Lab_Pandas_Indexing_and_Computations

May 5, 2020

0.1 Pandas Indexing and Selection Lab

0.2 Simple Series and DataFrames

Create a Series to store Temperature values for 1 week

```
In [ ]: import pandas as pd
```

```
In [ ]: temperature_trichy = pd.Series([40.2, 39.8, 36.3, 39.1, 41.3, 32.9, 36.6])
```

```
In [ ]: # show temperature values
        temperature_trichy
```

```
In [ ]: # What is the weather on 2nd day?
        temperature_trichy[1]
```

```
In [ ]: # Find all days and temperature where temperature over 40.0 degree Celsius
        temperature_trichy[temperature_trichy > 40.0]
```

```
In [ ]: # Find only day, not also temperature
        temperature_trichy[temperature_trichy > 40.0].index
```

0.2.1 Create a Dataframe for student details from List

```
In [ ]: students = [['DS01', 'Rex', '1msc'], ['DS02', 'peter', '2msc'], ['CS01', 'ann', '3bsc'
        df_stud = pd.DataFrame(students, columns=['rollno', 'name', 'class'])  # row index aut
```

```
In [ ]: df_stud
```

Display all column names

```
In [ ]: df_stud.columns
```

Add a new column address

```
In [ ]: address = ['Delhi', 'Bangalore', 'Chennai']
        df_stud['address'] = address
```

```
In [ ]: df_stud
```

0.2.2 Create a Dataframe for Phone book from Dictionary

```
In [ ]: phonebook = {'rex':[9942002764, 'rex@abc.com'], 'sam':[9932176542, 'sam@xyz.com'], 'pe
        df_phonebook = pd.DataFrame.from_dict(phonebook, orient='index', columns=['mobile', 'em
```

```
In [ ]: df_phonebook
```

0.3 Exploratory Data Analysis on Video Game Review Dataset

Import ign.csv dataset

```
In [ ]: reviews = pd.read_csv("ign.csv")
```

Show top-5 rows

```
In [ ]: reviews.head()
```

Show bottom 3 rows

```
In [ ]: reviews.tail(3)
```

How many rows and columns here?

```
In [ ]: reviews.shape
```

What are the datatypes?

```
In [ ]: reviews.dtypes
```

0.3.1 Selecting Columns

Select a single column, say title

```
In [ ]: reviews['title'].tail()
```

Select multiple columns, title and genre

```
In [ ]: reviews[['title', 'genre']].head(10)
```

0.3.2 Selection using Positions

Select top-5 rows and all columns, same as head()

```
In [ ]: reviews.iloc[:5, :]
```

Select rows from position 5 onwards, and columns from position 5 onwards.

```
In [ ]:
```

Select the first column, and all of the rows for the column

```
In [ ]:
```

the 10th row, and all of the columns for that row.

```
In [ ]:
```

First column is not useful. So remove it

```
In [ ]: reviews = reviews.iloc[:,1:]
        reviews.head()
```

0.3.3 Selection using Row and Column Labels

We have already created students dataframe as below. Let us access name column with loc()

```
In [ ]: students = [['DS01', 'Rex', '1msc'], ['DS02', 'peter', '2msc'], ['CS01', 'ann', '3bsc'
        df_stud = pd.DataFrame(students, columns=['rollno', 'name', 'class'])  # row index aut

In [ ]: df_stud

In [ ]: # Let us access name column by loc()
        df_stud.loc[:, "name"]
```

Let us come back to our reviews. Display the first five rows of reviews using the loc method

```
In [ ]: reviews.loc[0:5,:] # here 0:5 are labels, not integer values
```

Select score_phrase column

```
In [ ]: #reviews.loc[:, 'score_phrase']
        reviews.loc[:, 'score_phrase'].head()

In [ ]: reviews['score_phrase'].head(10)  # it is also same
```

Select rows from 5 to 15

```
In [ ]: some_reviews = reviews.iloc[5:15,]
        some_reviews.head()
```

Select scores of first 3 rows from score_phrase df

```
In [ ]: some_reviews.loc[5:7, 'score']  # scores of first 3 rows
```

Select "score", "genre", and "release_year" columns

```
In [ ]: #reviews[["score", "genre", "release_year"]]
        reviews[["score", "genre", "release_year"]].head()

In [ ]: # every column is a Series
        type(reviews["score"])
```

0.3.4 Aggregate Columns

Find mean value of score column

```
In [ ]: reviews['score'].mean()
```

Find mean value of all numeric columns

```
In [ ]: reviews.mean()
```

Find mean value for each numeric column

```
In [ ]: reviews.mean(axis=0)
```

Find mean value for each row containing numeric values

```
In [ ]: #reviews.mean(axis=1)
        reviews.mean(axis=1).head()
```

Find lowest, highest, median, standard deviation of a column

```
In [ ]: reviews['score'].median()
```

```
In [ ]: reviews['score'].min()
```

```
In [ ]: reviews['score'].max()
```

```
In [ ]: reviews['score'].std()
```

```
In [ ]: reviews['score'].count()  # how many non-null values
```

Describe() gives summary

```
In [ ]: reviews.describe()  #gives summary of all numeric columns by default
```

check if review score has any correlation with other columns

```
In [ ]: reviews.corr()
```

Review score has no correlation with other features. So, release timing doesn't linearly relate to review score

0.3.5 Math Operations on DF columns

Divide score by 2

```
In [ ]: #reviews['score'] / 2
        (reviews['score'] / 2).head()
```

0.3.6 Boolean Indexing in Pandas

Select all video games whose review score > 7

```
In [ ]: score_filter = reviews["score"] > 7
        #score_filter
        score_filter.head()
```

```
In [ ]: filtered_reviews = reviews[score_filter]
        filtered_reviews.head()
```

Show filtered_reviews shape and titles

```
In [ ]: filtered_reviews.shape
```

```
In [ ]: filtered_reviews['title'].head(10)
```

Find games released for the Xbox One platform that have a score of more than 7

```
In [ ]: xbox_one_filter = (reviews["score"] > 7) & (reviews["platform"] == "Xbox One")
        filtered_reviews2 = reviews[xbox_one_filter]
        filtered_reviews2.head()
```

```
In [ ]: filtered_reviews2.shape
```

How many video games are 'Action' genre?

```
In [ ]: action_reviews = reviews[reviews['genre'] == 'Action']
```

```
In [ ]: action_reviews.head()
```

```
In [ ]: action_reviews.shape
```

0.3.7 Plot Review Ratings of two Play Stations and Compare Which one has more ratings?

Now that we know how to filter, we can create plots to observe the review distribution for the Xbox One vs the review distribution for the PlayStation 4. This will help us figure out which console has better games.

We can do this via a histogram, which will plot the frequencies for different score ranges.

```
In [ ]: import matplotlib.pyplot as plt
        %matplotlib inline

        reviews[reviews["platform"] == "Xbox One"]["score"].plot(kind="hist")
```

Plot for Play Station4

```
In [ ]: reviews[reviews["platform"] == "PlayStation 4"]["score"].plot(kind="hist")
```

Therefore, it appears from our histograms that the PlayStation4 has many more highly rated games than the Xbox One.

```
In [ ]:
```