BTS | Barcelona Technology School

Alex Kumenius – Business Intelligence Data Scientist
Date : October 2019

RELATIONSHIPS BETWEEN VARIABLES

To answer research questions, data must be collected.

Analyses are motivated by ***looking*** for a ***relationship between two or more variables***.

Examining **summary statistics** could provide insights for each of the research questions about the study.

A **summary statistics** is a ***single number summarizing a large amount of data***. In other words, a **summary statistics** is a **value** computed from the **data**.

EXAMINING NUMERICAL DATA

We will be introduced to techniques for exploring and summarizing numerical variables, working with two datasets : '$email50$', '$county$' and '$cars$'.

In [ ]:
```
# importing libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

EXPLORING BIVARIATE VARIABLES WITH SCATTERPLOTS

A Scatterplot provides a case-by-case view of data for two **(bivariate)** numerical variables.

Scatterplots are helpful in quickly **spotting associations relating variables**, whether those associations come in the form of **simple trends** or whether those relationships are more **complex**.

We will use a Scatterplot to examine how $federal\ spending$ and $poverty$ are related in the $county$ dataset.

In [ ]:
```
# Open the choosen file
county = pd.read_csv('D:\\Documents\\EureCat\\Eurecat 2019\\BTS\\Datasets\\county.txt', sep='\t',
encoding='utf-8')
```

In [ ]:
```
county.shape
```

In [ ]:

```
# Create data
x = county.fed_spend
y = county.poverty
colors = 'Blue'
area = np.pi*5

plt.axis([0, 100, 0, 60])

# Plot
plt.scatter(x, y, s=area, c=colors, alpha=0.4, edgecolors='black')

plt.title('Federal Spending vs Poverty by County')
plt.ylabel('Federal Spending per Capita')
plt.xlabel('Poverty Rate (Percent)')
plt.show()
```

In any Scatterplot, each point represents a single ***case/observation***. Since there are **3.143** cases in $county$, there are **3.143** points

Now, We will compare the number of line breaks **(line_breaks)** and number of characters **(num_char)** in emails for the $email50$ dataset.

In [ ]:

```
dbe = pd.read_csv('D:\\Documents\\EureCat\\Eurecat 2019\\BTS\\Datasets\\email50.txt',
                  encoding='utf-8', sep='\t')
```

In [ ]:

```
dbe.shape
```

In [ ]:

```
# Create data
x = dbe.num_char
y = dbe.line_breaks

colors = "Blue"
area = np.pi*20
plt.axis([0, 70, 0, 1200])

# Plot
plt.scatter(x, y, s=area, c=colors, alpha=0.4, edgecolors='black')
plt.title('Spam email - # Lines vs # Characters')
plt.ylabel('Number of Lines')
plt.xlabel('Number of Characters (in thousands)')
plt.show()
```

To put the number of characters in perspective, this paragraph has **363** characters. Looking at scatterplot, it seems that some emails are incredibly verbose!. Upon further investigation, we would actually find that most of the long emails use the **HTML format**, which means most of the characters in those emails are used to **format the email** rather than **provide text**.

In [ ]:

```
dbcars = pd.read_csv('D:\\Documents\\EureCat\\Eurecat 2019\\BTS\\Datasets\\cars.txt',
                  encoding='utf-8', sep='\t')
```

Let's consider a new dataset $cars$ of 54 $cars$ with 6 variables. Create scatterplot to examine how $vehicle\ price$ and $weight$ are related.

What can be said about the relationship between these variables?

In [ ]:

```
dbcars.shape
```

In [ ]:

```
# Checking dataset variables
dbcars.dtypes
```

In [ ]:

```
dbcars.head()
```

In [ ]:

```
# Create data
x = dbcars.weight
y = dbcars.price

colors = "Blue"
area = np.pi*30

# Plot
plt.scatter(x, y, s=area, c=colors, alpha=0.4, edgecolors='black')
plt.title('Cars - Price vs Weight')
plt.ylabel('Price ($1000s)')
plt.xlabel('Weight (Pounds)')
```

The relationship is evidently nonlinear.

In [ ]:

```
fig = plt.figure(figsize=(12,4))

ax1 = fig.add_subplot(1, 3, 1)

# Create data
x = county.fed_spend
y = county.poverty
colors = 'Blue'
area = np.pi*5

plt.axis([0, 100, 0, 60])

# Plot
ax1.scatter(x, y, s=area, c=colors, alpha=0.4, edgecolors='black')

plt.title('County Dataset')
plt.ylabel('Federal Spending per Capita')
plt.xlabel('Poverty Rate (Percent)')

ax2 = fig.add_subplot(1, 3, 2)
# Create data
x = dbe.num_char
y = dbe.line_breaks

colors = "Blue"
area = np.pi*20
plt.axis([0, 70, 0, 1200])

# Plot
ax2.scatter(x, y, s=area, c=colors, alpha=0.4, edgecolors='black')
plt.title('Spam email Dataset')
plt.ylabel('# of Lines')
plt.xlabel('# of Characters (in thousands)')

ax3 = fig.add_subplot(1, 3, 3)
# Create data
x = dbcars.weight
y = dbcars.price

colors = "Blue"
area = np.pi*30
plt.axis([1500, 4300, 0, 65])

# Plot
ax3.scatter(x, y, s=area, c=colors, alpha=0.4, edgecolors='black')
plt.title('Cars Dataset')
plt.ylabel('Price ($1000s)')
plt.xlabel('Weight (Pounds)')

# plt.tight_layout(pad=0.4, w_pad=0.5, h_pad=1.0)
plt.tight_layout()
```

In [ ]:

```
sns.pairplot(county, diag_kind='kde', plot_kws={'alpha': 0.2})
```

HISTOGRAMS

Dot plots, like in scatterplot, show the **exact value for each observation**. This is useful for ***small datasets***, but they can become hard to read with **larger samples**.

Rather than showing the *value of each observation*, we prefer to think of the value as belonging to a **bin**.

These **bins** - ***(counts)*** **are plotted as bars into what is called a Histogram.**

Histogram provide a view of the data density. Higher bars represent where the data are relatively more common.

Histogram are especially convenient for describing the *shape of the data distribution*.

- **When data trail off to the right and have a longer right tail, the shape is said to be Right Skewed or also called Skewed to the Positive End.**

- **Contrary, data with the reverse characteristic – *a long, thin tail to the left* – are said to be Left Skewed. We also say that such a distribution has a long left tail.**

- **Data that show roughly equal trailing off in both directions are called Symmetric.**

```
In [ ]: dbe.hist(['num_char'], bins=14)
        plt.title('Spam email - # Characters')
        plt.ylabel('Frequency')
        plt.xlabel('# Characters (in thousands)')
```

> **Long tails to identify skew When data trail off in one direction, the distribution has a long tail. If a distribution has a long left tail, it is Left Skewed. If a distribution has a long right tail, it is Right Skewed.**

Modal Distribution

In addition to looking at whether a distribution is Skewed or Symmetric, histograms can be used to identify Modes.

A mode is the *value with the most occurrences*.

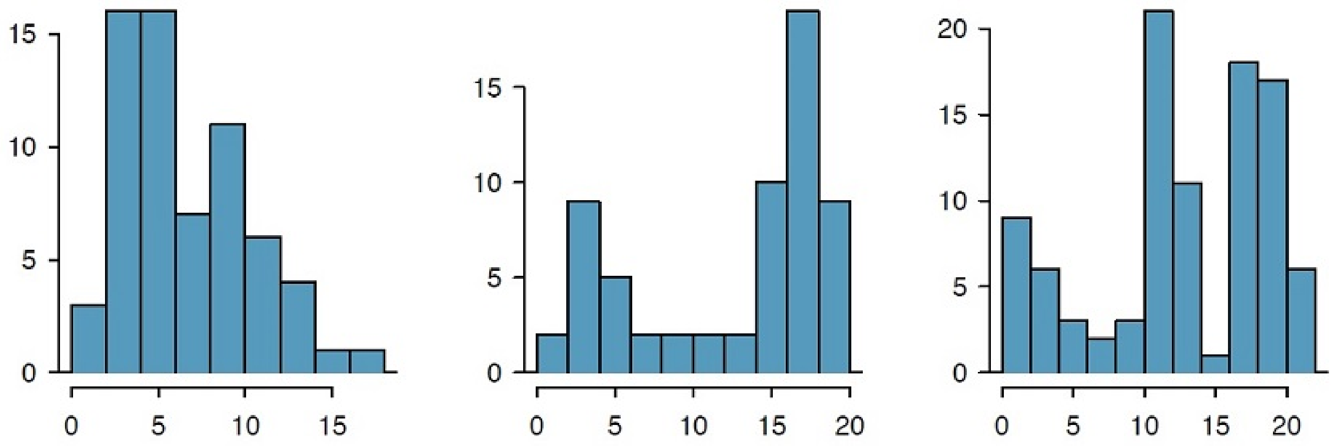
However, It is common to have no observations with the same value in a dataset, which makes, mode, useless for many real datasets.

A mode is represented by a prominent peak in the distribution. There is only one prominent peak in the histogram of num_char.

Histogram that have one, two, or three prominent peaks are called Unimodal, Bimodal, and Multimodal, respectively.

Any distribution with more than 2 prominent peaks is called Multimodal.

Notice that there was one prominent peak in the Unimodal distribution with a second less prominent peak that was not counted since it only differs from its neighboring bins by a few observations.



Looking for modes Looking for modes isn't about finding a clear and correct answer about the number of modes in a distribution. The important part of this examination is to better understand your data and how it might be structured.

SUMMARY STATISTICS

Mean - Average

The mean, sometimes called the average, is a common way to measure the center of a distribution of data.

To find the mean number of characters (num_char) in the 50 emails, we add up all the character counts and divide by the number of emails.

For computational convenience, the number of characters is listed in the thousands and rounded to the first decimal.

$$\bar{x} = \frac{21.7 + 7.0 + \ldots + 15.80}{50} = 11.6$$

```
In [ ]: dbe.num_char.mean().round(2)
```

The sample mean is often labeled $\bar{x}$. The letter x is being used as a generic placeholder for the variable of interest, num_char, and the *bar over on the x* communicates that the average number of characters in the 50 emails is 11,6.

Mean The sample mean $\bar{x}$ of a *numerical variable* is computed as the sum of all of the *observations* divided by the number of *observations*:

$$\bar{x} = \frac{x_1 + x_2 + \cdots + x_n}{n}$$

where $x_1, x_2, \ldots, x_n$ represent the n observed values. It is useful to think of the mean as the balancing point of the distribution.

EXERCISE - 3.1

Compare both Equations above.

- **What does x_1 correspond to ?,**
- **and x_2 ?**
- **Can you infer a general meaning to what x_i might represent?**
- **What was n in this sample of emails?**

> **Population Mean The Population mean has a special label : μ. The symbol μ is the $Greek$ letter mu and represents the average/mean of all observations in the Population. Sometimes a subscript, such as $_x$, is used to represent which variable the population mean refers to, e.g. μ_x**

EXERCISE - 3.2

The average number of characters across all emails (population) can be estimated using the sample data.

Based on the sample of 50 $emails$, what would be a reasonable estimate of μ_x, the mean number of characters in all emails in the email dataset? (Recall that $email50$ is a sample from $email$.)

Variance and Standard Deviation

Variance

The mean was introduced as a method to describe the center of a data set, but the variability in the data is also important.

We introduce two measures of variability: the Variance and the Standard Deviation. Both are very useful in data analysis.

The Standard Deviation describes how far away the typical observation is from the mean.

We call the *distance of an observation from its mean* its Deviation.

Below are the deviations for the 1st, 2nd, 3rd, and 50th observations in the num_char variable. For computational convenience, the number of characters is listed in the thousands and rounded to the first decimal.

$$\begin{aligned}
x_1 \; - \; \bar{x} &= 21.7 - 11.6 = 10.1 \\
x_2 \; - \; \bar{x} &= \;\; 7.0 - 11.6 = -4.6 \\
x_3 \; - \; \bar{x} &= \;\; 0.6 - 11.6 = -11.0 \\
&\vdots \\
x_{50} - \bar{x} &= 15.8 - 11.6 = 4.2
\end{aligned}$$

If we square these deviation and then take an average, the result is about equal to the sample variance, denoted by s^2:

$$s^2 \;=\; \frac{10.1^2 + (-4.6)^2 + (-11.0)^2 + \cdots + 4.2^2}{50 - 1} = 172,44$$

Sample Variance s^2 We divide by $n \; - \; 1$, rather than dividing by n, when computing the Variance. squaring the deviations does two things: - First, it makes large values much larger, seen by comparing 10.1^2, $(-4.6)^2$, $(-11.0)^2$, and 4.2^2. - Second, it gets rid of any negative signs. The variance is roughly the average squared distance from the mean.

Standard Deviation

Standard Seviation The Standard Deviation is defined as the square root of the Variance :

$$s = \sqrt{172.44} = 13.13$$

The Standard Deviation is useful when considering how close the data are to the Mean. Formulas and methods used to compute the Variance and Standard Deviation for a Population are similar to those used for a sample *(The only difference is that the Population Variance has a division by n instead of $n - 1$).* However, like the Mean, the Population values have special symbols : - σ^2 for the Variance and - σ for the Standard Deviation. The symbol σ is the $Greek$ letter $sigma$.

```
In [ ]: dbe.num_char.iloc[[0], ].std()
```

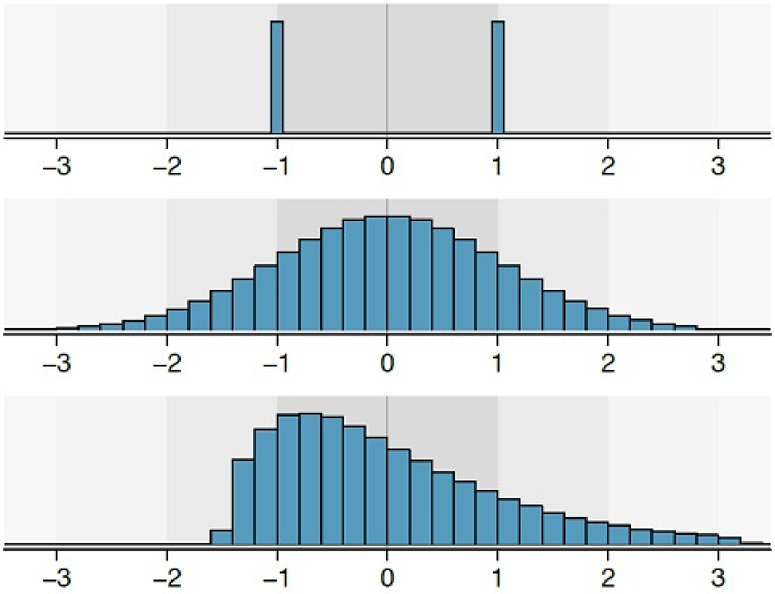
Standard Deviation describes Variability, so focus on the conceptual meaning of the Standard Deviation as a descriptor of Variability rather than the formulas.

Usually 70% of the data will be within one standard deviation of the mean and about 95% will be within two standard deviations two standard deviations. However, these percentages are not strict rules.

EXERCISE - 3.6

A good description of the shape of a distribution should include modality and whether the distribution is symmetric or skewed to one side.

Explore the figure as an example, explain why such a description is important :



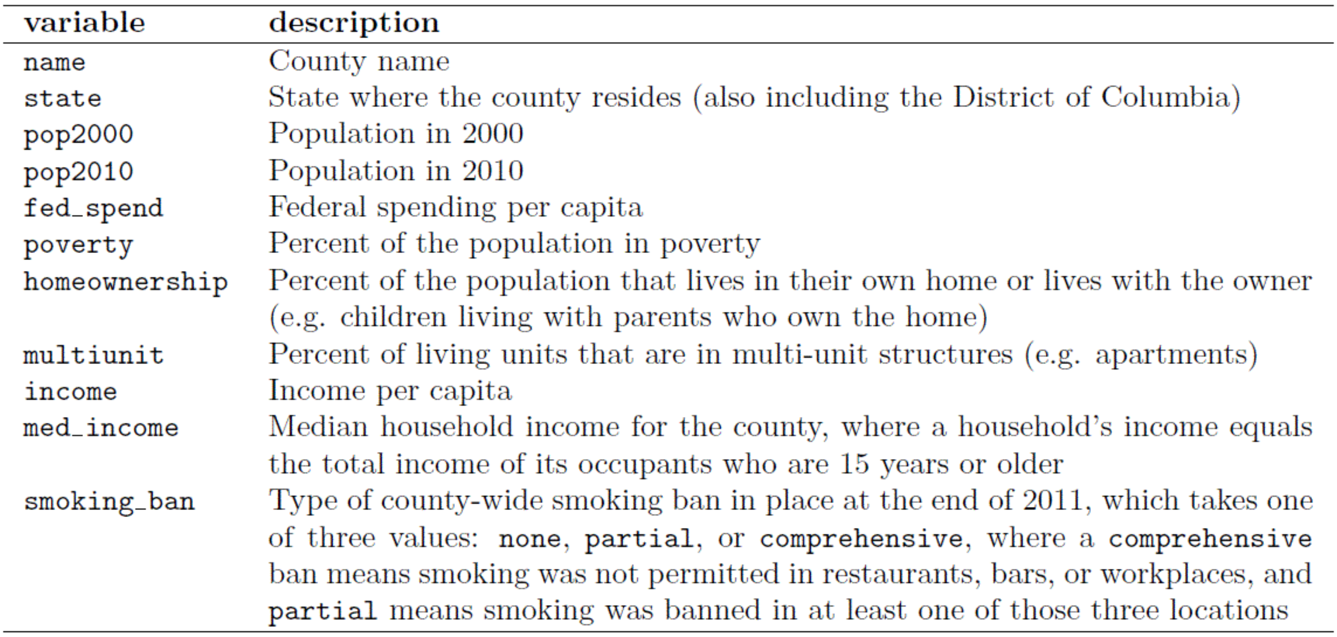
```
In [ ]: dbe.hist(['num_char'], bins=14)
        plt.title('Spam email - # Characters')
        plt.ylabel('Frequency')
        plt.xlabel('# Characters (in thousands)')
```

EXERCISE - 3.7

Describe the distribution of the num_char variable using the histogram display above.

The description should incorporate the center, variability, and shape of the distribution, and it should also be placed in context: the number of characters in emails. Also note any especially unusual cases.

We will use the Variance and Standard Deviation to assess how close the Sample Mean ($\bar{x}$) is to the Population Mean (μ).



variable	description
name	County name
state	State where the county resides (also including the District of Columbia)
pop2000	Population in 2000
pop2010	Population in 2010
fed_spend	Federal spending per capita
poverty	Percent of the population in poverty
homeownership	Percent of the population that lives in their own home or lives with the owner (e.g. children living with parents who own the home)
multiunit	Percent of living units that are in multi-unit structures (e.g. apartments)
income	Income per capita
med_income	Median household income for the county, where a household's income equals the total income of its occupants who are 15 years or older
smoking_ban	Type of county-wide smoking ban in place at the end of 2011, which takes one of three values: none, partial, or comprehensive, where a comprehensive ban means smoking was not permitted in restaurants, bars, or workplaces, and partial means smoking was banned in at least one of those three locations

In [ ]:
```
fig = plt.figure(figsize=(10,8))

ax1 = fig.add_subplot(2, 2, 1)

ax1.hist(county['multiunit'], bins=25)
plt.title('County - 2010 Population')
plt.ylabel('Frequency')
plt.xlabel('multi unit (%)')

ax2 = fig.add_subplot(2, 2, 2)

ax2.hist(county['income'], bins=25)

plt.title('2010 County Population')
plt.ylabel('Frequency')
plt.xlabel('Per Capita Income')

ax3 = fig.add_subplot(2, 2, 3)

ax3.hist(county['homeownership'], bins=25)
plt.title('2010 County Population')
plt.ylabel('Frequency')
plt.xlabel('Homeownership (%)')

ax4 = fig.add_subplot(2, 2, 4)

ax4.hist(county['med_income'], bins=25)

plt.title('2010 County Population')
plt.ylabel('Frequency')
plt.xlabel('Median Household Imcome')

plt.tight_layout()
```

In [ ]:

```
fig = plt.figure(figsize=(20,5))

ax1 = fig.add_subplot(1, 4, 1)

ax1.hist(county['multiunit'], bins=25)
plt.title('2010 County Population')
plt.ylabel('Frequency')
plt.xlabel('multi unit (%)')

ax2 = fig.add_subplot(1, 4, 2)

ax2.hist(county['income'], bins=25)

plt.title('2010 County Population')
plt.ylabel('Frequency')
plt.xlabel('Per Capita Income')

ax3 = fig.add_subplot(1, 4, 3)

ax3.hist(county['homeownership'], bins=25)
plt.title('2010 County Population')
plt.ylabel('Frequency')
plt.xlabel('Homeownership (%)')

ax4 = fig.add_subplot(1, 4, 4)

ax4.hist(county['med_income'], bins=25)

plt.title('2010 County Population')
plt.ylabel('Frequency')
plt.xlabel('Median Household Imcome')

# plt.tight_layout(pad=0.4, w_pad=0.5, h_pad=1.0)
plt.tight_layout()
```

BOX PLOTS

A Box Plot summarizes a dataset using *five statistics* while also plotting unusual observations - Anomalies or Outliers.

Quartiles, and the Median

In [ ]:

```
(dbe['num_char']).describe()
```

The median (6,890), splits the data into the bottom 50% and the top 50%, marked in the dot plot by horizontal dashes and open circles, respectively.

In [ ]:

```
(dbe['num_char']).median().round(3)
```

The first step in building a box plot is drawing a dark line denoting the median, which splits the data in half. 50% of the data falling below the median and other 50% falling above the median.

There are 50 character counts in the dataset (an even number) so the data are perfectly split into two groups of 25. We take the median in this case to be the average of the two observations closest to the 50th percentile:

$$(6,768 + 7,012)/2 = 6,890.$$

When there are an odd number of observations, there will be exactly one observation that splits the data into two halves, and in such a case that observation is the median (no average needed).

Median If the data are ordered from smallest to largest, the median is the observation right in the middle. If there are an even number of observations, there will be two values in the middle, and the median is taken as their average.

The second step in building a box plot is drawing a rectangle to represent the middle 50 of the data. The total length of the box, is called the interquartile range (IQR). It, like the Standard Deviation, is a measure of Variability in data. The more variable the data, the larger the Standard Deviation and IQR.

The two boundaries of the box are called the first quartile (the 25^{th} percentile), i.e. 25 of the data fall below this value and the third quartile (the 75^{th} percentile), and these are often labeled $Q1$ and $Q3$, respectively.

> Interquartile range (IQR) The IQR is the length of the box in a box plot. It is computed as
> $$IQR = Q3 - Q1$$
> where $Q1$ and $Q3$ are the 25^{th} and 75^{th} percentiles.

```
In [ ]: sns.set(style="whitegrid")
        ax = sns.boxplot(x=dbe["num_char"], color='lightblue', fliersize=5,  orient='v', linewidth=1 , wid
        th=0.3)
```

```
In [ ]: sns.stripplot(x=dbe["num_char"], orient='v', color='darkblue')
```

```
In [ ]: ax = sns.boxplot(y="num_char", data=dbe,  color='lightblue', fliersize=5,  orient='v', linewidth=1
        , width=0.3)
        ax = sns.stripplot(x=dbe["num_char"], orient='v', color='darkblue')
```

```
In [ ]: sns.set(style="whitegrid")
        ax = sns.boxplot(x=dbe["num_char"], color='lightblue', fliersize=5,  orient='v', linewidth=1 , wid
        th=0.3)
```

```
In [ ]: sns.swarmplot(x=dbe["num_char"], orient='v', color='darkblue')
```

```
In [ ]: ax = sns.boxplot(y="num_char", data=dbe,  color='lightblue', fliersize=5,  orient='v', linewidth=1
        , width=0.3)
        ax = sns.swarmplot(y="num_char", data=dbe, color="darkblue", orient="v", size=4)
```

EXERCISE - 3.8

1. What percent of the data fall between Q1 and the median?
2. What percent is between the median and Q3?

Extending out from the box, the whiskers attempt to capture the data outside of the box, however, their reach is never allowed to be more than $1.5\ x\ IQR$

They capture everything within this reach. The upper whisker does not extend to the last three points, which is beyond $Q3\ +\ 1.5\ x\ IQR$, and so it extends only to the last point below this limit.

The lower whisker stops at the lowest value, 33, since there is no additional data to reach; the lower whisker's limit is not shown in the figure because the plot does not extend down to $Q1\ -\ 1.5\ x\ IQR$. In a sense, the box is like the body of the box plot and the whiskers are like its arms trying to reach the rest of the data.

Any observation that lies beyond the whiskers is labeled with a dot. The purpose of labeling these points – instead of just extending the whiskers to the minimum and maximum observed values – is to help identify any observations that appear to be unusually distant from the rest of the data. Unusually distant observations are called Outliers.

In this case, it would be reasonable to classify the emails with character counts of 41,623, 42,793, and 64,401 as outliers since they are numerically distant from most of the data.

Outlier An **outlier** is an *observation* that appears **extreme** relative to the rest of the **data**. Why it is important to look for outliers Examination of data for possible **outliers** serves many useful purposes, including : 1. Identifying strong **skew** in the distribution. 2. Identifying data collection or **entry errors**. For instance, we re-examined the email purported to have 64,401 characters to ensure this value was accurate. 3. Providing **insight** into interesting **properties** of the **data**.

EXERCISE - 3.9

estimate the following values for num_char in the $email50$ dataset:

a).- $Q1$,
b).- $Q3$, and
c).- IQR