Stat 341 Assignment 1

2022-09-26

Question 1: Basic R Calculations

1a)

```
3^4
```

```
## [1] 81
```

1b)

```
log(100, base = 7) # 1.b)
```

```
## [1] 2.366589
```

1c)

```
x <- seq(1, 100)
sum(sapply(x, function(x) {1/(x^2)}))
```

```
## [1] 1.634984
```

1d)

```
100 %% 7
```

```
## [1] 2
```

1e)

```
dx_steps <- 0.001
x_val <- seq(0, pi/2, by = dx_steps)
sum(sapply(x_val, function(x){ sin(x) * dx_steps }))
```

```
## [1] 0.9997036
```

1f)

```
dx_steps <- 0.001
x_val <- seq(0, 3, by = dx_steps)
sum(sapply(x_val, function(x){ dexp(x, rate = 1/2) * dx_steps }))
```

```
## [1] 0.7771756
```

1g)

```r
f <- function(x) {
  return (x^2 + 3)
}

dx_steps <- 0.0001
x_val <- seq(-2, 2, by = dx_steps)
sum(sapply(x_val, function(x){ f(x) * dx_steps }))
```

```
## [1] 17.33403
```

Question 2: Comparing Spread Attributes

2a)

$$
\begin{aligned}
SD(\mathcal{P}+b) &= \sqrt{\frac{\sum_{u \in \mathcal{P}+b} \left(y_u - (mean(\mathcal{P}+b)\right)^2}{N}} \\
&= \sqrt{\frac{\sum_{u \in \mathcal{P}} \left((y_u + b) - (\overline{y} + b)\right)^2}{N}} \\
&= \sqrt{\frac{\sum_{u \in \mathcal{P}} \left((y_u - \overline{y} + b - b)\right)^2}{N}} \\
&= \sqrt{\frac{\sum_{u \in \mathcal{P}} \left((y_u - \overline{y})\right)^2}{N}} \\
&= SD(\mathcal{P})
\end{aligned}
$$

Hence, Standard Deviation is location invariant.

$$
\begin{aligned}
a(\mathcal{P}+b) = MAD(\mathcal{P}+b) &= \underset{u \in \mathcal{P}+b}{\text{median}} \left| y_u - (\underset{u \in \mathcal{P}+b}{\text{median}} \; y_u) \right|. \\
&= \underset{u \in \mathcal{P}}{\text{median}} \left| y_u + b - (\underset{u \in \mathcal{P}}{\text{median}} \; y_u + b) \right|. \\
&= \underset{u \in \mathcal{P}}{\text{median}} \left| (y_u - \underset{u \in \mathcal{P}}{\text{median}} \; y_u) + b - b \right| \\
&= MAD(\mathcal{P})
\end{aligned}
$$

Hence, Median Absolute Deviation is location invariant.

2b)

$$
\begin{aligned}
SD(\alpha \times \mathcal{P}) &= \sqrt{\frac{\sum_{u \in \alpha \times \mathcal{P}} \left(y_u - (mean(\alpha \times \mathcal{P})\right)^2}{N}} \\
&= \sqrt{\frac{\sum_{u \in \mathcal{P}} \left((\alpha \times y_u) - (\alpha \times \overline{y})\right)^2}{N}} \\
&= \sqrt{\frac{\sum_{u \in \mathcal{P}} \left(\alpha \times (y_u - \overline{y})\right)^2}{N}} \\
&= \sqrt{\frac{\sum_{u \in \mathcal{P}} \alpha^2 \times \left((y_u - \overline{y})\right)^2}{N}} \\
&= \alpha \times SD(\mathcal{P})
\end{aligned}
$$

Hence, Standard Deviation is scale equivariant.

$$
\begin{aligned}
a(\alpha \times \mathcal{P}) = MAD(\alpha \times \mathcal{P}) &= \underset{u \in \alpha \times \mathcal{P}}{\text{median}} \left| y_u - (\underset{u \in \alpha \times \mathcal{P}}{\text{median}} \ y_u) \right|. \\
&= \underset{u \in \mathcal{P}}{\text{median}} \left| (\alpha \times y_u - \underset{u \in \mathcal{P}}{\text{median}} \ \alpha \times y_u) \right| \\
&= \underset{u \in \mathcal{P}}{\text{median}} \left| (\alpha \times (y_u - \underset{u \in \mathcal{P}}{\text{median}} \ y_u)) \right| \\
&= \alpha \times MAD(\mathcal{P})
\end{aligned}
$$

Hence, Median Absolute Deviation is scale equivariant.

2c)

$$
\begin{aligned}
SD(\mathcal{P}^k) &= \sqrt{\frac{\sum_{u \in \mathcal{P}^k} (y_u - (mean(\mathcal{P}^k))^2}{N}} \\
&= \sqrt{\frac{\sum_{u \in \mathcal{P}} (y_u - \overline{y})^2}{N}} \\
&= SD(\mathcal{P})
\end{aligned}
$$

Hence, Standard Deviation is replication invariant.

$$
\begin{aligned}
MAD(\mathcal{P}^k) &= \underset{u \in \mathcal{P}^k}{\text{median}} \left| y_u - (\underset{u \in \mathcal{P}^k}{\text{median}} \ y_u) \right|. \\
&= \underset{u \in \mathcal{P}}{\text{median}} \left| (y_u - \underset{u \in \mathcal{P}}{\text{median}} \ y_u) \right| \\
&= MAD(\mathcal{P})
\end{aligned}
$$

Hence, Median Absolute Deviation is replication invariant.

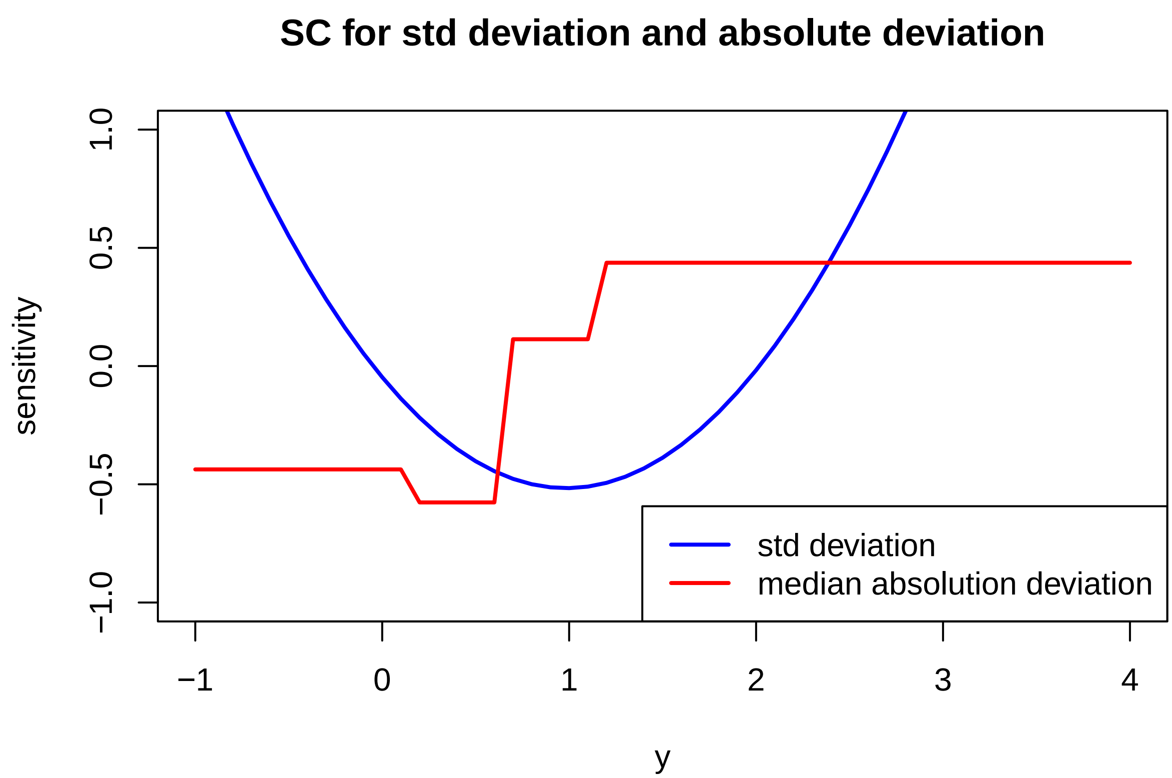
2d)

```r
SD <- function(y) {
  return (sqrt(sum((y - mean(y))^2 / length(y))))
}

MAD <- function(y) {
  return (median(abs(y - median(y))))
}
```

2e)

```r
set.seed(341)
sc <- function(pop, y, attr){
  N <- length(pop) + 1
  sapply (y, function(y.new){ N*(attr(c(y.new, pop)) - attr(pop)) })
}
```

```
set.seed(341)
pop = rexp(1000)

y_val <- seq(-1, 4, by=0.1)

plot(y_val, sc(pop, y_val, SD), type="l", lwd = 2,
     main="SC for std deviation and absolute deviation", ylab="sensitivity", xlab="y",
     xlim=c(-1,4), ylim=c(-1, 1), col="blue")
lines(y_val, sc(pop, y_val, MAD), type="l", lwd = 2, main="Sensitivity curve for the median absolute de

legend(x = "bottomright",          # Position
       legend = c("std deviation", "median absolution deviation"),  # Legend texts
       col = c("blue", "red"),           # Line colors
       lwd = 2)
```



2f)

SD and MAD are both location invariant, scale equivariant, and replication invariant.

SD is however much more sensitive to extreme values since it's sensitivity curve increases without bounds as $y \to \infty$ or $y \to -\infty$.

MAD is not very sensitive to extreme values and it's sensitivity will be bounded and hence, it is a much more robust measure because of its high breakdown point. On the other hand, SD is a fragile attribute and has a very low breakdown point.

It is advantageous to use MAD over SD when we want to limit the effect of outliers on the statistic. It is advantageous to use SD over MAD to gain clarity over the range of variation within the dataset.

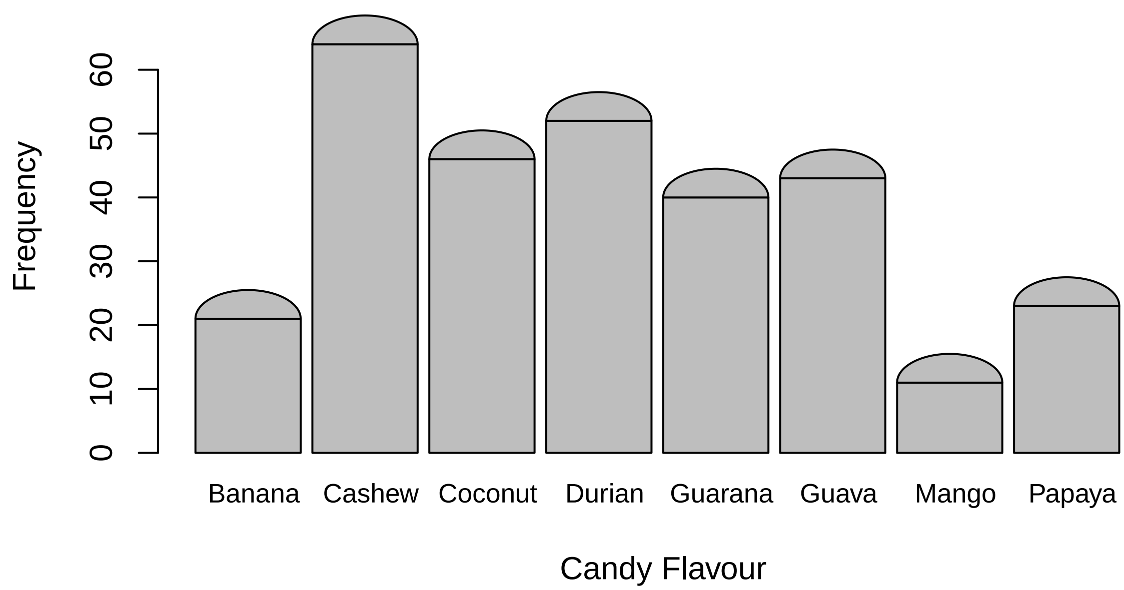
Question 3: Write a rounded-barplot-making function

3a)

```
rounded.barplot <- function(x, xlab){
  table_x <- table(x)
  categories <- names(table_x)
  categories_frequencies <- as.numeric(table_x)

  plot.new()
  plot(NULL, type="n", xlim=c(0, 10*length(categories_frequencies)), ylim=c(0, max(categories_frequenci

  axis(2, at=seq(from=0, to=max(categories_frequencies), by=10))
  mtext(xlab, side=1, line=2)
  mtext("Frequency", side=2, line=3)

  x_semi <- seq(-4.5, 4.5, by=0.01)
  y_semi <- sqrt(20.25-x_semi^2)

  for (i in c(1: length(categories_frequencies))){
    rect(10*(i-1), 0, 10*i-1, categories_frequencies[i], col = "gray", border = "black")
    mtext(categories[i], 1, at=10*i-5, cex=0.85)
    polygon(x_semi + 4.5 + 10*(i-1), y_semi + categories_frequencies[i], col = "gray")
  }
}
```

3b)

```
set.seed(12345)
flavours = c("Mango","Papaya","Banana","Coconut","Guava","Guarana","Durian","Cashew")
candies = sample(flavours, size=300, prob=(1:8)/sum(1:8), replace=TRUE)

rounded.barplot(candies, xlab="Candy Flavour")
```


Frequency
0
10
20
30
40
50
60
Banana
Cashew
Coconut
Durian
Guarana
Guava
Mango
Papaya
Candy Flavour

Question 4: R Analysis Question

4a)

```
setwd("C:/Users/2baja/OneDrive/Desktop/STAT 341/A1")
apartment_eval <- read.csv("Apartment_Building_Evaluation.csv")

score_90 <- apartment_eval[,"SCORE"] >= 90
sum(score_90)
```

```
## [1] 410
```

4b)

```
# 4.b)
davenport <- which(apartment_eval[,"WARDNAME"] == "Davenport")
davenport_apartments <- apartment_eval[davenport,]
davenport_apartments_sorted_addresses <- davenport_apartments[order(-davenport_apartments$SCORE),"SITE_
davenport_apartments_sorted_addresses[c(1:5)]
```

```
## [1] "1544 DUNDAS ST W"  "1544 DUNDAS ST W"  "1289 DUNDAS ST W"
## [4] "19-21 RUSHOLME RD" "410 DOVERCOURT RD"
```

4c)

Scarborough North has the highest score on average: 81.5. River-Black Creek has the lowest score on average: 68.79.

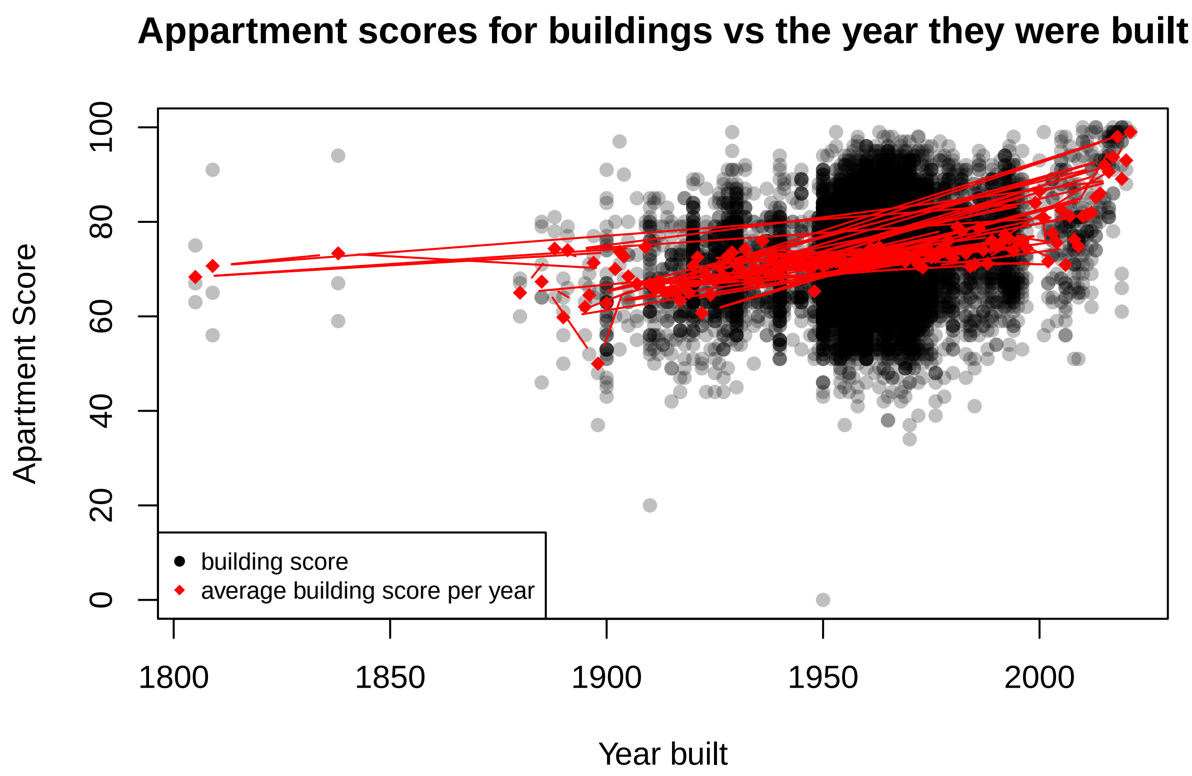
```
unique_wardnames <- unique(apartment_eval[,"WARDNAME"])
sapply(unique_wardnames, function(name) { mean(apartment_eval[which(apartment_eval$WARDNAME == name), "S
```

```
##    Scarborough Southwest       Eglinton-Lawrence    Scarborough-Agincourt
##                 72.03354                72.17902                 78.33333
##        Beaches-East York               Davenport        Spadina-Fort York
##                 72.44581                68.86260                 75.14400
##         Toronto-Danforth          Toronto Centre       Toronto-St. Paul's
##                 73.21563                71.90877                 73.62217
##      University-Rosedale       York South-Weston Humber River-Black Creek
##                 71.81912                70.28017                 68.79331
##               Willowdale    Scarborough-Guildwood       Scarborough Centre
##                 76.86667                72.28054                 74.51587
##         Etobicoke Centre          Don Valley East              York Centre
##                 72.14054                76.30913                 71.53305
##          Don Valley West        Parkdale-High Park      Etobicoke-Lakeshore
##                 76.69196                69.34385                 71.47331
##          Etobicoke North        Scarborough North        Don Valley North
##                 69.30645                81.50000                 79.19310
##   Scarborough-Rouge Park
##                 75.05479
```

4d)

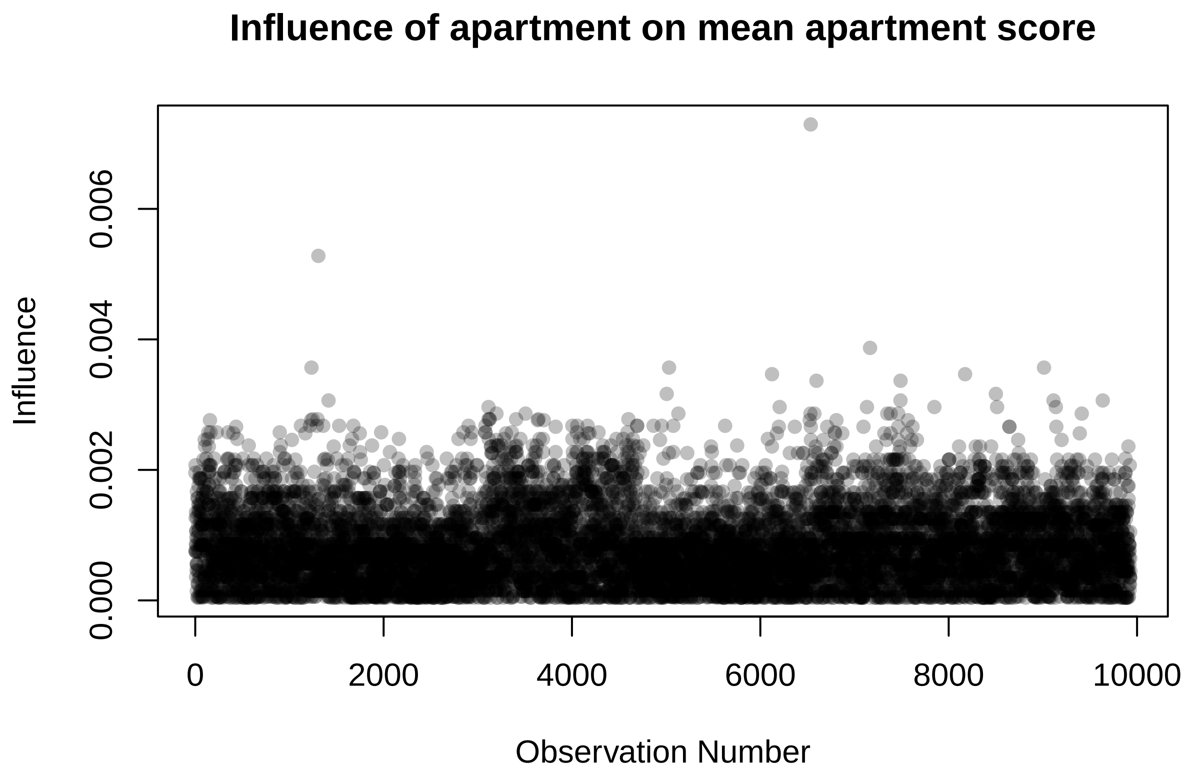
```
plot(apartment_eval$YEAR_BUILT, apartment_eval$SCORE, pch = 16, col=adjustcolor("black", alpha = 0.25),

unique_years <- unique(apartment_eval[,"YEAR_BUILT"])
```

```r
average_score_by_year <- sapply(unique_years, function(year_built) { mean(apartment_eval[which(apartmen

lines(unique_years, average_score_by_year, pch = 18, col="red", type="b")

legend(x = "bottomleft",          # Position
       legend = c("building score", "average building score per year"),  # Legend texts
       col = c("black", "red"),           # Line colors
       cex = 0.75,
       pch = c(16, 18))
```



4e)

```r
influence <- function(pop, attribute){
  N <- length(pop)
  attribute_total_pop <- attribute(pop)

  return (sapply(1:N, function(x) { abs(attribute_total_pop - attribute(pop[-x])) }))
}

plot(1:length(apartment_eval$SCORE), influence(apartment_eval$SCORE, mean), pch = 16, col=adjustcolor("
```



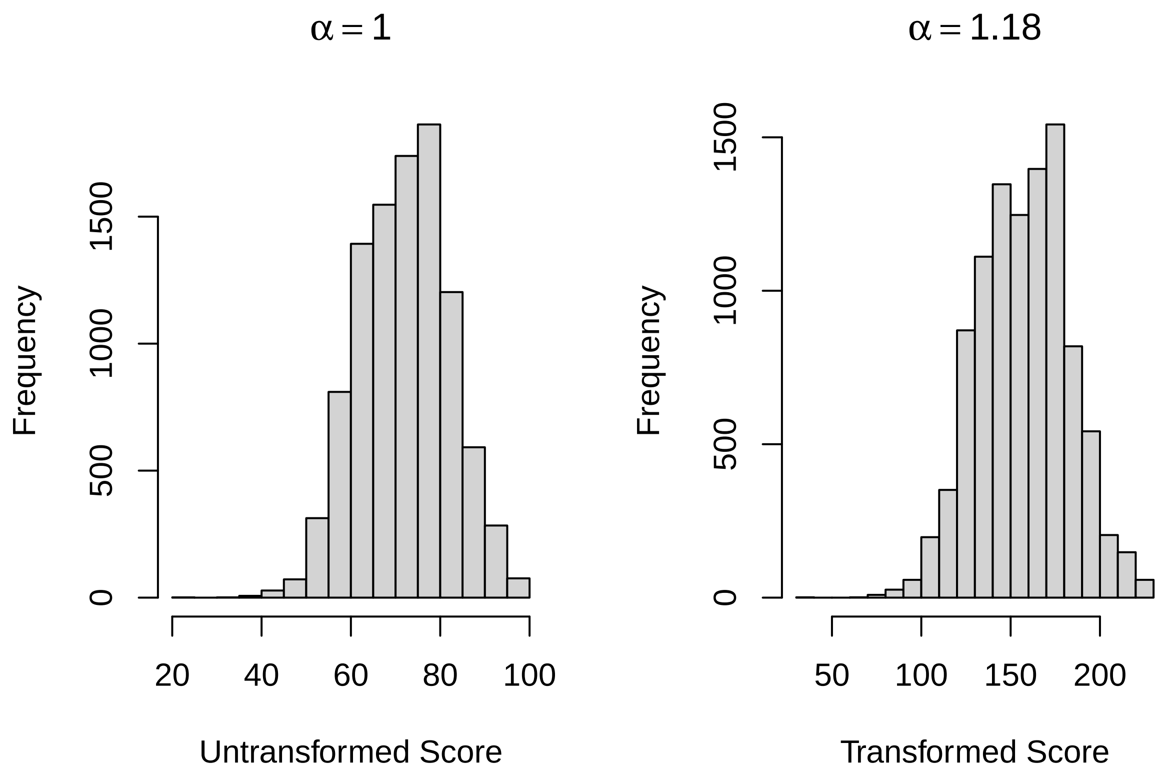
The building with the largest influence has the following observation number:

```
which.max(influence(apartment_eval$SCORE, mean))
```

```
## [1] 6535
```

4.f)

```
apartment_eval_without_outlier <- apartment_eval[-which.max(influence(apartment_eval$SCORE, mean)),]

powerfun <- function(y, alpha) {
  if(sum(y <= 0) > 0) stop("y must be positive")
  if (alpha == 0)
    log(y)
  else if (alpha > 0) {
    y^alpha
  } else -(y^alpha)
}

par(mfrow=c(1,2))
hist(powerfun(apartment_eval_without_outlier$SCORE, 1), main=bquote(alpha == .(1)), xlab="Untransformed
hist(powerfun(apartment_eval_without_outlier$SCORE, 1.18), main=bquote(alpha == .(1.18)), xlab="Transfo
```



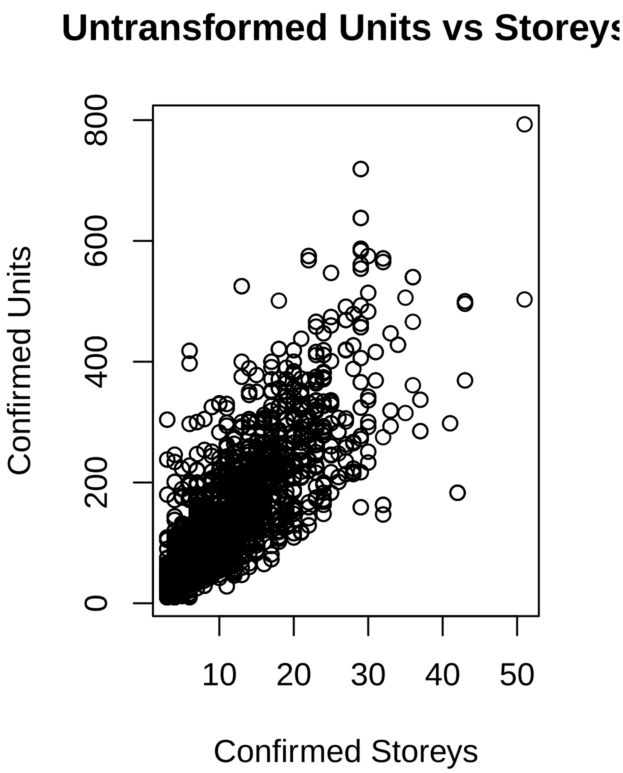
The α that makes the SCORE distribution more symmetric is chosen as 1.18 since the transformed histogram has a skewness that is very close to 0.

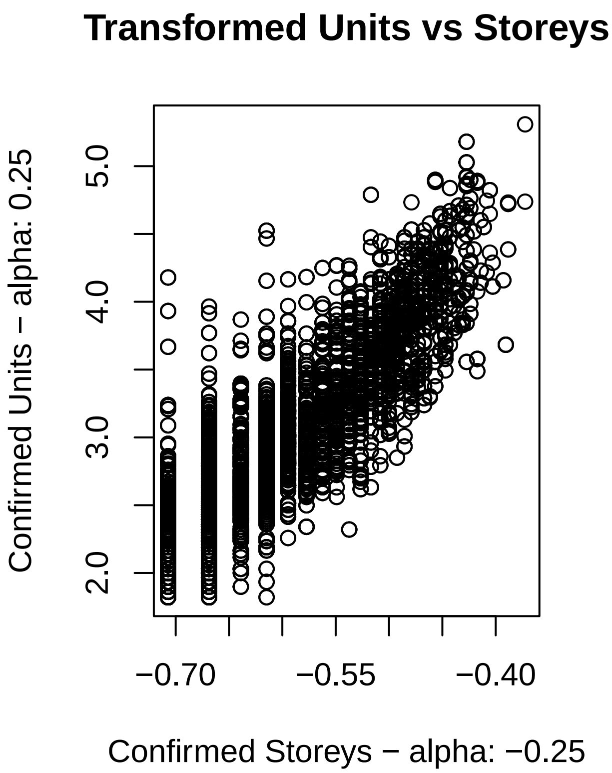
```
library("moments")
skewness(powerfun(apartment_eval_without_outlier$SCORE, 1.18))
```

```
## [1] 0.0006098378
```

4.g)

```
par(mfrow=c(1,2))
plot(apartment_eval_without_outlier$CONFIRMED_STOREYS, apartment_eval_without_outlier$CONFIRMED_UNITS, m
plot(powerfun(apartment_eval_without_outlier$CONFIRMED_STOREYS + 1, -0.25), powerfun(apartment_eval_witl
```





α_x is chosen as -0.25 and α_y is chosen as 0.25.

The corresponding linear regression model shows that transformed plot has an $r^2 = 0.8003$ which depicts a closer linear relationship than the untransformed plot which has an $r^2 = 0.7469$.