Day 3 Exercises-Key

Question 1

Access the given URL (https://github.com/awesomedata/awesome-public-datasets) to get access to the dataset available in this page. Use the given “Clone with HTTPS” link to bring the data in your colab. The given dataset consists of 12 variables and 891 data points.

- First, write a python program that access the csv file in the given Zip file.
- Read the csv file to a dataframe called “org_data”.
- Use the describe() function to provide a statistical information about the variables in the dataset.
- Extract four columns (Name, Sex, Age and Embarked) to create a new dataframe called “sub_data”. Use the “Name” column as an index of the “sub_data” dataframe.
- Use the following conditions to extract a subset of the data stored in “sub_data” datfarme:
- All male passengers under 18 who are in the Embarked class Q or S.
- Export all the last two dataframes in csv files (use the same names given to the dataframes)

```
# bring data
! git clone https://github.com/awesomedata/awesome-public-datasets.git
```

```
# access the zip file path
cd awesome-public-datasets
```

```
cd Datasets
```

```
# unzip the zip file

from zipfile import ZipFile
filename="titanic.csv.zip"

#opening zip file in read mode
with ZipFile(filename,'r') as zip:
  zip.extractall()
```

```
#import data from CSV
import pandas as pd
org_data=pd.read_csv('titanic.csv', index_col="Name")

# get summary of the data
print(org_data.describe())

# select some columns
sub_data= org_data[['Sex', 'Age', 'Embarked']]

# use conditions
options = ['Q', 'S']
# selecting rows based on two conditions
selected_data = sub_data[(sub_data['Sex'] == 'male') &
                   (sub_data['Age'] < 18) &
               sub_data['Embarked'].isin(options)]

sub_data.to_csv("sub_data.csv")
selected_data.to_csv("selected_data.csv")

print("\n Done... \n Two Excel sheets have been created..")
```

Question 2

Write Pandas program to read the given “stud_scores.csv” file to a dataframe called “scores”.

- Sort the data ascendingly using the final project score and keep the results in a new dataframe “sorted_scores”. – Add a new column “Total” to the “scores” dataframe to keep the total scores for each student.
- Rank the dataframe according to the new “Total” column.

```
# First of all, upload the file to COLAB
from google.colab import files
uploaded = files.upload()
```

```
import pandas as pd
#read csv file
scores = pd.read_csv("stud_scores.csv", index_col="Name")

# sort data using the final project values
sorted_scores=scores.sort_values("Final Project")

#add a new column to keep the scores total for each student
scores["Total"]= scores.sum(axis=1)

# rank the dataframe using the new Total column
scores['t_rank']=scores["Total"].rank(ascending=0)
final = scores.set_index("t_rank")
final_sorted= final.sort_index()

final_sorted
```

Question 3

Two different datasets for wart treatment are presented in “UCI Machine Learning Repository”.

- Cryotherapy Dataset (https://archive.ics.uci.edu/ml/datasets/Cryotherapy+Dataset+)
- Immunotherapy Dataset (https://archive.ics.uci.edu/ml/datasets/Immunotherapy+Dataset)

Read the two datasets to dataframes, then find the correlation matrix for each dataset. Although the datasets have the same variables, the dataset has an additional variable “induration_diameter”. Thus, drop the additional column from the Cryotherapy Dataset. Then merge the two datsets together in one dataframe.

Finally, find the correlation matrix again for the merged dataset and compare it with the two generated earlier.

```
# First go the provided links and download the excel sheets to your local drive
# Upload the two files to your COLAB workspace

# First of all, upload the file to COLAB
from google.colab import files
uploaded = files.upload()
```

```
# Read the first dataset to imm datafarme
import pandas as pd
Immun = pd.read_excel("Immunotherapy.xlsx")
Cryoth= pd.read_excel("Cryotherapy.xlsx")

# find the correlation matrix for each dataset
print("Correlation Matrix of Immunotherapy.xlsx: \n ", Immun.corr())
print("Correlation Matrix of Cryotherapy.xlsx: \n ", Cryoth.corr())

# drop the extra column from the second dataset
Immun_ =Immun.drop(["induration_diameter"],axis=1)

# combine the two datasets and print the result
combined =pd.concat([Immun_, Cryoth])
print("The final combined dataset: \n", combined)

# find the correlation for the combined dataset
print("The correlation matrix of the combined dataset: \n", combined.corr())
```