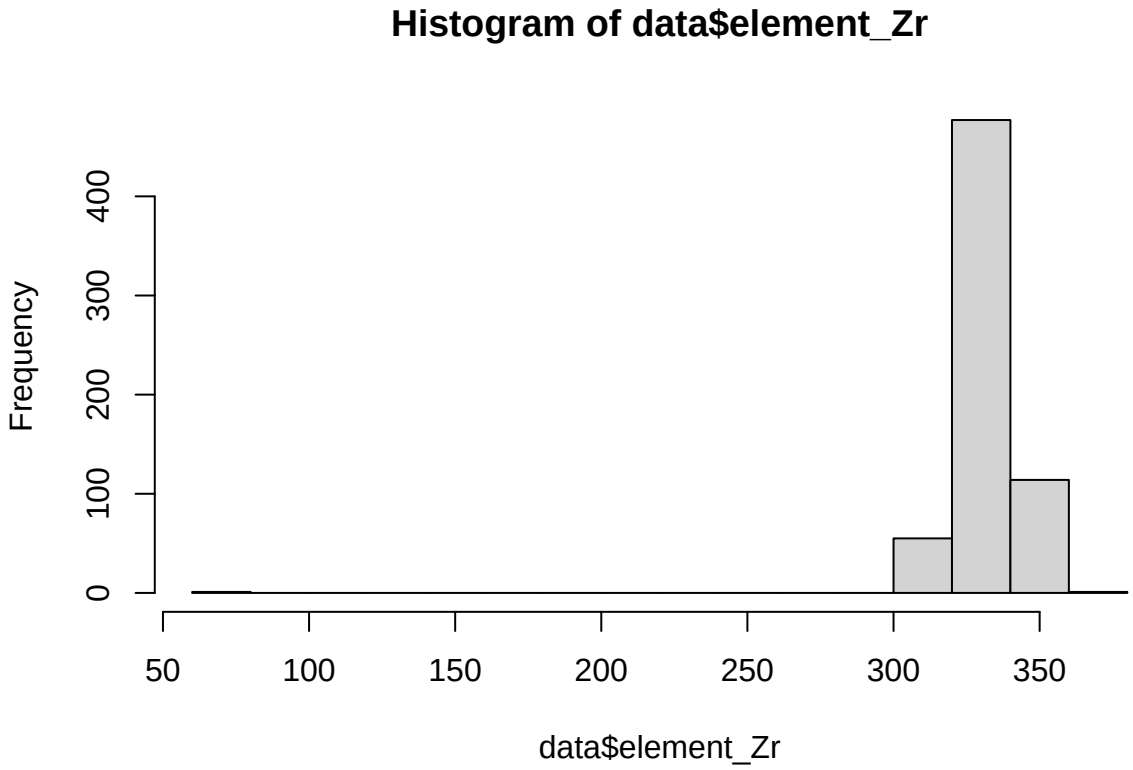

```
hist(data$element_Sr)
```




```
hist(data$element_Y)
```




```
hist(data$element_Zr)
```



Rb looks fine, but I think the others have outliers we can get rid of, which are probably just mis-entered data.

```
data[which(data$element_Zr<100 | data$element_Y>50 | data$element_Sr<20), ]
```

```
##          ID       mass  type         site element_Rb element_Sr element_Y
## 628 297078L -2.6036902 Blade Chagha Sefid        234         35        62
## 652 297110b -0.7571525 Blade Chagha Sefid        215         10        23
##     element_Zr
## 628        303
## 652         65
```

I will just delete these two

```
data <- data[-which(data$element_Zr<100 | data$element_Y>50 | data$element_Sr<20), ]
```

```
summary(data)
```

```
##       ID                  mass             type               site
##  Length:646         Min.   :-3.4420   Length:646         Length:646
##  Class :character   1st Qu.:-1.5500   Class :character   Class :character
##  Mode  :character   Median :-0.8651   Mode  :character   Mode  :character
##                     Mean   :-0.9077
##                     3rd Qu.:-0.3671
##                     Max.   : 2.2379
##    element_Rb      element_Sr      element_Y       element_Zr
##  Min.   :206.0   Min.   :39.00   Min.   :22.00   Min.   :307.0
##  1st Qu.:231.0   1st Qu.:45.00   1st Qu.:28.00   1st Qu.:326.0
```

```
##  Median :240.5   Median :47.00   Median :29.00   Median :332.0
##  Mean   :241.3   Mean   :47.04   Mean   :29.41   Mean   :332.4
##  3rd Qu.:250.0   3rd Qu.:49.00   3rd Qu.:30.00   3rd Qu.:338.8
##  Max.   :291.0   Max.   :65.00   Max.   :34.00   Max.   :365.0
```

The data looks clean-ish now.

So we move onto Step 3, inserting missing data or uncertain data. We have some NAs to fill in, as well as some uncertain types and sites which we will impute by mean.

For the sites, we see that the uncertain objects are both blades, so compare their masses to the masses of the blades found at the two common sites.

```
mean(data[which(data$site == "Ali Kosh" & data$type == "Blade"), ]$mass)
```

```
## [1] -1.436775
```

```
mean(data[which(data$site == "Chagha Sefid" & data$type == "Blade"), ]$mass)
```

```
## [1] -0.7985774
```

Both of the two uncertain sites seem closer to the mean of Ali Kosh, so I will reassign them there.

```
data$site[data$site == "Ali Kosh/Chaga Sefid" | data$site == "Hulailan Tepe Guran"] <- "Ali Kos
```

Now, we do the same for the uncertain types.

```
mean(data[which(data$type == "Blade"), ]$mass)
```

```
## [1] -1.051365
```

```
mean(data[which(data$type == "Flake"), ]$mass)
```

```
## [1] -0.8208057
```

```
mean(data[which(data$type == "Core"), ]$mass)
```

```
## [1] 0.5643912
```

```
data[which(data$type != "Blade" & data$type != "Flake" & data$type != "Core" ), ]
```

```
##              ID       mass        type     site element_Rb element_Sr element_Y
## 60  288275.002i -1.3318062  Flake/Core Ali Kosh        227         46        27
## 77      288276c -0.5621189 Blade/Flake Ali Kosh        245         47        29
## 78      288276e -0.1554849 Blade/Flake Ali Kosh        234         43        28
## 79      288276f -1.4229583 Blade/Flake Ali Kosh        249         47        30
## 212    288284oL  0.8135933  Flake/Core Ali Kosh        236         45        29
##     element_Zr
## 60         326
## 77         340
## 78         331
## 79         339
## 212        329
```

Manually assign them to the one their mean is closer to in the two choices.

```
data$type[data$ID == "288275.002i"] <- "Flake"
data$type[data$ID == "288276c"] <- "Flake"
data$type[data$ID == "288276e"] <- "Flake"
data$type[data$ID == "288276f"] <- "Blade"
data$type[data$ID == "288284oL"] <- "Core"
```

With our missing/uncertain values imputed, let us look at the data for one last time.

```
summary(data)
```

```
##       ID                  mass              type               site
##  Length:646         Min.   :-3.4420   Length:646         Length:646
##  Class :character   1st Qu.:-1.5500   Class :character   Class :character
##  Mode  :character   Median :-0.8651   Mode  :character   Mode  :character
##                     Mean   :-0.9077
##                     3rd Qu.:-0.3671
##                     Max.   : 2.2379
##    element_Rb      element_Sr      element_Y       element_Zr
##  Min.   :206.0   Min.   :39.00   Min.   :22.00   Min.   :307.0
##  1st Qu.:231.0   1st Qu.:45.00   1st Qu.:28.00   1st Qu.:326.0
##  Median :240.5   Median :47.00   Median :29.00   Median :332.0
##  Mean   :241.3   Mean   :47.04   Mean   :29.41   Mean   :332.4
##  3rd Qu.:250.0   3rd Qu.:49.00   3rd Qu.:30.00   3rd Qu.:338.8
##  Max.   :291.0   Max.   :65.00   Max.   :34.00   Max.   :365.0
```

Looks good!

Step 4: Model Selection

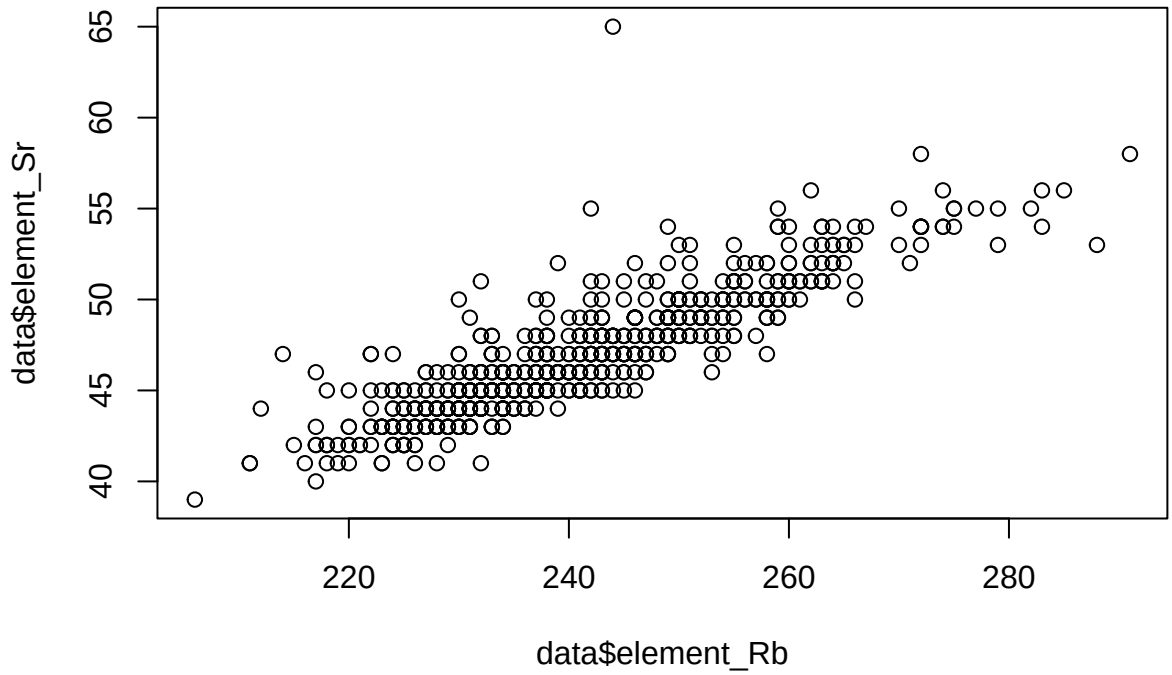
```
model = lm(formula = mass ~ type*site*(element_Sr+element_Y+element_Rb + element_Zr)**2, data =
anova(model)
```

```
## Analysis of Variance Table
##
## Response: mass
##                                  Df  Sum Sq Mean Sq  F value    Pr(>F)
## type                              2  64.951  32.476 108.7799 < 2.2e-16 ***
## site                              1  57.887  57.887 193.8982 < 2.2e-16 ***
## element_Sr                        1 161.030 161.030 539.3838 < 2.2e-16 ***
## element_Y                         1   0.142   0.142   0.4758  0.490598
## element_Rb                        1  27.363  27.363  91.6535 < 2.2e-16 ***
## element_Zr                        1  18.539  18.539  62.0973 1.599e-14 ***
## type:site                         2   2.359   1.179   3.9503  0.019763 *
## element_Sr:element_Y              1   0.200   0.200   0.6707  0.413151
## element_Sr:element_Rb             1   5.406   5.406  18.1080 2.430e-05 ***
## element_Sr:element_Zr             1   0.289   0.289   0.9670  0.325824
## element_Y:element_Rb              1   6.076   6.076  20.3510 7.791e-06 ***
## element_Y:element_Zr              1   0.003   0.003   0.0086  0.925973
## element_Rb:element_Zr             1   0.221   0.221   0.7408  0.389756
## type:element_Sr                   2   0.212   0.106   0.3555  0.700985
```

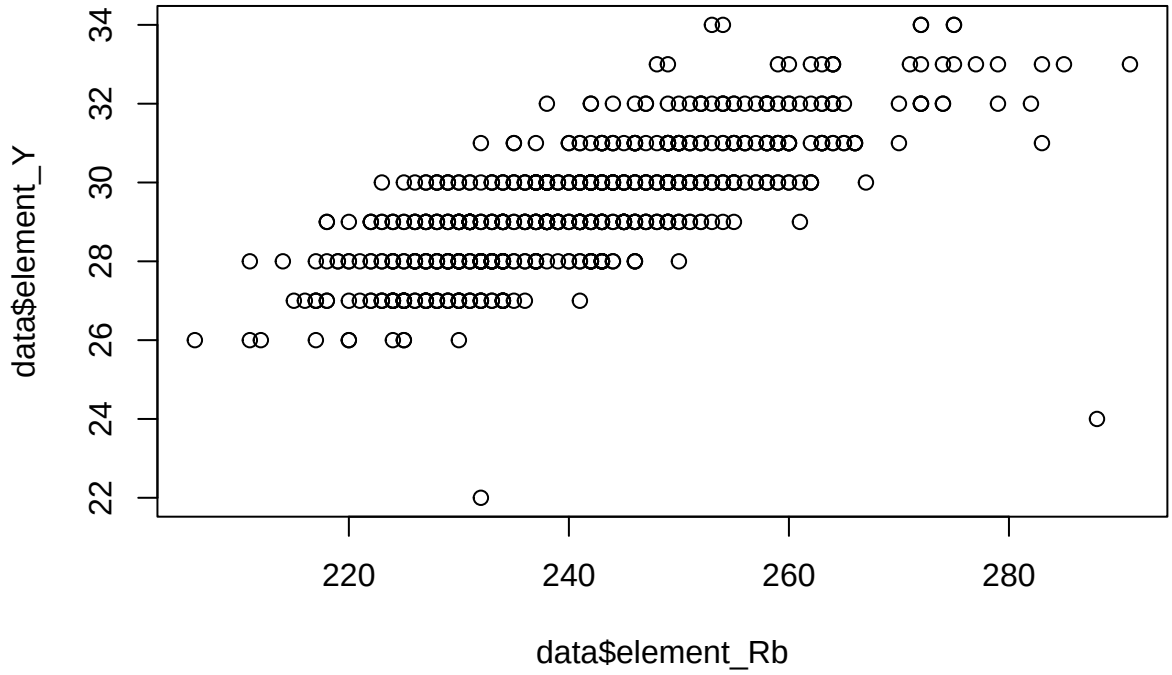
```
## type:element_Y                    2   3.990   1.995   6.6820  0.001351 **
## type:element_Rb                   2   0.280   0.140   0.4697  0.625453
## type:element_Zr                   2   2.929   1.464   4.9051  0.007716 **
## site:element_Sr                   1   0.001   0.001   0.0038  0.950554
## site:element_Y                    1   0.148   0.148   0.4958  0.481653
## site:element_Rb                   1   0.590   0.590   1.9760  0.160343
## site:element_Zr                   1   0.070   0.070   0.2348  0.628185
## type:element_Sr:element_Y         2   0.796   0.398   1.3327  0.264553
## type:element_Sr:element_Rb        2   2.105   1.052   3.5247  0.030087 *
## type:element_Sr:element_Zr        2   2.225   1.113   3.7269  0.024640 *
## type:element_Y:element_Rb         2   1.605   0.802   2.6876  0.068885 .
## type:element_Y:element_Zr         2   1.433   0.717   2.4004  0.091573 .
## type:element_Rb:element_Zr        2   1.076   0.538   1.8014  0.165981
## site:element_Sr:element_Y         1   0.451   0.451   1.5108  0.219503
## site:element_Sr:element_Rb        1   0.964   0.964   3.2284  0.072885 .
## site:element_Sr:element_Zr        1   1.344   1.344   4.5022  0.034270 *
## site:element_Y:element_Rb         1   0.298   0.298   0.9986  0.318060
## site:element_Y:element_Zr         1   0.896   0.896   3.0011  0.083732 .
## site:element_Rb:element_Zr        1   1.552   1.552   5.1974  0.022979 *
## type:site:element_Sr              2   0.142   0.071   0.2386  0.787791
## type:site:element_Y               2   0.753   0.377   1.2616  0.283973
## type:site:element_Rb              2   0.641   0.321   1.0739  0.342334
## type:site:element_Zr              2   1.359   0.680   2.2769  0.103512
## type:site:element_Sr:element_Y    1   0.953   0.953   3.1911  0.074555 .
## type:site:element_Sr:element_Rb   1   0.662   0.662   2.2179  0.136955
## type:site:element_Sr:element_Zr   1   0.010   0.010   0.0348  0.852151
## type:site:element_Y:element_Rb    1   0.726   0.726   2.4302  0.119560
## type:site:element_Y:element_Zr    1   0.011   0.011   0.0383  0.844948
## type:site:element_Rb:element_Zr   1   1.986   1.986   6.6532  0.010140 *
## Residuals                       586 174.947   0.299
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#plot(data$element_Rb, model$residuals)
```

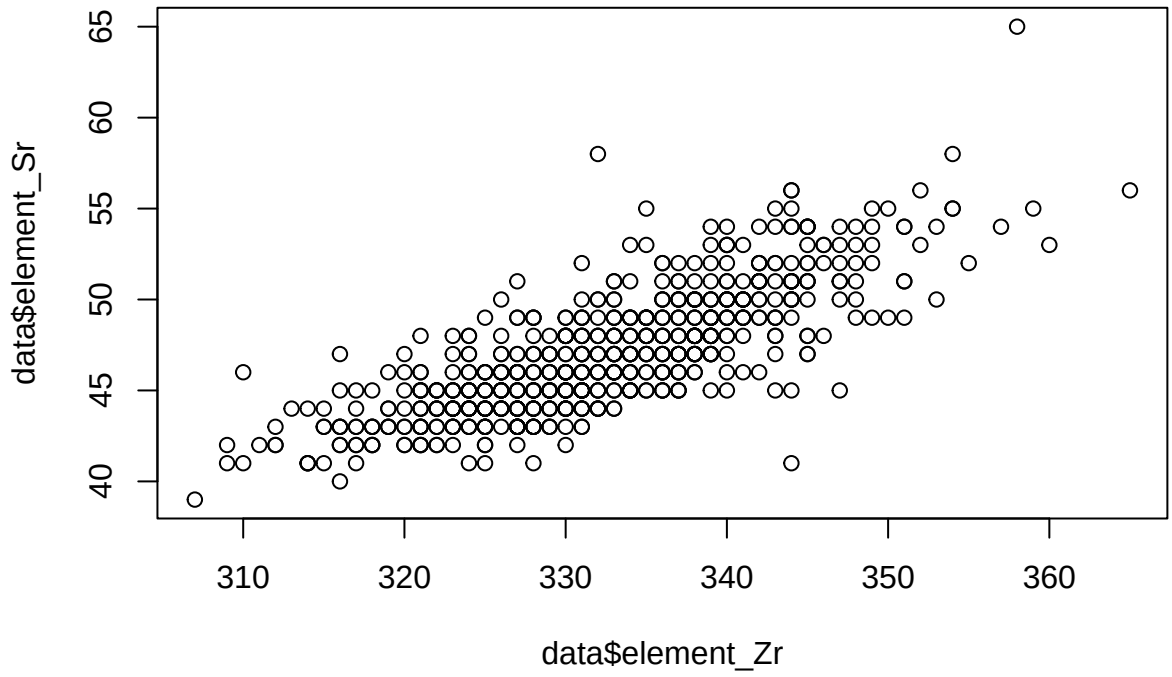
```
plot(data$element_Rb, data$element_Sr)
```



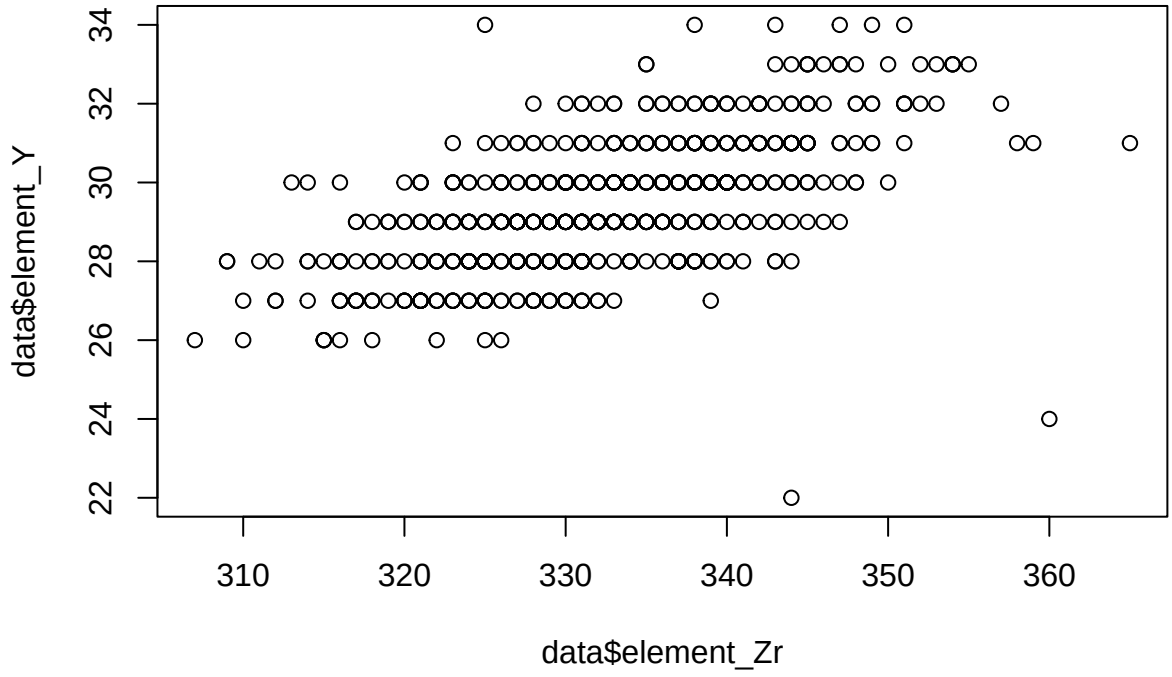

```
plot(data$element_Rb, data$element_Y)
```




```
plot(data$element_Zr, data$element_Sr)
```




```
plot(data$element_Zr, data$element_Y)
```




```
#plot(data$type, data$element_Sr)
```