MINI – PROJECT 1

IST – 652

SCRIPTING FOR DATA ANALYSIS

FALL 2022

STRUCTURED DATA

PREPARED BY : **HENDI KUSHTA**

Table of Contents

Data and Source

The main outline of the assignment is to write a program that reads data in found in formats such as .csv, .tsv, .txt or file saved from excel. The data that I have chosen is named netflix_titles.csv. It has been a Kaggle.com dataset, but after that was hidden, and could only be red by using API. I found the data in github link :
https://github.com/prasertcbs/basic-dataset/blob/master/netflix_titles.csv.
There are 6234 rows and 12 columns in total.
show_id -- the unique key for each show.
type – is either Movie or TV Show.
title – show title.
director – name of the show director.
cast – main actors.
country – where is the show played.
date_added – when the show was added in Netflix databases.
release_year – year when was the show created.
rating – there are 14 different rating.
listed_in – what is the genre of the show.
description – show description.

Data Exploration and Cleaning

Firstly I used head() and tail() functions to check first and last rows.
Next step was to check the number of rows and columns.
After that I checked the dataset column names in case I needed to change any label.
I changed the label listed_in to genre which seems to be better.
My next step was to check the column data types, and after that, I checked if there was any empty value.
I decided to drop all the empty values, even though most of the empty values were on director, cast, country, date_added and rating columns.
My next step was to check if there was any duplicated value.
Since I thought to make analysis based on time too, I converted the date_added column into datetime format and then separated this column into 3 other columns, year, month and days.
There are some other data cleaning and exploration in 3 questions and analysis I have prepared in this mini project.

Data Analysis - Questions

There are three questions I have tried to answer in this first mini-project.
1. Number of Movies and TV Shows added in Netflix over years.
2. Number of Movies and TV Shows created in top 10 Countries.
3. Average Movie duration time for top 10 movie genres

Scripting

Import the necessary packages that I have used in my mini project.
Then I read the data from netflix_titles.csv.

Check first and last rows and look how many rows and columns are there.

```
# import the necessary packages for my project

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

```
# read the csv data file

netflix_df = pd.read_csv('netflix_titles.csv')
```

```
# print dataset first rows

netflix_df.head()
```

```
# print dataset last rows

netflix_df.tail()
```

```
netflix_df.shape # check number of columns and rows.
```

Check columns names if any of the labels needs to be changed.
Change column name listed_in to genre. Then check if there is any null values in the columns and I drop the null values. Get some statistics about the object data types. Check if there are any duplicated values and how many unique values are for each column.

```
netflix_df.columns # print column names of netflix dataset.
```

```
# rename column listed_in to genre
netflix_df.rename(columns = {'listed_in':'genre'}, inplace = True)
```

```
# check for null values
netflix_df.isnull().sum()
```

```
# drop null values
netflix_df = netflix_df.dropna()
```

```
# check columns data types
netflix_df.dtypes
```

```
#getting the categorical fields based on dtype
netflix_df.dtypes[netflix_df.dtypes == 'object']
```

```
# get an index object of categorical fields

categorical = netflix_df.dtypes[netflix_df.dtypes == 'object'].index
categorical
```

```
# describe only the categorical fields, notice the descriptors
netflix_df[categorical].describe()
```

```
# print dataset shape after droping all null values

netflix_df.shape
```

```
# check for duplicate values
netflix_df.duplicated().sum()
```

```
# check for unique values
netflix_df.nunique()
```

Since i will make analysis based on time too, i have to convert all columns of time,in date time formats. Create 3 new columns - month added - what month its added, day_added and year_added.

```
# since i will make analysis based on time too, i have to convert all columns of time,
# in date time formats. Create 3 new columns - month added - what month its added, day_added
# and year_added.

netflix_df["date_added"] = pd.to_datetime(netflix_df['date_added'])
netflix_df['day_added'] = netflix_df['date_added'].dt.day
netflix_df['year_added'] = netflix_df['date_added'].dt.year
netflix_df['month_added']= netflix_df['date_added'].dt.month
netflix_df['year_added'].astype(int);
netflix_df['day_added'].astype(int);
```

Analysis 1 - Number of Movies and TV Shows added in Netflix over years.

First of all, I choose only the columns I need to answer the question by using double square brackets [[‘’]] and write the name of the columns inside.
Secondly I grouped by the columns and found the count of shows during years through count() aggregate function.

```
# In this example i need to use only type and year_added columns
# from netflix dataset

movies_shows_over_years_df = netflix_df[['type', 'year_added']]
```

```
movies_shows_over_years_df
```

```
# count how many movies and tv shows are in the dataset

movies_shows_over_years_df['type'].value_counts()
```

```
# groupby the new dataset using 2 columns type and year.
# find the number of movies shown through years using count
# aggregate function.
# give a name to the column of the aggregate function
# reset index so the dataset can appear as below.

grouped_multiple = movies_shows_over_years_df.groupby(['type', 'year_added']).agg({'type': ['count']})
grouped_multiple.columns = ['nr_of_movies_shows_added_through_years']
grouped_multiple = grouped_multiple.reset_index()
print(grouped_multiple)
```

Then I pivoted the table to have a better representation. Type as rows, year_added as columns and number of shows as values. I replaced the null values with 0 and converted table data type in integer values.
Last, I created a plot for the dataset.

```
# pivot dataset
grouped_multiple_pivot = pd.pivot_table(grouped_multiple, values='nr_of_movies_shows_added_through_years',
                                        index='type', columns='year_added')

# replace null values in the pivoted table with 0
grouped_multiple_pivot = grouped_multiple_pivot.replace(np.nan,0)
```

```
# converted to type integer
grouped_multiple_pivot = grouped_multiple_pivot.astype(int)
```

```
grouped_multiple_pivot
```

```
# save pivoted table as a csv file
grouped_multiple_pivot.to_csv('Analysis1_After_Pivot.csv')
```

```
# plot the graph of number of Movies or TV Shows added in netflix
# over years.

grouped_multiple_pivot.plot(kind='barh', figsize=(15,8))
plt.xlabel("Number of Movies or TV Shows")
plt.ylabel("Type")
plt.title("Number Of Movies Or TV Shows Added In Netflix Over Years")
plt.savefig('Analysis1.png', dpi=300)
plt.show()
```

Analysis 2 - Number of Movies and TV Shows created in top 10 Countries.

First of all, I choose only the columns I need to answer the question by using double square brackets [[‘’]] and write the name of the columns inside.
Split the countries column in different columns since there are 2-3 countries in a cell. Drop the country column and merge all countries in the same column.
Created 2 new dataframes on with countries and the number of shows in each conutry, one for country, show type and show count for movies and TV Shows. I merged this two dataframes based on country. After merging, I found the percentage of movies and tv shows in each country from the total shows.

```
# select title, type and country
netflix_country = netflix_df[['title','type', 'country']]

# drop column country and split the countries in different rows
netflix_country = (netflix_country.drop('country', axis=1).join
            (netflix_country.country.str.split(', ',expand=True).stack()
            .reset_index(drop=True, level=1).rename('country')
            ))
```

```
netflix_country
```

```
#create the new dataframe for countries and their count
netflix_country_1 = netflix_country[['title', 'country']].groupby(['country'])['title']
       .count().reset_index().sort_values(
                        'title', ascending= False).head(10).rename(columns = {'title': 'movies_count'})
netflix_country_1
```

```
#Create and merge the dataframes
netflix_country_2 = netflix_country[['title', 'country']].groupby(['country'])['title']
                .count().reset_index().sort_values('title', ascending= False).head(10)

netflix_country_3 = netflix_country[['title', 'type', 'country']].groupby(['country', 'type'])['title']
                    .count().reset_index().sort_values('title', ascending= False).rename(columns = {
                                            'title': 'movies_count'})

netflix_country_chart = netflix_country_2.merge(netflix_country_3, how = 'left', left_on = 'country',
                                            right_on = 'country')

# create a new column % which shows the percentages of movies or
# tv shows from all amount of program.
netflix_country_chart['%'] = ((netflix_country_chart['movies_count']/netflix_country_chart['title'])*100)
                                            .round(2)
```

Next step, I pivoted the table type as row, countries as columns and number of shows as values.
Replaced empty rows with 0 and converted data types in integers.
Last, I plotted a graph.

```
# create a pivot table with types and countries

netflix_country_chart = pd.pivot_table(netflix_country_chart, values='movies_count', index='type',
                                       columns='country')
```

```
netflix_country_chart
```

```
# replace empty with 0 and convert to integer

netflix_country_chart = netflix_country_chart.replace(np.nan,0)
netflix_country_chart = netflix_country_chart.astype(int)
netflix_country_chart
```

```
# save pivoted table as a csv file
netflix_country_chart.to_csv('Analysis2_After_Pivot.csv')
```

```
# plot a graph for the dataset

netflix_country_chart.plot(kind='bar', figsize=(15,8))
plt.ylabel("Number of Movies or TV Shows")
plt.xlabel("Country")
plt.title("Number Of Movies Or TV Shows Added In Netflix In Countries")
plt.savefig('Analysis2.png', dpi=300)
plt.show()
```

Analysis 3 - Number of Movies and TV Shows created in top 10 Countries.

Choose only rows where type is equal to movies.
Since duration was a string, I split the column in two parts and select only the numeric one. When there us no duration, I assigned to 0.
I choose only the columns I need to answer the question by using double square brackets [[‘’]] and write the name of the columns inside.
Since there are 2-3 genres in the same cells for some movies, I created 3 new columns and drop column named genre.

```
# choose only where type is Movie

netflix_movies = netflix_df[netflix_df['type'] == 'Movie']
```

```
# if duration is null, fill it with 0
netflix_movies['duration'] = netflix_movies['duration'].fillna("0")
# after spliting, get only the first element and convert to integer.
netflix_movies['duration'] = netflix_movies['duration'].str.split(" ").str[0].astype(int)
```

```
netflix_movies.head(2)
```

```
# check unique values of the new dataset

netflix_movies.nunique()
```

```
# select genre and duration columns to be in the new dataset
netflix_movies = netflix_movies[['genre', 'duration']]
```

```
netflix_movies.head(10)
```

```
netflix_movies.shape
```

```
# split genres in 3 different columns

netflix_movies[['genre1', 'genre2', 'genre3']] = netflix_movies['genre'].str.split(',', expand=True)
```

```
netflix_movies.head(2)
```

```
# drop genre column

netflix_movies = netflix_movies.drop('genre', axis=1)
```

Next step, I unpivot the created dataset, to put all 3 newly created columns to a single one and drop empty values.
Then I grouped the dataset by genre and found the average duration time for each movie using aggregate function. I sorted the values and choose top 10 genres.

```
# drop null values

netflix_movies = netflix_movies.dropna()
```

```
# drop variable column

netflix_movies = netflix_movies.drop('variable', axis=1)
```

```
netflix_movies
```

```
# group by genre and find the average duration for each genre.
# give a column name, reset index and round the average to 2
# decimal numbers

netflix_movies_grouped = netflix_movies.groupby(['genre']).agg({'duration': ['mean']})
netflix_movies_grouped.columns = ['average_duration']
netflix_movies_grouped = netflix_movies_grouped.reset_index()
netflix_movies_grouped = netflix_movies_grouped.round(2)
```

```
# sort the dataset in descending.

netflix_movies_grouped = netflix_movies_grouped.sort_values(by=['average_duration'], ascending=False)
```

```
# select only top 10 rows with highest duration

final_df = netflix_movies_grouped.head(10)
```

Last, I plotted a graph.

```
# plot a figure for most watched genre in movies.

ax = final_df.plot.bar(legend=True, x = 'genre',
                               title='Average Movie Duration For Top 10 Categories',
                               color='orange')
ax.set_xlabel("Genres")
ax.set_ylabel("Average Duration")

plt.savefig('Analysis3.png', dpi=300)

plt.show()
```

Output Files.

- Analysis 1:
    - 1 csv file before pivoting dataset, but just grouped one.

	type	year_added	nr_of_movies_shows_added_through_years
0	Movie	2008	1
1	Movie	2009	2
2	Movie	2010	1
3	Movie	2011	13
4	Movie	2012	4
5	Movie	2013	6
6	Movie	2014	14
7	Movie	2015	47
8	Movie	2016	204
9	Movie	2017	778
10	Movie	2018	1122
11	Movie	2019	1347
12	Movie	2020	139
13	TV Show	2013	1
14	TV Show	2015	3
15	TV Show	2016	7
16	TV Show	2017	27
17	TV Show	2018	18
18	TV Show	2019	39
19	TV Show	2020	1

    - 1 csv file after pivoting dataset.

type	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020
Movie	1	2	1	13	4	6	14	47	204	778	1122	1347	139
TV Show	0	0	0	0	0	1	0	3	7	27	18	39	1

    - 1 plotted graph

- Analysis 2:
    - 1 csv file before pivoting dataset, but just grouped one.

	country	title	type	movies_count	%
0	United States	1718	Movie	1685	98.08
1	United States	1718	TV Show	33	1.92
2	India	762	Movie	757	99.34
3	India	762	TV Show	5	0.66
4	United Kingdom	341	Movie	323	94.72
5	United Kingdom	341	TV Show	18	5.28
6	Canada	194	Movie	189	97.42
7	Canada	194	TV Show	5	2.58
8	France	191	Movie	188	98.43
9	France	191	TV Show	3	1.57
10	Spain	120	Movie	116	96.67
11	Spain	120	TV Show	4	3.33
12	Germany	113	Movie	112	99.12
13	Germany	113	TV Show	1	0.88
14	Hong Kong	91	Movie	91	100
15	Japan	80	Movie	70	87.5
16	Japan	80	TV Show	10	12.5
17	China	79	Movie	78	98.73
18	China	79	TV Show	1	1.27

    - 1 csv file after pivoting dataset.

type	Canada	China	France	Germany	Hong Kong	India	Japan	Spain	United Kingdom	United States
Movie	189	78	188	112	91	757	70	116	323	1685
TV Show	5	1	3	1	0	5	10	4	18	33

    - 1 plotted graph

- Analysis 3:

    - 1 csv file

	genre	average_duration
2	Classic Movies	122.18
12	Music & Musicals	117.99
21	Classic Movies	117.19
6	Dramas	116.61
1	Children & Family Movies	116
18	Action & Adventure	114.26
10	International Movies	113.6
28	International Movies	113.24
25	Dramas	112.42
13	Romantic Movies	111.9

    - 1 plotted graph