Final Project Submission

Please fill out:

- Student name: Dorothy Chomba
- Student pace: Part time
- Scheduled project review date/time: Monday 19, 2024
- Instructor name: William Okomba
- Blog post URL:

In [1]:

```
# Your code here - remember to use markdown cells for comments as well!
# identifying the list of files available
!ls
```

```
Phase 1 Project- Dorothy Presentation.pdf
bom.movie_gross.csv
student- Dorothy.ipynb
title.ratings.csv
```

In [2]:

```
import pandas as pd
df= pd.read_csv(r'title.ratings.csv')
```

In [3]:

```
df.head()
```

Out[3]:

	tconst	averagerating	numvotes
0	tt10356526	8.3	31
1	tt10384606	8.9	559
2	tt1042974	6.4	20
3	tt1043726	4.2	50352
4	tt1060240	6.5	21

In [4]:

```
df.tail()
```

Out[4]:

	tconst	averagerating	numvotes
73851	tt9805820	8.1	25
73852	tt9844256	7.5	24
73853	tt9851050	4.7	14
73854	tt9886934	7.0	5
73855	tt9894098	6.3	128

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 73856 entries, 0 to 73855
Data columns (total 3 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   tconst         73856 non-null  object 
 1   averagerating  73856 non-null  float64
 2   numvotes       73856 non-null  int64  
dtypes: float64(1), int64(1), object(1)
memory usage: 1.7+ MB
```

In [6]:
```
df.shape
```

Out[6]:
```
(73856, 3)
```

In [7]:
```
df.describe()
```

Out[7]:

	averagerating	numvotes
count	73856.000000	7.385600e+04
mean	6.332729	3.523662e+03
std	1.474978	3.029402e+04
min	1.000000	5.000000e+00
25%	5.500000	1.400000e+01
50%	6.500000	4.900000e+01
75%	7.400000	2.820000e+02
max	10.000000	1.841066e+06

Missing values

In [8]:
```
df.dropna().sum
```

Out[8]:
```
<bound method DataFrame.sum of            tconst  averagerating  numvotes
0      tt10356526            8.3        31
1      tt10384606            8.9       559
2       tt1042974            6.4        20
3       tt1043726            4.2     50352
4       tt1060240            6.5        21
...           ...            ...       ...
73851   tt9805820            8.1        25
73852   tt9844256            7.5        24
73853   tt9851050            4.7        14
73854   tt9886934            7.0         5
73855   tt9894098            6.3       128

[73856 rows x 3 columns]>
```

In [9]:
```
df.isna()
```

Out[9]:

	tconst	averagerating	numvotes
0	False	False	False
1	False	False	False
2	False	False	False
3	False	False	False
4	False	False	False
...	...	...	...
73851	False	False	False
73852	False	False	False
73853	False	False	False
73854	False	False	False
73855	False	False	False

73856 rows × 3 columns

duplicated values

In [10]:
```
df.duplicated()
```

Out[10]:
```
0        False
1        False
2        False
3        False
4        False
         ...  
73851    False
73852    False
73853    False
73854    False
73855    False
Length: 73856, dtype: bool
```

In [11]:
```
df['averagerating'].min()
```

Out[11]:
```
1.0
```

In [12]:
```
df['averagerating'].max()
```

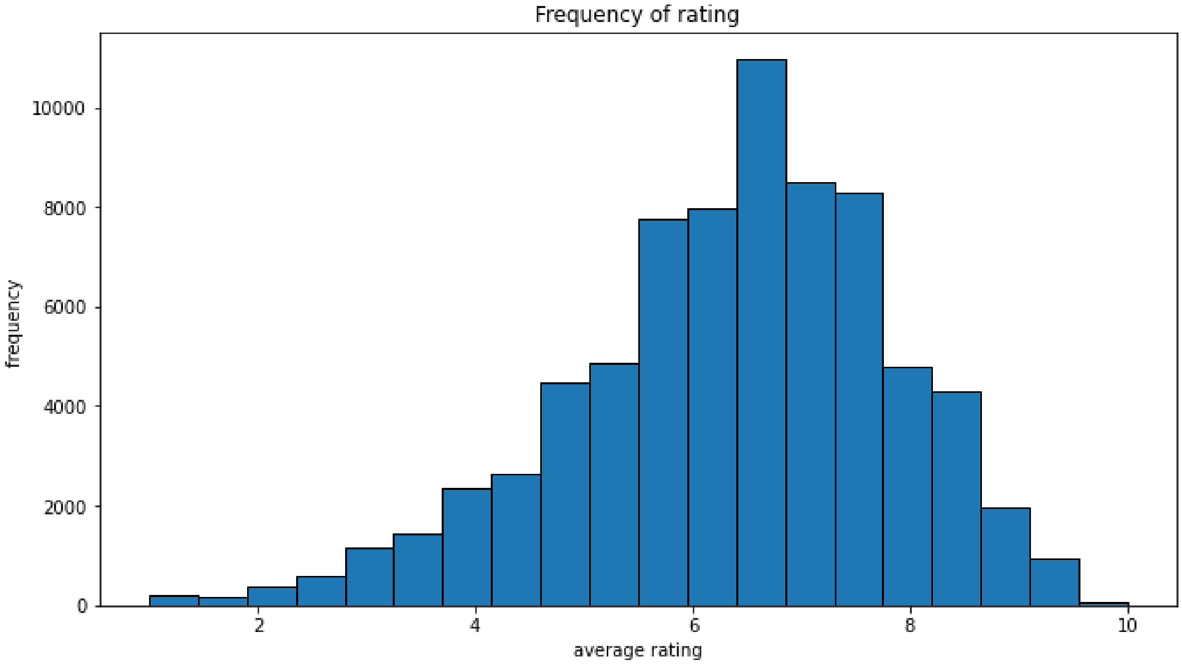
Out[12]:
```
10.0
```

Visualisation

In [13]:
```
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

In [14]:
```
x=df['averagerating']
fig, axes =plt.subplots(figsize=(11, 6))
axes.set_facecolor('white')
axes.set_title('Frequency of rating')
# average_rating_bins=[1,2,3,4,5,6,7,8,9,10]
# df['averagerating_group']= pd.cut(df['averagerating'],bins=average_rating_bi
plt.hist (x,bins=20,edgecolor='black')
plt.xlabel('average rating')
plt.ylabel('frequency ')
plt.show
```

Out[14]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



In [15]:

```
# x=df['averagerating']
# y=df['numvotes']
# plt.figure(figsize=(10,6))

# plt.bar(x, y, label='relationship between rating and votes')
# average_rating_bins=[1,2,3,4,5,6,7,8,9,10]
# df['averagerating_group']= pd.cut(df['averagerating'],bins=average_rating_bir
# plt.xlabel('average rating')
# plt.ylabel('number of votes')

# plt.title('relationship between rating and votes')
# plt.legend()

# # Output the final plot
# plt.show()
```

```
import matplotlib.pyplot as plt
import pandas as pd
df['averagerating_group']= pd.cut(df['averagerating'], bins=
[1,2,3,4,5,6,7,8,9,10])
# Assuming df is your DataFrame containing columns 'averagerating' and
'numvotes'

# Group the data by 'averagerating_group' and sum the 'numvotes' in each
group
grouped_data = df.groupby('averagerating_group')['numvotes'].sum()

# Plot the grouped data
plt.figure(figsize=(10, 6))
grouped_data.plot(kind='bar', color='skyblue', label='relationship between
rating and votes')

plt.xlabel('Average Rating Group')
plt.ylabel('Number of Votes')
plt.title('Relationship between Rating and Votes')
plt.legend()

# Output the final plot
plt.show()
```

In [16]:

```
!ls
```

```
Phase 1 Project- Dorothy Presentation.pdf
bom.movie_gross.csv
student- Dorothy.ipynb
title.ratings.csv
```

In [17]:
```
pd.read_csv(r'bom.movie_gross.csv')
```

Out[17]:

	title	studio	domestic_gross	foreign_gross	year
0	Toy Story 3	BV	415000000.0	652000000	2010
1	Alice in Wonderland (2010)	BV	334200000.0	691300000	2010
2	Harry Potter and the Deathly Hallows Part 1	WB	296000000.0	664300000	2010
3	Inception	WB	292600000.0	535700000	2010
4	Shrek Forever After	P/DW	238700000.0	513900000	2010
...	...	...	...	...	...
3382	The Quake	Magn.	6200.0	NaN	2018
3383	Edward II (2018 re-release)	FM	4800.0	NaN	2018
3384	El Pacto	Sony	2500.0	NaN	2018
3385	The Swan	Synergetic	2400.0	NaN	2018
3386	An Actor Prepares	Grav.	1700.0	NaN	2018

3387 rows × 5 columns

In [18]:
```
df1=pd.read_csv(r'bom.movie_gross.csv')
```

In [19]:
```
df1.describe()
```

Out[19]:

	domestic_gross	year
count	3.359000e+03	3387.000000
mean	2.874585e+07	2013.958075
std	6.698250e+07	2.478141
min	1.000000e+02	2010.000000
25%	1.200000e+05	2012.000000
50%	1.400000e+06	2014.000000
75%	2.790000e+07	2016.000000
max	9.367000e+08	2018.000000

In [20]:
```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3387 entries, 0 to 3386
Data columns (total 5 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   title           3387 non-null   object 
 1   studio          3382 non-null   object 
 2   domestic_gross  3359 non-null   float64
 3   foreign_gross   2037 non-null   object 
 4   year            3387 non-null   int64  
dtypes: float64(1), int64(1), object(3)
memory usage: 132.4+ KB
```

In [21]:
```
df1.head()
```

Out[21]:

	title	studio	domestic_gross	foreign_gross	year
0	Toy Story 3	BV	415000000.0	652000000	2010
1	Alice in Wonderland (2010)	BV	334200000.0	691300000	2010
2	Harry Potter and the Deathly Hallows Part 1	WB	296000000.0	664300000	2010
3	Inception	WB	292600000.0	535700000	2010
4	Shrek Forever After	P/DW	238700000.0	513900000	2010

In [22]:
```
df1.dropna().sum()
```

Out[22]:
```
title             Toy Story 3Alice in Wonderland (2010)Harry Pot...
studio            BVBVWBWBP/DWSum.Par.BVUni.P/DWWBFoxWein.BVSony...
domestic_gross                                          9.43688e+10
foreign_gross     652000000691300000664300000535700000513900003...
year                                                        4041107
dtype: object
```

In [23]:
```
df1.shape
```

Out[23]:
```
(3387, 5)
```

In [24]:
```
df1.isna().sum()
```

Out[24]:
```
title                0
studio               5
domestic_gross      28
foreign_gross     1350
year                 0
dtype: int64
```

In [25]:
```
df1=df1.dropna(axis=0)
```

In [26]:
```
df1.shape
```

Out[26]:
```
(2007, 5)
```

In [27]:
```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2007 entries, 0 to 3353
Data columns (total 5 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   title           2007 non-null   object 
 1   studio          2007 non-null   object 
 2   domestic_gross  2007 non-null   float64
 3   foreign_gross   2007 non-null   object 
 4   year            2007 non-null   int64  
dtypes: float64(1), int64(1), object(3)
memory usage: 94.1+ KB
```

In [28]:
```
type(df1['foreign_gross'])
```

Out[28]:
```
pandas.core.series.Series
```

In [29]:
```
df1['foreign_gross'] = pd.to_numeric(df1['foreign_gross'], errors='coerce')
df1['foreign_gross']= df1['foreign_gross'].astype(float)
```

In [30]:
```
df1.describe()
```

Out[30]:

	domestic_gross	foreign_gross	year
count	2.007000e+03	2.002000e+03	2007.000000
mean	4.701984e+07	7.597967e+07	2013.506228
std	8.162689e+07	1.383001e+08	2.597997
min	4.000000e+02	6.000000e+02	2010.000000
25%	6.700000e+05	4.000000e+06	2011.000000
50%	1.670000e+07	1.960000e+07	2013.000000
75%	5.605000e+07	7.645000e+07	2016.000000
max	9.367000e+08	9.605000e+08	2018.000000

```
In [31]: df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2007 entries, 0 to 3353
Data columns (total 5 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   title           2007 non-null   object 
 1   studio          2007 non-null   object 
 2   domestic_gross  2007 non-null   float64
 3   foreign_gross   2002 non-null   float64
 4   year            2007 non-null   int64  
dtypes: float64(2), int64(1), object(2)
memory usage: 94.1+ KB
```

```
In [32]: df1.isna().sum()
```

```
Out[32]: title             0
         studio            0
         domestic_gross    0
         foreign_gross     5
         year              0
         dtype: int64
```

```
In [33]: df1.duplicated()
```

```
Out[33]: 0       False
         1       False
         2       False
         3       False
         4       False
                 ...  
         3275    False
         3286    False
         3309    False
         3342    False
         3353    False
         Length: 2007, dtype: bool
```

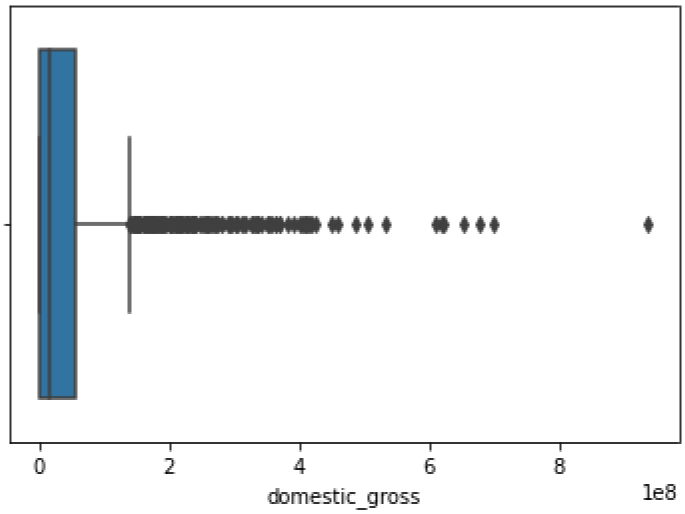
In [34]:

```
import seaborn as sns
sns.boxplot(df1['domestic_gross'])
```

```
C:\Users\earlysense.LENOVOE31-PC\anaconda3\envs\learn-env\lib\site-packages\s
eaborn\_decorators.py:36: FutureWarning: Pass the following variable as a key
word arg: x. From version 0.12, the only valid positional argument will be `d
ata`, and passing other arguments without an explicit keyword will result in
an error or misinterpretation.
  warnings.warn(
```

Out[34]:

```
<AxesSubplot:xlabel='domestic_gross'>
```



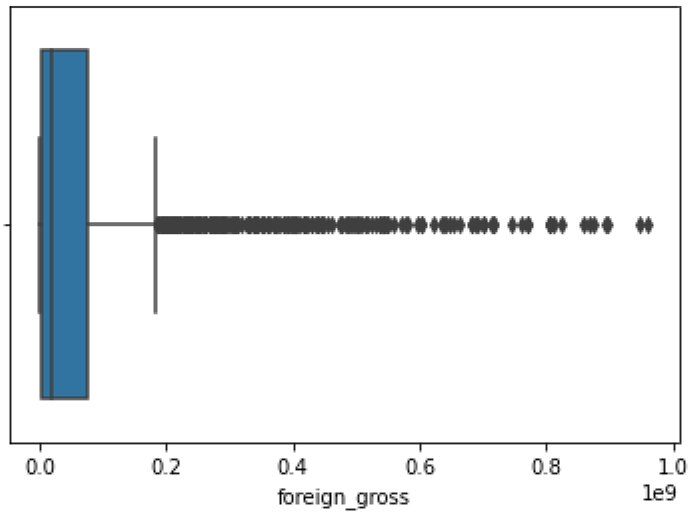
In [35]:

```
sns.boxplot(df1['foreign_gross'])
```

```
C:\Users\earlysense.LENOVOE31-PC\anaconda3\envs\learn-env\lib\site-packages\s
eaborn\_decorators.py:36: FutureWarning: Pass the following variable as a key
word arg: x. From version 0.12, the only valid positional argument will be `d
ata`, and passing other arguments without an explicit keyword will result in
an error or misinterpretation.
  warnings.warn(
```

Out[35]:

```
<AxesSubplot:xlabel='foreign_gross'>
```



In [36]:

```
# df1['foreign_gross']= df1['foreign_gross'].fillna(median, inplace=true)
q1=df1['foreign_gross'].quantile(0.25)
q3=df1['foreign_gross'].quantile(0.75)
iqr=q3-q1
lower_bound= q1- 1.5*iqr
upper_bound= q3+ 1.5*iqr
df1['foreign_gross']=df1['foreign_gross'].clip(lower_bound, upper_bound)
```

In [37]:

```
q1=df1['domestic_gross'].quantile(0.25)
q3=df1['domestic_gross'].quantile(0.75)
iqr=q3-q1
lower_bound= q1- 1.5*iqr
upper_bound= q3+ 1.5*iqr
df1['domestic_gross']=df1['domestic_gross'].clip(lower_bound, upper_bound)
```

In [38]:

```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2007 entries, 0 to 3353
Data columns (total 5 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   title           2007 non-null   object 
 1   studio          2007 non-null   object 
 2   domestic_gross  2007 non-null   float64
 3   foreign_gross   2002 non-null   float64
 4   year            2007 non-null   int64  
dtypes: float64(2), int64(1), object(2)
memory usage: 94.1+ KB
```

In [39]:

```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2007 entries, 0 to 3353
Data columns (total 5 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   title           2007 non-null   object 
 1   studio          2007 non-null   object 
 2   domestic_gross  2007 non-null   float64
 3   foreign_gross   2002 non-null   float64
 4   year            2007 non-null   int64  
dtypes: float64(2), int64(1), object(2)
memory usage: 94.1+ KB
```

In [40]:

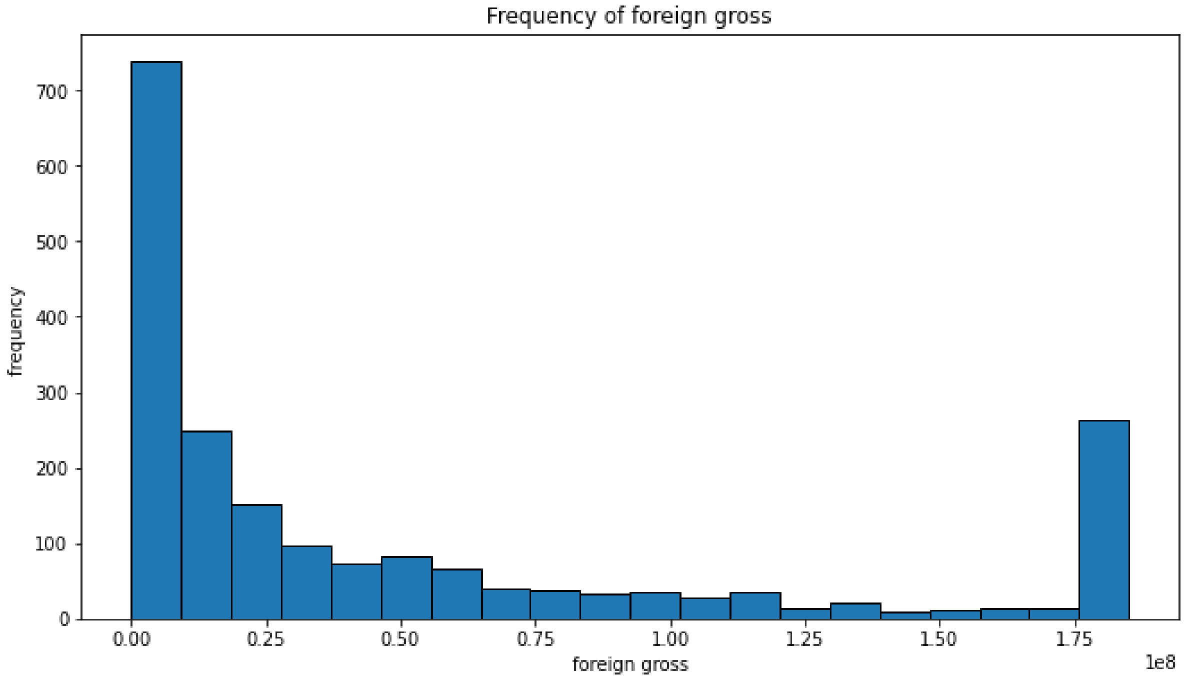
```
median= df1['foreign_gross'].median()
df1['foreign_gross']= df1['foreign_gross'].fillna(median)
```

In [41]:
```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2007 entries, 0 to 3353
Data columns (total 5 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   title           2007 non-null   object 
 1   studio          2007 non-null   object 
 2   domestic_gross  2007 non-null   float64
 3   foreign_gross   2007 non-null   float64
 4   year            2007 non-null   int64  
dtypes: float64(2), int64(1), object(2)
memory usage: 94.1+ KB
```

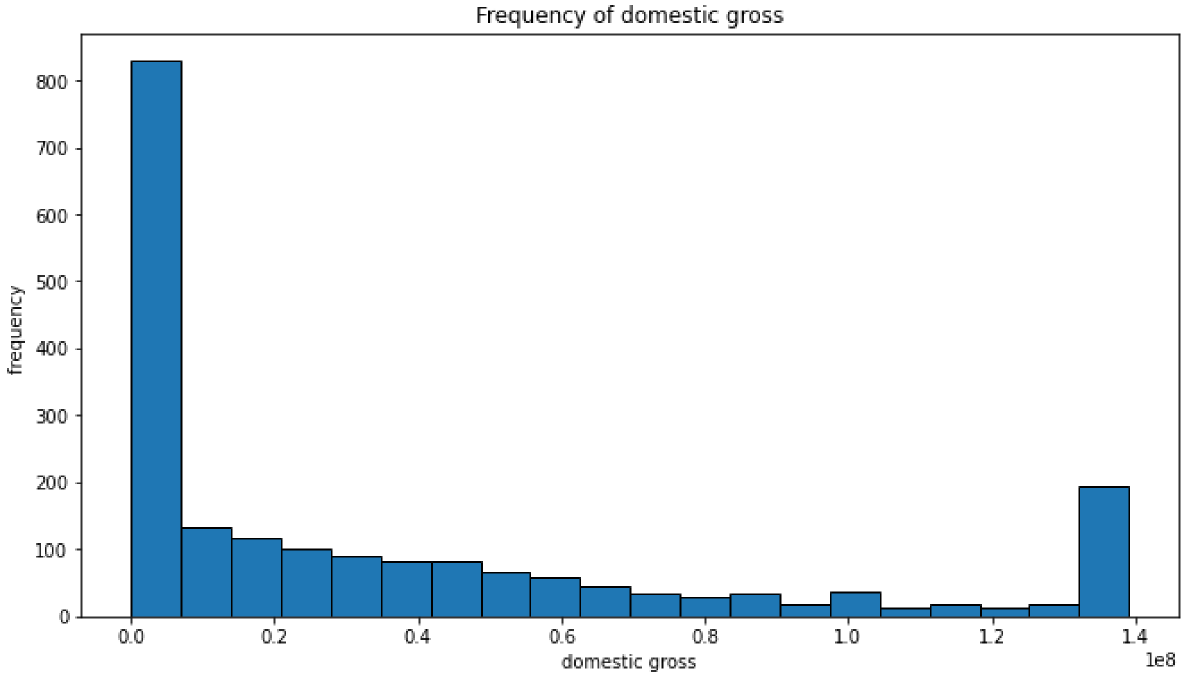
In [42]:
```
x=df1['foreign_gross']
fig, axes =plt.subplots(figsize=(11, 6))
axes.set_facecolor('white')
axes.set_title('Frequency of foreign gross')
# average_rating_bins=[1,2,3,4,5,6,7,8,9,10]
# df['averagerating_group']= pd.cut(df['averagerating'],bins=average_rating_bi
plt.hist (x,bins=20,edgecolor='black' )
plt.xlabel('foreign gross')
plt.ylabel('frequency ')
plt.show
```

Out[42]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```

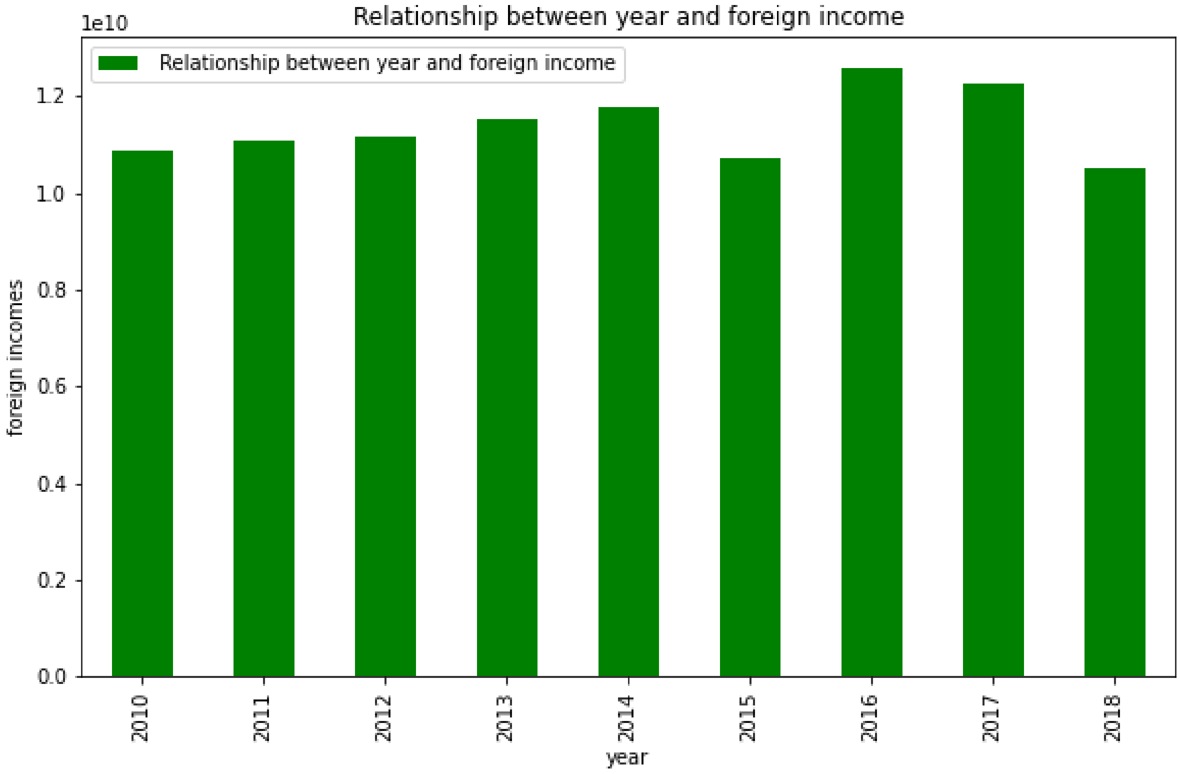


In [43]:
```
x=df1['domestic_gross']
fig, axes =plt.subplots(figsize=(11, 6))
axes.set_facecolor('white')
axes.set_title('Frequency of domestic gross')
# average_rating_bins=[1,2,3,4,5,6,7,8,9,10]
# df['averagerating_group']= pd.cut(df['averagerating'],bins=average_rating_bir
plt.hist (x,bins=20,edgecolor='black')
plt.xlabel('domestic gross')
plt.ylabel('frequency ')
plt.show
```

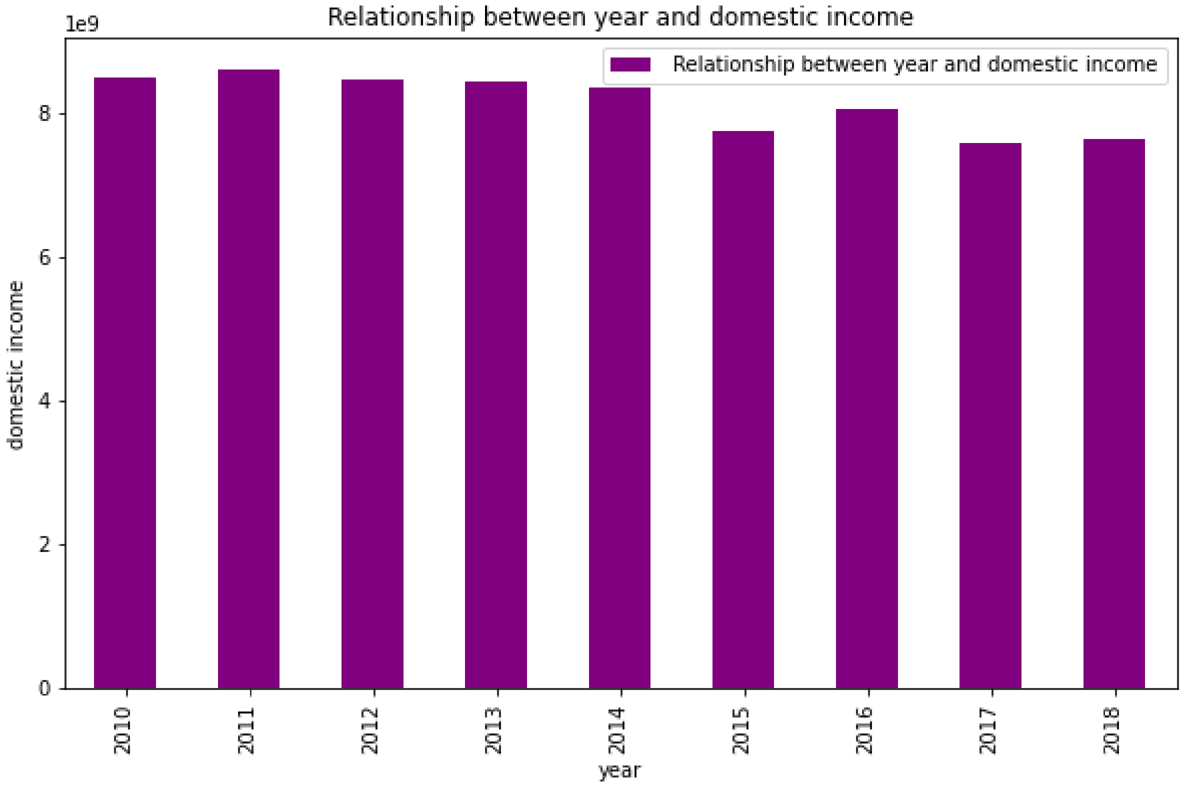
Out[43]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



In [44]:
```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
# df1['year_group']= pd.cut(df1['year'], bins=10)

# Group the data by 'year' and sum the 'foreign gross' in each group
grouped_data = df1.groupby("year")['foreign_gross'].sum()


# Plot the grouped data
plt.figure(figsize=(10, 6))
grouped_data.plot(kind='bar', color='green', label=' Relationship between year

plt.xlabel('year')
plt.ylabel('foreign incomes')
plt.title('Relationship between year and foreign income')
plt.legend()

# Output the final plot
plt.show()
```

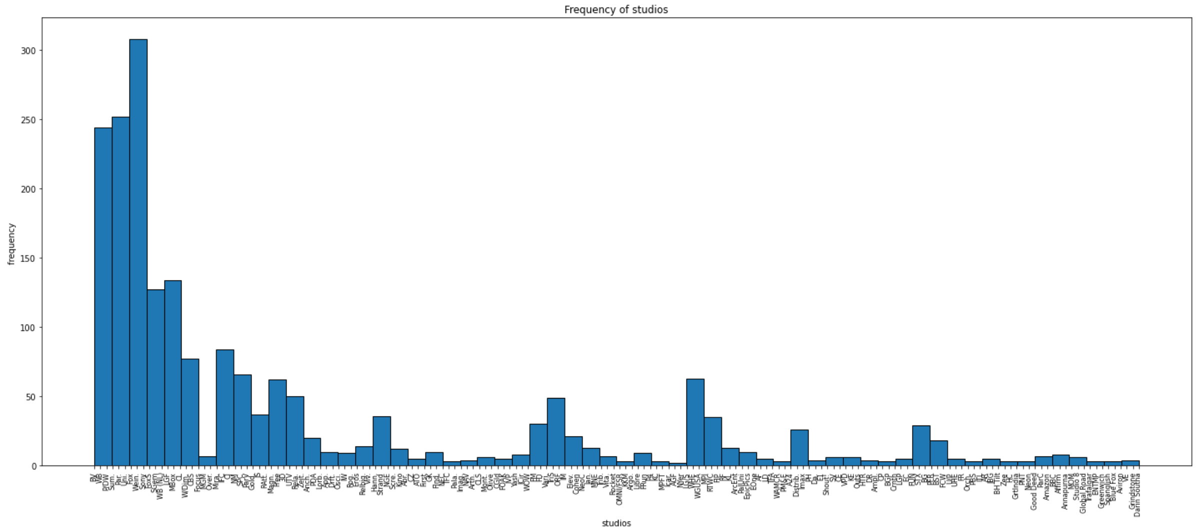


In [45]:
```python
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
# df1['year_group']= pd.cut(df1['year'], bins=10)

# Group the data by 'year' and sum the 'foreign gross' in each group
grouped_data = df1.groupby("year")['domestic_gross'].sum()


# Plot the grouped data
plt.figure(figsize=(10, 6))
grouped_data.plot(kind='bar', color='purple', label=' Relationship between year

plt.xlabel('year')
plt.ylabel('domestic income')
plt.title('Relationship between year and domestic income')
plt.legend()

# Output the final plot
plt.show()
```



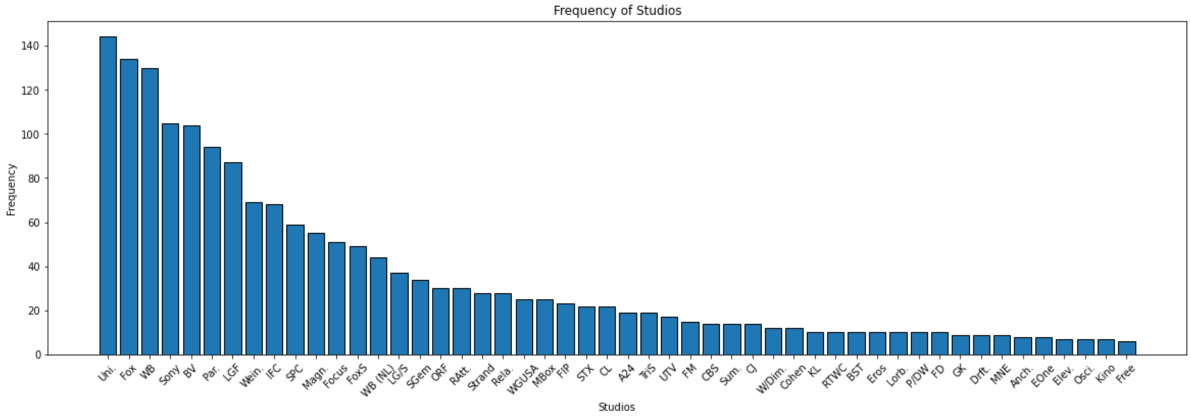
In [46]:
```
x=df1['studio']
fig, axes =plt.subplots(figsize=(20, 9))
axes.set_facecolor('white')
axes.set_title('Frequency of studios')
plt.hist(x,bins=60,edgecolor='black')
plt.xlabel('studios')
plt.ylabel('frequency ')
plt.xticks(rotation=90, fontsize=8)
plt.tight_layout()
plt.show
```

Out[46]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



In [47]:

```
# Count the frequency of each studio
studio_counts = df1['studio'].value_counts()

# Extract studios and their frequencies
studios = studio_counts.index[:50]
frequencies = studio_counts.values[:50]

# Plot the frequencies
plt.figure(figsize=(20, 6))
plt.bar(studios, frequencies, edgecolor='black')
plt.title('Frequency of Studios')
plt.xlabel('Studios')
plt.ylabel('Frequency')
plt.xticks(rotation=45, fontsize=10)
plt.show()
```



In [48]:

```
studio_counts = df1['studio'].value_counts()
studio_counts
```

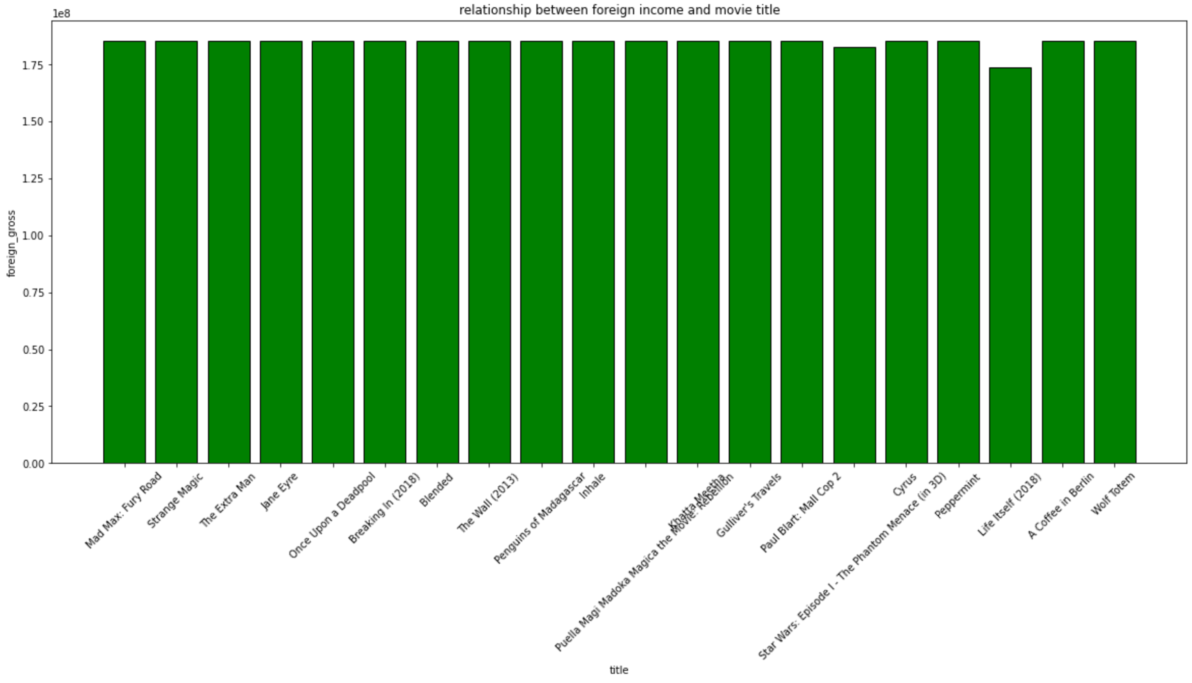
Out[48]:

```
Uni.            144
Fox             134
WB              130
Sony            105
BV              104
               ...
ICir              1
LGP               1
Darin Southa      1
PalUni            1
NYer              1
Name: studio, Length: 172, dtype: int64
```

In [49]:
```
df1.info()
```

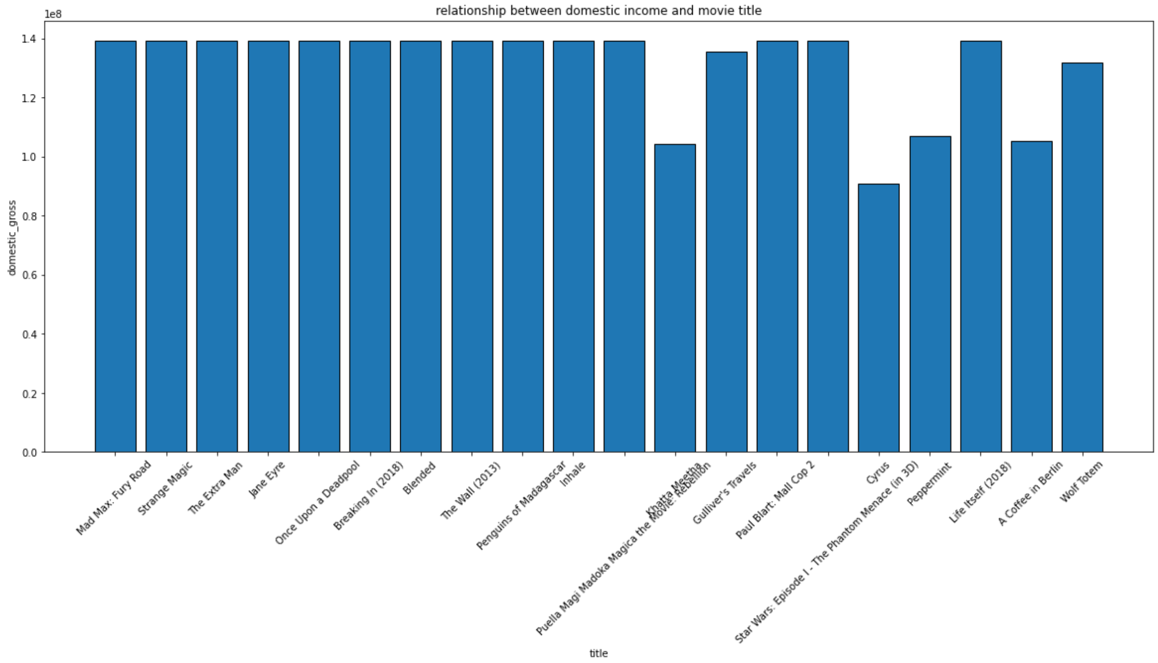
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2007 entries, 0 to 3353
Data columns (total 5 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   title           2007 non-null   object 
 1   studio          2007 non-null   object 
 2   domestic_gross  2007 non-null   float64
 3   foreign_gross   2007 non-null   float64
 4   year            2007 non-null   int64  
dtypes: float64(2), int64(1), object(2)
memory usage: 94.1+ KB
```

In [50]:
```
title_counts = df1['title'].value_counts()
title = title_counts.index[:20]
foreign_gross=df1['foreign_gross'][:20]

# Plot the frequencies
plt.figure(figsize=(20, 8))
plt.bar(title,foreign_gross, edgecolor='black', color='green')
plt.title('relationship between foreign income and movie title')
plt.xlabel('title')
plt.ylabel('foreign_gross')
plt.xticks(rotation=45, fontsize=10)
plt.show()
```



In [51]:

```
title_counts = df1['title'].value_counts()
title = title_counts.index[:20]
domestic_gross=df1['domestic_gross'][:20]

# Plot the frequencies
plt.figure(figsize=(20, 8))
plt.bar(title,domestic_gross, edgecolor='black')
plt.title('relationship between domestic income and movie title')
plt.xlabel('title')
plt.ylabel('domestic_gross')
plt.xticks(rotation=45, fontsize=10)
plt.show()
```



In [52]:

```
pip install nbconvert
```

```
Requirement already satisfied: nbconvert in c:\users\earlysense.lenovoe31-pc
\anaconda3\envs\learn-env\lib\site-packages (6.0.7)
Requirement already satisfied: pygments>=2.4.1 in c:\users\earlysense.lenovoe
31-pc\anaconda3\envs\learn-env\lib\site-packages (from nbconvert) (2.7.1)
Requirement already satisfied: bleach in c:\users\earlysense.lenovoe31-pc\ana
conda3\envs\learn-env\lib\site-packages (from nbconvert) (3.2.1)
Requirement already satisfied: jupyterlab-pygments in c:\users\earlysense.len
ovoe31-pc\anaconda3\envs\learn-env\lib\site-packages (from nbconvert) (0.1.2)
Requirement already satisfied: pandocfilters>=1.4.1 in c:\users\earlysense.le
novoe31-pc\anaconda3\envs\learn-env\lib\site-packages (from nbconvert) (1.4.
2)
Requirement already satisfied: nbclient<0.6.0,>=0.5.0 in c:\users\earlysense.
lenovoe31-pc\anaconda3\envs\learn-env\lib\site-packages (from nbconvert) (0.
5.1)
Requirement already satisfied: testpath in c:\users\earlysense.lenovoe31-pc\a
naconda3\envs\learn-env\lib\site-packages (from nbconvert) (0.4.4)
Requirement already satisfied: mistune<2,>=0.8.1 in c:\users\earlysense.lenov
oe31-pc\anaconda3\envs\learn-env\lib\site-packages (from nbconvert) (0.8.4)
Requirement already satisfied: nbformat>=4.4 in c:\users\earlysense.lenovoe31
-pc\anaconda3\envs\learn-env\lib\site-packages (from nbconvert) (5.0.8)
Requirement already satisfied: jinja2>=2.4 in c:\users\earlysense.lenovoe31-p
c\anaconda3\envs\learn-env\lib\site-packages (from nbconvert) (2.11.2)
Requirement already satisfied: traitlets>=4.2 in c:\users\earlysense.lenovoe3
1-pc\anaconda3\envs\learn-env\lib\site-packages (from nbconvert) (5.0.5)
Requirement already satisfied: entrypoints>=0.2.2 in c:\users\earlysense.leno
voe31-pc\anaconda3\envs\learn-env\lib\site-packages (from nbconvert) (0.3)
Requirement already satisfied: jupyter-core in c:\users\earlysense.lenovoe31-
pc\anaconda3\envs\learn-env\lib\site-packages (from nbconvert) (4.6.3)
Requirement already satisfied: defusedxml in c:\users\earlysense.lenovoe31-pc
\anaconda3\envs\learn-env\lib\site-packages (from nbconvert) (0.6.0)
Requirement already satisfied: packaging in c:\users\earlysense.lenovoe31-pc
\anaconda3\envs\learn-env\lib\site-packages (from bleach->nbconvert) (20.4)
Requirement already satisfied: six>=1.9.0 in c:\users\earlysense.lenovoe31-pc
\anaconda3\envs\learn-env\lib\site-packages (from bleach->nbconvert) (1.15.0)
Requirement already satisfied: webencodings in c:\users\earlysense.lenovoe31-
pc\anaconda3\envs\learn-env\lib\site-packages (from bleach->nbconvert) (0.5.
1)
Requirement already satisfied: async-generator in c:\users\earlysense.lenovoe
31-pc\anaconda3\envs\learn-env\lib\site-packages (from nbclient<0.6.0,>=0.5.0
->nbconvert) (1.10)
Requirement already satisfied: jupyter-client>=6.1.5 in c:\users\earlysense.l
enovoe31-pc\anaconda3\envs\learn-env\lib\site-packages (from nbclient<0.6.0,>
=0.5.0->nbconvert) (6.1.7)
Requirement already satisfied: nest-asyncio in c:\users\earlysense.lenovoe31-
pc\anaconda3\envs\learn-env\lib\site-packages (from nbclient<0.6.0,>=0.5.0->n
bconvert) (1.4.1)
Requirement already satisfied: ipython-genutils in c:\users\earlysense.lenovo
e31-pc\anaconda3\envs\learn-env\lib\site-packages (from nbformat>=4.4->nbconv
ert) (0.2.0)
Requirement already satisfied: jsonschema!=2.5.0,>=2.4 in c:\users\earlysens
e.lenovoe31-pc\anaconda3\envs\learn-env\lib\site-packages (from nbformat>=4.4
->nbconvert) (3.2.0)
Requirement already satisfied: MarkupSafe>=0.23 in c:\users\earlysense.lenovo
e31-pc\anaconda3\envs\learn-env\lib\site-packages (from jinja2>=2.4->nbconver
t) (1.1.1)
Requirement already satisfied: pywin32>=1.0; sys_platform == "win32" in c:\us
ers\earlysense.lenovoe31-pc\anaconda3\envs\learn-env\lib\site-packages (from
jupyter-core->nbconvert) (227)
Requirement already satisfied: pyparsing>=2.0.2 in c:\users\earlysense.lenovo
e31-pc\anaconda3\envs\learn-env\lib\site-packages (from packaging->bleach->nb
convert) (2.4.7)
Requirement already satisfied: pyzmq>=13 in c:\users\earlysense.lenovoe31-pc
\anaconda3\envs\learn-env\lib\site-packages (from jupyter-client>=6.1.5->nbcl
ient<0.6.0,>=0.5.0->nbconvert) (19.0.2)
Requirement already satisfied: tornado>=4.1 in c:\users\earlysense.lenovoe31-
pc\anaconda3\envs\learn-env\lib\site-packages (from jupyter-client>=6.1.5->nb
client<0.6.0,>=0.5.0->nbconvert) (6.0.4)
Requirement already satisfied: python-dateutil>=2.1 in c:\users\earlysense.le
novoe31-pc\anaconda3\envs\learn-env\lib\site-packages (from jupyter-client>=
6.1.5->nbclient<0.6.0,>=0.5.0->nbconvert) (2.8.1)
Requirement already satisfied: attrs>=17.4.0 in c:\users\earlysense.lenovoe31
-pc\anaconda3\envs\learn-env\lib\site-packages (from jsonschema!=2.5.0,>=2.4-
>nbformat>=4.4->nbconvert) (20.2.0)
Requirement already satisfied: pyrsistent>=0.14.0 in c:\users\earlysense.leno
voe31-pc\anaconda3\envs\learn-env\lib\site-packages (from jsonschema!=2.5.0,>
=2.4->nbformat>=4.4->nbconvert) (0.17.3)
Requirement already satisfied: setuptools in c:\users\earlysense.lenovoe31-pc
\anaconda3\envs\learn-env\lib\site-packages (from jsonschema!=2.5.0,>=2.4->nb
format>=4.4->nbconvert) (50.3.0.post20201103)
Note: you may need to restart the kernel to use updated packages.
```

In [ ]: