11_Descriptive_Statistics

March 31, 2020

0.1 What is Statistics?

Statistics is an area of applied mathematics concerned with data collection, analysis, interpretation, and presentation.

This area of mathematics deals with understanding how data can be used to solve complex problems. Here are a couple of example problems that can be solved by using statistics:

- Your company has created a new drug that may cure cancer. How would you conduct a test to confirm the drug's effectiveness?
- You and a friend are at a baseball game, and out of the blue, he offers you a bet that neither team will hit a home run in that game. Should you take the bet?
- The latest sales data have just come in, and your boss wants you to prepare a report for management on places where the company could improve its business. What should you look for? What should you not look for?

0.1.1 1. Statistical Terms

There are various statistical terms that one should be aware of while dealing with statistics.

Popolation: A collection of all probable observations of a specific characteristic of interest. It is a group from which data are collected.

```
Example: All learners taking data science course.
```

Sample: A subset of population

```
Example: A group of 20 learners selected for a quiz
```

Variable: An item of interest that can acquire various numerical values

```
Example: The number of defective items manufactured in a factory
```

0.1.2 Example:

[1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

[2]:

```
population = np.random.randint(10, 20, 100)
population
```

```
[2]: array([13, 19, 17, 19, 18, 10, 16, 15, 13, 15, 19, 11, 17, 14, 10, 12, 19,
            14, 16, 18, 13, 19, 15, 11, 11, 15, 11, 14, 19, 11, 18, 18, 17, 10,
            13, 17, 15, 11, 13, 10, 12, 11, 14, 12, 12, 12, 18, 12, 12, 19, 10,
            14, 10, 13, 12, 16, 13, 18, 18, 19, 18, 16, 14, 10, 10, 12, 11, 11,
            13, 17, 18, 17, 14, 16, 12, 11, 16, 18, 18, 17, 19, 17, 14, 18, 17,
            17, 13, 15, 14, 18, 17, 13, 13, 11, 19, 15, 19, 12, 17, 18])
```

```
[3]: np.random.randint?
```

```
[4]: sample1 = np.random.choice(population, 15)
     sample2 = np.random.choice(population, 15)
     sample3 = np.random.choice(population, 15)
```

```
[5]: print("Sample 1: ", sample1)
     print("Sample 2: ", sample2)
     print("Sample 3: ", sample3)
```

```
Sample 1:  [17 11 14 13 13 12 14 17 11 13 17 12 11 19 13]
Sample 2:  [17 18 10 17 10 17 15 17 19 18 10 11 12 15 17]
Sample 3:  [16 19 11 10 13 18 18 10 13 13 19 18 13 18 11]
```

0.1.3 2. Categories Of Data

Data can be categorized into two sub-categories: 1. Qualitative Data 2. Quantitative Data

2.1 Qualitative Data: Qualitative data deals with characteristics and descriptors that can't be easily measured, but can be observed subjectively. Qualitative data is further divided into two types of data:

Nominal Data: Data with no inherent order or ranking such as gender or race.

Ordinal Data: Data with an ordered series of information is called ordinal data.

2.2 Quantitative Data: Quantitative data deals with numbers and things you can measure objectively. This is further divided into two:

Discrete Data: Also known as categorical data, it can hold a finite number of possible values. Example: Number of students in a class.

Continuous Data: Data that can hold an infinite number of possible values. Example: Weight of a person.

0.2 4. Types of Statistics

There are two well-defined types of statistics:

1. Descriptive Statistics
2. Inferential Statistics

0.2.1 4.1 Descriptive Statistics

Descriptive statistics is a method used to describe and understand the features of a specific data set by giving short summaries about the sample and measures of the data.

Descriptive Statistics is mainly focused upon the main characteristics of data. It provides a graphical summary of the data.

Suppose you want to gift all your classmate's t-shirts. To study the average shirt size of students in a classroom, in descriptive statistics you would record the shirt size of all students in the class and then you would find out the maximum, minimum and average shirt size of the class.

Descriptive Statistics is broken down into two categories:

1. Measures of Central Tendency
2. Measures of Variability (spread)

0.2.2 4.1.1 Measures of Central Tendency

Measures of center are statistics that give us a sense of the "middle" of a numeric variable. Common measures of center include: - mean - median - mode

0.2.3 Mean

Arithmetic average of a range of values or quantities, computed by dividing the total of all values by the number of values.

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

[6]:
```
df1 = pd.DataFrame(dict(eid = range(6), age = np.random.randint(18, 31, size = 6)))
df1
```

[6]:
```
   eid  age
0    0   20
1    1   25
2    2   24
3    3   26
4    4   23
5    5   25
```

[7]:
```
df1.age.mean()
```

[7]:
```
23.833333333333332
```

[8]:
```
df1["age"].mean()
```

[8]:
```
23.833333333333332
```

```
[9]: df1.mean()
```

```
[9]: eid     2.500000
     age    23.833333
     dtype: float64
```

Median Denotes value or quantity lying at the midpoint of a frequency distribution of observed values or quantities, such that there is an equal probability of falling above or below it. Simply put, it is the *middle* value in the list of numbers.

If count is odd, the median is the value at $\frac{(n+1)}{2}$,

else it is the average of $\frac{n}{2}$ and $\frac{(n+1)}{2}$

```
[10]: df1.age.median()
```

```
[10]: 24.5
```

Mode It is the number which appears most often in a set of numbers.

```
[11]: df1.age.mode()
```

```
[11]: 0    25
      dtype: int32
```

```
[12]: df1['age'].mode()
```

```
[12]: 0    25
      dtype: int32
```

Using Pandas built-in function

```
[13]: df1.describe() # summary of data
```

```
[13]:             eid        age
      count  6.000000   6.000000
      mean   2.500000  23.833333
      std    1.870829   2.136976
      min    0.000000  20.000000
      25%    1.250000  23.250000
      50%    2.500000  24.500000
      75%    3.750000  25.000000
      max    5.000000  26.000000
```

0.2.4 4.1.2 Measures of Spread

Measures of spread (dispersion) are statistics that describe how data varies. While measures of center give us an idea of the typical value, measures of spread give us a sense of how much the data tends to diverge from the typical value. The measures of spread are: - Range - Standard deviation - Variance - Interquartile range

Range Range is the difference between the maximum and minimum observations.

$Range = max(x) - min(x)$

[14]:
```
df1
```

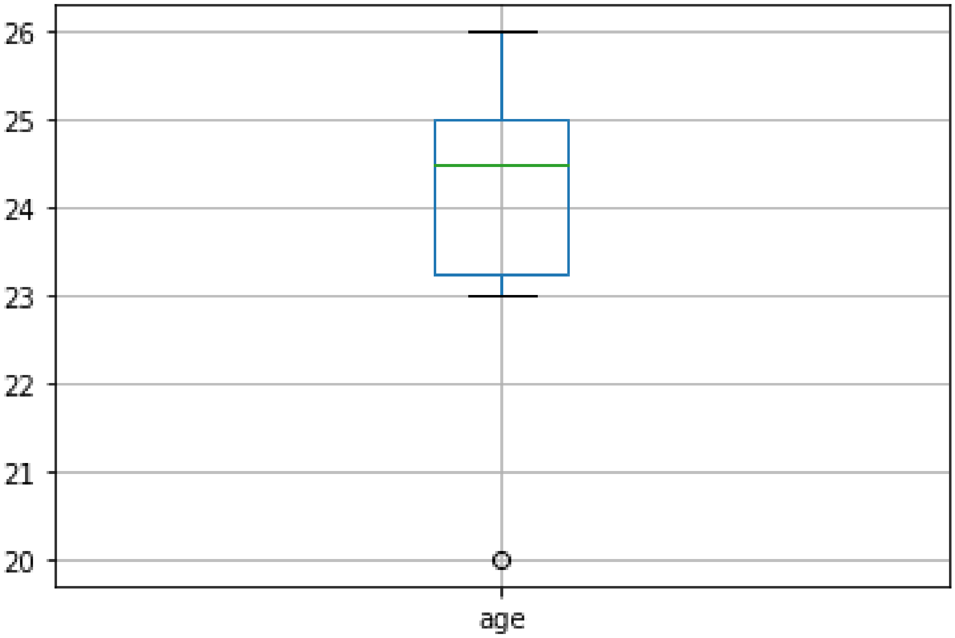
[14]:
```
   eid  age
0    0   20
1    1   25
2    2   24
3    3   26
4    4   23
5    5   25
```

[15]:
```
r = df1.age.max() - df1.age.min() ## range
r
```

[15]:
```
6
```

[16]:
```
df1.boxplot(column = 'age', return_type = 'axes')
```

[16]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x2532b3d0f88>
```



Variance It's the average distance of the data values from the *mean*. Variance can be calculated by using the below formula:

$Variance = S^2 = \sum \frac{(X-\overline{X})^2}{n}$

Here,

X: Individual data points n: Total number of data points x: Mean of data points

```
[17]: df1.age.var() # variance
```

```
[17]: 4.566666666666667
```

Interquartile Range(IQR) It is the measure of statistcal dispersion, being equal to the difference between upper and lower quartile.

$IQR = Q3 - Q1$

Quartiles tell us about the spread of a data set by breaking the data set into quarters, just like the median breaks it in half.

```
[18]: q75, q25 = df1.age.quantile([0.75, 0.25])
      IQR = q75-q25
      IQR
```

```
[18]: 1.75
```

Standard Deviation It is the square root of variance. This will have the same units as the data and mean. It can be calculated by using the below formula:

$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$

```
[19]: df1.age.std() ## standard deviance
```

```
[19]: 2.136976056643281
```

Skewness and Kurtosis Beyond measures of center and spread, descriptive statistics include measures that give you a sense of the shape of a distribution.

Skewness is a measure of the asymmetry of a data distribution. Skewness is asymmetry in a statistical distribution, in which the curve appears distorted or skewed either to the left or to the right. Skewness can be quantified to define the extent to which a distribution differs from a normal distribution.

Skewness measures the skew while kurtosis measures the "peakedness" of a distribution.

We won't go into the exact calculations behind skewness and kurtosis, but they are essentially just statistics that take the idea of variance a step further: while variance involves squaring deviations from the mean, skewness involves cubing deviations from the mean and kurtosis involves raising deviations from the mean to the 4th power.

Pandas has built in functions for checking skewness and kurtosis, df.skew() and df.kurt() respectively:

```
[20]: df1["age"].skew()
```

```
[20]: -1.3389545754235674
```

```
[21]: df1["age"].kurt()
```

```
[21]: 1.8780968618466556
```

0.3 2.2 Inferential Statistics

Inferential statistics generalize the larger dataset and applies probability theory to draw a conclusion.

It allows you to infer population parameters based on sample statistics and to model relationships within data.

0.3.1 2.2.1 Correlation

A correlation is a statistical test of association between variables that is measured on a -1 to 1 scale. The closer the correlation value is to -1 or 1 the stronger the association, the closer to 0, the weaker the association. It measures how change in one variable is associated with change in another variable.

There are a few common types of tests to measure the level of correlation: **Pearson, Spearman, and Kendall**.

Each have their own assumptions about the data that needs to be meet in order for the test to be able to accurately measure the level of correlation. Each type of correlation test is testing the following hypothesis.

Extent to which two or more variables fluctuate together. A **positive correlation** indicates the extent to which those variables increase or decrease in parallel; a **negative correlation** indicates the extent to which one variable increases as the other decreases.

$r = \frac{1}{n-1}\sum(\frac{x-\bar{x}}{s_x})(\frac{y-\bar{y}}{s_y})$

Pearson correlation assumptions

Pearson correlation test is a parametric test that makes assumption about the data. In order for the results of a Pearson correlation test to be valid, the data must meet these assumptions:

- The sample is independently and randomly drawn
- A linear relationship between the two variables is present
  - When plotted, the lines form a line and is not curved
- There is homogeneity of variance

The variables being used in the correlation test should be continuous and measured either on a ratio or interval sale, each variable must have equal number of non-missing observations, and there should be no outliers present.

Spearman Rank correlation assumptions

The Spearman rank correlation is a non-parametric test that does not make any assumptions about the distribution of the data. The assumption for the Spearman rank correlation test is:

- There is a monotonic relationship between the variables being tested
- A monotonic relationship exists when one variable increases so does the other

For the Spearman rank correlation, the data can be used on ranked data, if the data is not normally distributed, and even if the there is not homogeneity of variance.

Kendall's Tau correlation assumptions

The Kendall's Tau correlation is a non-parametric test that does not make any assumptions about the distribution of the data. The only assumption is:

- There should be a monotonic relationship between the variables being tested

The data should be measured on either an ordinal, ratio, or interval scale.

```
[22]: df1.corr()
```

```
[22]:            eid       age
      eid  1.000000  0.525274
      age  0.525274  1.000000
```

```
[23]: df1['eid'].corr(df1['age']) # default is pearson
```

```
[23]: 0.5252736513086527
```

```
[24]: df1['eid'].corr(df1['age'], method="spearman")
```

```
[24]: 0.3768511731740915
```

```
[25]: df1['eid'].corr(df1['age'], method="kendall")
```

```
[25]: 0.27602622373694163
```

```
[26]: df1['eid'].corr(df1['age'], method="pearson")
```

```
[26]: 0.5252736513086527
```

```
[ ]:
```