STATISTICS : Informed Decisions Using Data

Chapter - 2 : Organizing and Summarizing Data

Shivam Agrawal

19 March 2019

Defination 1

A **frequency distribution** lists each category of data and the number of occurrences for each category of data.

Example 1

Organizing Qualitative Data into a Frequency Distribution.

```
# Data

table_1 <- c("Back","Back", "Hand", "Wrist", "Back", "Groin", "Elbow", "Back", "Back", "Back", "Shoulder",
"Shoulder", "Hip", "Knee", "Hip", "Neck", "Knee", "Knee", "Shoulder", "Shoulder", "Back", "Back", "Back", "
Back", "Knee", "Knee", "Back", "Hand", "Back", "Wrist")

# Convert data into factor

table_1 <- as.factor(table_1)

# Freuency table

freq_table <- data.frame(table(table_1))

# Output

freq_table
```

Defination 2

The **relative frequency** is the proportion (or percent) of observations within a category and is found using the formula

$$ \ Relative \ frequency = \frac{frequency}{sum\ of\ all \ frequencies} $$

A **relative frequency distribution** lists each category of data together with the relative frequency.

Example 2

Constructing a Relative Frequency Distribution of Qualitative Data.

```
# Total frequency

t_f <- sum(freq_table$Freq)

# Relative Frequency

r_f <- round(freq_table$Freq/t_f, 2)
freq_table <- cbind(freq_table, r_f)

# Output

freq_table
```

Defination 3

A **bar graph** is constructed by labeling each category of data on either the horizontal or vertical axis and the frequency or relative frequency of the category on the other axis. Rectangles of equal width are drawn for each category. The height of each rectangle represents the category's frequency or relative frequency.

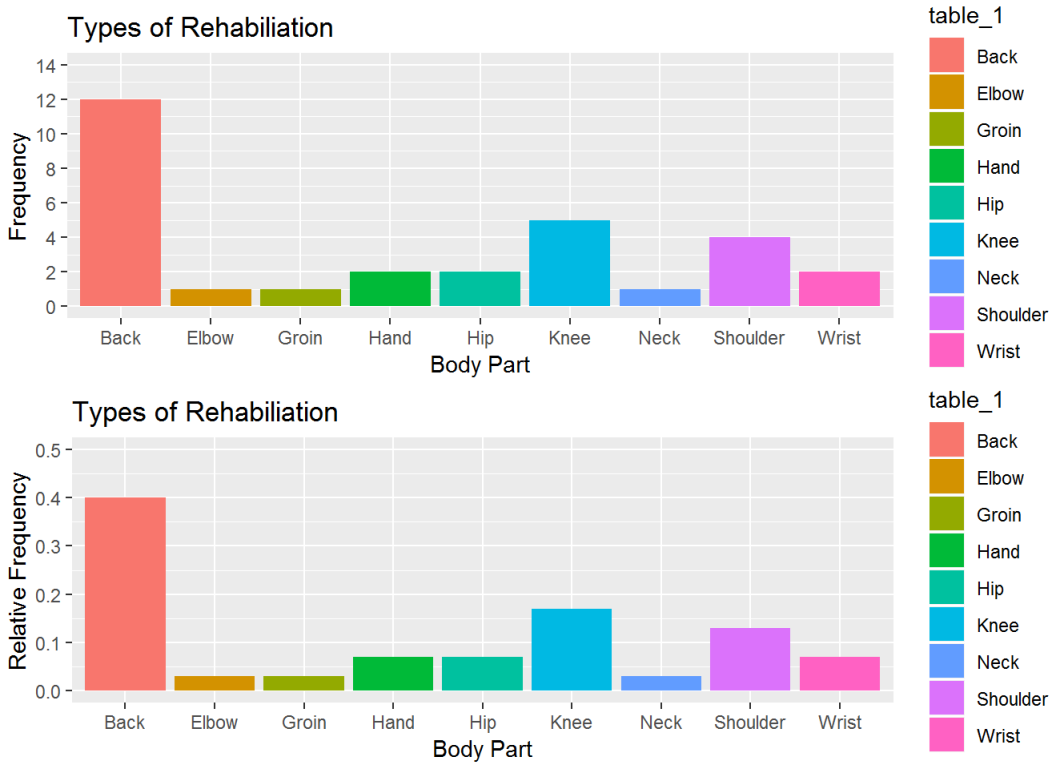
Example 3

Constructing a Frequency and a Relative Frequency Bar Graph.

```r
# Library

library(ggplot2)

# Frequency Bar Graph

p1 <- ggplot(data = freq_table, aes(x = table_1, y = Freq, fill = table_1)) +
  geom_bar(stat = "identity") +
  ggtitle("Types of Rehabiliation") +
  xlab("Body Part") +
  ylab("Frequency") +
  scale_y_continuous(limits = c(0,14), breaks = seq(from = 0, to = 14, by = 2))

# Relative Frequency Bar Graph

p2 <- ggplot(data = freq_table, aes(x = table_1, y = r_f, fill = table_1)) +
  geom_bar(stat = "identity") +
  ggtitle("Types of Rehabiliation") +
  xlab("Body Part") +
  ylab("Relative Frequency") +
  scale_y_continuous(limits = c(0,0.5), breaks = seq(from = 0, to = 0.50, by = 0.10))

# Output

require(gridExtra)
```

```
## Loading required package: gridExtra
```

```r
grid.arrange(p1, p2, nrow = 2)
```



Defination 4

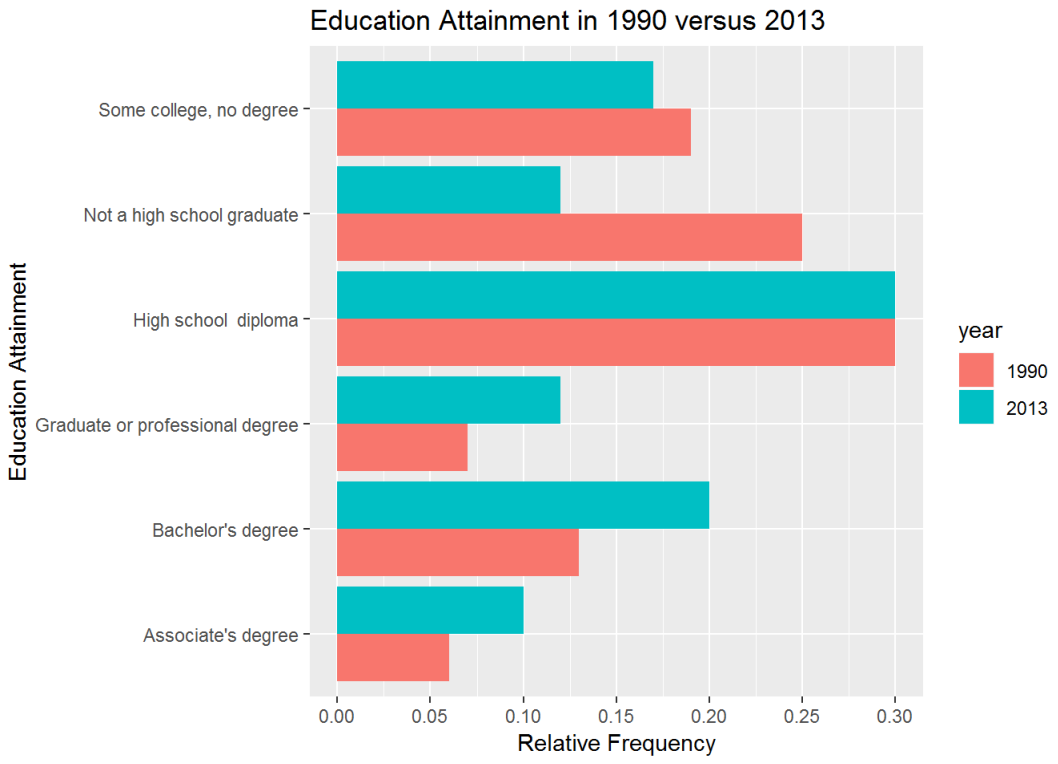
A **Pareto chart** is a bar graph whose bars are drawn in decreasing order of frequency or relative frequency.

Example 4

Comparing Two Data Sets.

```r
# Data

table_2 <- data.frame(Education_Attainment = c("Not a high school graduate",
                                               "High school  diploma",
                                               "Some college, no degree",
                                               "Associate's degree",
                                               "Bachelor's degree",
                                               "Graduate or professional degree"),
                      y_1990 = c(39344, 47643, 29780, 9792, 20833, 11478),
                      y_2013 = c(24517, 61704, 34805, 20367, 41575, 23931))

# Relative Frequencies

t_f_y1990 <- sum(table_2$y_1990)
r_f_y1990 <- round(table_2$y_1990/t_f_y1990, 2)

t_f_y2013 <- sum(table_2$y_2013)
r_f_y2013 <- round(table_2$y_2013/t_f_y2013, 2)

table_2 <- cbind(table_2, r_f_y1990, r_f_y2013)

table_2
```

```r
# Side-by-side Bar Graphs

Education_Attainment <- rep(table_2$Education_Attainment,2)
year <- c(rep(1990, 6), rep(2013, 6))
year <- as.factor(year)
values <- c(table_2$y_1990, table_2$y_2013)
r_f <- c(table_2$r_f_y1990, table_2$r_f_y2013)

table_3 <- data.frame(Education_Attainment, year, values, r_f)

ggplot(table_3, aes(x = Education_Attainment, r_f, fill = year)) +
  geom_bar(stat = "identity", position = "dodge") +
  xlab("Education Attainment") +
  ylab("Relative Frequency") +
  ggtitle("Education Attainment in 1990 versus 2013") +
  scale_y_continuous(limits = c(0, 0.30), breaks = seq(from = 0, to = 0.30, by = 0.05)) +
  coord_flip()
```



Defination 5

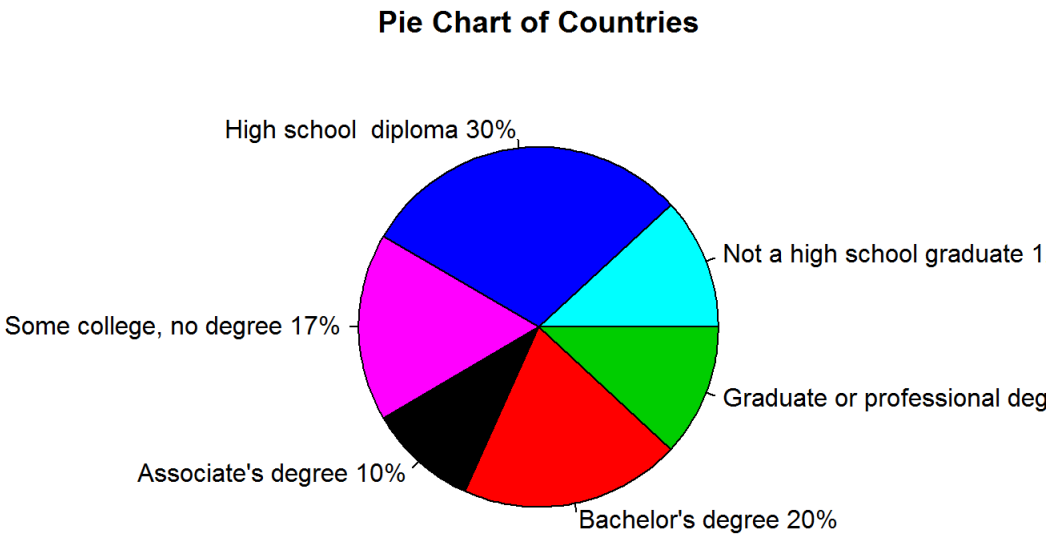
A **pie chart** is a circle divided into sectors. Each sector represents a category of data. The area of each sector is proportional to the frequency of the category.

Example 5

Constructing a Pie Chart.

```
# Data

table_4 <- data.frame(Education_Attainment = c("Not a high school graduate",
                                               "High school  diploma",
                                               "Some college, no degree",
                                               "Associate's degree",
                                               "Bachelor's degree",
                                               "Graduate or professional degree"),
                      y_2013 = c(24517, 61704, 34805, 20367, 41575, 23931))

# Pie Chart

r_f <- round(table_4$y_2013/sum(table_4$y_2013), 2)
degree <- round(r_f * 360)

table_4 <- cbind(table_4, r_f, degree)
table_4
```

```
pct <- round(table_4$r_f/sum(table_4$r_f)*100)
lbls <- paste(table_4$Education_Attainment, pct)
lbls <- paste(lbls,"%",sep="")

pie(table_4$r_f,
    labels = lbls,
    col=table_4$Education_Attainment,
    main="Pie Chart of Countries")
```



Example 6

Constructing Frequency and Relative Frequency Distributions from Discrete Data.

```
# Data

arrivals <- c(7, 6, 6, 6, 4, 6, 2, 6, 5, 6, 6, 11, 4, 5, 7, 6, 2, 7, 1, 2, 4, 8, 2, 6, 6, 5, 5,
3, 7, 5, 4, 6, 2, 2, 9, 7, 5, 9, 8, 5)

DF <- data.frame(table(arrivals))
r_f <- round(DF$Freq/sum(DF$Freq), 3)
DF <- cbind(DF, r_f)

# Output

DF
```

Defination 6

A **histogram** is constructed by drawing rectangles for each class of data. The height of each rectangle is the frequency or relative frequency of the class. The width of each rectangle is the same and the rectangles touch each other.

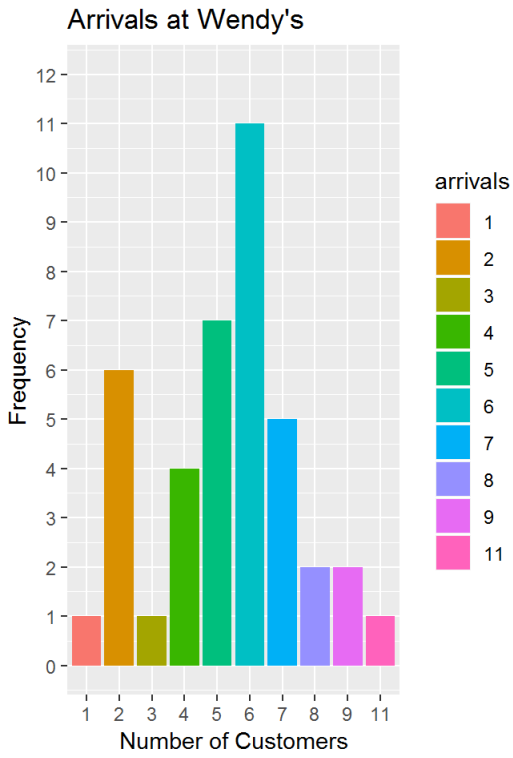
Example 7

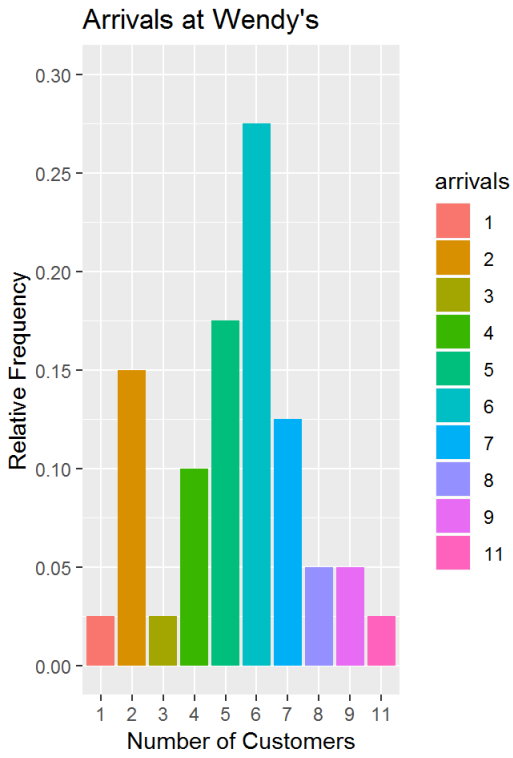
Drawing a Histogram for Discrete Data.

```
# Frequency Histogram

p1 <- ggplot(data = DF, aes(x = arrivals, y = Freq, fill = arrivals)) +
  geom_histogram(stat = "identity") +
  ggtitle("Arrivals at Wendy's") +
  xlab("Number of Customers") +
  ylab("Frequency") +
  scale_y_continuous(limits = c(0,12), breaks = seq(from = 0, to = 12, by = 1))
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```

```
# Relative Frequency Histogram

p2 <- ggplot(data = DF, aes(x = arrivals, y = r_f, fill = arrivals)) +
  geom_bar(stat = "identity") +
  ggtitle("Arrivals at Wendy's") +
  xlab("Number of Customers") +
  ylab("Relative Frequency") +
  scale_y_continuous(limits = c(0,0.3), breaks = seq(from = 0, to = 0.3, by = 0.05))

# Output

require(gridExtra)
grid.arrange(p1, p2, ncol = 2)
```





Example 8

Organizing Continuous Data into a Frequency and Relative Frequency Distribution.

```
# Data

funds <- c(10.94, 14.60, 12.80, 16.00, 11.93, 15.68, 9.03, 13.40, 10.53, 13.98, 13.86, 12.36,
13.54, 9.94, 13.93, 13.63, 14.12, 14.88, 14.77, 13.13, 8.28, 19.43, 12.98, 13.16,
           12.26, 14.20, 14.80, 13.26, 13.67, 10.08, 14.86, 8.71, 12.17, 10.26, 15.22, 13.36,
13.55, 13.90, 15.64, 12.80)

# Procedure for Continuous Data Frequency

range(funds)
```

```
## [1]  8.28 19.43
```

```
breaks <- seq(8, 20, by = 0.99)
cuts <- cut(funds, breaks, right =  T)
DF2 <- data.frame(table(cuts))

r_f <- round(DF2$Freq/sum(DF2$Freq), 3)
DF2 <- cbind(DF2, r_f)

# Output

DF2    # Output is different from the book
```

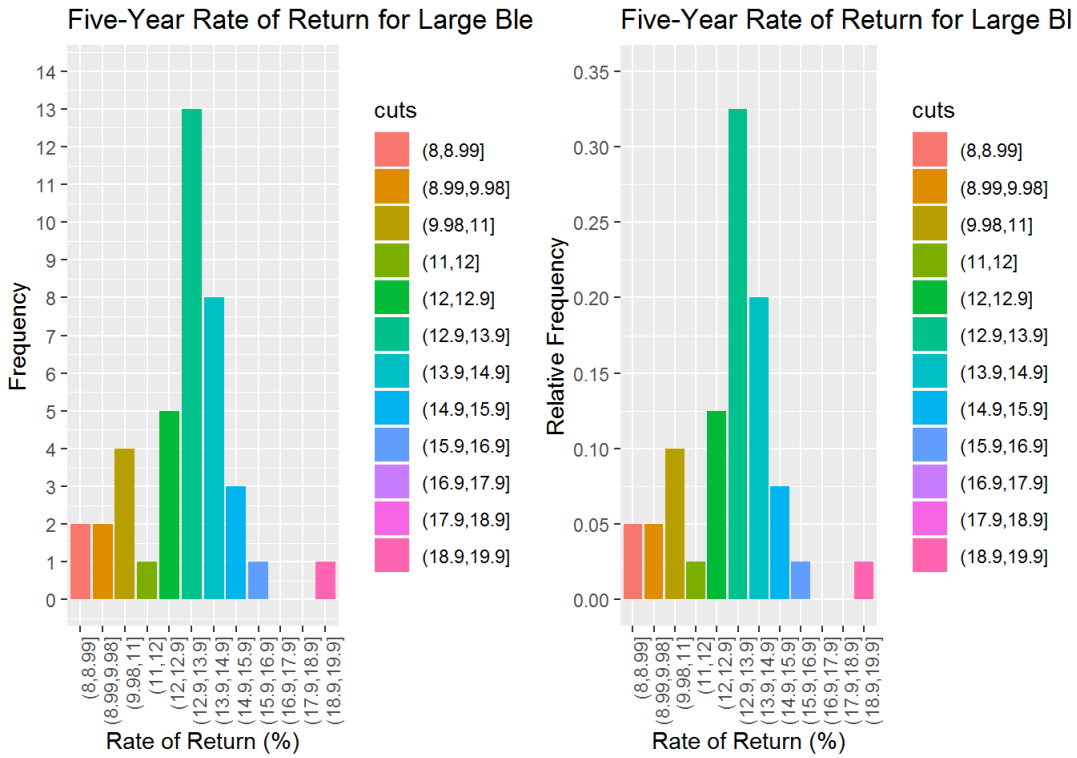
Example 9

Drawing a Histogram for Continuous Data.

```
# Frequency Histogram

p1 <- ggplot(data = DF2, aes(x = cuts, y = Freq, fill = cuts)) +
  geom_histogram(stat = "identity") +
  ggtitle("Five-Year Rate of Return for Large Blended Mutual Funds") +
  xlab("Rate of Return (%)") +
  ylab("Frequency") +
  scale_y_continuous(limits = c(0,14), breaks = seq(from = 0, to = 14, by = 1)) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```

```
# Relative Frequency Histogram

p2 <- ggplot(data = DF2, aes(x = cuts, y = r_f, fill = cuts)) +
  geom_bar(stat = "identity") +
  ggtitle("Five-Year Rate of Return for Large Blended Mutual Funds") +
  xlab("Rate of Return (%)") +
  ylab("Relative Frequency") +
  scale_y_continuous(limits = c(0,0.35), breaks = seq(from = 0, to = 0.35, by = 0.05)) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))

# Output

require(gridExtra)
grid.arrange(p1, p2, ncol = 2)
```



Example 10

Constructing a Stem-and-Leaf Plot.

```
# Data

percent <- c(16.4, 18.9, 10.6, 10.5, 20.2, 14.5, 19.6, 12.5, 16.0, 18.6, 10.1, 14.3, 15.4, 11.6,
13.1, 11.2, 14.1, 13.6, 10.8, 11.0, 16.3, 13.7, 22.2, 11.5, 19.9, 14.5, 18.4, 15.1,              14.0, 16.9
, 17.2, 11.6, 9.6, 12.5, 16.6, 10.0, 13.7, 8.6, 10.5, 12.9, 10.2, 11.8,               13.4, 21.0, 17.0, 10.
5, 15.9, 11.2, 13.6, 17.9, 10.7)

stem(percent, scale = 2)
```

```
##
##   The decimal point is at the |
##
##    8 | 6
##    9 | 6
##   10 | 012555678
##   11 | 0225668
##   12 | 559
##   13 | 146677
##   14 | 01355
##   15 | 149
##   16 | 03469
##   17 | 029
##   18 | 469
##   19 | 69
##   20 | 2
##   21 | 0
##   22 | 2
```

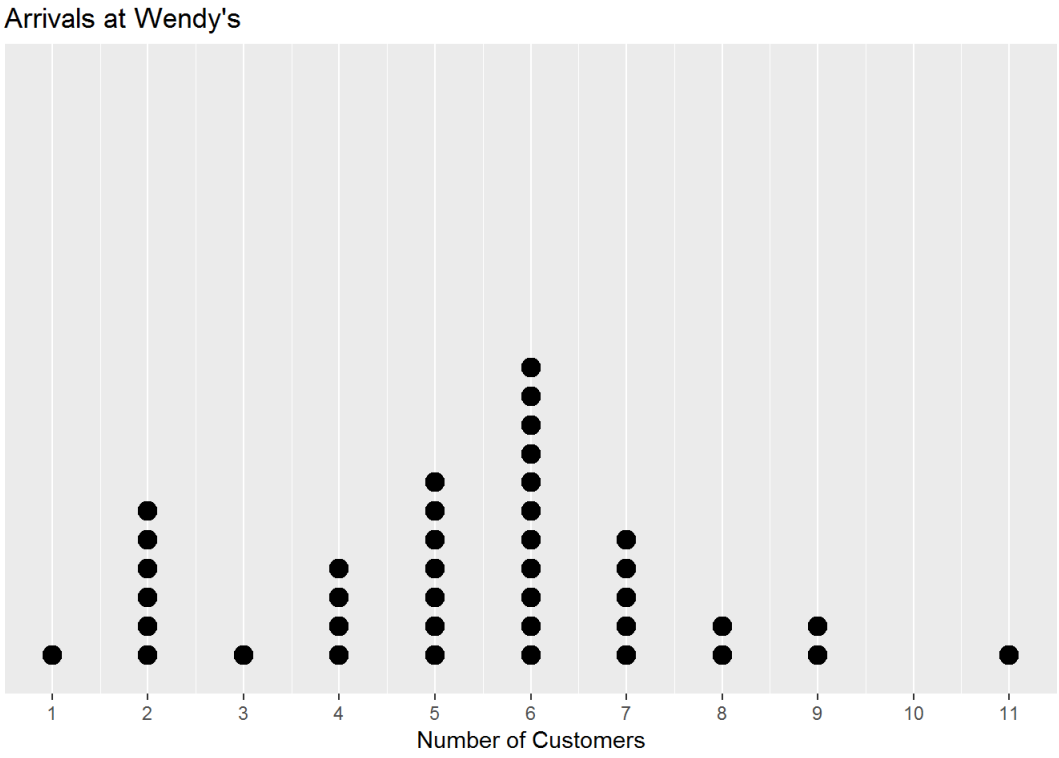
Example 11

Drawing a Dot Plot.

```
# Data

arrivals
```

```
##  [1]  7  6  6  6  4  6  2  6  5  6  6 11  4  5  7  6  2  7  1  2  4  8  2
## [24]  6  6  5  5  3  7  5  4  6  2  2  9  7  5  9  8  5
```

```
ggplot(data = data.frame(arrivals), aes(x = arrivals)) +
  geom_dotplot(binwidth = 0.2, stackratio = 1.5) +
  scale_y_continuous(NULL, breaks = NULL) +
  scale_x_continuous(limits = c(1, 11), breaks = c(seq(from = 1, to = 11, by = 1))) +
  xlab("Number of Customers") +
  ggtitle("Arrivals at Wendy's")
```



Defination 7

A **class midpoint** is the sum of consecutive lower class limits divided by 2.

Defination 8

A **frequency polygon** is a graph that uses points, connected by line segments, to represent the frequencies for the classes. It is constructed by plotting a point above each class midpoint on a horizontal axis at a height equal to the frequency of the class. Next, line segments are drawn connecting consecutive points. Two additional line segments are drawn connecting each end of the graph with the horizontal axis.

Defination 9

A **cumulative frequency distribution** displays the aggregate frequency of the category. In other words, for discrete data, it displays the total number of observations less than or equal to the category. For continuous data, it displays the total number of observations less than or equal to the upper class limit of a class.

A **cumulative relative frequency distribution** displays the proportion (or percentage) of observations less than or equal to the category for discrete data and the proportion (or percentage) of observations less than or equal to the upper class limit for continuous data.

Defination 10

An **ogive** (read as “oh jive”) is a graph that represents the cumulative frequency or cumulative relative frequency for the class. It is constructed by plotting points whose x-coordinates are the upper class limits and whose y-coordinates are the cumulative frequencies or cumulative relative frequencies of the class. Then line segments are drawn connecting consecutive points. An additional line segment is drawn connecting the first point to the horizontal axis at a location representing the upper limit of the class that would precede the first class (if it existed).

Defination 11

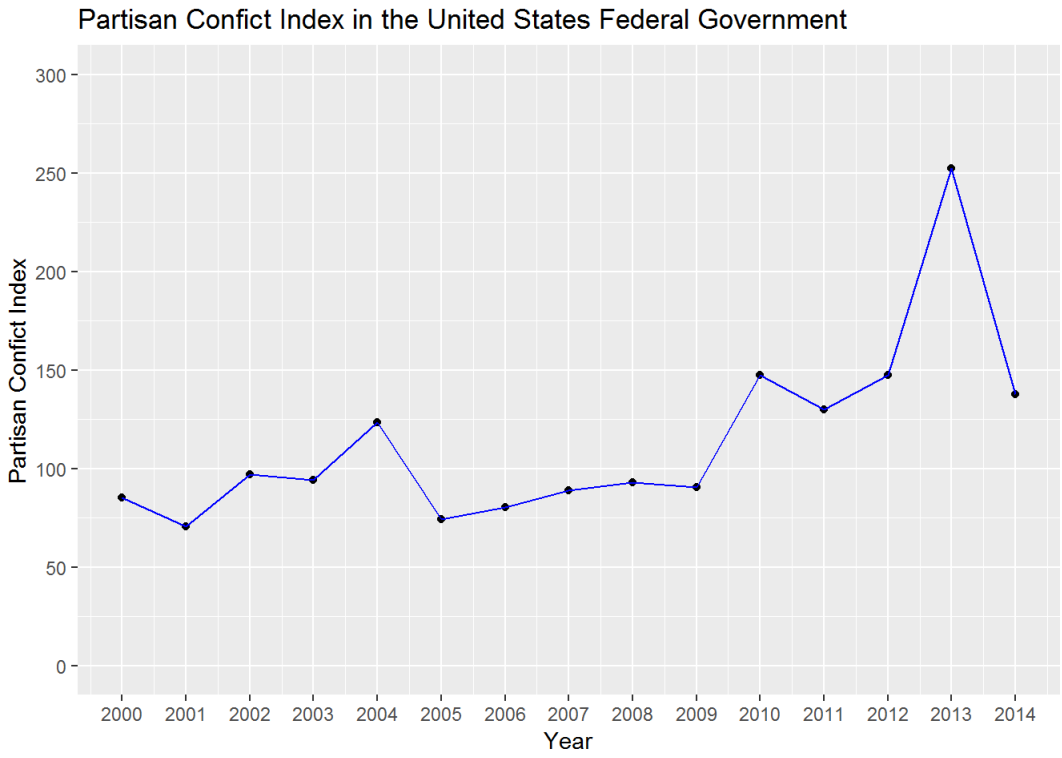
A **time-series plot** is obtained by plotting the time in which a variable is measured on the horizontal axis and the corresponding value of the variable on the vertical axis. Line segments are then drawn connecting the points.

Example 12

Drawing a Time-Series Plot.

```
# Data

year <- c(2000:2014)
PCI <- c(85.12, 70.46, 96.87, 94.18, 123.41, 74.39, 80.16, 88.88, 93.00, 90.54, 147.52, 129.75,
         147.55, 252.10, 137.58)

DF3 <- data.frame(year, PCI)
DF3
```

```
# Plot

ggplot(data = DF3, aes(x = year, y = PCI)) +
  geom_point() +
  geom_line(color = "blue") +
  scale_x_continuous(limits = c(2000, 2014), breaks = c(seq(from = 2000, to = 2014, by = 1))) +
  scale_y_continuous(limits = c(0, 300), breaks = c(seq(from = 0, to = 300, by = 50))) +
  ggtitle("Partisan Confict Index in the United States Federal Government") +
  xlab("Year") +
  ylab("Partisan Confict Index")
```


Partisan Confict Index in the United States Federal Government
Partisan Conflict Index
300
250
200
150
100
50
0
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
Year