Statistics in R: Task 0

Liuaza Etezova

```
library(ggplot2)
library(ggpubr)
```

1. Measures of center

```
x <- c(175, 176, 182, 165, 167, 172, 175, 196, 158, 172)
```

Mode

```
mode_my <- function(sample) {
  tab <- table(sample)
  as.numeric(names(tab)[tab == max(tab)])
}

mode_my_x <- mode_my(x)
mode_my_x
```

```
## [1] 172 175
```

Median

```
median_my <- function(sample) {
  sorted <- sort(sample)
  (sorted[length(sample) / 2] + sorted[length(sample) / 2 + 1]) / 2
}
median_my_x <- median_my(x)

median_builtin_x <- median(x)

median_builtin_x
```

```
## [1] 173.5
```

```
median_my_x
```

```
## [1] 173.5
```

Mean

```
mean_my <- function(sample) {
  sum(sample) / length(sample)
}
mean_my_x <- mean_my(x)

mean_builtin_x <- mean(x)
mean_trim_x <- mean(x, trim = 0.1)

mean_builtin_x
```

```
## [1] 173.8
```

```
mean_my_x
```

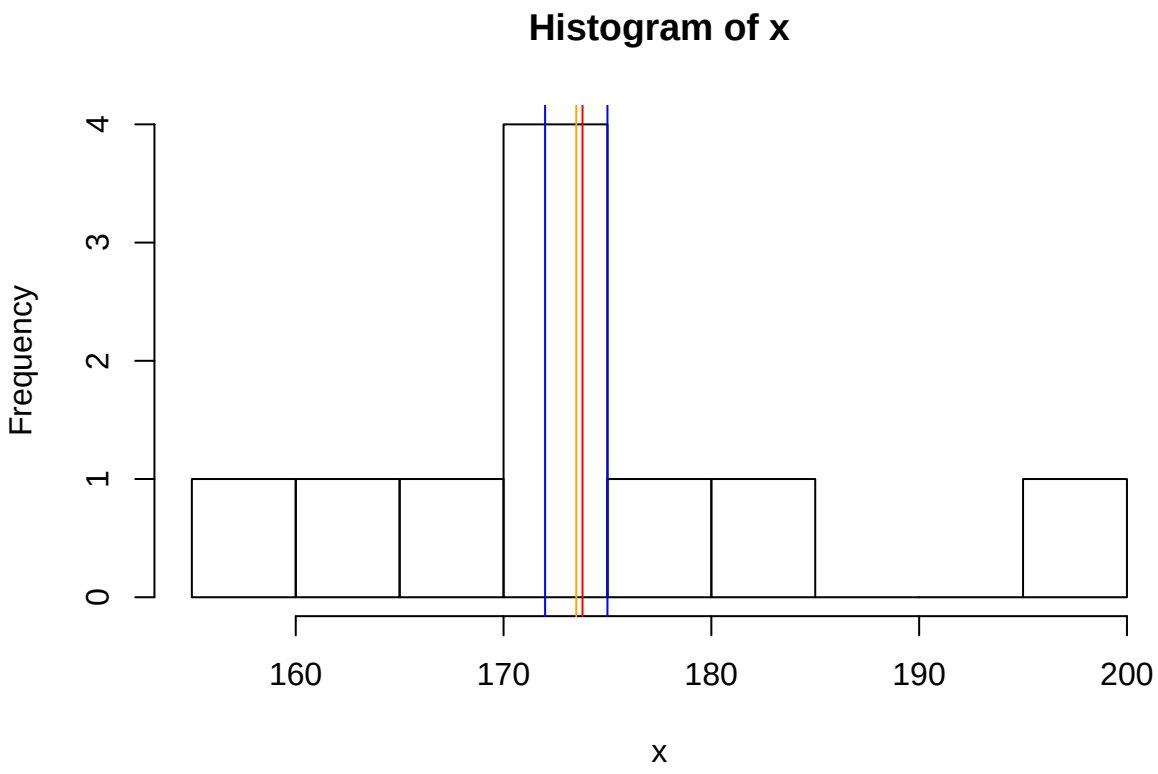
```
## [1] 173.8
```

```
mean_trim_x
```

```
## [1] 173
```

Histogram

```
hist(x, breaks=10)
abline(v = mode_my_x, col="blue")
abline(v = median_my_x, col="orange")
abline(v = mean_my_x, col="red")
```



Sample with an outlier

```
x_outl <- c(x, 2)

mode_my_outl <- mode_my(x_outl)

median_builtin_outl <- median(x_outl)
median_my_outl <- median_my(x_outl)

mean_builtin_outl <- mean(x_outl)
mean_my_outl <- mean_my(x_outl)
mean_trim_outl <- mean(x_outl, trim = 0.1)

mode_my_outl
```

```
## [1] 172 175
```

```
median_builtin_outl
```

```
## [1] 172
```

```
median_my_outl
```

```
## [1] 172
```

```
mean_builtin_outl
```

```
## [1] 158.1818
```

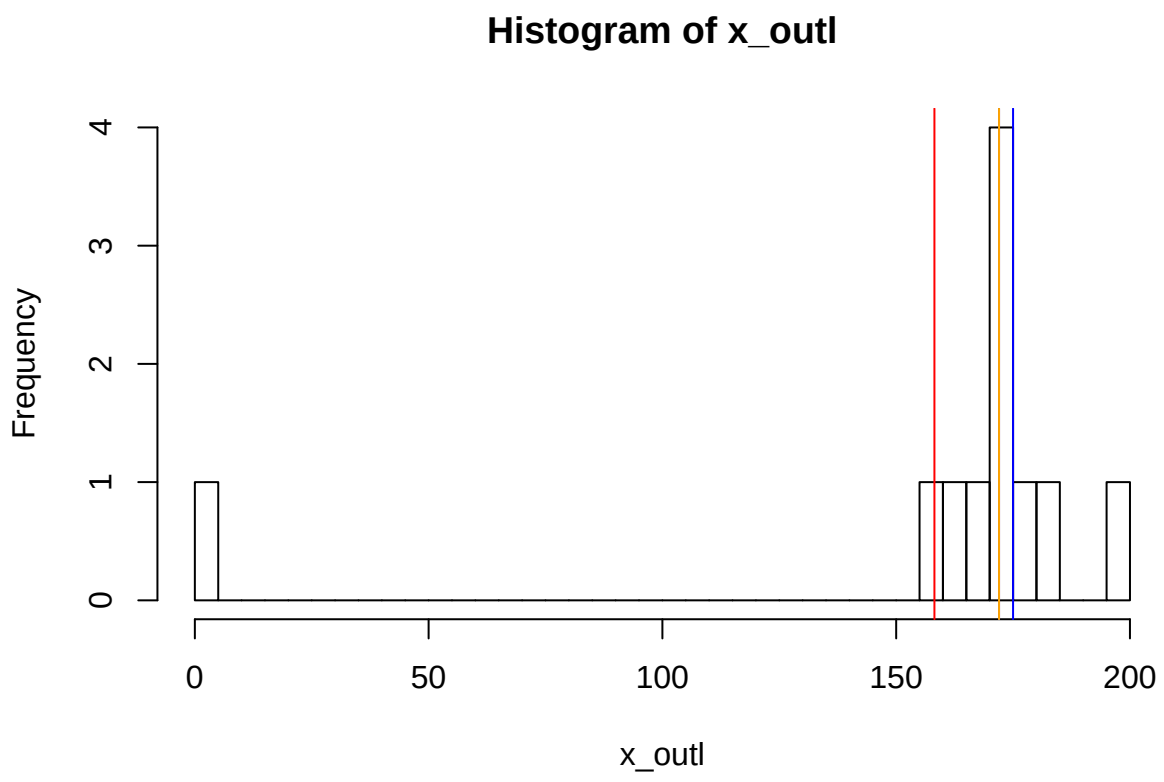
```
mean_my_outl
```

```
## [1] 158.1818
```

```
mean_trim_outl
```

```
## [1] 171.3333
```

```
hist(x_outl, breaks=30)
abline(v = mode_my_outl, col="blue")
abline(v = median_my_outl, col="orange")
abline(v = mean_my_outl, col="red")
```



One of the modes and the median are overlapping.

2. Measures of spread

Variance

```
var_builtin_x <- var(x)

var_my <- function(sample) {
  sum <- 0
  for (i in 1:length(sample)) {
    sum = sum + (sample[i] - mean_builtin_x)^2
  }
  sum / (length(sample) - 1)
}
var_my_x <- var_my(x)

var_builtin_x
```

```
## [1] 105.2889
```

```
var_my_x
```

```
## [1] 105.2889
```

SD

```
sd_builtin_x <- sd(x)
sd_my_x <- sqrt(var_my_x)

sd_builtin_x
```

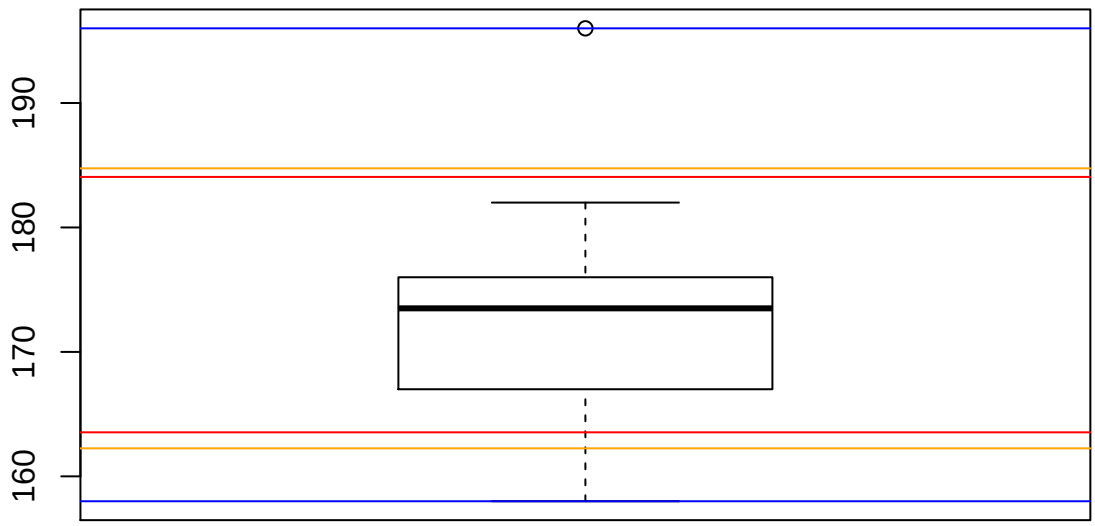
```
## [1] 10.26104
```

```
sd_my_x
```

```
## [1] 10.26104
```

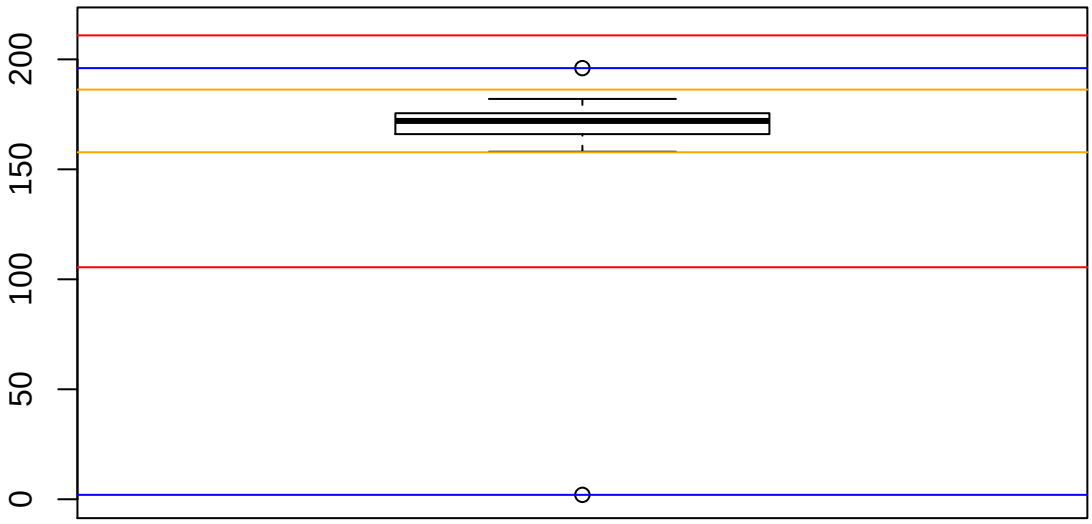
Boxplot

```
iqr_x <- IQR(x)

boxplot(x)
abline(h = range(x), col="blue")
abline(h = median_builtin_x + 1.5 * iqr_x, col="orange")
abline(h = median_builtin_x - 1.5 * iqr_x, col="orange")
abline(h = mean_builtin_x + sd_builtin_x, col="red")
abline(h = mean_builtin_x - sd_builtin_x, col="red")
```



Sample with an outlier

```
iqr_outl <- IQR(x_outl)
sd_outl <- sd(x_outl)

boxplot(x_outl, ylim=c(0, 215))
abline(h = range(x_outl), col="blue")
abline(h = median_builtin_outl + 1.5 * iqr_outl, col="orange")
abline(h = median_builtin_outl - 1.5 * iqr_outl, col="orange")
abline(h = mean_builtin_outl + sd_outl, col="red")
abline(h = mean_builtin_outl - sd_outl, col="red")
```



3. Properties

Checking

```
sub <- x - 100
div <- x / 100

all.equal(mean(sub), mean(x) - 100)
```

```
## [1] TRUE
```

```
all.equal(mean(div), mean(x) / 100)
```

```
## [1] TRUE
```

```
abs(sum(x - mean(x)) - 0) < 0.000000001
```

```
## [1] TRUE
```

```r
all.equal(var(sub), var(x))
```

```
## [1] TRUE
```

```r
all.equal(var(div), var(x) / 10000)
```

```
## [1] TRUE
```

```r
all.equal(sd(div), sd(x) / 100)
```

```
## [1] TRUE
```

Visualization

Table

```r
prop_df <- matrix(c(mean(x), mean(sub), mean(div),
                    var(x), var(sub), var(div),
                    sd(x), sd(sub), sd(div)), ncol=3,byrow=TRUE)
colnames(prop_df) <- c('x','x-100','x/100')
rownames(prop_df) <- c("mean","var","sd")
prop_df <- as.data.frame(prop_df)
prop_df
```

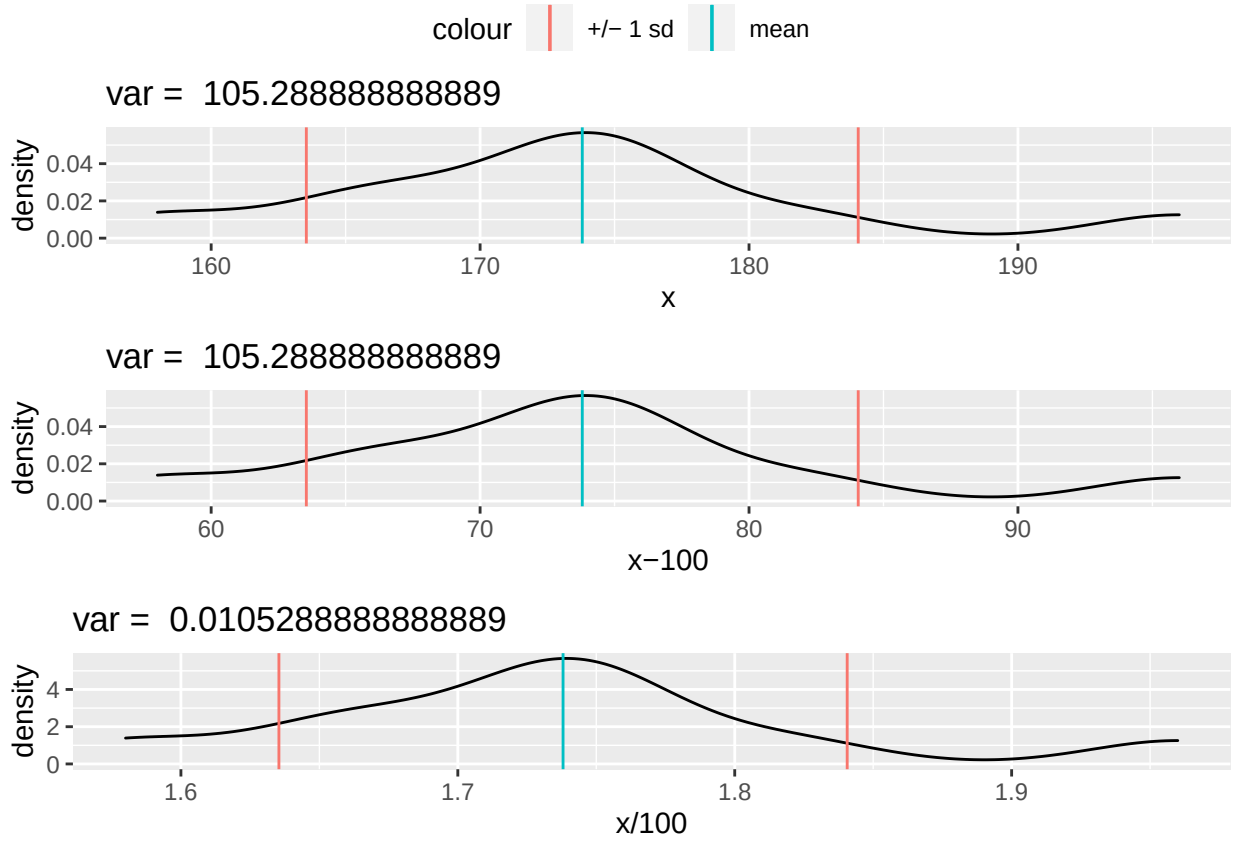
```
##              x     x-100      x/100
## mean 173.80000  73.80000 1.73800000
## var  105.28889 105.28889 0.01052889
## sd    10.26104  10.26104 0.10261037
```

Plot

```r
data_df <- cbind(x, sub, div)
samples_ids <- c('x','x-100','x/100')
colnames(data_df) <- samples_ids
prop_df <- rbind(prop_df, data_df)
```

```r
plot <- function(sample_id) {
  column <- prop_df[[sample_id]]
  ggplot() +
    aes(column[4:13]) +
    geom_density() +
    geom_vline(aes(xintercept=column[[1]], color="mean")) +
    geom_vline(aes(xintercept=column[[1]] - column[[3]], color="+/- 1 sd")) +
    geom_vline(aes(xintercept=column[[1]] + column[[3]], color="+/- 1 sd")) +
    xlab(sample_id) +
    ggtitle(paste("var = ", column[[2]]))
```

```
}

# not working with cycle; with cycle through samples_ids instead of apply
# it will use the last sample_id in all plots, so all plots will be the same
plots <- lapply(samples_ids, plot)
ggarrange(plotlist=plots, ncol=1, common.legend = TRUE)
```



4. Normal distribution

```
set.seed(42)
```

```
# p(x < 156)
pnorm(156, mean=175, sd=10)
```

```
## [1] 0.02871656
```

```
# p(x > 198)
pnorm(198, mean=175, sd=10, lower=FALSE)
```

```
## [1] 0.01072411
```

```r
# p(168 < x < 172)
obs <- 1e5
sample <- rnorm(obs, mean=175, sd=10)
sum(sample > 168 & sample < 172) / obs
```

```
## [1] 0.13851
```

Standard normal distribution

```r
# mean = 0, sd = 1
sample <- rnorm(obs)

# +/- 1 sd: 68%
sum(sample > -1 & sample < 1) / obs
```

```
## [1] 0.68135
```

```r
pnorm(1) - pnorm(-1)
```

```
## [1] 0.6826895
```

```r
# +/- 2 sd: 95%
sum(sample > -2 & sample < 2) / obs
```

```
## [1] 0.95408
```

```r
pnorm(2) - pnorm(-2)
```

```
## [1] 0.9544997
```

```r
# +/- 3 sd: 99.7%
sum(sample > -3 & sample < 3) / obs
```

```
## [1] 0.99679
```

```r
pnorm(3) - pnorm(-3)
```

```
## [1] 0.9973002
```

Standardization

```r
sample <- rnorm(obs, mean=175, sd=10)
mean(sample)
```

```
## [1] 174.992
```

```r
sd(sample)
```

```
## [1] 10.00118
```

```r
sample <- scale(sample, center=TRUE, scale=TRUE)
mean(sample)
```

```
## [1] 1.44957e-15
```

```r
sd(sample)
```

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

5. Central Limit Theorem

```r
stat_by_k <- function(population, k) {
  samples <- replicate(n, sample(population, k))
  means <- colMeans(samples)
  c(k, mean(means), sd(means), sd(means)/sqrt(n), means)
}

obs <- 1e6
population <- rnorm(obs)
ks <- c(10, 50, 100, 500)
n <- 30
stat <- data.frame()
distr <- vector(mode="list", length=length(ks))
names(distr) <- ks
for (k in ks) {
  stat_temp <- stat_by_k(population, k)
  stat <- rbind(stat, stat_temp[1:4])
  distr[[as.character(k)]] <- stat_temp[5:n+4]
}
colnames(stat) <- c('k','mean','sd', 'SE')

stat
```

```
##     k        mean         sd         SE
## 1  10 0.134680648 0.34136140 0.06232378
## 2  50 0.002981947 0.13294629 0.02427256
## 3 100 0.022223126 0.08261959 0.01508420
## 4 500 0.002207400 0.04512560 0.00823877
```

```r
plot <- function(k) {
  ggplot() +
    aes(distr[[as.character(k)]]) +
    geom_histogram() +
    geom_vline(aes(xintercept=stat[stat$k == k, 2], color="mean")) +
```

```
    geom_vline(aes(xintercept=stat[stat$k == k, 2] - stat[stat$k == k, 4],
                   color="+/- 1 SE")) +
    geom_vline(aes(xintercept=stat[stat$k == k, 2] + stat[stat$k == k, 4],
                   color="+/- 1 SE")) +
    xlab(paste(k, " observations")) +
    xlim(-0.5, 0.5)
}

plots <- lapply(ks, plot)
ggarrange(plotlist=plots, common.legend = TRUE)
```