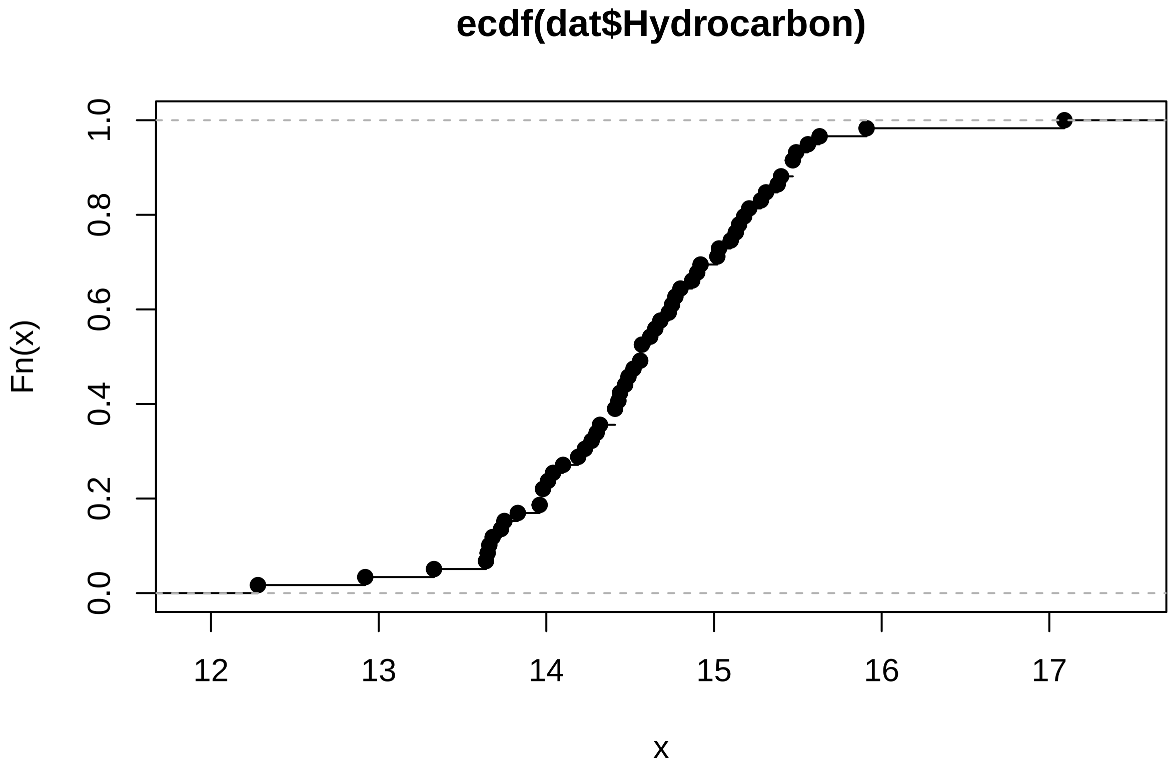
Homework 6

Nicholas Lai

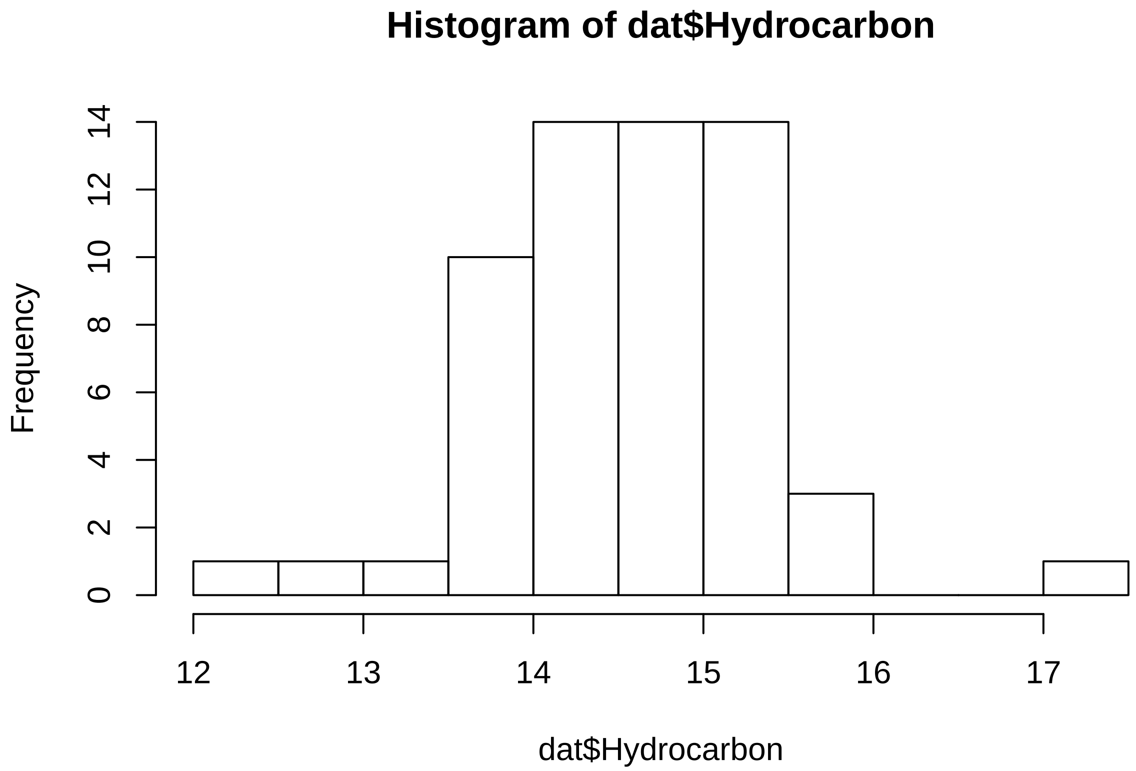
April 5, 2018

Question 6

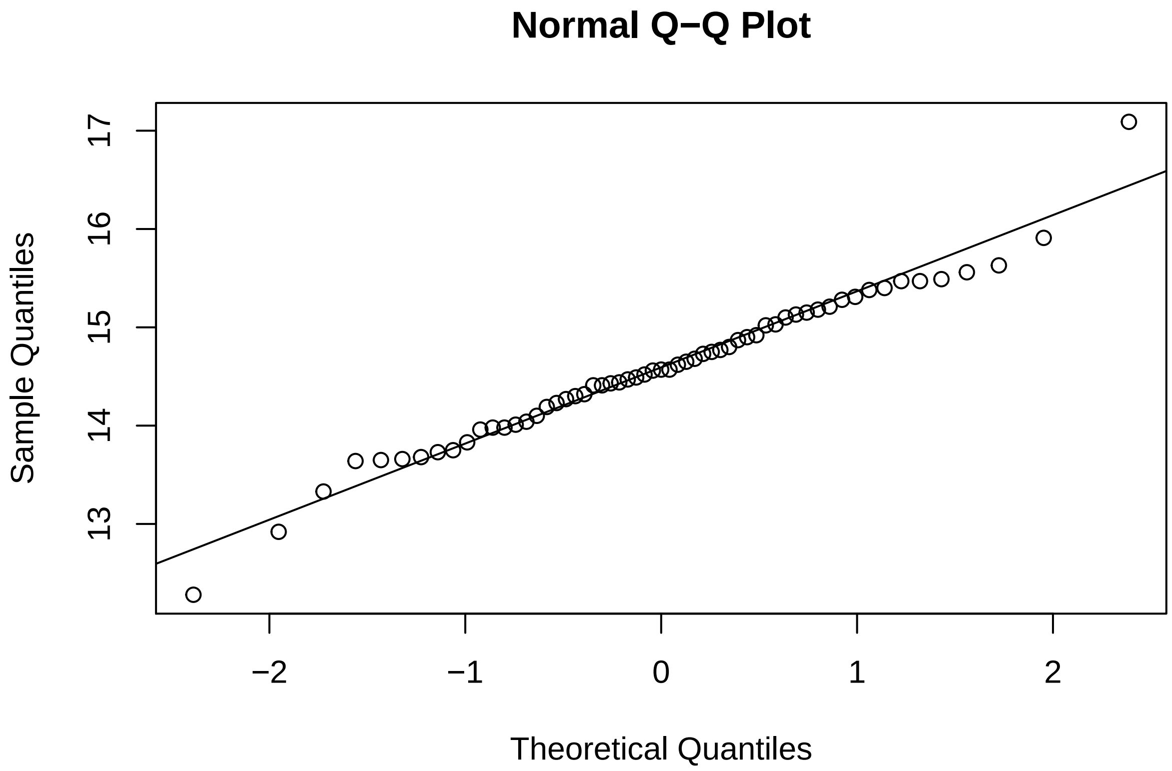
```r
dat <- read.csv("beeswax.txt")
plot(ecdf(dat$Hydrocarbon))
```



```r
hist(dat$Hydrocarbon)
```



```
qqnorm(dat$Hydrocarbon)
qqline(dat$Hydrocarbon)
```



```
quantile(dat$Hydrocarbon, probs = c(.9, .75, .50, .25, .10))
```

```
##    90%    75%    50%    25%    10%
## 15.470 15.115 14.570 14.070 13.676
```

```
waxMean<- mean(dat$Hydrocarbon)
sd(dat$Hydrocarbon)
```

```
## [1] 0.7764197
```

```
waxMean - .99*waxMean+(.01*.85)
```

```
## [1] 0.1543
```

```
waxMean - .97*waxMean+(.03*.85)
```

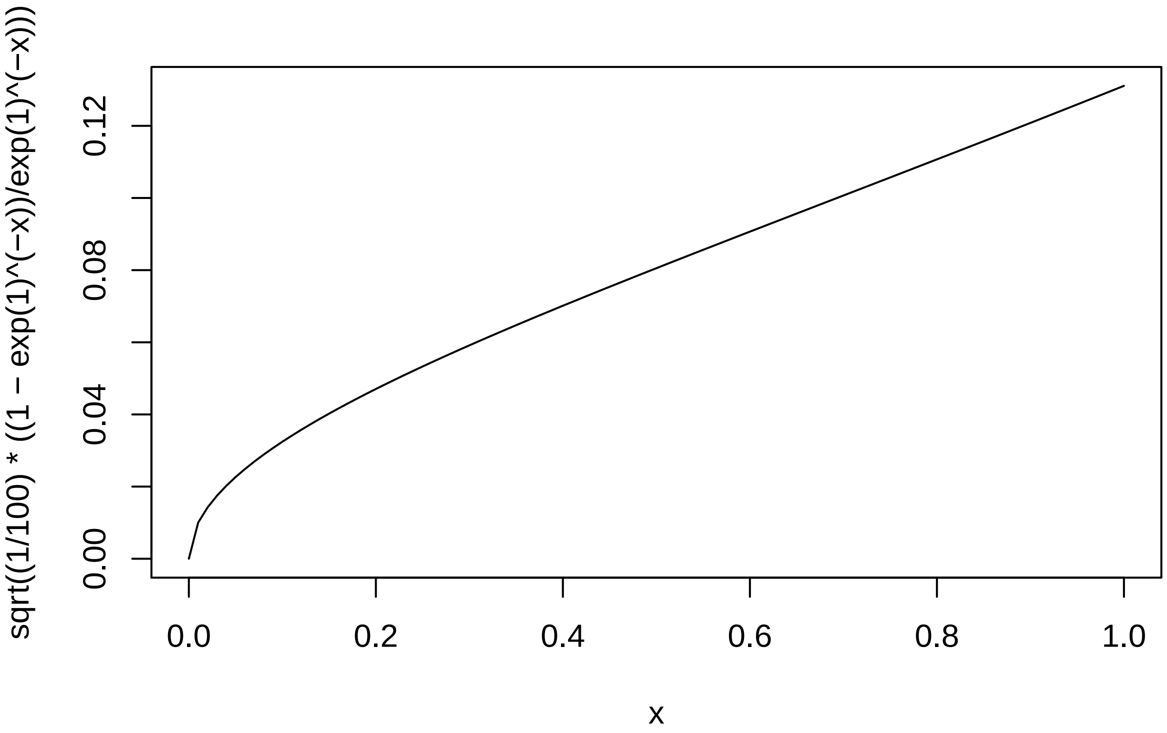
```
## [1] 0.4629
```

```
waxMean - .95*waxMean+(.05*.85)
```

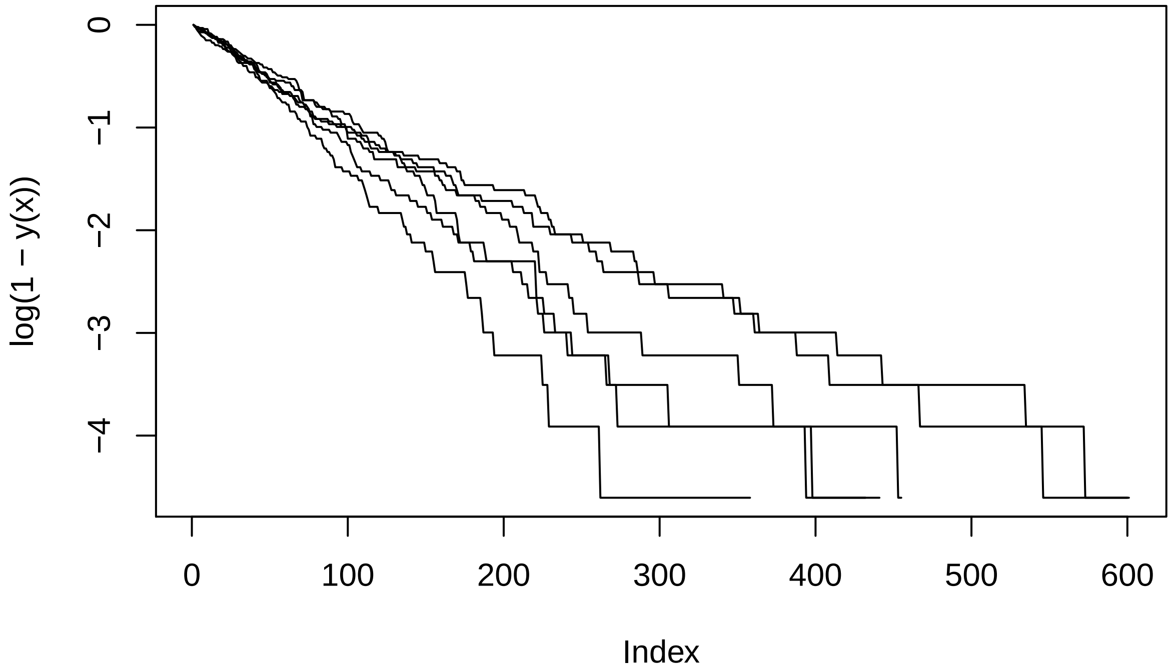
```
## [1] 0.7715
```

Question 8

```
curve(sqrt((1/100)*((1-exp(1)^(-x))/exp(1)^(-x))))
```

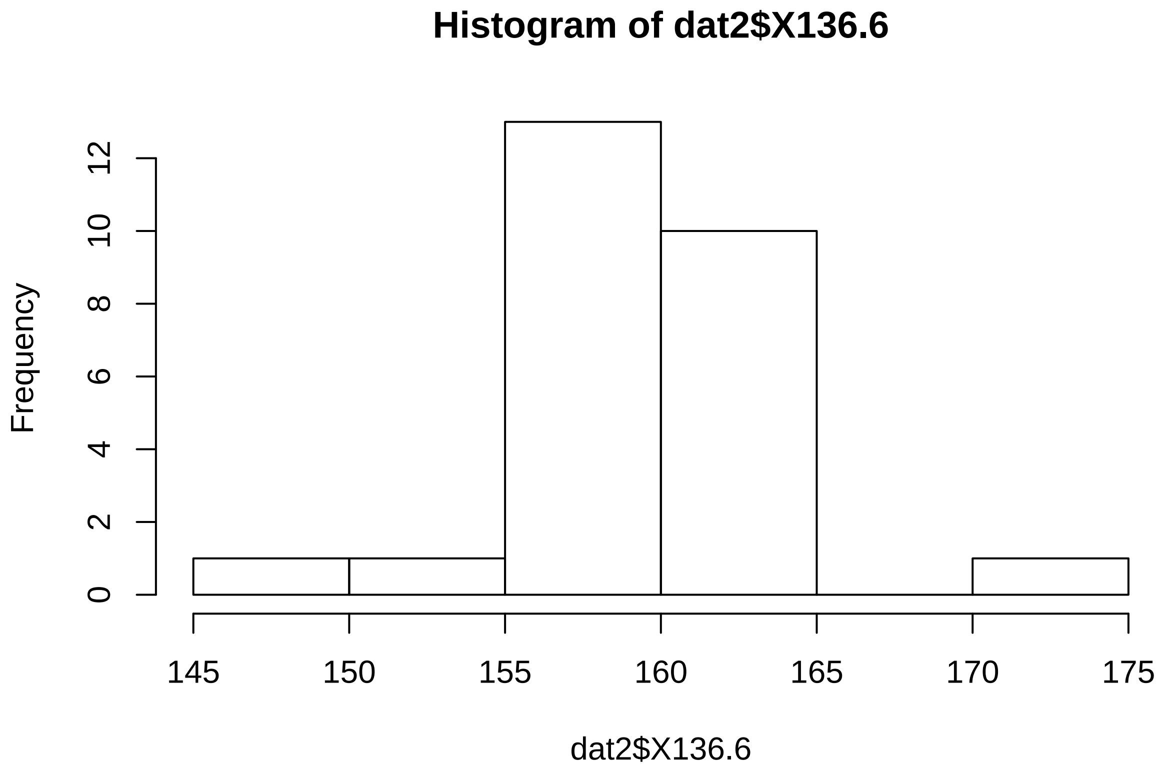


```
data1 <- rexp(n=100, rate = 1)

y = ecdf(data1)

x=seq(0, 6, by=.01)

plot(log(1-y(x)), type = 'l')

for (i in 1:5){
data1 <- rexp(n=100, rate = 1)

y = ecdf(data1)

x=seq(0, 6, by=.01)

lines(log(1-y(x)))
}
```

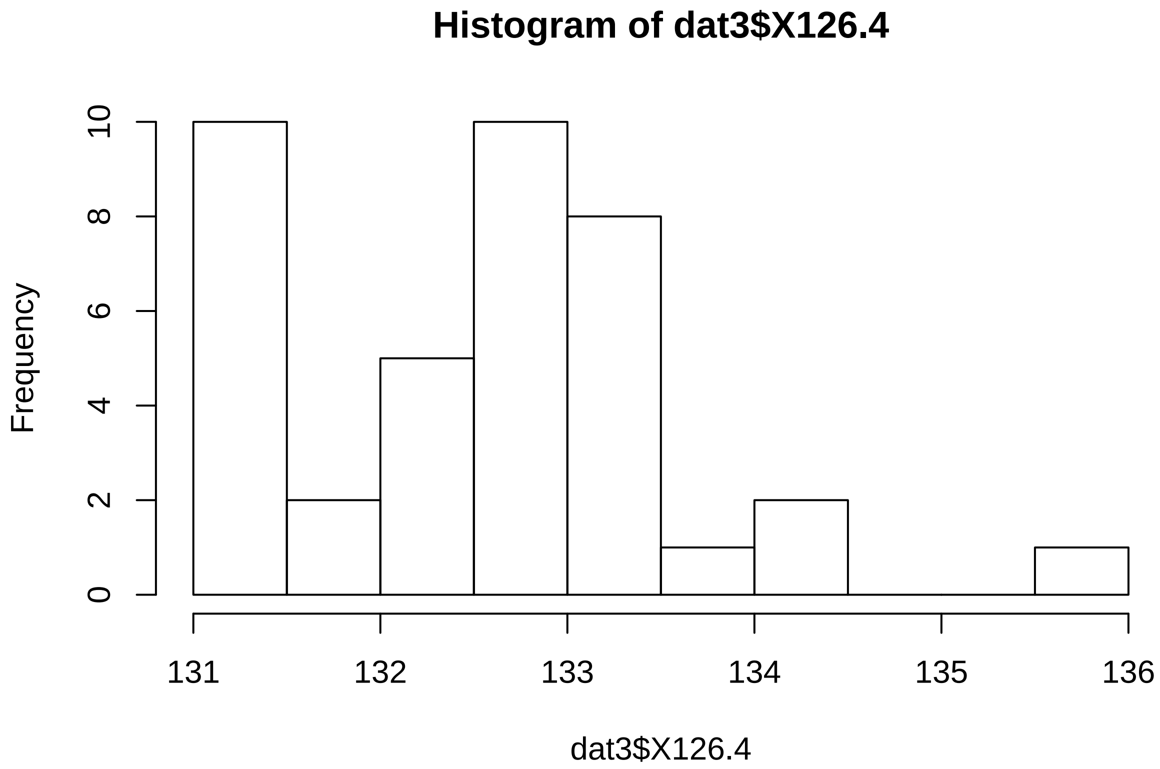


Question 26

```
dat2 <- read.csv("iridium.txt")
dat3 <- read.csv("rhodium.txt")
hist(dat2$X136.6)
```



```
hist(dat3$X126.4)
```



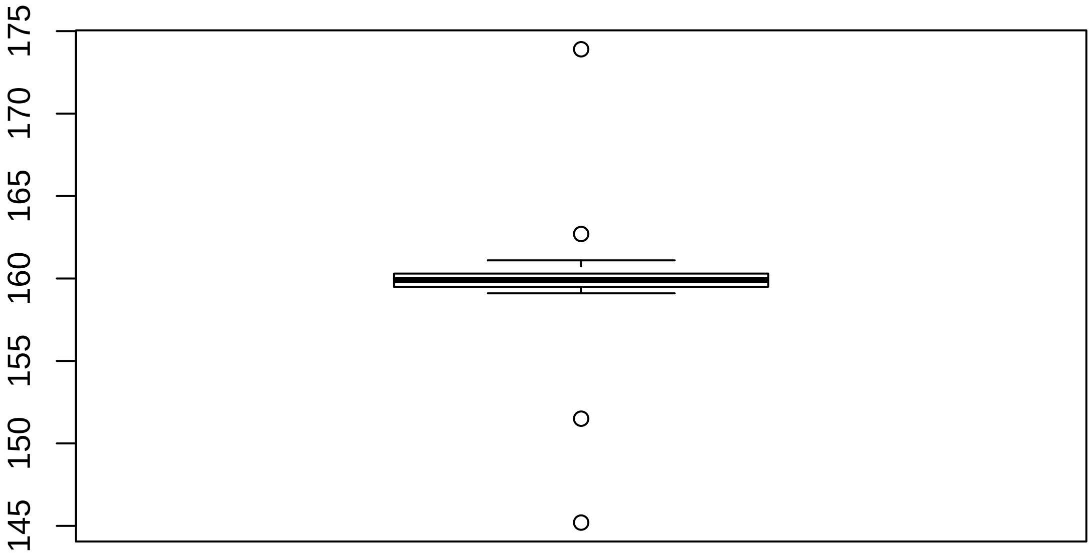
```
stem(dat2$X136.6)
```

```
##
##   The decimal point is 1 digit(s) to the right of the |
##
##   14 | 5
##   15 | 2
##   15 | 999
##   16 | 00000000000000001113
##   16 |
##   17 | 4
```

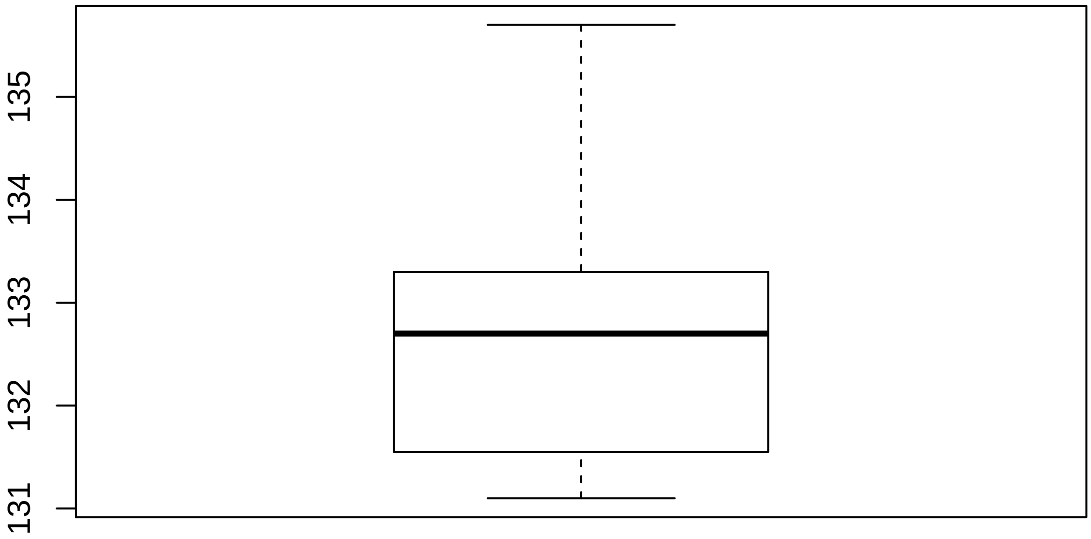
```
stem(dat3$X126.4)
```

```
##
##   The decimal point is at the |
##
##   131 | 111112234
##   131 | 569
##   132 | 1234
##   132 | 56677899
##   133 | 0003334
##   133 | 55558
##   134 | 12
##   134 |
##   135 |
##   135 | 7
```

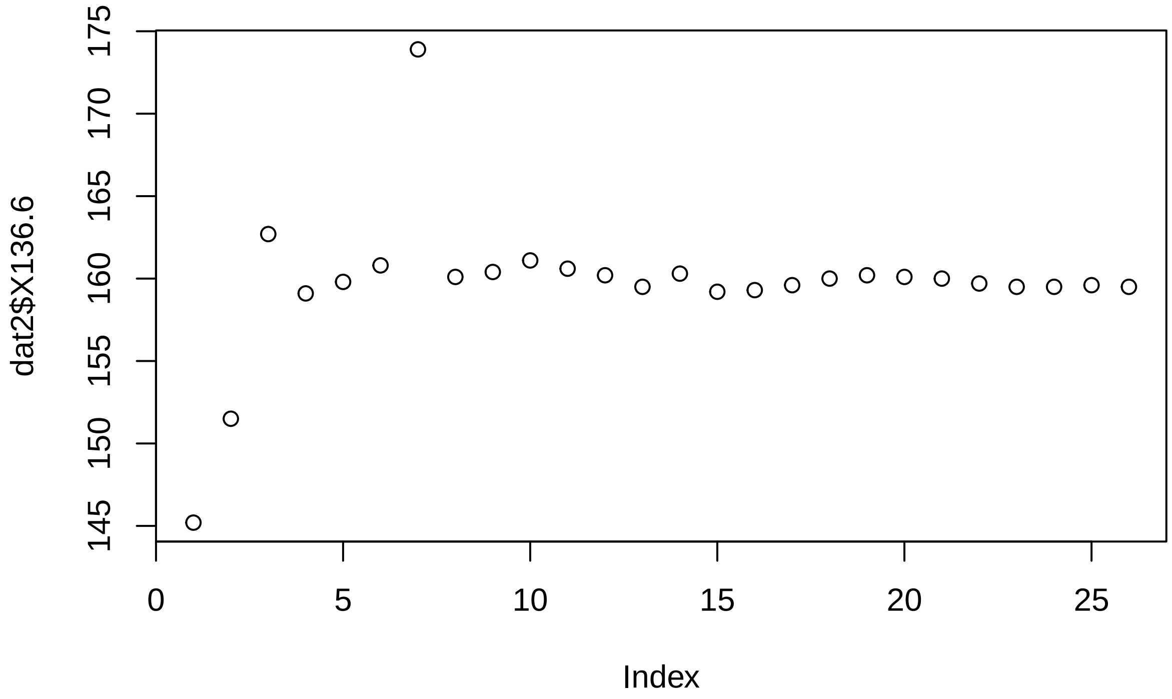
```
boxplot(dat2$X136.6)
```



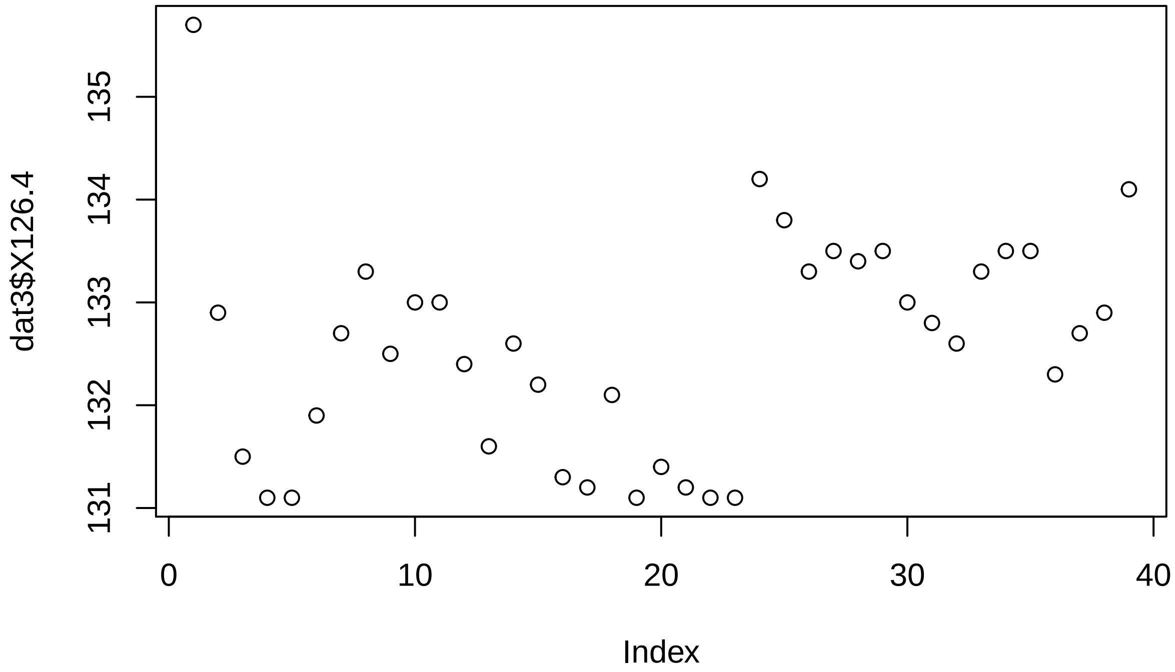
```
boxplot(dat3$X126.4)
```



```r
plot(dat2$X136.6)
```



```
plot(dat3$X126.4)
```



```
mean(dat2$X136.6)
```

```
## [1] 159.6692
```

```
mean(dat2$X136.6, trim = 0.1)
```

```
## [1] 159.9136
```

```
mean(dat2$X136.6, trim = 0.2)
```

```
## [1] 159.875
```

```
median(dat2$X136.6)
```

```
## [1] 159.9
```

```
mean(dat3$X126.4)
```

```
## [1] 132.5744
```

```
mean(dat3$X126.4, trim = 0.1)
```

```
## [1] 132.5182
```

```
mean(dat3$X126.4, trim = 0.2)
```

```
## [1] 132.568
```

```
median(dat3$X126.4)
```

```
## [1] 132.7
```

```
sd(dat2$X136.6)/sqrt(26)
```

```
## [1] 0.8725695
```

```
sd(dat3$X126.4)/sqrt(39)
```

```
## [1] 0.171235
```

```
mean(dat2$X136.6) + sd(dat2$X136.6)/sqrt(26)
```

```
## [1] 160.5418
```

```
mean(dat2$X136.6) - sd(dat2$X136.6)/sqrt(26)
```

```
## [1] 158.7967
```

```
mean(dat3$X126.4) + sd(dat3$X126.4)/sqrt(39)
```

```
## [1] 132.7456
```

```
mean(dat3$X126.4) - sd(dat3$X126.4)/sqrt(39)
```

```
## [1] 132.4031
```

```
ci.median(dat2$X136.6, .90)
```

```
##
## 90% Confidence interval for population median
## Estimate       5%      95%
##    159.9    159.5    160.2
```

```
ci.median(dat3$X126.4, .90)
```

```
##
## 90% Confidence interval for population median
## Estimate       5%      95%
##    132.7    132.2    133.0
```

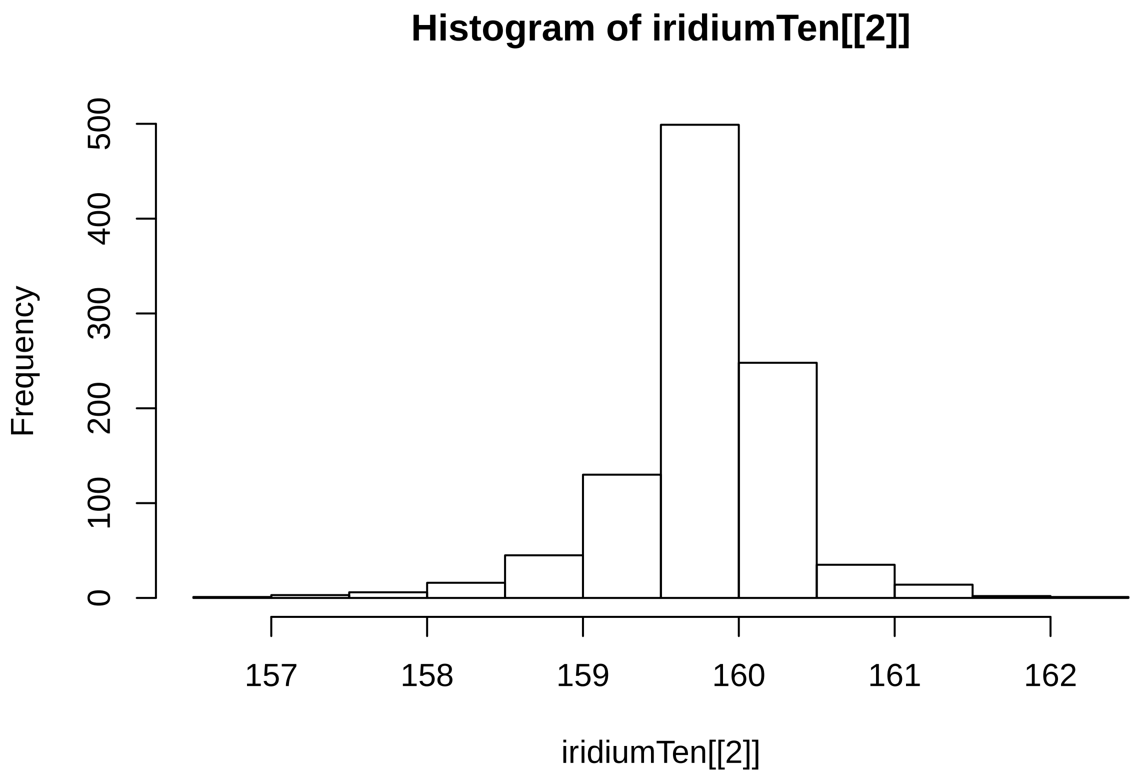
```
bootStrap = function(mySample, popSize = NULL, B = 1000, repl = FALSE){
  if (repl) {
    # Bootstrap should be done the same way as original sample, usually without rep
    return(replicate(B, mean(sample(mySample, length(mySample), TRUE))))
  } else {
    vals = sort(unique(mySample))
    counts = table(mySample)
    # makes the bootstrap pop as rounded version of sample, not quite right
    bootPop = rep(vals, round(counts * popSize / length(mySample)))
    return(list(bootPop,
                bootSamps = replicate(B,mean(sample(bootPop, length(mySample), FALSE), trim = 0.1)))
           )
  }
}

iridiumTen <- bootStrap(dat2$X136.6, 1300)
mean(iridiumTen[[2]])
```

```
## [1] 159.7971
```

```
sd(iridiumTen[[2]])
```

```
## [1] 0.5380054
```

```
hist(iridiumTen[[2]])
```



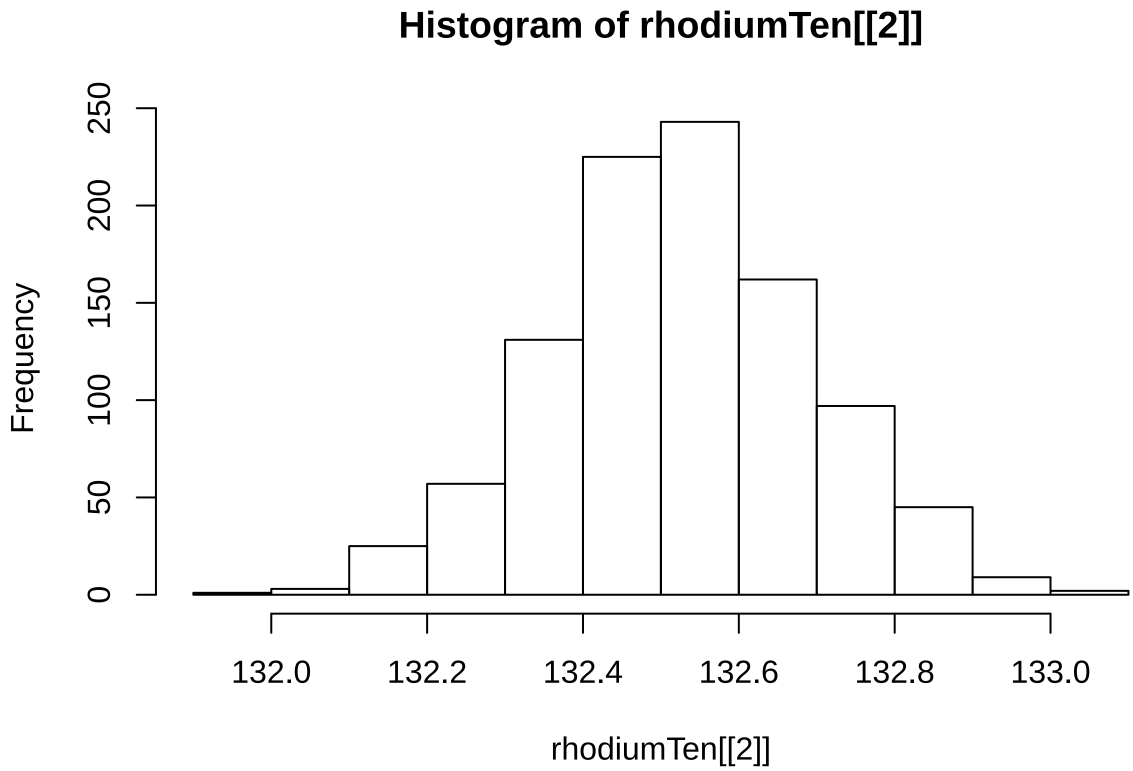
```
rhodiumTen <- bootStrap(dat3$X126.4, 1950)
mean(rhodiumTen[[2]])
```

```
## [1] 132.5269
```

```
sd(rhodiumTen[[2]])
```

```
## [1] 0.1672449
```

```
hist(rhodiumTen[[2]])
```



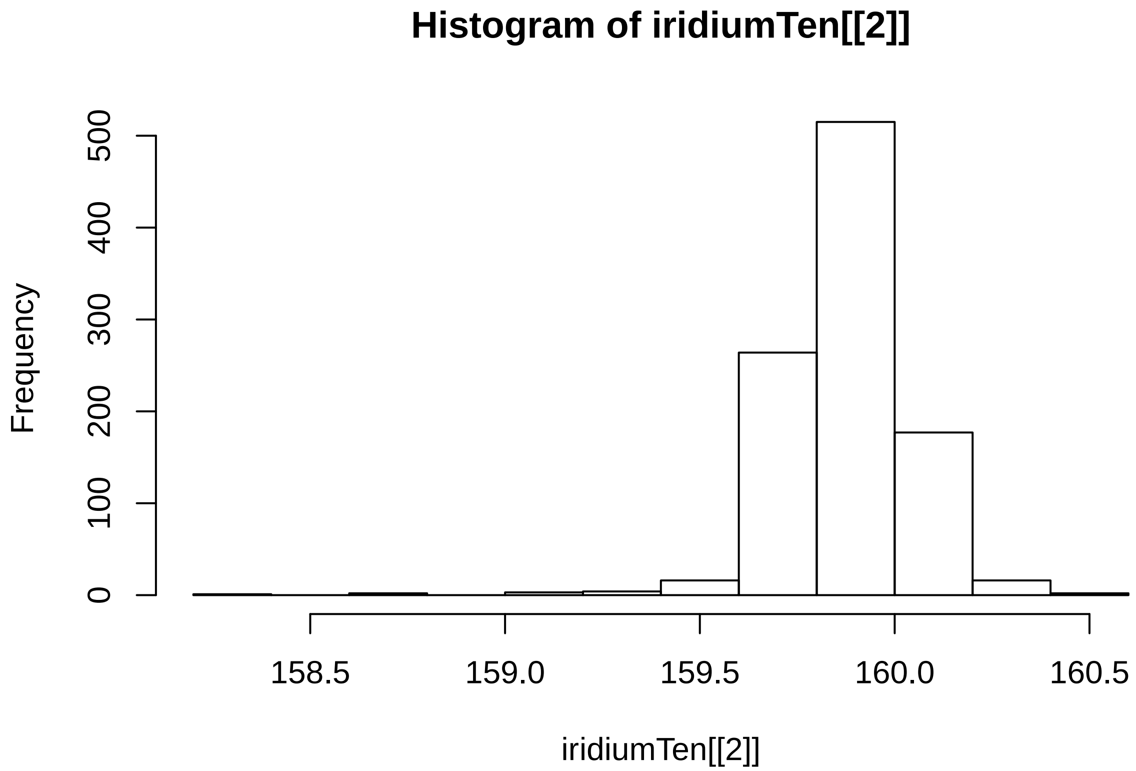
```
bootStrap = function(mySample, popSize = NULL, B = 1000, repl = FALSE){
  if (repl) {
    # Bootstrap should be done the same way as original sample, usually without rep
    return(replicate(B, mean(sample(mySample, length(mySample), TRUE))))
  } else {
    vals = sort(unique(mySample))
    counts = table(mySample)
    # makes the bootstrap pop as rounded version of sample, not quite right
    bootPop = rep(vals, round(counts * popSize / length(mySample)))
    return(list(bootPop,
                bootSamps = replicate(B,mean(sample(bootPop, length(mySample), FALSE), trim = 0.2)))
           )
  }
}
iridiumTen <- bootStrap(dat2$X136.6, 1300)
mean(iridiumTen[[2]])
```

```
## [1] 159.875
```

```
sd(iridiumTen[[2]])
```

```
## [1] 0.1668958
```

```
hist(iridiumTen[[2]])
```



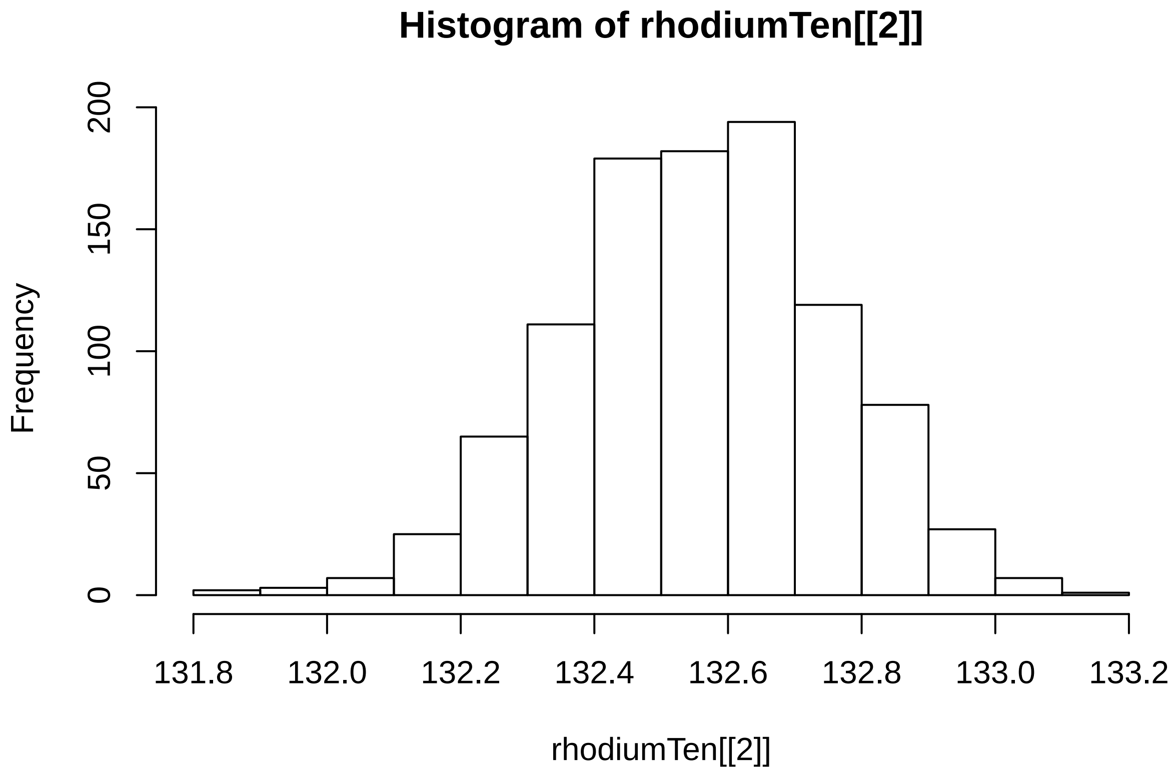
```r
rhodiumTen <- bootStrap(dat3$X126.4, 1950)
mean(rhodiumTen[[2]])
```

```
## [1] 132.5578
```

```r
sd(rhodiumTen[[2]])
```

```
## [1] 0.1984552
```

```r
hist(rhodiumTen[[2]])
```



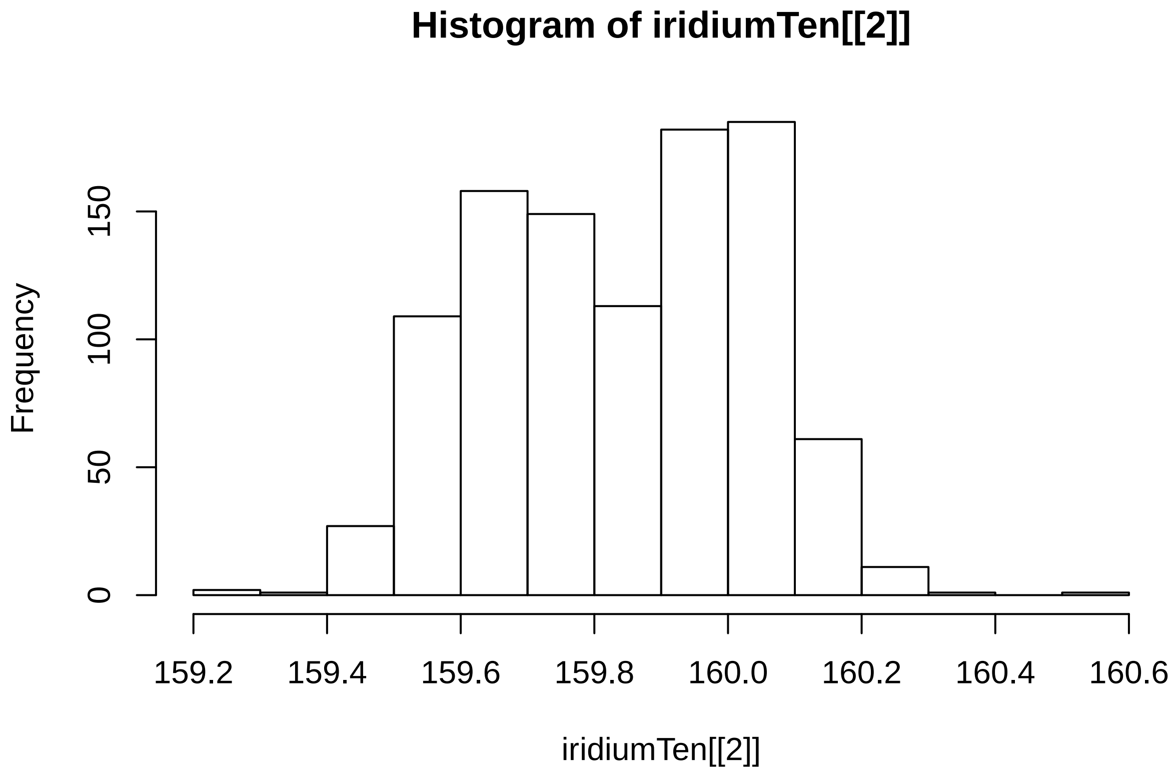
```
bootStrap = function(mySample, popSize = NULL, B = 1000, repl = FALSE){
  if (repl) {
    # Bootstrap should be done the same way as original sample, usually without rep
    return(replicate(B, mean(sample(mySample, length(mySample), TRUE))))
  } else {
    vals = sort(unique(mySample))
    counts = table(mySample)
    # makes the bootstrap pop as rounded version of sample, not quite right
    bootPop = rep(vals, round(counts * popSize / length(mySample)))
    return(list(bootPop,
                bootSamps = replicate(B,median(sample(bootPop, length(mySample), FALSE))))
           )
  }
}
iridiumTen <- bootStrap(dat2$X136.6, 1300)
mean(iridiumTen[[2]])
```

```
## [1] 159.8712
```

```
sd(iridiumTen[[2]])
```

```
## [1] 0.1998127
```

```
hist(iridiumTen[[2]])
```



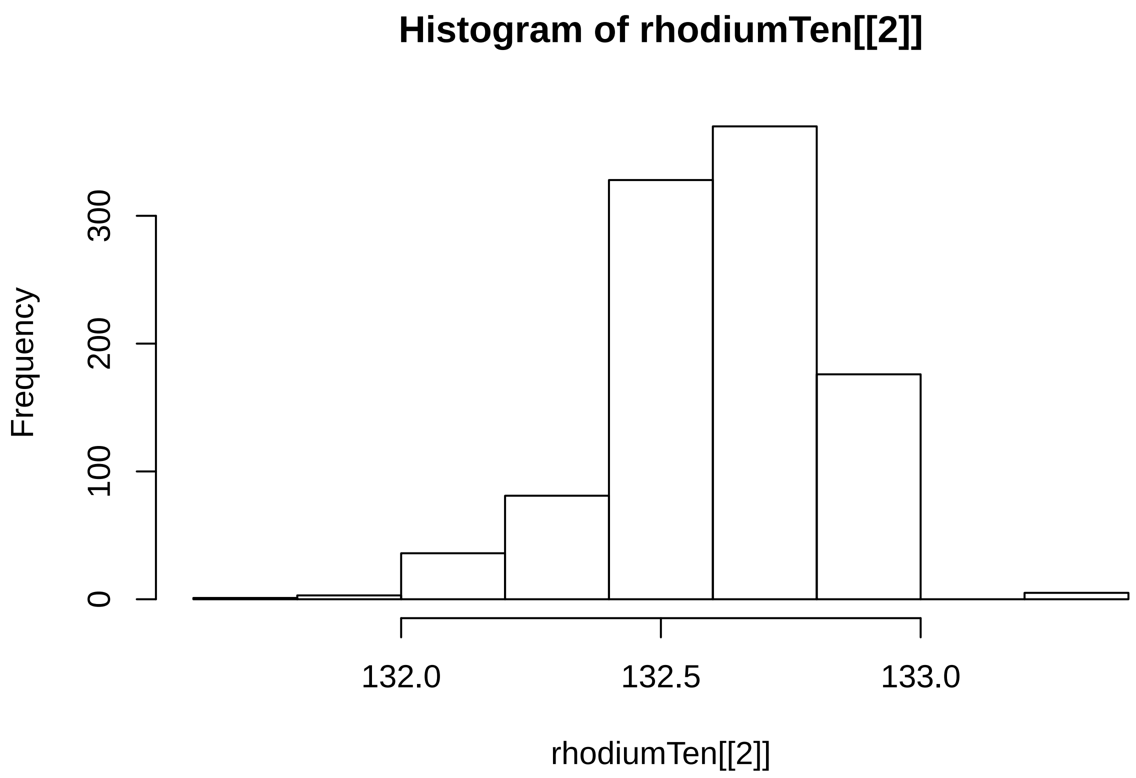
```
rhodiumTen <- bootStrap(dat3$X126.4, 1950)
mean(rhodiumTen[[2]])
```

```
## [1] 132.6608
```

```
sd(rhodiumTen[[2]])
```

```
## [1] 0.2004084
```

```
hist(rhodiumTen[[2]])
```



```
MeanCI(dat2$X136.6, trim = .1, conf.level = .9)
```

```
##     mean   lwr.ci   upr.ci
## 159.9136 159.6481 160.1791
```

```
MeanCI(dat2$X136.6, trim = .2, conf.level = .9)
```

```
##     mean   lwr.ci   upr.ci
## 159.8750 159.6573 160.0927
```

```
MeanCI(dat3$X126.4, trim = .1, conf.level = .9)
```

```
##     mean   lwr.ci   upr.ci
## 132.5182 132.2020 132.8344
```

```
MeanCI(dat3$X126.4, trim = .2, conf.level = .9)
```

```
##     mean   lwr.ci   upr.ci
## 132.5680 132.1818 132.9542
```