Exploratory Data Analysis

SW

2025-08-05

Estimates of Location

The data

```
# Get the data
csv <- "https://raw.githubusercontent.com/stevenkhwun/Online-resources/refs/heads/main/Data/Practical-S
state <- read.csv(csv)
# View the dataset
dim(state)
```

```
## [1] 50  4
```

```
head(state, n = 5)
```

```
##        State Population Murder.Rate Abbreviation
## 1    Alabama    4779736         5.7           AL
## 2     Alaska     710231         5.6           AK
## 3    Arizona    6392017         4.7           AZ
## 4   Arkansas    2915918         5.6           AR
## 5 California   37253956         4.4           CA
```

```
summary(state)
```

```
##     State             Population        Murder.Rate     Abbreviation
##  Length:50          Min.   :  563626   Min.   : 0.900   Length:50
##  Class :character   1st Qu.: 1833004   1st Qu.: 2.425   Class :character
##  Mode  :character   Median : 4436370   Median : 4.000   Mode  :character
##                     Mean   : 6162876   Mean   : 4.066
##                     3rd Qu.: 6680312   3rd Qu.: 5.550
##                     Max.   :37253956   Max.   :10.300
```

Mean, trimmed mean, and median

Compute the mean, trimmed mean, and median for the population using *R*:

```
# Mean
mean(state[["Population"]])
```

```
## [1] 6162876
```

```
# Trimmed mean
mean(state[["Population"]], trim = 0.1)
```

```
## [1] 4783697
```

```
# Median
median(state[["Population"]])
```

```
## [1] 4436370
```

Weighted mean and weighted median

```
# Weighted mean
weighted.mean(state[["Murder.Rate"]], w=state[["Population"]])
```

```
## [1] 4.445834
```

> Notes on *R*: Since base *R* doesn't have a function for weighted median, We need to install a package such as `matrixStats`.

```
library("matrixStats")
weightedMedian(state[["Murder.Rate"]], w=state[["Population"]])
```

```
## [1] 4.4
```

Estimates of Variability

Key Terms for Variability Metrics

- Deviations (errors, residuals): The difference between the observed values and the estimate of location.
- Variance (mean-squared-error): The sum of squared deviations from the mean divided by $n-1$ where n is the number of data values.
- Standard deviation: The square root of the variance.
- Mean absolute deviation (l1-norm, Manhattan norm): The mean of the absolute values of the deviations from the mean.
- Median absolute deviation from the median (MAD): The median of the absolute values of the deviations from the median.
- Range: The difference between the largest and the smallest value in a data set.
- Order statistics (ranks): Metrics based on the data values sorted from smallest to biggest.
- Percentile (quantile): The value such that P percent of the values take on this value or less and $(100-P)$ percent take on this value or more.
- Interquartile range (IQR): The difference between 75th percentile and the 25th percentile.

Neither the variance, the standard deviation, nor the mean absolute deviation is robust to outliers and extreme values. The variance and standard deviation are especially sensitive to outliers since they are based on the squared deviations. A robust estimate of variability is the *median absolute deviation from the median* or **MAD**.

Using *R*'s built-in functions for the standard deviation, the interquartile range (IQR), and the median absolute deviation from the median (MAD)

```
# Standard deviation
sd(state[["Population"]])
```

```
## [1] 6848235
```

```
# Interquartile range
IQR(state[["Population"]])
```

```
## [1] 4847308
```

```
# Median absolute deviation from the median
mad(state[["Population"]])
```

```
## [1] 3849870
```

The standard deviation is almost twice as large as the **MAD**. This is not surprising since the standard deviation is sensitive to outliers.

Exploring the Data Distribution

Key Terms for Exploring the Distribution

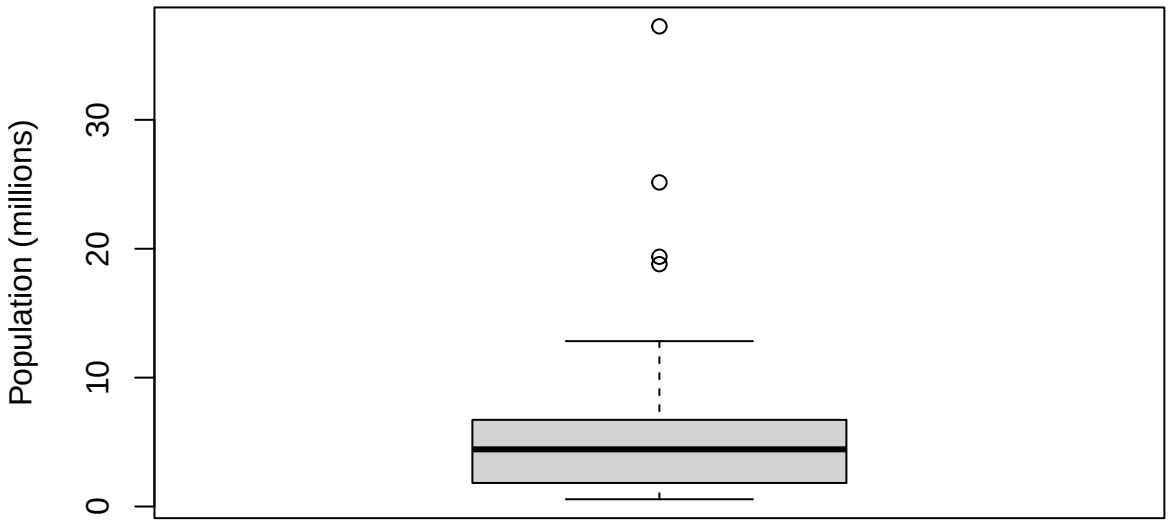
- Boxplot (box and whiskers plot): A plot introduced by Tukey as a quick way to visualize the distribution of data.
- Frequency table: A tally of the count of numeric data values that fall into a set of intervals (bins).
- Histogram: A plot of the frequency table with the bins on the x-axis and the count (or proportion) on the y-axis.
- Density plot: A smoothed version of the histogram, often based on a *kernel density estimate*.

Percentiles and Boxplots

```
# Percentiles
quantile(state[["Murder.Rate"]], p = c(.05, .25, .5, .75, .95))
```

```
##    5%   25%   50%   75%   95%
## 1.600 2.425 4.000 5.550 6.510
```

```
# Boxplot
boxplot(state[["Population"]]/1000000, ylab = "Population (millions)")
```

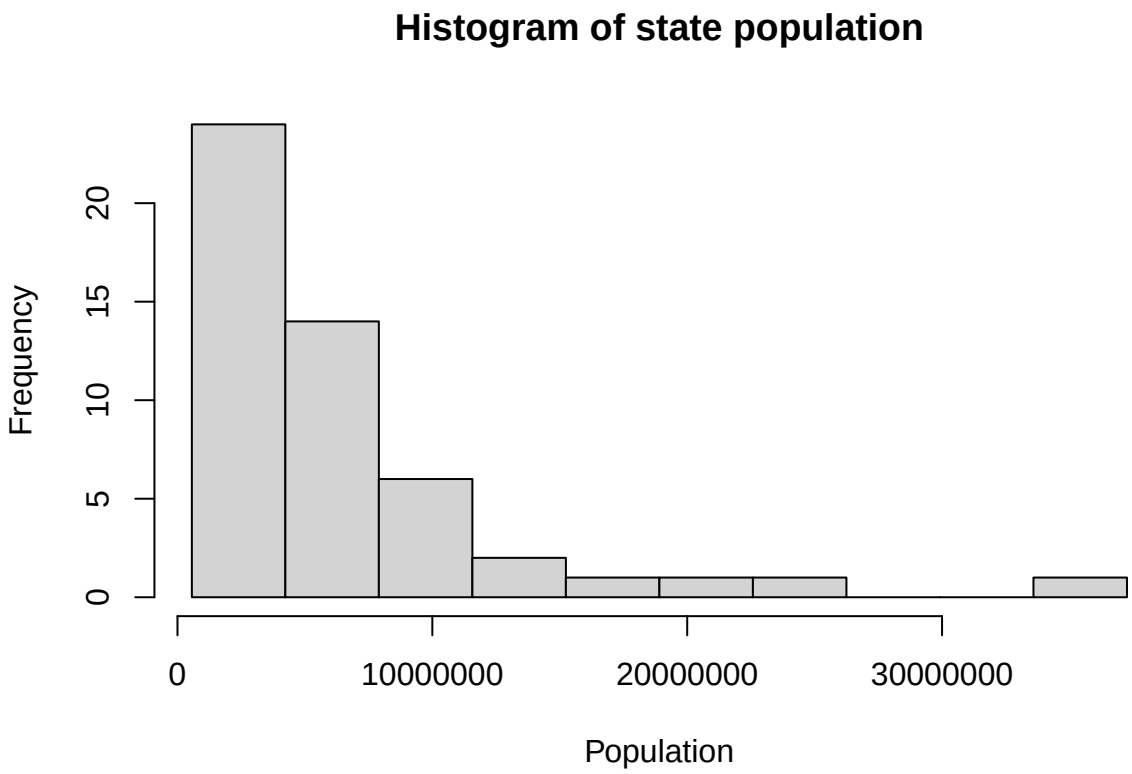


Frequency Tables and Histograms

```
# Frequency table
breaks <- seq(from = min(state[["Population"]]),
              to = max(state[["Population"]]), length = 11)
pop_freq <- cut(state[["Population"]], breaks = breaks,
                right = TRUE, include.lowest = TRUE)
table(pop_freq)
```

```
## pop_freq
## [5.64e+05,4.23e+06]  (4.23e+06,7.9e+06]  (7.9e+06,1.16e+07] (1.16e+07,1.52e+07]
##                  24                   14                   6                   2
## (1.52e+07,1.89e+07] (1.89e+07,2.26e+07] (2.26e+07,2.62e+07] (2.62e+07,2.99e+07]
##                   1                    1                   1                   0
## (2.99e+07,3.36e+07] (3.36e+07,3.73e+07]
##                   0                   1
```

```
# Histogram
options(scipen = 5)
hist(state[["Population"]], breaks = breaks, main = "Histogram of state population",
     xlab = "Population")
```



Notes on *R*: The option `options(scipen = 5)` determines how likely *R* is to switch to scientific notation in the plot. The higher the number, the less likely *R* will switch to scientific notation.

Statistical Moments

In statistical theory, location and variability are referred to as the first and second *moments* of a distribution. The third and fouth moments are called *skewness* and *kurtosis*. **Skewness** refers to whether the data is skewed to larger or smaller values, and **kurtosis** indicates the propensity of the data to have extreme values. Generally, metrics are not used to measure skewness and kurtosis; instead, these are discovered through visual displays.

Density Plots and Estimates

Related to the histogram is a density plot, which shows the distribution of data values as a continuous line. A density plot can be thought of as a smoothed histogram, although it is typically computed directly from the data through a *kernel density estimate*. In *R*, you can compute a density estimate using the `density` function:

```
hist(state[["Murder.Rate"]], freq = FALSE, main = "State Murder Rate",
          xlab = "Murder rate (murders per 100,000 people)")
lines(density(state[["Murder.Rate"]]), lwd = 3, col = "blue")
```

Exploring Binary and Categorical Data

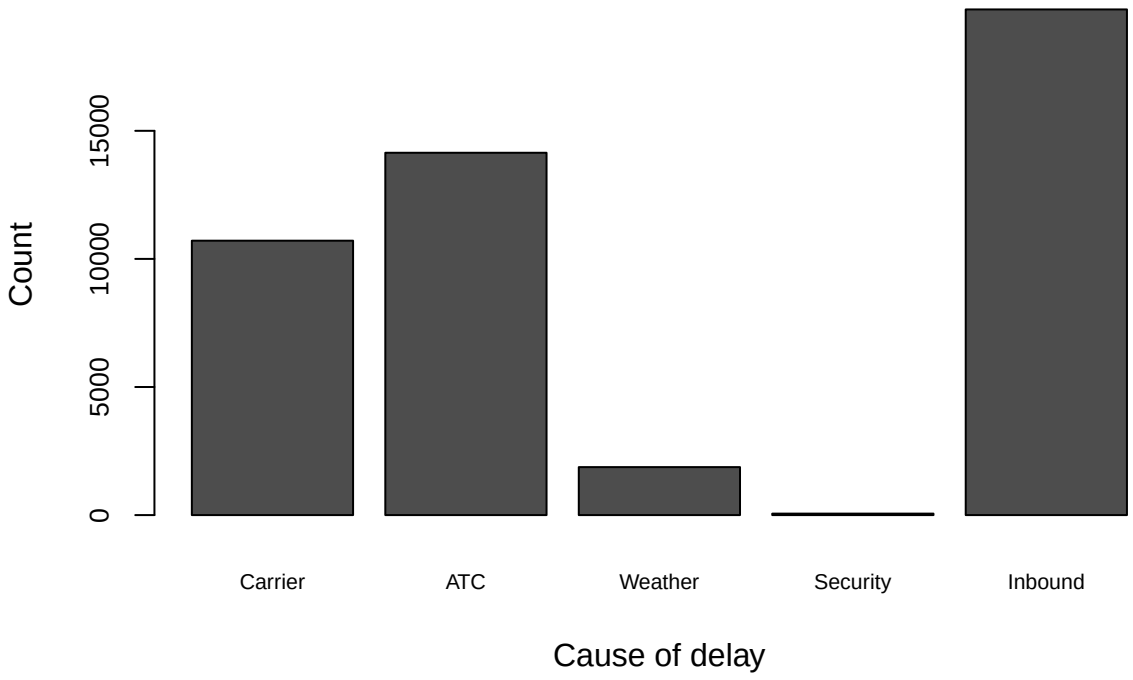
The data

```
# Get the data
csv <- "https://raw.githubusercontent.com/stevenkhwun/Online-resources/refs/heads/main/Data/Practical-S
dfw <- read.csv(csv)
dfw
```

```
##    Carrier     ATC  Weather Security  Inbound
## 1 64263.16 84856.5 11235.42   343.15 118427.8
```

Bar Chart

```
# Bar chart
barplot(as.matrix(dfw) / 6, cex.axis = 0.8, cex.names = 0.7,
        xlab = "Cause of delay", ylab = "Count")
```



Note that a bar chart resembles a histogram; in a bar chart the x-axis represents different categories of a factor variable, while in a histogram the x-axis represents values of a single variable on a numeric scale. In a histogram, the bars are typically shown touching each other, with gaps indicating values that did not occur in the data. In a bar chart, the bars are shown separae from one another.

Correlation

Key Terms for Correlation

- Correlation coefficient: A metric that measures the extent to which numeric variables are associated with one another (ranges from -1 to +1)
- Correlation matrix: A table where the variables are shown on both rows and columns, and the cell values are the correlations between the variables.
- Scatterplot: A plot in which the x-axis is the value of one variable, and the y-axis the value of another.

The data

`sp500_px` data

```
# Access the data from my Dell Laptop
data_path <- "C:/Users/steve/GitHub/Online-resources/Data/Practical-Statistics"
sp500_px <- read.csv(file.path(data_path, "sp500_data.csv.gz"), row.names = 1)
dim(sp500_px)
```

```
## [1] 5647  517
```

`sp500_sym` data

```
# Access the data from GitHub
csv <- "https://raw.githubusercontent.com/stevenkhwun/Online-resources/refs/heads/main/Data/Practical-S
sp500_sym <- read.csv(csv, stringsAsFactors = FALSE)
dim(sp500_sym)
```

```
## [1] 517   4
```

Correlation matrix

```
# Telecommunication stocks data
telecom <- sp500_px[, sp500_sym[sp500_sym$sector == 'telecommunications_services', 'symbol']]
head(telecom, n = 5)
```

```
##                      T         CTL         FTR          VZ LVLT
## 1993-01-29 -0.21626771 -0.46391947  0.00000000 -0.06407607    0
## 1993-02-01  0.09611898  0.17397279  0.01449721  0.14951282    0
## 1993-02-02  0.07208924  0.08698639  0.02899756  0.08543676    0
## 1993-02-03  0.00000000  0.00000000 -0.02899733  0.10679658    0
## 1993-02-04 -0.04805949  0.28995152  0.02899756 -0.10679231    0
```

```
# Create the correlation matrix
telecom <- telecom[row.names(telecom) > '2012-07-01',]
telecom_cor <- cor(telecom)
telecom_cor
```

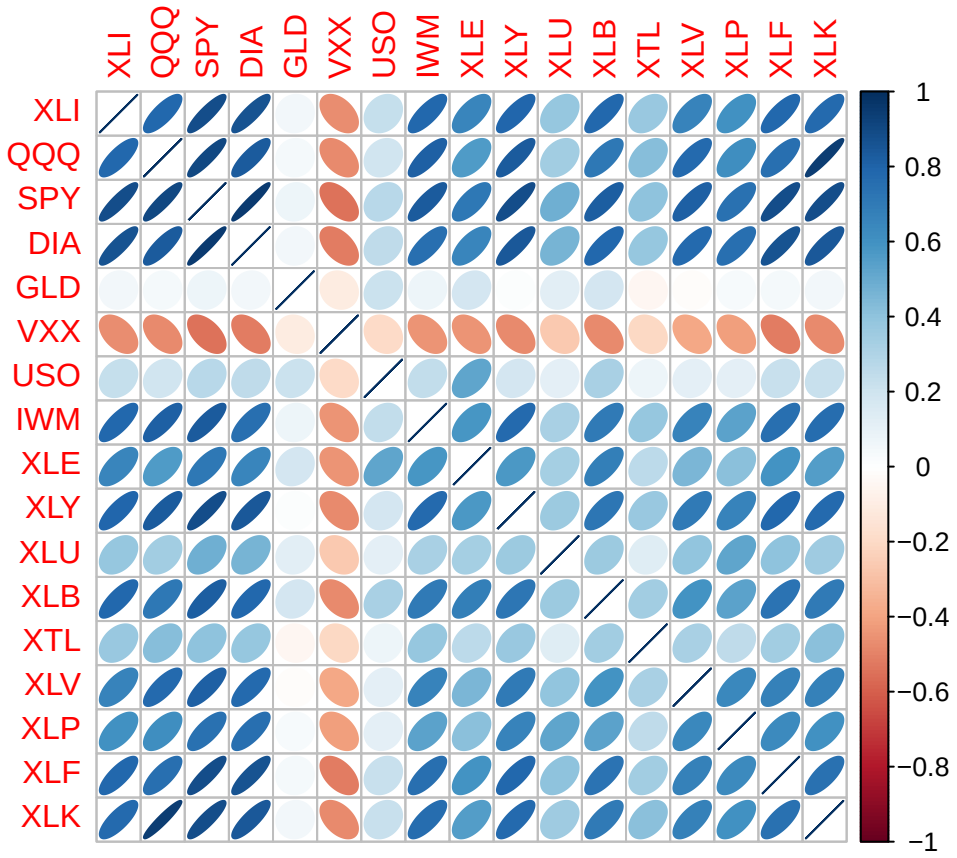
```
##              T       CTL       FTR        VZ      LVLT
## T    1.0000000 0.4746828 0.3277670 0.6776125 0.2786259
## CTL  0.4746828 1.0000000 0.4197567 0.4166045 0.2866655
## FTR  0.3277670 0.4197567 1.0000000 0.2873864 0.2600678
## VZ   0.6776125 0.4166045 0.2873864 1.0000000 0.2421985
## LVLT 0.2786259 0.2866655 0.2600678 0.2421985 1.0000000
```

Notes on *R*: In *R*, we can easily create a table of correlations using the package `corrplot`.

```
# Load the library
library(corrplot)
```

The following gigure shows the correlation between the daily returns for major exchange-traded funds (ETFs).

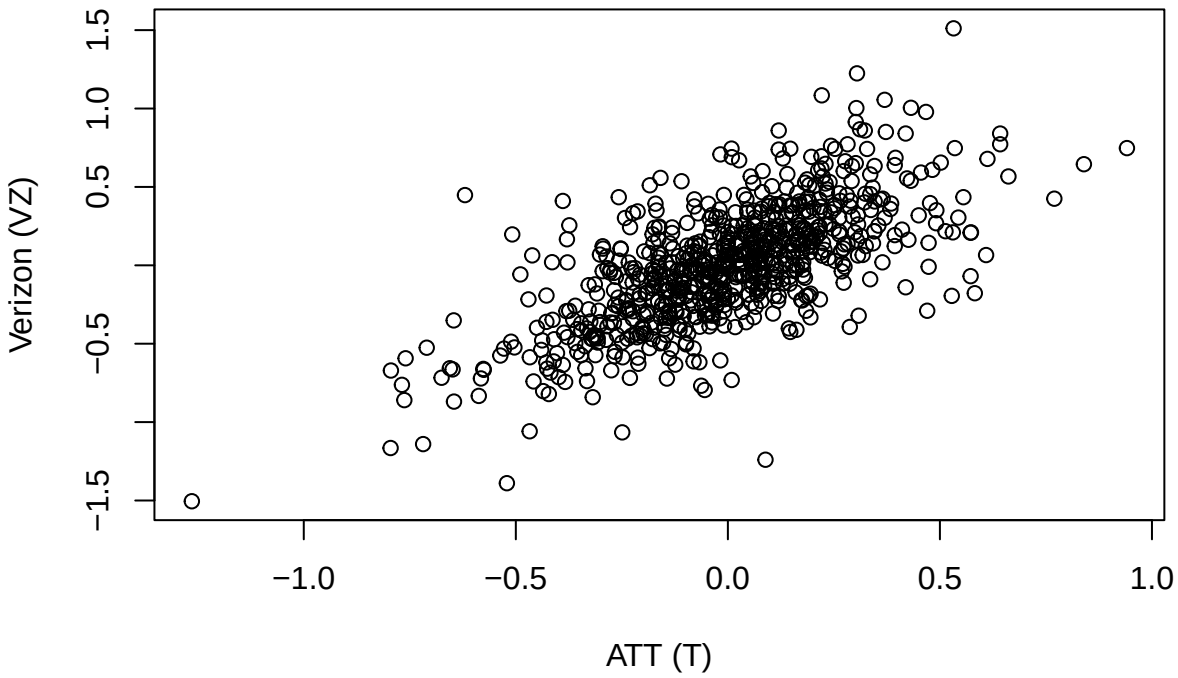
```
# Create the data
etfs <- sp500_px[row.names(sp500_px) > '2012-07-01',
                 sp500_sym[sp500_sym$sector == 'etf', 'symbol']]
# Create the plot
corrplot(cor(etfs), method = 'ellipse')
```



Scatterplots

The standard way to visualize the relationship between two measured data variables is with a scatterplot. The x-axis represents one variable and the y-axis another, and each point on the graph is a record.

```
# Scatterplot
plot(telecom$T, telecom$VZ, xlab = "ATT (T)", ylab = "Verizon (VZ)")
```


Verizon (VZ)
1.5
1.0
0.5
−0.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
ATT (T)