

**School of Built Environment, Engineering and Computing**

**Week 1: Part 2 – Descriptive Statistics**

****

**By**

**Prof. Ah-Lian Kor**

**Descriptive Statistics**

The descriptive statistics in this section will be provided by the describe() function.

You could provide the descriptive statistics for all the quantitative columns or selected columns

**Read in student\_marks.xlsx file.**

**Type the following:**

# Standard import for pandas, numpy and matplot

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

# Read in the excel file and display some of the basic info

marks=pd.read\_excel("student\_marks.xlsx")

print ("Data types in the file:\n")

print (marks.dtypes)

print("\n")

print ("Summary of the input file:")

print (marks.describe())

print("\n")

print ("student marks stats:")

print (marks['Subject1'].describe())

print("\n")

**Use titanic dataset from Github** [**https://raw.githubusercontent.com/pandas-dev/pandas/master/doc/data/titanic.csv**](https://raw.githubusercontent.com/pandas-dev/pandas/master/doc/data/titanic.csv)

**Data Dictionary** [**https://pandas.pydata.org/docs/getting\_started/intro\_tutorials/06\_calculate\_statistics.html**](https://pandas.pydata.org/docs/getting_started/intro_tutorials/06_calculate_statistics.html)

**Read in the titanic.txt file.**

**Part 1: Display the top 5 records in the dataset**

**Type the following (note: NaN is missing value):**

import pandas as pd

#read in the titanic.txt file

titanic = pd.read\_csv("titanic.txt")

titanic.head()

**Part 2: Display the Descriptive Statistics of titanic dataset**

**Type the following:**

import pandas as pd

#read in the titanic.txt file

titanic = pd.read\_csv("titanic.txt")

# Descriptive Statistics of titanic dataset

titanic.describe()

**Part 3: Display the Descriptive Statistics of selected columns of the titanic dataset**

**Type the following:**

import pandas as pd

#read in the titanic.txt file

titanic = pd.read\_csv("titanic.txt")

# Only choose descriptive statistics for Age and Fare

titanic[["Age", "Fare"]].describe()

**Part 4: More Descriptive Functions**

**Type the following:**

import pandas as pd

#read in the titanic.txt file

titanic = pd.read\_csv("titanic.txt")

print("Median for Age and Fare")

print(titanic[["Age", "Fare"]].median())

print("\n")

print("Mean for Age and Fare")

print(titanic[["Age", "Fare"]].mean())

print("\n")

print("Aggregated descriptive functions")

print(titanic.agg({'Age': ['min', 'mean', 'max', 'median', 'skew'],

'Fare': ['min', 'max', 'median', 'mean', 'skew']}))

**Part 5: Descriptive Statistics by Groups**

**Find the mean of female and male in the titanic dataset**

**Type the following:**

import pandas as pd

#read in the titanic.txt file

titanic = pd.read\_csv("titanic.txt")

print("Find the means for numeric columns grouped by sex (female and male)")

print(titanic.groupby("Sex").mean())

print("\n")

print("Find the means for the Age column grouped by sex (female and male)")

print(titanic.groupby("Sex")["Age"].mean())

print("\n")

print("Find the means for the Age column for grouped by sex (female and male) and cabin class (Class 1, 2, 3)")

titanic.groupby(["Sex", "Pclass"])["Age"].mean()

**Part 6: Frequency Counts**

**Type the following:**

import pandas as pd

#read in the titanic.txt file

titanic = pd.read\_csv("titanic.txt")

print("How many passengers are there for each cabin class?")

print(titanic["Pclass"].value\_counts())

print("\n")

print("How many males and females are there for each cabin class?")

print(titanic.groupby("Sex")["Pclass"].count())

**Resources**

<https://pandas.pydata.org/docs/getting_started/intro_tutorials/06_calculate_statistics.html>

<https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.describe.html>

<https://towardsdatascience.com/a-quick-guide-on-descriptive-statistics-using-pandas-and-seaborn-2aadc7395f32>

<https://pythonhow.com/accessing-dataframe-columns-rows-and-cells/>

<https://www.tutorialspoint.com/python_pandas/python_pandas_descriptive_statistics.htm>

**Cumulative Plots**

**Dataset 1: Number of students that progress for each level for an undergraduate and postgraduate course (see table below). Note the codes are adapted from** [**https://dfrieds.com/data-visualizations/when-use-cumulative-frequency-graph.html**](https://dfrieds.com/data-visualizations/when-use-cumulative-frequency-graph.html)

|  |  |
| --- | --- |
| **Level** | **Pass\_Frequency** |
| **4** | **1670** |
| **5** | **1255** |
| **6** | **950** |
| **7** | **546** |
| **8** | **400** |
| **9** | **100** |

**Part 1: Read the dataset in Python and display all the records.**

**Type the following:**

import pandas as pd

import matplotlib.pyplot as plt

%matplotlib inline

dataset = {'Level': [4, 5, 6, 7, 8, 9],

'Pass\_Frequency': [1670, 1255, 950, 546, 400, 100]}

df = pd.DataFrame(data=dataset)

df

**Part 2: Plot graphs for the dataset**

**Type the following for Scatterplot**

import pandas as pd

import matplotlib.pyplot as plt

%matplotlib inline

dataset = {'Level': [4, 5, 6, 7, 8, 9],

'Pass\_Frequency': [1670, 1255, 950, 546, 400, 100]}

df = pd.DataFrame(data=dataset)

df.plot(kind='scatter', x='Level', y='Pass\_Frequency',

figsize=(10, 8), legend=True, marker= "x", color='red', rot=0);

plt.title("A Graph for the Number of Passes for Each Level", fontsize=20)

plt.xlabel("Level of Study", labelpad=15)

plt.ylabel("Frequency", labelpad=15)

plt.grid()

plt.show()

**Type the following (Bar Chart)**

import pandas as pd

import matplotlib.pyplot as plt

%matplotlib inline

dataset = {'Level': [4, 5, 6, 7, 8, 9],

'Pass\_Frequency': [1670, 1255, 950, 546, 400, 100]}

df = pd.DataFrame(data=dataset)

df.plot(kind='bar', x='Level', y='Pass\_Frequency',

figsize=(10, 8), legend=True, color='green', rot=0);

plt.title("A Graph for the Number of Passes for Each Level", fontsize=20)

plt.xlabel("Level of Study", labelpad=10, fontsize=15)

plt.ylabel("Frequency", labelpad=10, fontsize=15)

plt.show()

**Part 3: Cumulative Frequency Plot (or Cumulative Frequency Distribution)**

**Add a new Cumulative Frequency Column to the dataset**

**Type the following:**

#Cumulative Frequency Plot

import pandas as pd

import matplotlib.pyplot as plt

%matplotlib inline

dataset = {'Level': [4, 5, 6, 7, 8, 9],

'Pass\_Frequency': [1670, 1255, 950, 546, 400, 100]}

df = pd.DataFrame(data=dataset)

#Add a new Column Cumulative Frequency

df['Cumulative Frequency'] = df['Pass\_Frequency'].cumsum()

df

**Plot the Cumulative Frequency Graph or Distribution**

**Type the following:**

#Cumulative Frequency Plot

import pandas as pd

import matplotlib.pyplot as plt

%matplotlib inline

dataset = {'Level': [4, 5, 6, 7, 8, 9],

'Pass\_Frequency': [1670, 1255, 950, 546, 400, 100]}

df = pd.DataFrame(data=dataset)

#Add a new Column Cumulative Frequency

df['Cumulative Frequency'] = df['Pass\_Frequency'].cumsum()

#note: rx- red colour, marker x, continuous line

df.plot(kind='line', x='Level', y='Cumulative Frequency',

figsize=(10, 8), legend=True, style='rx-', label="Cumulative Frequency")

plt.title("A Cumulative Frequency Plot for Passes", fontsize=20)

plt.xlabel("Levels of Study", labelpad=15, fontsize=15)

plt.ylabel("Cumulative Frequency", fontsize=15, labelpad=15)

plt.legend()

**Type 2: Cumulative Percentile Graph**

**Step 1: Find the total of the Pass\_Frequency column**

**Step 2: Add a new Cumulative Frequency Column**

**Step 3: Convert the Cumulative Frequency Column to Cumulative Percentile**

**Step 4: Convert values in Step 3 to 2 decimal places**

**Step 5 Display the dataset with additional columns**

**Type the following codes:**

#Cumulative Percentile Plot

import pandas as pd

import numpy as py

import matplotlib.pyplot as plt

%matplotlib inline

dataset = {'Level': [4, 5, 6, 7, 8, 9],

'Pass\_Frequency': [1670, 1255, 950, 546, 400, 100]}

df = pd.DataFrame(data=dataset)

# Get the total frequency for the Pass\_Frequency Column

total = df['Pass\_Frequency'].sum()

print('The total number of pass is:', total)

#Add a new Column Cumulative Frequency

df['Cumulative Frequency'] = df['Pass\_Frequency'].cumsum()

#Convert Cumulative Frequency to Percentage and Add a Cumulative Percentile Column

df['Cumulative Percentile'] = df['Cumulative Frequency'] \* 100/total

#Specify the format (i.e. 2 decimal places for the Cumulative Percentile Column)

df['Cumulative Percentile'] = df['Cumulative Percentile'].apply('{:.2f}%'.format)

df

**Step 6: Plot the Cumulative Percentile Graph**

**Type the following codes:**

#Cumulative Percentile Plot

import pandas as pd

import matplotlib.pyplot as plt

%matplotlib inline

dataset = {'Level': [4, 5, 6, 7, 8, 9],

'Pass\_Frequency': [1670, 1255, 950, 546, 400, 100]}

df = pd.DataFrame(data=dataset)

# Get the total frequency for the Pass\_Frequency Column

total = df['Pass\_Frequency'].sum()

print('The total number of pass is:', total)

#Add a new Column Cumulative Frequency

df['Cumulative Frequency'] = df['Pass\_Frequency'].cumsum()

#Convert Cumulative Frequency to Percentage and Add a Cumulative Percentile Column

df['Cumulative Percentile'] = df['Cumulative Frequency'] \* 100/total

df.plot(kind='line', x='Level', y='Cumulative Percentile',

figsize=(10, 8), legend=True, style='bx-', label="Cumulative Percentile")

plt.title("A Cumulative Percentile Plot for Passes", fontsize=20)

plt.xlabel("Levels of Study", labelpad=15, fontsize=15)

plt.ylabel("Cumulative Percentile", fontsize=15, labelpad=15)

plt.legend()

**Resources**

[**https://dfrieds.com/data-visualizations/when-use-cumulative-frequency-graph.html**](https://dfrieds.com/data-visualizations/when-use-cumulative-frequency-graph.html)

<https://www.nintyzeros.com/2017/05/plotting-with-python-matplotlib-pandas-dataframe.html>

<https://pandas.pydata.org/pandas-docs/version/0.23.1/generated/pandas.Series.cumsum.html>

<https://www.geeksforgeeks.org/cumulative-percentage-of-a-column-in-pandas-python/>

**Meaning of %matplotlib inline**

<https://stackoverflow.com/questions/43027980/purpose-of-matplotlib-inline#:~:text=%25matplotlib%20inline%20sets%20the%20backend,stored%20in%20the%20notebook%20document>.

The output of plotting commands is displayed **inline** within frontends like the Jupyter notebook, directly below the code cell that produced it. The resulting plots will then also be stored in the notebook document.