PANDAS

> Here is the every pandas function and tricks that I have learn.

About

> This booklet is written by Ahmad, who is currently pursuing a bachelor degree in Data Science , He is 20 year old and is very passionate about they whole thing. Here's how you can reach him out

> contact Email: 93245ahmad@gmail.com

Preface

For pandas we mainly use VS code editor because it provide all the functionality we need to make programs and to create application, and to analyze data. It has the extension of every programming language there is.
We will be using Jupyter Notebook
You can install its extension in VS code

Chapter one

> **installing the necessary libraries**

We can install pandas perhaps any library by writing the following command in our terminal or cmd

```
pip install pandas
pip install seaborn
pip install numpy
```

Chapter 2

> **importing the libraries**

In this step we import the libraries to our notebook

```
import pandas as pd
import seaborn as sns
import numpy as np
```

Chapter 3

> **learning about Pandas**

Pandas is a library in python which is widely used for data analysis because it provide the tools such is making tables, making dictionaries ,filtering the data,
pandas is like excel of python

Basic terminologies

Series

In pandas we call the column a series because it is a series of data

> To create a series

```
pd.Series([1,2,3,4,5,6])
```

Ìndex

In pandas we call Rows Index

R: Index alway start from 0.

DataFrame

In pandas we call DataFrame to a two dimensional data e.g Table

```
df =  pd.DateFrame({"column_1":[1,2,3,4,5], 'column_2':[6,7,8,9,10]})
df
```

chapter 4

> **creating arrays with numpy**

We can also create a data frame with numpy

```
array_1 = np.array([5,6,7,8],[55,66,77,88])
pd.DataFrame(array_1)
```

creating a random array

Creating a random array with np.random

```
df = pd.DataFrame(np.random.rand(3,4), columns=list('abc'))
```

you can enter (rows , columns) how many you wanna generate

Chapter 5

> **Viewing dataset**

you can view the DataFrame by just Enter the variable name

viewing head

In head command just can enter the parameter or just can leave it is it is

```
df.head()
```

`viewing tail`

In tail you can either enter the parameter or leave it

```
df.tail()
```

Chapter 6

> **Basic commands of DataFrame**

To get information about the dataset we usually use the following the commands

`info`

it will tell us about the no. of entries and the memory weight and datatype

```
df.info()
```

`describe`

it will statistically describe our dataset

```
df.describe()
```

`dtypes`

to find out about the data types of the columns

```
df.dtypes
```

`shape`

to the shape of the dataset (rows * columns)

```
df.shape
```

`len`

we use len to know the length of the dataset

```
len(df)
```

`columns`

to get information about the columns

```
df.columns
```

`unique`

this tell us about the no. of unique values in the columns

```
df.column_name.unique()
```

`count`

it will count the no. of entries in each column

```
df.count
```

Chapter 7

> **Renaming the columns**

There are several ways to rename the columns

`Method 1`

This method is used when we manually select and rename the columns

```
pd.rename(column={
    "column_1": 'col_1',
    'column_2': 'col_2'},
    inplace=True)
```

This inplace make sure the change is placed

`Method 2`

This method is simple but you have to change the name of every column there is

```
df.columns=['col_1',"col_2"]
```

`Method 3`

In this method we replace some symbol with another

```
df.column.str.replace('-' , ' ')
```

this will replace the underscore with space

Chapter 8

Adding the prefix and suffix

`prefix`

to add prefix to the columns

```
df.add_prefix('no.')
```

`suffix`

to add suffix to the columns

```
df.add_suffix('done')
```

Chapter 9

Changing the data types

`method 1`

```
df['col_N'] = df['col_N'].astype(datatype)
```

`method 2`

```
df.astype({
    "col_1" : 'datatype',
    'col_2' : 'datatype'
})
```

Chapter 10

Loading the Dataset

To load the dataset that you have on the local setup

`reading data`

```
pd.read_extension('file_name.extension')
```

`loading a template`

we can load datasets from seaborn library

N: There are total 15 such datasets in the seaborn library

```
sns.load_dataset('dataset_name')
# e.g
sns.load_dataset('tips')
```

Chapter 11

> **saving the dataset**

to save a dataset that we have created in pandas or saving the dataset of sns library

```
df.to_xlsx('world_cup.xlsx')
```

Chapter 12

> **Reversing the row and column order**

Sometime to analyze data we might reverser the rows columns or rows to columns

`tranpose`

Transpose will change the rows to columns and col to rows

```
df.T
```

`Reversing`

To reverse the row order or columns

```
# for index
df.loc[::-1]

#for columns
df.loc[:,::-1]

#for reversing back the rows
df.loc[::-1].reset_index(drop=True)
#or
df.loc[:,::1]
```

Chapter 13

> **Selections**

To select the columns or index that you wanna show

`method 1`

to select entries by its datatype

```
df.select.dtypes(include=['number']).head
#this will cells with numbers
```

`method 2`

we use this to include the columns that we wanna show

```
df.select_dtypes(include=['int64'])
```

`Method 3`

To select the column that we want show

```
df[['column_1', 'column_2']]
```

`Method 4`

taking samples out of data

```
df.sample(5)
#this will take 5 sample of the whole data just 5 random
```

Chapter 14

Reducing DataFrame size \

`using sample method`

```
df.sample(frac=0.5)
```

`spliting the dataframe into two serparate dataframes`

We will be using sample and drop method for that

```
df_1 = df.sample(frac=0.5, random_state=1)
df_2 = df.drop(df_1.index)
df_2.shape
```

Note :
sample + frac == spliting
drop == for dropping

Chapter 15

`Combinig two data sets`

R: Many old videos and toturial will tell you to use 'append' for joining but in new version that doesn't work because that keyword is removed. The new keyword is 'concat'

```
j_df = pd.concat([df_1 , df_2], ignore_index= True)
```

Chapter 16

Filtering data

`Filtering the data`

```
#checking the unique values in the columns
df.column_1.unique

#filtering the unique value of the column
df[(df.column_1=="filter")]
```

`Filtering by catagoreis`

```
#count the no of things in the data
df.column_1.value_count()

#filtering the largest counts
df.column_1.value_count().nlargest()
```

``Spliting string into multiple columns`

```
df[["coln1","coln2"]] = df.column_1.str.split(' '), Expand=True
#it will split the string after the space
```

Chapter 17

Aggregate by multiple group (search by group of data)

```
df.groupby('column_1').count
```

Reshape Multi index series

```
# We are using titanic dataset for this session
df = sns.load_dataset('titanic')

# to get the mean of the column (data)
# df.column.mean()
df.age.mean()


#groupby

# to get the mean age grouped by sex(male and female)
df.groupby('sex').age.mean()

#another
#to get the data of the survived people according to their class and gender
df.groupby(['sex','class']).age.min()
```