qbs120_ps7_correction_gibran

Gibran Erlangga

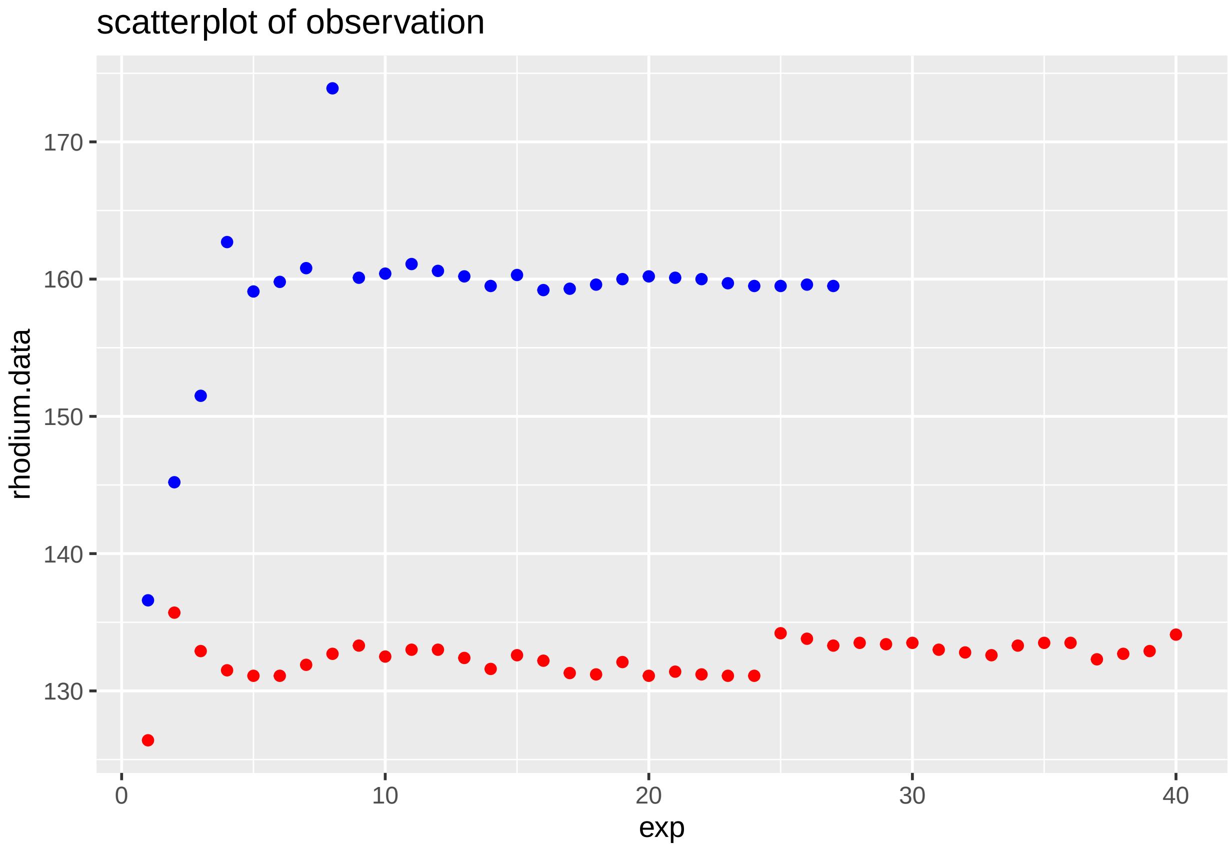
11/7/2021

Question 1

```
library(ggplot2)
library(sfsmisc)
library(tidyverse)
library(reshape2)

iridium.data = c(136.6, 145.2, 151.5, 162.7, 159.1, 159.8, 160.8, 173.9, 160.1, 160.4,
                 161.1, 160.6, 160.2, 159.5, 160.3, 159.2, 159.3, 159.6, 160.0, 160.2,
                 160.1, 160.0, 159.7, 159.5, 159.5, 159.6, 159.5)

rhodium.data = c(126.4, 135.7, 132.9, 131.5, 131.1, 131.1, 131.9, 132.7, 133.3, 132.5,
                 133.0, 133.0, 132.4, 131.6, 132.6, 132.2, 131.3, 131.2, 132.1, 131.1,
                 131.4, 131.2, 131.1, 131.1, 134.2, 133.8, 133.3, 133.5, 133.4, 133.5,
                 133.0, 132.8, 132.6, 133.3, 133.5, 133.5, 132.3, 132.7, 132.9, 134.1)
```

a. my original solution was correct.
b. my original solution was correct.
c. my original solution was correct.
d.

```
rhodium_df <- data.frame(rhodium.data)
rhodium_df$exp <- c(1:length(rhodium.data))
iridium_df <- data.frame(iridium.data)
iridium_df$exp <- c(1:length(iridium.data))

ggplot() +
  geom_point(data=rhodium_df, aes(x=exp, y=rhodium.data), color="red") +
  geom_point(data=iridium_df, aes(x=exp, y=iridium.data), color="blue") +
  labs(title="scatterplot of observation",
       xlab="experiment",
       ylab="value")
```



e. No, it does not seem reasonable. When the results are plotted according to the order of the experiments, there appears to be some sequential correlation. For iridium, looks like the researchers had calibration (or other) technical issues for the first few experiments but then achieved stable results. For rhodium, there was also some initial instability as well as an upward bias that appears after the 25th experiment.

f. my original solution was correct.
g.

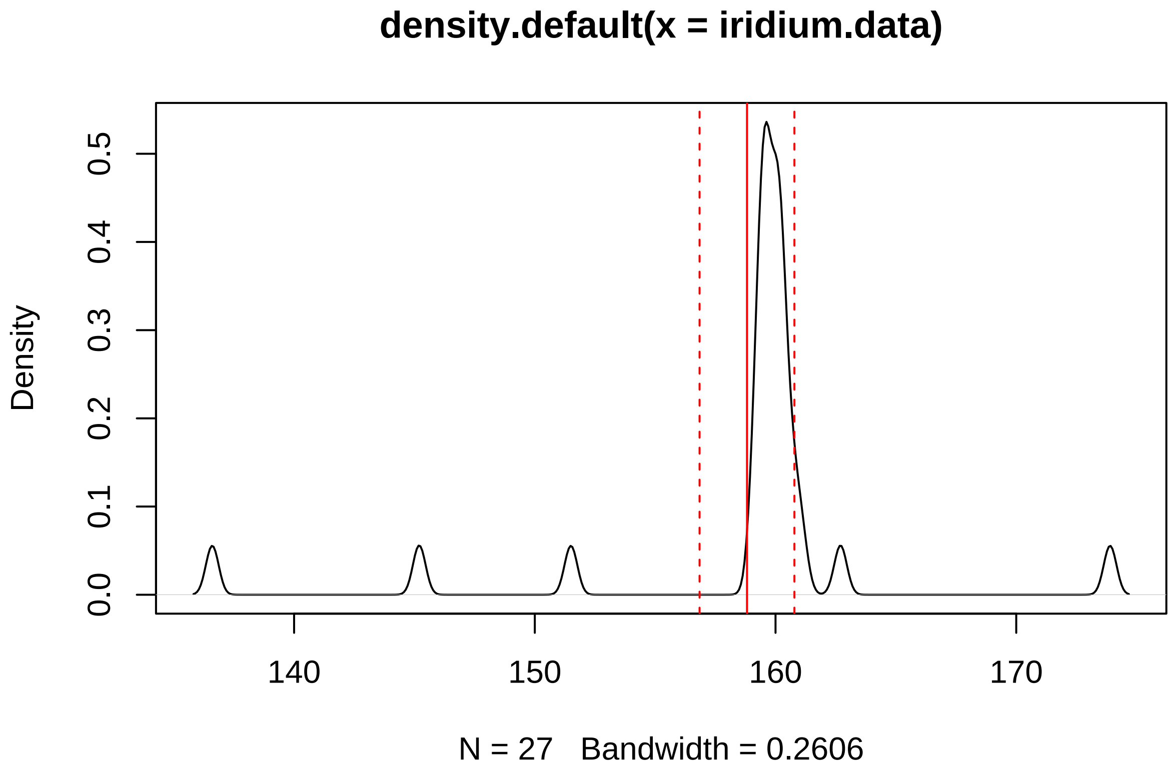
```
# iridium
n=length(iridium.data)
(mean.ci.low.iridium =mean(iridium.data) - sd(iridium.data) * qnorm(0.95)/sqrt(n))
```

```
## [1] 156.8444
```

```
(mean.ci.high.iridium =mean(iridium.data) + sd(iridium.data) * qnorm(0.95)/sqrt(n))
```

```
## [1] 160.7852
```

```
plot(density(iridium.data))
abline(v=mean(iridium.data), col="red")
abline(v=mean.ci.low.iridium, col="red", lty="dashed")
abline(v=mean.ci.high.iridium, col="red", lty="dashed")
```



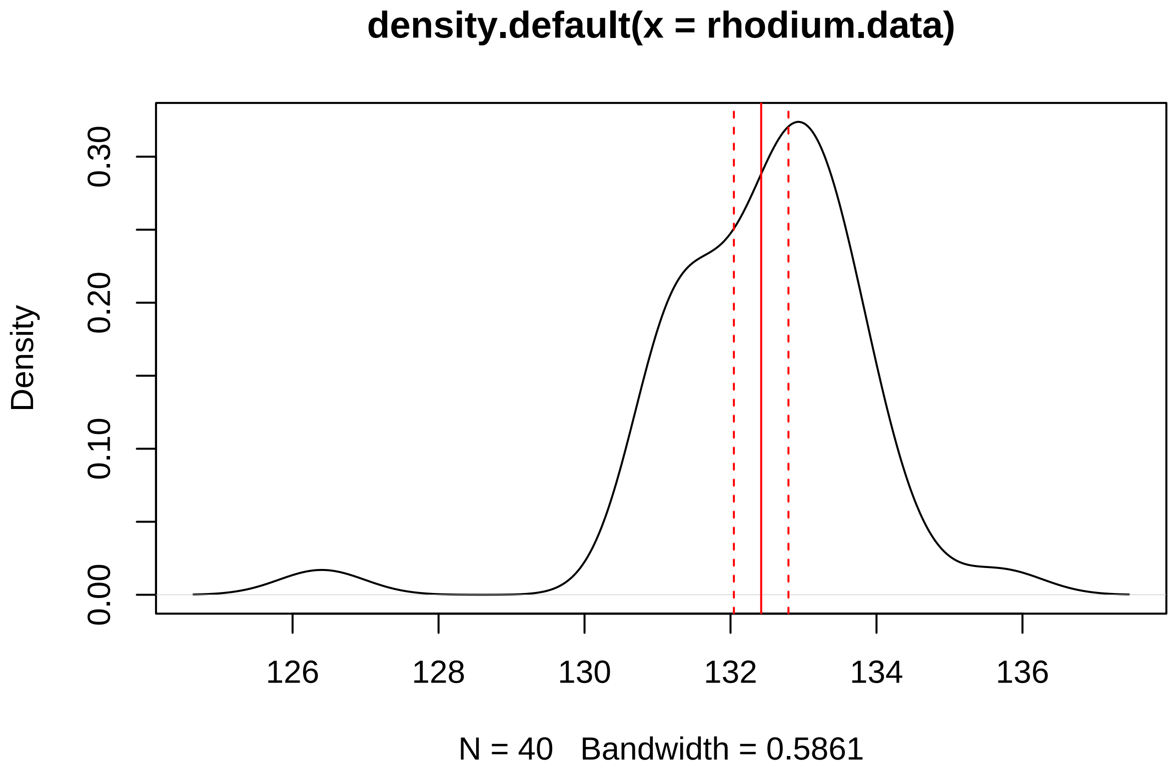
```
# rhodium
n=length(rhodium.data)
(mean.ci.low.rhodium =mean(rhodium.data) - sd(rhodium.data) * qnorm(0.95)/sqrt(n))
```

```
## [1] 132.0461
```

```
(mean.ci.high.rhodium =mean(rhodium.data) + sd(rhodium.data) * qnorm(0.95)/sqrt(n))
```

```
## [1] 132.7939
```

```
plot(density(rhodium.data))
abline(v=mean(rhodium.data), col="red")
abline(v=mean.ci.low.rhodium, col="red", lty="dashed")
abline(v=mean.ci.high.rhodium, col="red", lty="dashed")
```

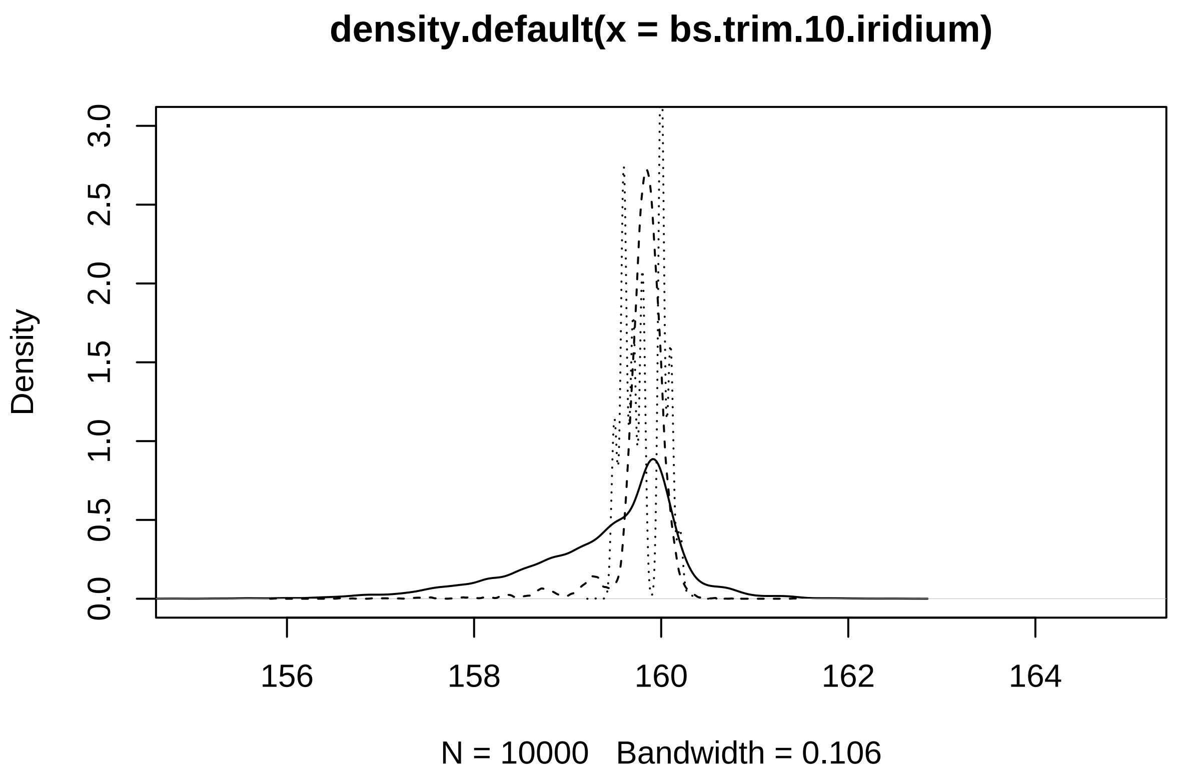


h.

```
computeBootstrapDist = function(data, estimator, B=10000) {
  n=length(data)
  bootstrap.data = matrix(sample(data, B*n, replace=T), nrow=B, ncol=n)
  bs.estimates = apply(bootstrap.data, 1, function(x) {
    estimator(x)
  })
  return(bs.estimates)
  }

# iridium
bs.trim.10.iridium = computeBootstrapDist(iridium.data, function(x){mean(x, trim=0.1)})

se.trim.10 = sd(bs.trim.10.iridium)

bs.trim.20.iridium = computeBootstrapDist(iridium.data, function(x){mean(x, trim=0.2)})
                                                          se.trim.20 = sd(bs.trim.20.iridium)
bs.median.iridium =computeBootstrapDist(iridium.data, function(x){median(x)})
se.median = sd(bs.median.iridium)
(ses = list(se.trim.10=se.trim.10, se.trim.20=se.trim.20, se.median=se.median))
```

```
## $se.trim.10
## [1] 0.8517902
##
## $se.trim.20
## [1] 0.306332
##
## $se.median
## [1] 0.2104986
```

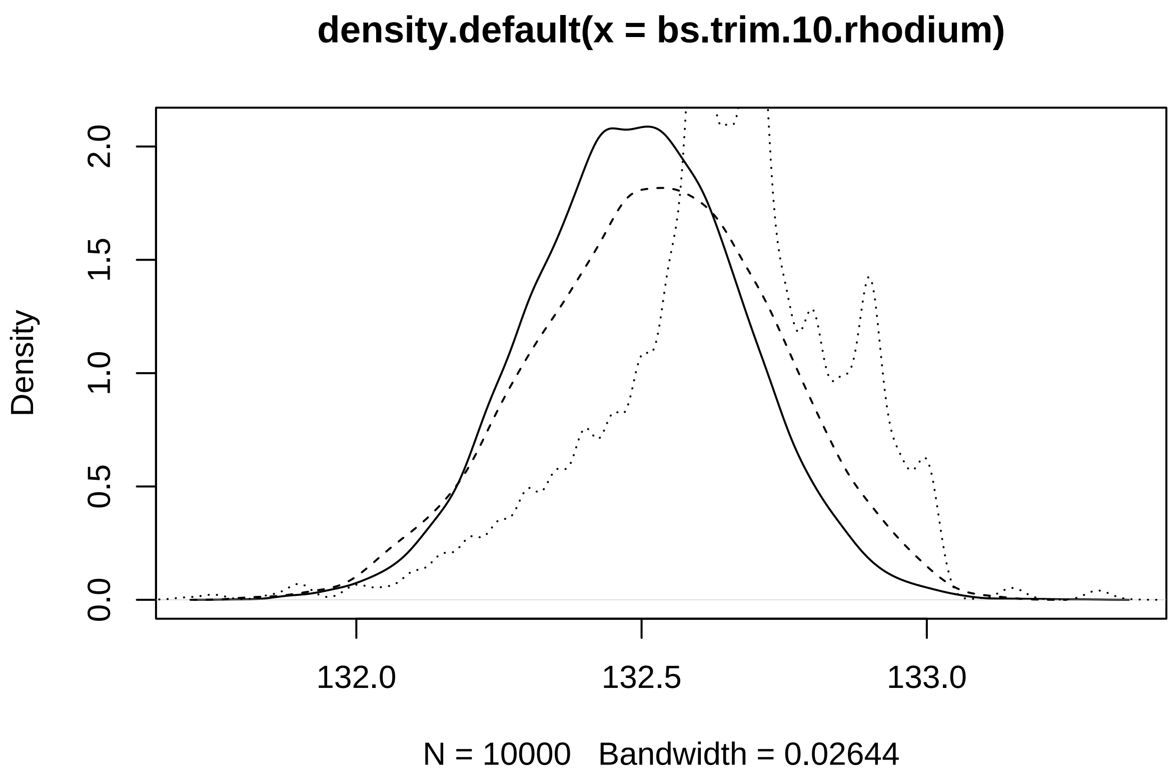
```
plot(density(bs.trim.10.iridium), ylim=c(0,3), xlim=c(155,165))
lines(density(bs.trim.20.iridium), lty="dashed")
lines(density(bs.median.iridium), lty="dotted")
```



```
# rhodium
bs.trim.10.rhodium =computeBootstrapDist(rhodium.data, function(x){mean(x, trim=0.1)})
se.trim.10 = sd(bs.trim.10.rhodium)
bs.trim.20.rhodium =computeBootstrapDist(rhodium.data, function(x){mean(x, trim=0.2)})
se.trim.20 = sd(bs.trim.20.rhodium)
bs.median.rhodium =computeBootstrapDist(rhodium.data, function(x){median(x)})
se.median = sd(bs.median.rhodium)
(ses = list(se.trim.10=se.trim.10, se.trim.20=se.trim.20, se.median=se.median))
```

```
## $se.trim.10
## [1] 0.1853878
##
## $se.trim.20
## [1] 0.2121215
##
## $se.median
## [1] 0.2172929
```

```
plot(density(bs.trim.10.rhodium))
lines(density(bs.trim.20.rhodium), lty="dashed")
lines(density(bs.median.rhodium), lty="dotted")
```



Question 2

my original solution was correct.

Question 3

my original solution was correct.

Question 4

my original solution was correct.