GLIB DOLOTOV (U14734371)

Problem Set 1

Problem Set 1: Assigned 01/28/19, Due week of 02/04/19

1. Oscar winners by Genre: From 1927 to 2015, the genres of the films winning Academy Awards can be broken down as follows:

Action-Adventure: 5, Comedy: 11, Drama: 35, Epic: 12, Musical: 9, War: 3, Western: 3

- Construct a table with the class frequency and class percentage for each genre

- Construct a pie chart and a Pareto diagram of this data

- Interpret these plots. Does this suggest a bias on the part of academy voters toward dramas or can you think of another explanation?

2. Textbook problem 2.46, "State SAT scores", page 52.

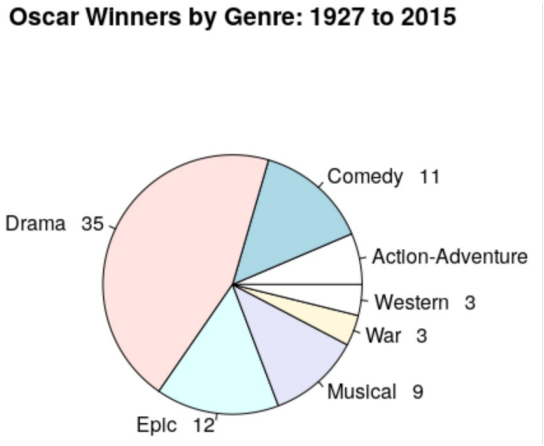
3. Textbook problem 2.66, "Ranking driving performance of professional golfers", page 62.

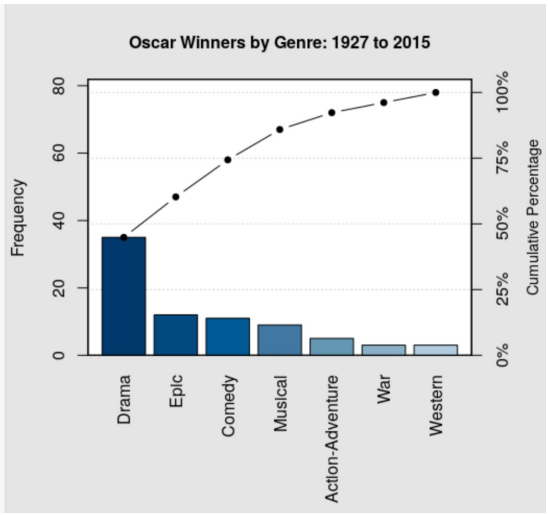
4. Textbook problem 2.72, "Active nuclear power plants", page 64.

5. Using the same PGA data that you used in Problem 3, compute the variances and standard deviations for the set of driving distances, driving accuracy, and driving performance index values (make sure each of your answers indicate the correct units).

1)

```r
library("qcc")

data <- data.frame(
  c("Action-Adventure",
    "Comedy",
    "Drama",
    "Epic",
    "Musical",
    "War",
    "Western"),
  c(5,11,35,12,9,3,3)
)

colnames(data) <- c("genre", "count")
data$percentage <- data$count / sum(data$count) * 100

pie(data$count,
    paste(data$genre, " ", data$count),
    main = "Oscar Winners by Genre: 1927 to 2015")

pareto.data <- data$count
names(pareto.data) <- data$genre
pareto.chart(pareto.data, main = "Oscar Winners by Genre: 1927 to 2015")
```





Alone, these graphs are not enough to suggest a bias towards dramas. It is possible that each year, more dramas get put out than any other film genre, thus skewing the results.

2)

2.46 State SAT scores. Educators are constantly evaluating the efficacy of public schools in the education and training of U.S. students. One quantitative assessment of change over time is the difference in scores on the SAT, which has been used for decades by colleges and universities as one criterion for admission. The **SAT** file contains average SAT scores for each of the 50 states and the District of Columbia for 2011 and 2014. Selected observations are shown in the following table:

State	2011	2014
Alabama	1623	1608
Alaska	1513	1495
Arizona	1539	1551
Arkansas	1692	1697
California	1513	1505
⋮	⋮	⋮
Wisconsin	1767	1771
Wyoming	1692	1757

Based on College Entrance Examination Board, 2014.

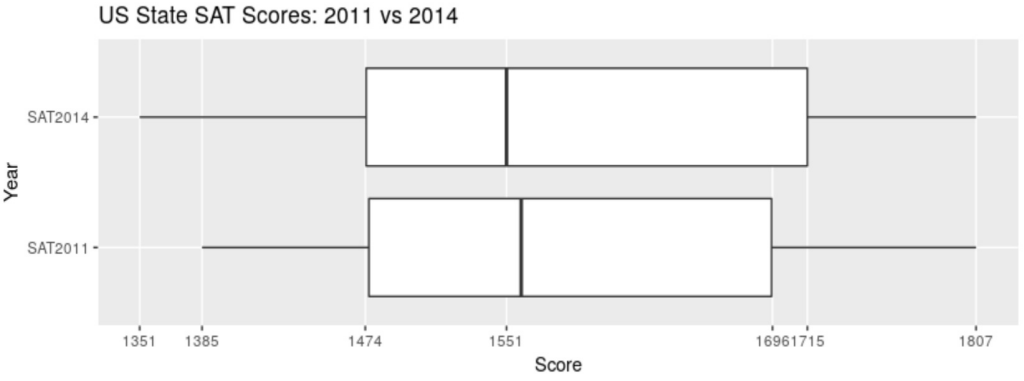
a. Use graphs to display the two SAT score distributions. How have the distributions of state scores changed from 2011 to 2014?

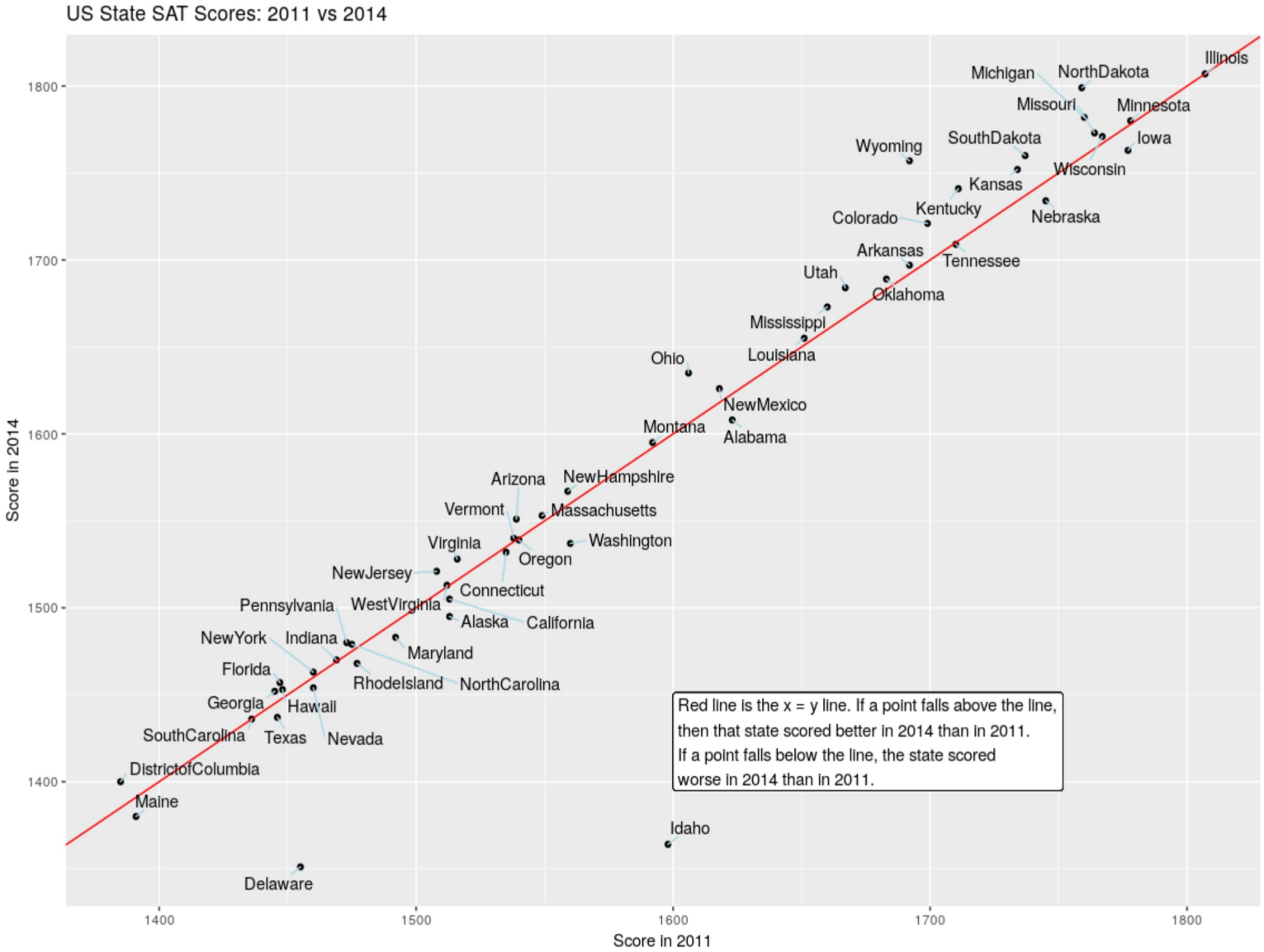
b. As another method of comparing the 2011 and 2014 average SAT scores, compute the **paired difference** by subtracting the 2011 score from the 2014 score for each state. Summarize these differences with a graph.

c. Interpret the graph you made in part **b**. How do your conclusions compare with those of part **a**?

d. Identify the state with the largest improvement in the SAT score between 2011 and 2014.

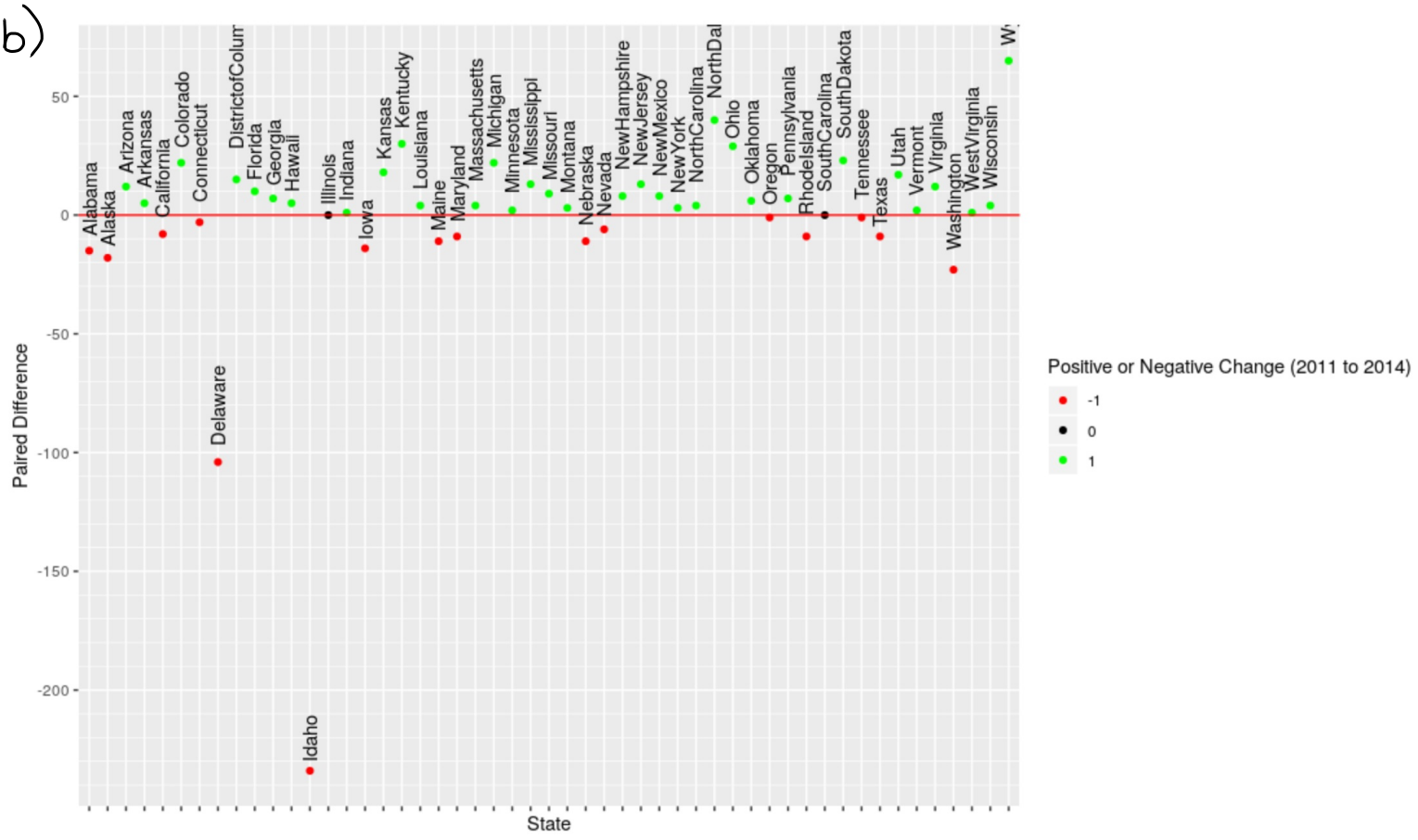
a)

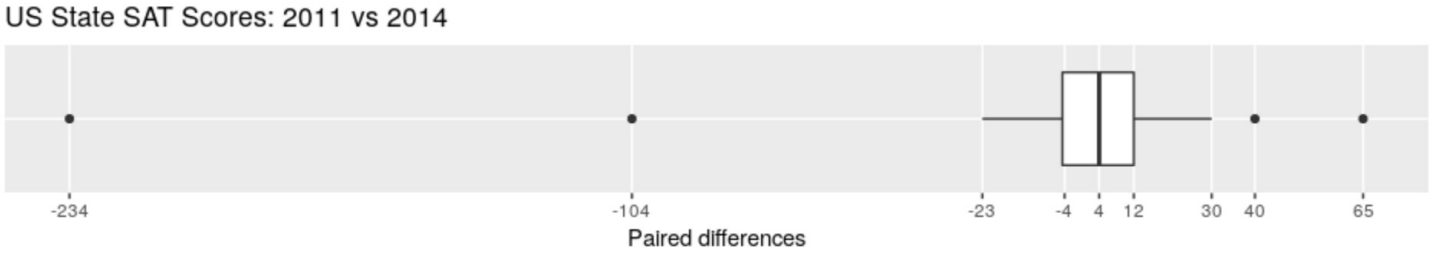




Most states' SAT scores increased, however the median still decreased from 2011 to 2014

b)





c) The second barplot shows us that, for the most part, states' scores improved by a median of 4 points.

hw1.R*

Source on Save | Run | Source

```r
# 2)
library(ggplot2)
library(dplyr)
library(tidyr)
library(ggrepel)

data <- read.csv("SAT.csv")
boxplot(select(data, SAT2011, SAT2014))
# Attempt to do same thing w/ ggplot

ggplot(gather(data, year, score, SAT2011, SAT2014),
       aes(x = year, y = score)) +
  geom_boxplot()

# Box-and-whisker plots
p <- ggplot(gather(data, Year, Score, SAT2011, SAT2014),
            aes(x = Year, y = Score)) + geom_boxplot()
p <- p + coord_flip()
# Select the critical values for axis-lines (quartiles, outliers)
b <- ggplot_build(p)$data %>%
  as.data.frame() %>%
  select(ymin, lower, middle, upper, ymax, outliers) %>%
  unlist() %>%
  as.vector %>% round() %>% sort()
# Remove breaks that would be too close to one another
b <- b[abs(b - head(c(0,b), length(b))) > 10]
p <- p + scale_y_continuous(
  breaks = b,
  minor_breaks = NULL) +
  ggtitle("US State SAT Scores: 2011 vs 2014")
p

ggplot(data, aes(x = data$SAT2011, y = data$SAT2014)) +
  geom_point() +
  geom_abline(slope = 1, col = "red") +
  geom_label(aes(x=1600, y=1395, hjust = 0, vjust = 0,
       label = c('Red line is the x = y line. If a point falls above the line
then that state scored better in 2014 than in 2011.
If a point falls below the line, the state scored
worse in 2014 than in 2011.'))) +
  geom_text_repel(aes(label = data$STATE),
                  box.padding = 0.4,
                  min.segment.length = 0.2,
                  segment.colour = "lightblue",
                  position = position_dodge(width = 0.5)) +
  xlab("Score in 2011") + ylab("Score in 2014") +
  ggtitle("US State SAT Scores: 2011 vs 2014")

```

72:48 (Top Level) R Script

hw1.R*

Source on Save | Run | Source

```r

data$paired.diff <- data$SAT2014 - data$SAT2011
# Box-and-whisker plots
p <- ggplot(data, aes(x = "score", y = paired.diff)) + geom_boxplot() +
  ylab("Paired differences")
p <- p + coord_flip() +
  theme(axis.title.y = element_blank(),
        axis.text.y.left = element_blank(),
        axis.ticks.y = element_blank())
p <- p + scale_y_continuous(
  breaks = ggplot_build(p)$data %>%
    as.data.frame() %>%
    select(ymin, lower, middle, upper, ymax, outliers) %>%
    unlist() %>%
    as.vector %>% round(),
  minor_breaks = NULL) +
  ggtitle("US State SAT Scores: 2011 vs 2014")
p

boxplot(data$paired.diff)

ggplot(data, aes(x = data$STATE, data$paired.diff)) +
  geom_point(aes(colour = as.character(sign(data$paired.diff)))) +
  scale_colour_manual(values = setNames(c('green','red', "black"),
                                        c("1", "-1", "0"))) +
  theme(axis.text.x = element_blank()) +
  geom_hline(yintercept = 0, color = "red") +
  geom_text(aes(label = data$STATE), angle = 90, hjust = -0.2) +
  scale_y_continuous(breaks = seq(-250, 150, 50),
                     minor_breaks = seq(-240, 150, 10)) +
  labs(colour = "Positive or Negative Change (2011 to 2014)") +
  ylab("Paired Difference") + xlab("State")

```

72:48 (Top Level) R Script

2.66 Ranking driving performance of professional golfers. D PGA A group of Northeastern University researchers developed a new method for ranking the total driving performance of golfers on the Professional Golf Association (PGA) tour (*The Sport Journal*, Winter 2007). The method requires knowing a golfer's average driving distance (yards) and driving accuracy (percent of drives that land in the fairway). The values of these two variables are used to compute a driving performance index. Data for the top 40 PGA golfers (ranked by the new method) are saved in the **PGA** file. The first five and last five observations are listed in the accompanying table.

Rank	Player	Driving Distance (yards)	Driving Accuracy (%)	Driving Performance Index
1	Woods	316.1	54.6	3.58
2	Perry	304.7	63.4	3.48
3	Gutschewski	310.5	57.9	3.27
4	Wetterich	311.7	56.6	3.18
5	Hearn	295.2	68.5	2.82
⋮	⋮	⋮	⋮	⋮
36	Senden	291	66	1.31
37	Mickelson	300	58.7	1.30
38	Watney	298.9	59.4	1.26
39	Trahan	295.8	61.8	1.23
40	Pappas	309.4	50.6	1.17

Based on Wiseman, F., et. al. "A new method for ranking total driving performance on the PGA Tour," *Sports Journal*, Vol. 10, No. 1, Winter 2007 (Table 2).

a. Find the mean, median, and mode for the 40 driving performance index values.
b. Interpret each of the measures of central tendency calculated in part **a**.
c. Use the results from part **a** to make a statement about the type of skewness in the distribution of driving performance indexes. Support your statement with a graph.

a) mean = 1.927
median = 1.755
mode = 1.40

b) mean: if another golfer's data were added, we'd expect their DPI to be about 1.755

median: most likely DPI of any given golfer.

```
# 3)
data <- read.csv("PGA.csv")
summary(data$INDEX)
sort(data$INDEX)
```

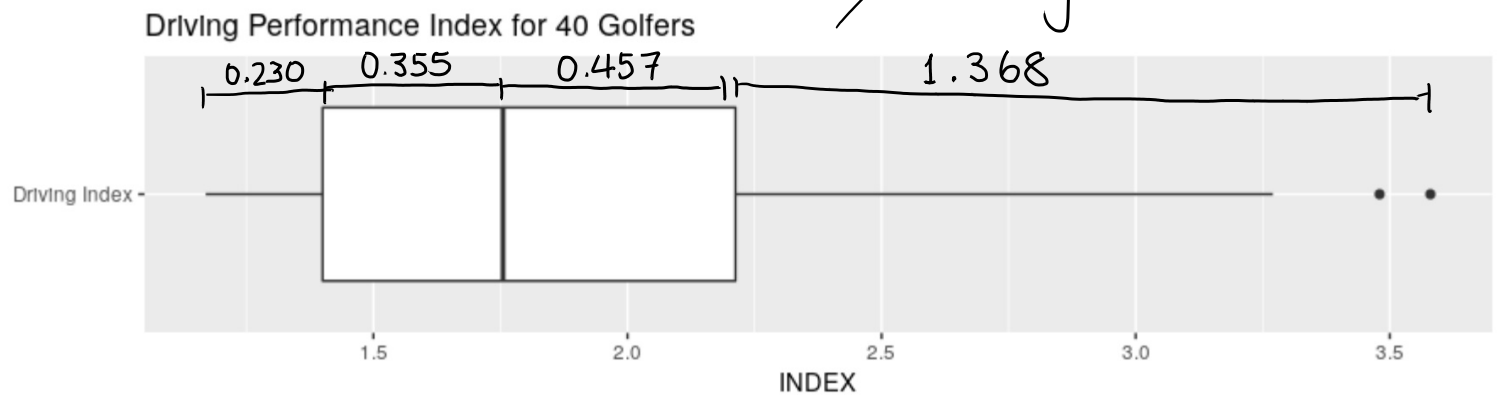
b)

```
> # 3)
> data <- read.csv("PGA.csv")
> summary(data$INDEX)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.170   1.400   1.755   1.927   2.212   3.580
> sort(data$INDEX)
 [1] 1.17 1.23 1.26 1.30 1.31 1.34 1.36 1.37 1.40 1.40 1.40 1.42 1.43 1.49 1.50 1.52 1.56 1.58 1.71 1.75 1.76 1.76 1.85 1.89 1.90
[26] 1.92 2.02 2.20 2.21 2.21 2.22 2.27 2.55 2.74 2.74 2.82 3.18 3.27 3.48 3.58
>
```

c) measures of central tendency:
mean > median ⇒ rightward skewness

```
ggplot(data, aes(x = "Driving Index", y = INDEX)) +
  geom_boxplot() +
  coord_flip() +
  theme(axis.title.y = element_blank()) +
  ggtitle("Driving Performance Index for 40 Golfers")
```

data in top 50% is more widely dispersed ⇒ rightward skewness



2.72 Active nuclear power plants. The U.S. Energy Information Administration monitors all nuclear power plants operating in the United States. The table below lists the number of active nuclear power plants operating in each of a sample of 20 states.

D NUC

a. Find the mean, median, and mode of this data set.

Data for Exercise 2.72

State	Number of Power Plants	State	Number of Power Plants
Alabama	5	New Hampshire	1
Arizona	3	New York	6
California	4	North Carolina	5
Florida	5	Ohio	3
Georgia	4	Pennsylvania	9
Illinois	11	South Carolina	7
Kansas	1	Tennessee	3
Louisiana	2	Texas	4
Massachusetts	1	Vermont	1
Mississippi	1	Wisconsin	3

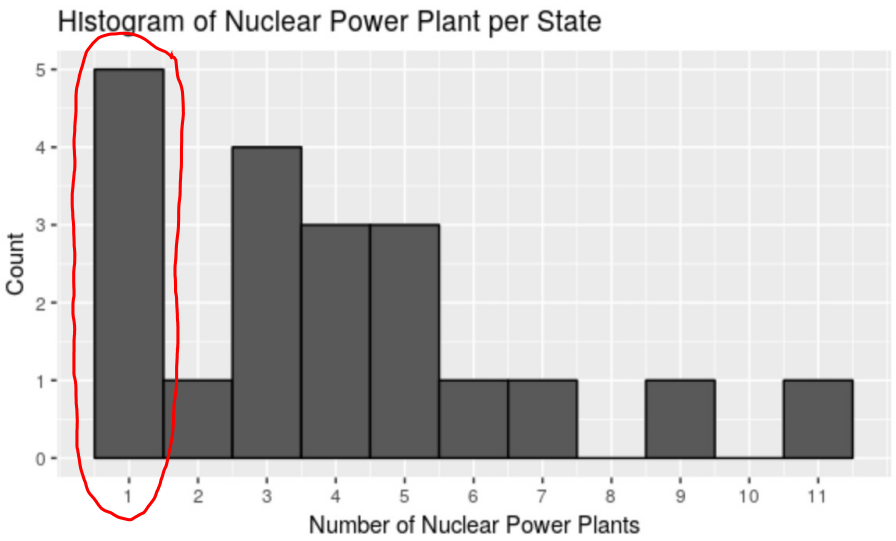
Based on *Statistical Abstract of the United States,* 2012 (Table 942). U.S. Energy Information Administration, *Electric Power Annual.*

b. Eliminate the largest value from the data set and repeat part **a.** What effect does dropping this measurement have on the measures of central tendency found in part **a**?

c. Arrange the 20 values in the table from lowest to highest. Next, eliminate the lowest two values and the highest two values from the data set, and find the mean of the remaining data values. The result is called a *10% trimmed mean*, since it is calculated after removing the highest 10% and the lowest 10% of the data values. What advantages does a trimmed mean have over the regular arithmetic mean?

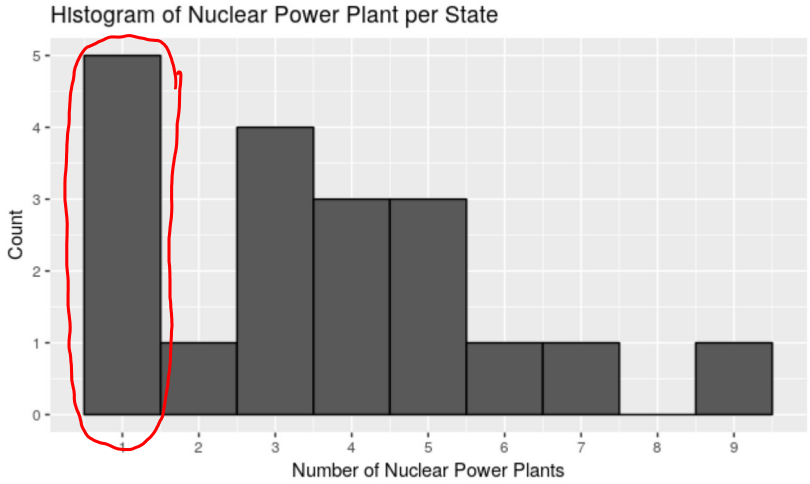
```
> library(ggplot2)
> library(dplyr)
> data <- read.csv("NUC.csv")
> summary(data$PLANTS)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1.00    1.75    3.50    3.95    5.00   11.00
> ggplot(data, aes(x = data$PLANTS)) +
+   geom_histogram(binwidth = 1, col = "black") +
+   scale_x_continuous(breaks = seq(min(data$PLANTS), max(data$PLANTS), 1)) +
+   xlab("Number of Nuclear Power Plants") + ylab("Count") +
+   ggtitle("Histogram of Nuclear Power Plant per State")
```

a) mean: 3.95 mode: 1
median: 3.50




```
> # b) Remove maximum, redo part (a)
> data <- data %>% filter(data$PLANTS < max(data$PLANTS))
> summary(data$PLANTS)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   1.500   3.000   3.579   5.000   9.000
> ggplot(data, aes(x = data$PLANTS)) +
+   geom_histogram(binwidth = 1, col = "black") +
+   scale_x_continuous(breaks = seq(min(data$PLANTS), max(data$PLANTS), 1)) +
+   xlab("Number of Nuclear Power Plants") + ylab("Count") +
+   ggtitle("Histogram of Nuclear Power Plant per State")
```

b) mean: 3.579
median: 3.000
mode: 1
dropping the highest measurement decreased the measures of central tendency



c) Trimmed statistics are likely to eliminate outliers, possibly producing a more accurate guess of the population parameter.

```
> # c) Trimmed mean
> data <- read.csv("NUC.csv")
> sort(data$PLANTS)
 [1]  1  1  1  1  1  2  3  3  3  3  4  4  4  5  5  5  6  7  9 11
> sort(data$PLANTS)[3:18]
 [1] 1 1 1 2 3 3 3 3 4 4 4 5 5 5 6 7
> (1+1+1+2+3+3+3+3+4+4+4+5+5+5+6+7)/16
[1] 3.5625
> mean(data$PLANTS, trim = 0.1)
[1] 3.5625
> mean(sort(data$PLANTS)[3:18])
[1] 3.5625
```

5)

```
> # 5)
> data <- read.csv("PGA.csv")
> var(data$DISTANCE)
[1] 56.62718
> var(data$ACCURACY)
[1] 27.30882
> var(data$INDEX)
[1] 0.4358892
> sd(data$DISTANCE)
[1] 7.525103
> sd(data$ACCURACY)
[1] 5.225784
> sd(data$INDEX)
[1] 0.660219
```

$$s^2 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n - 1}$$

$$s^2 = \frac{\sum_{i=1}^{n} x_i^2 - \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n}}{n - 1}$$

```
> v <- sum((data$DISTANCE - mean(data$DISTANCE))^2) / (length(data$DISTANCE) - 1)
> v2 <- (sum(data$DISTANCE^2) - sum(data$DISTANCE)^2/length(data$DISTANCE)) /
+   (length(data$DISTANCE) - 1)
> c(v,v2)
[1] 56.62718 56.62718
> sd <- sqrt(v)
> sd
[1] 7.525103
```

DISTANCE: $s^2 = 56.627\ yd^2$

$s = 7.525\ yds.$

```
> v <- sum((data$ACCURACY - mean(data$ACCURACY))^2) / (length(data$ACCURACY) - 1)
> v2 <- (sum(data$ACCURACY^2) - sum(data$ACCURACY)^2/length(data$ACCURACY)) /
+   (length(data$ACCURACY) - 1)
> c(v,v2)
[1] 27.30882 27.30882
> sd <- sqrt(v)
> sd
[1] 5.225784
```

ACCURACY: $s^2 = 27.309\ (\%^2)$

$s = 5.226\ \%$

```
> v <- sum((data$INDEX - mean(data$INDEX))^2) / (length(data$INDEX) - 1)
> v2 <- (sum(data$INDEX^2) - sum(data$INDEX)^2/length(data$INDEX)) /
+   (length(data$INDEX) - 1)
> c(v,v2)
[1] 0.4358892 0.4358892
> sd <- sqrt(v)
> sd
[1] 0.660219
```

INDEX: $s^2 = 0.439$

$s = 0.660$