# Business Problem

Microsoft sees all the big companies creating original video content and they want to get in on the fun. They have decided to create a new movie studio, but they don't know anything about creating movies. You are charged with exploring what types of films are currently doing the best at the box office.In the following project, I will attempt to translate those findings into actionable insights that the head of Microsoft's new movie studio can use to help decide what type of films to create.

In [19]:
```
#Start by importing the necessary libraries

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

In [20]:
```
#Open csv file as a Pandas DataFrame to load the data

df = pd.read_csv('title_basics.csv')
df
```

Out[20]:

	tconst	primary_title	original_title	start_year	runtime_minutes	genres
0	tt0063540	Sunghursh	Sunghursh	2013	175.0	Action,Crime,Drama
1	tt0066787	One Day Before the Rainy Season	Ashad Ka Ek Din	2019	114.0	Biography,Drama
2	tt0069049	The Other Side of the Wind	The Other Side of the Wind	2018	122.0	Drama
3	tt0069204	Sabse Bada Sukh	Sabse Bada Sukh	2018	NaN	Comedy,Drama
4	tt0100275	The Wandering Soap Opera	La Telenovela Errante	2017	80.0	Comedy,Drama,Fantasy
...	...	...	...	...	...	...
146139	tt9916538	Kuambil Lagi Hatiku	Kuambil Lagi Hatiku	2019	123.0	Drama
146140	tt9916622	Rodolpho Teóphilo - O Legado de um Pioneiro	Rodolpho Teóphilo - O Legado de um Pioneiro	2015	NaN	Documentary
146141	tt9916706	Dankyavar Danka	Dankyavar Danka	2013	NaN	Comedy
146142	tt9916730	6 Gunn	6 Gunn	2017	116.0	NaN
146143	tt9916754	Chico Albuquerque - Revelações	Chico Albuquerque - Revelações	2013	NaN	Documentary

146144 rows × 6 columns

In [21]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 146144 entries, 0 to 146143
Data columns (total 6 columns):
 #   Column           Non-Null Count   Dtype  
---  ------           --------------   -----  
 0   tconst           146144 non-null  object 
 1   primary_title    146143 non-null  object 
 2   original_title   146122 non-null  object 
 3   start_year       146144 non-null  int64  
 4   runtime_minutes  114405 non-null  float64
 5   genres           140736 non-null  object 
dtypes: float64(1), int64(1), object(4)
memory usage: 6.7+ MB
```

In [22]:
```
#The columns primary_title and original_title have have almost identical rows,
#Having noted row similarities in the two columns drop the primary_title

# Drop a column using axis=1
column_to_drop = 'primary_title'
newdf = df.drop('primary_title', axis=1)

# Display the resulting DataFrame
print(newdf)
```

```
           tconst                               original_title  start_year  \
0       tt0063540                                    Sunghursh        2013   
1       tt0066787                              Ashad Ka Ek Din        2019   
2       tt0069049                   The Other Side of the Wind        2018   
3       tt0069204                              Sabse Bada Sukh        2018   
4       tt0100275                         La Telenovela Errante        2017   
...           ...                                          ...         ...   
146139  tt9916538                          Kuambil Lagi Hatiku        2019   
146140  tt9916622  Rodolpho Teóphilo - O Legado de um Pioneiro        2015   
146141  tt9916706                              Dankyavar Danka        2013   
146142  tt9916730                                       6 Gunn        2017   
146143  tt9916754               Chico Albuquerque - Revelações        2013   

        runtime_minutes                genres  
0                 175.0    Action,Crime,Drama  
1                 114.0       Biography,Drama  
2                 122.0                 Drama  
3                   NaN          Comedy,Drama  
4                  80.0  Comedy,Drama,Fantasy  
...                 ...                   ...  
146139            123.0                 Drama  
146140              NaN           Documentary  
146141              NaN                Comedy  
146142            116.0                   NaN  
146143              NaN           Documentary  

[146144 rows x 5 columns]
```

In [23]:
```
# Proceed to rename the column original_title to title
newdf = newdf.rename(columns={'original_title': 'title'})

# Display the DataFrame after renaming the column
newdf.head()
```

Out[23]:

	tconst	title	start_year	runtime_minutes	genres
0	tt0063540	Sunghursh	2013	175.0	Action,Crime,Drama
1	tt0066787	Ashad Ka Ek Din	2019	114.0	Biography,Drama
2	tt0069049	The Other Side of the Wind	2018	122.0	Drama
3	tt0069204	Sabse Bada Sukh	2018	NaN	Comedy,Drama
4	tt0100275	La Telenovela Errante	2017	80.0	Comedy,Drama,Fantasy

In [24]:
```
#Open csv file as a Pandas DataFrame to load the data

df1 = pd.read_csv('title_ratings.csv')
df1
```

Out[24]:

	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 × 3 columns

In [25]:
```
#Open csv file as a Pandas DataFrame to load the data

df2 = pd.read_csv('bom_movie_gross.csv')
df2
```

Out[25]:

	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 [26]:
```
# Combing title_basics and title_ratings

#To merge the dataframes, we use the following code:
merged_df = pd.merge(newdf, df1, on='tconst', how='outer')
merged_df.head()
```

Out[26]:

	tconst	title	start_year	runtime_minutes	genres	averagerating	numvotes
0	tt0063540	Sunghursh	2013	175.0	Action,Crime,Drama	7.0	77.0
1	tt0066787	Ashad Ka Ek Din	2019	114.0	Biography,Drama	7.2	43.0
2	tt0069049	The Other Side of the Wind	2018	122.0	Drama	6.9	4517.0
3	tt0069204	Sabse Bada Sukh	2018	NaN	Comedy,Drama	6.1	13.0
4	tt0100275	La Telenovela Errante	2017	80.0	Comedy,Drama,Fantasy	6.5	119.0

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 146144 entries, 0 to 146143
Data columns (total 7 columns):
 #   Column           Non-Null Count   Dtype  
---  ------           --------------   -----  
 0   tconst           146144 non-null  object 
 1   title            146122 non-null  object 
 2   start_year       146144 non-null  int64  
 3   runtime_minutes  114405 non-null  float64
 4   genres           140736 non-null  object 
 5   averagerating    73856 non-null   float64
 6   numvotes         73856 non-null   float64
dtypes: float64(3), int64(1), object(3)
memory usage: 7.8+ MB
```

In [28]:
```
# Check for missing values
merged_df.isna().sum()
```

Out[28]:
```
tconst                 0
title                 22
start_year             0
runtime_minutes    31739
genres              5408
averagerating      72288
numvotes           72288
dtype: int64
```

In [29]:
```
# Drop null values

merged_df.dropna(inplace=True)
merged_df.head()
```

Out[29]:

	tconst	title	start_year	runtime_minutes	genres	averagerating	numvotes
0	tt0063540	Sunghursh	2013	175.0	Action,Crime,Drama	7.0	77.0
1	tt0066787	Ashad Ka Ek Din	2019	114.0	Biography,Drama	7.2	43.0
2	tt0069049	The Other Side of the Wind	2018	122.0	Drama	6.9	4517.0
4	tt0100275	La Telenovela Errante	2017	80.0	Comedy,Drama,Fantasy	6.5	119.0
7	tt0137204	Joe Finds Grace	2017	83.0	Adventure,Animation,Comedy	8.1	263.0

In [30]:
```
merged_df.isna().sum()
```

Out[30]:
```
tconst             0
title              0
start_year         0
runtime_minutes    0
genres             0
averagerating      0
numvotes           0
dtype: int64
```

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

```
<class 'pandas.core.frame.DataFrame'>
Index: 65720 entries, 0 to 146134
Data columns (total 7 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   tconst           65720 non-null  object 
 1   title            65720 non-null  object 
 2   start_year       65720 non-null  int64  
 3   runtime_minutes  65720 non-null  float64
 4   genres           65720 non-null  object 
 5   averagerating    65720 non-null  float64
 6   numvotes         65720 non-null  float64
dtypes: float64(3), int64(1), object(3)
memory usage: 4.0+ MB
```

In [32]:
```
#To merge merged_df and df2 we use the following code
merged_df4 = pd.merge(merged_df, df2, on='title', how='outer')
merged_df4.head()
```

Out[32]:

	tconst	title	start_year	runtime_minutes	genres	averagerating	numvotes	studio	domestic_gross	foreign_gross
0	tt0063540	Sunghursh	2013.0	175.0	Action,Crime,Drama	7.0	77.0	NaN	NaN	NaN
1	tt0066787	Ashad Ka Ek Din	2019.0	114.0	Biography,Drama	7.2	43.0	NaN	NaN	NaN
2	tt0069049	The Other Side of the Wind	2018.0	122.0	Drama	6.9	4517.0	NaN	NaN	NaN
3	tt0100275	La Telenovela Errante	2017.0	80.0	Comedy,Drama,Fantasy	6.5	119.0	NaN	NaN	NaN
4	tt0137204	Joe Finds Grace	2017.0	83.0	Adventure,Animation,Comedy	8.1	263.0	NaN	NaN	NaN

In [33]:
```
merged_df4.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 66975 entries, 0 to 66974
Data columns (total 11 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   tconst           65720 non-null  object 
 1   title            66975 non-null  object 
 2   start_year       65720 non-null  float64
 3   runtime_minutes  65720 non-null  float64
 4   genres           65720 non-null  object 
 5   averagerating    65720 non-null  float64
 6   numvotes         65720 non-null  float64
 7   studio           3651 non-null   object 
 8   domestic_gross   3623 non-null   float64
 9   foreign_gross    2209 non-null   object 
 10  year             3656 non-null   float64
dtypes: float64(6), object(5)
memory usage: 5.6+ MB
```

In [34]:
```
# Check for missing values
merged_df4.isna().sum()
```

Out[34]:
```
tconst              1255
title                  0
start_year          1255
runtime_minutes     1255
genres              1255
averagerating       1255
numvotes            1255
studio             63324
domestic_gross     63352
foreign_gross      64766
year               63319
dtype: int64
```

In [35]:
```
# Drop null values

merged_df4.dropna(inplace=True)
merged_df4.head()
```

Out[35]:

	tconst	title	start_year	runtime_minutes	genres	averagerating	numvotes	studio	domestic_gross	foreign_gross
31	tt0337692	On the Road	2012.0	124.0	Adventure,Drama,Romance	6.1	37886.0	IFC	744000.0	8000000
32	tt4339118	On the Road	2014.0	89.0	Drama	6.0	6.0	IFC	744000.0	8000000
33	tt5647250	On the Road	2016.0	121.0	Drama	5.7	127.0	IFC	744000.0	8000000
38	tt0359950	The Secret Life of Walter Mitty	2013.0	114.0	Adventure,Comedy,Drama	7.3	275300.0	Fox	58200000.0	129900000
42	tt0365907	A Walk Among the Tombstones	2014.0	114.0	Action,Crime,Drama	6.5	105116.0	Uni.	26300000.0	26900000

In [36]:
```
merged_df4.isna().sum()
```

Out[36]:
```
tconst             0
title              0
start_year         0
runtime_minutes    0
genres             0
averagerating      0
numvotes           0
studio             0
domestic_gross     0
foreign_gross      0
year               0
dtype: int64
```

In [37]:
```
merged_df4.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 1518 entries, 31 to 64832
Data columns (total 11 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   tconst           1518 non-null   object 
 1   title            1518 non-null   object 
 2   start_year       1518 non-null   float64
 3   runtime_minutes  1518 non-null   float64
 4   genres           1518 non-null   object 
 5   averagerating    1518 non-null   float64
 6   numvotes         1518 non-null   float64
 7   studio           1518 non-null   object 
 8   domestic_gross   1518 non-null   float64
 9   foreign_gross    1518 non-null   object 
 10  year             1518 non-null   float64
dtypes: float64(6), object(5)
memory usage: 142.3+ KB
```

Data cleaning

In this section, we clean the data to generate quality data.

###We look out for unnecessary data

In [38]:
```
# Drop unneeded column

merged_df5 = merged_df4.drop('tconst', axis=1)
merged_df5.head()
```

Out[38]:

	title	start_year	runtime_minutes	genres	averagerating	numvotes	studio	domestic_gross	foreign_gross	year
31	On the Road	2012.0	124.0	Adventure,Drama,Romance	6.1	37886.0	IFC	744000.0	8000000	2012.0
32	On the Road	2014.0	89.0	Drama	6.0	6.0	IFC	744000.0	8000000	2012.0
33	On the Road	2016.0	121.0	Drama	5.7	127.0	IFC	744000.0	8000000	2012.0
38	The Secret Life of Walter Mitty	2013.0	114.0	Adventure,Comedy,Drama	7.3	275300.0	Fox	58200000.0	129900000	2013.0
42	A Walk Among the Tombstones	2014.0	114.0	Action,Crime,Drama	6.5	105116.0	Uni.	26300000.0	26900000	2014.0

##We check for missing values ###Missing values would make our data inaccurate which may lead to bias in decision making.

In [39]:
```
# Check for missing values

merged_df5.isna().sum()

#There are no missing values
```

Out[39]:
```
title              0
start_year         0
runtime_minutes    0
genres             0
averagerating      0
numvotes           0
studio             0
domestic_gross     0
foreign_gross      0
year               0
dtype: int64
```

We check for duplicates

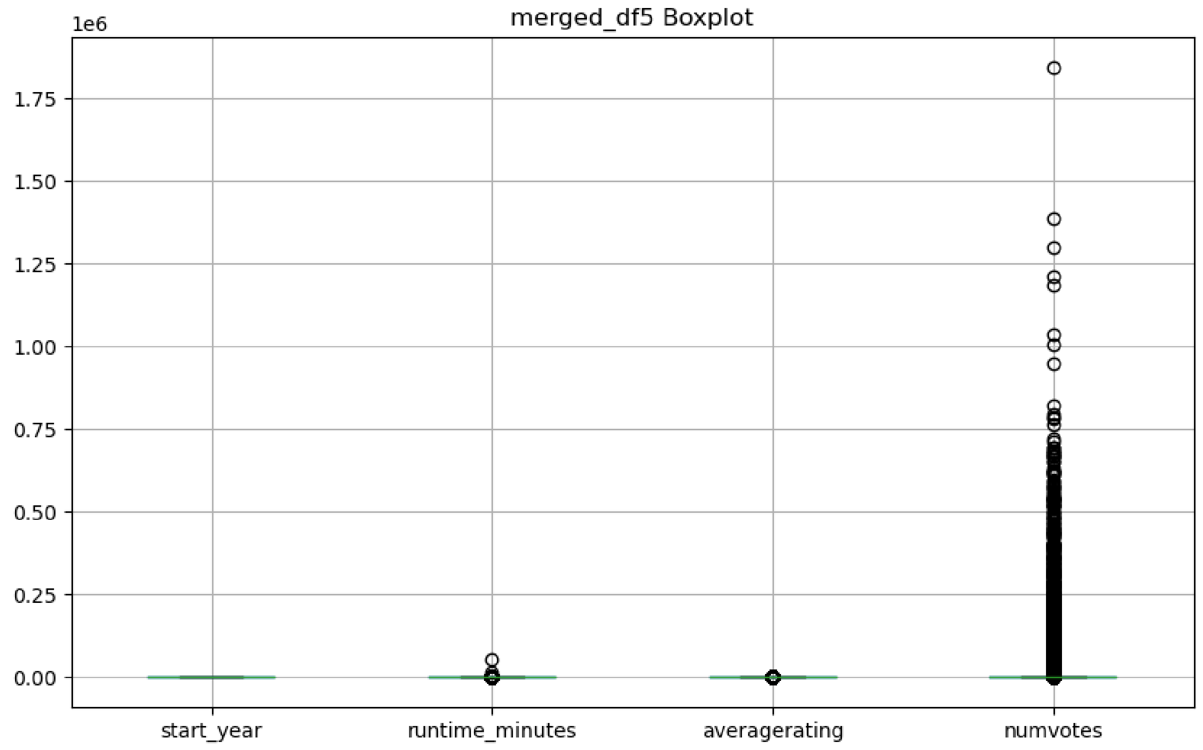
In [40]:
```
#To check for duplicates we use the code
merged_df5.duplicated().sum()

#No duplicates exist, we proceed
```

Out[40]:
```
0
```

Handling outliers

```
Outliers are extreme values that greatly stand out  an overall pattern of values. They may manifest as a typo
during data entry.
They are mainly dependent on context, proporton domain
```

In [51]:
```
# Plot a boxplot
fig, ax = plt.subplots(figsize=(10, 6))  # Adjust the width and height as needed
merged_df5.boxplot(ax=ax)
plt.title('merged_df5 Boxplot')
plt.show()
```



# Exploratory Data Analysis

This will help us get a good perspective on the company's data

In [53]:
```
# numvotes analysis
merged_df5.numvotes.value_counts()
```

Out[53]:
```
numvotes
6.0         2231
5.0         2076
7.0         1923
8.0         1702
9.0         1514
            ...
8442.0         1
10512.0        1
123577.0       1
130084.0       1
19632.0        1
Name: count, Length: 7349, dtype: int64
```

In [49]:
```
# average rating analysis
merged_df5.averagerating.value_counts()
```

Out[49]:
```
averagerating
7.0     0.031132
6.5     0.030949
6.6     0.030919
7.2     0.030615
6.8     0.030386
          ...
9.6     0.000228
10.0    0.000198
9.7     0.000167
9.8     0.000167
9.9     0.000076
Name: proportion, Length: 91, dtype: float64
```

In [54]:
```
# runtime_minutes analysis
merged_df5.runtime_minutes.value_counts()
```

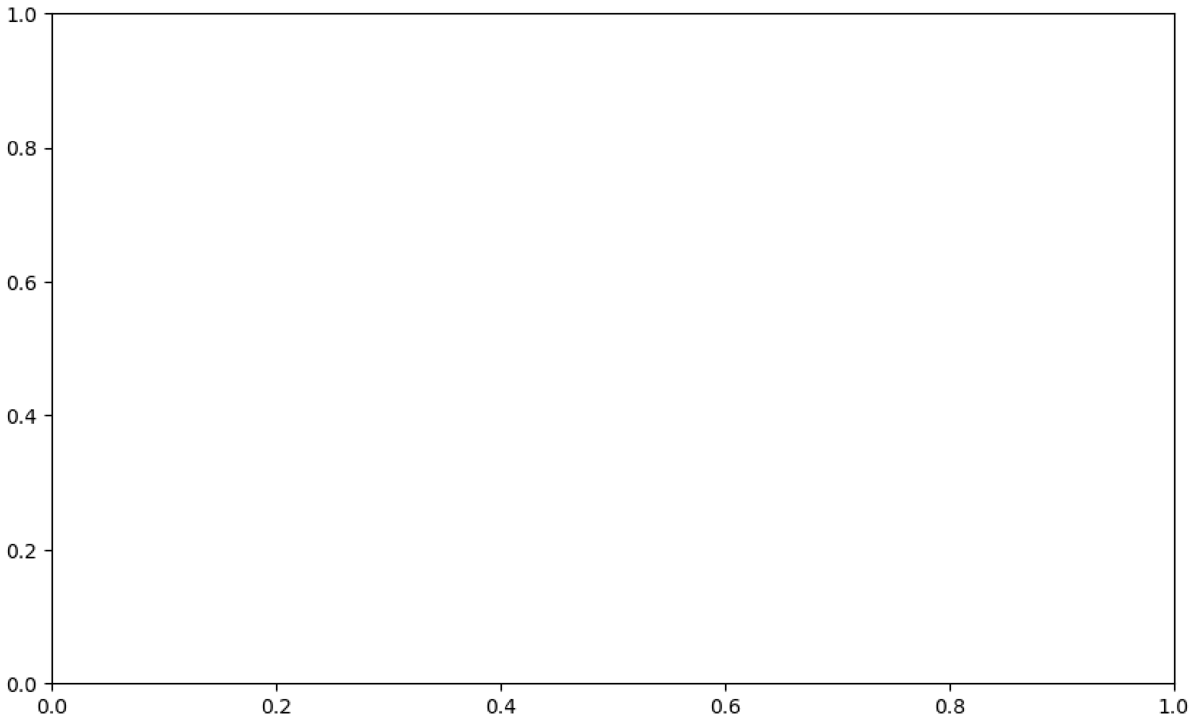
Out[54]:
```
runtime_minutes
90.0     4718
80.0     2142
85.0     2048
100.0    1954
95.0     1919
         ...
202.0       1
319.0       1
350.0       1
476.0       1
261.0       1
Name: count, Length: 289, dtype: int64
```

In [56]:
```
# startyear analysis
merged_df5.start_year.value_counts()
```

Out[56]:
```
start_year
2016    7785
2017    7718
2015    7650
2014    7528
2013    7216
2012    6866
2018    6573
2011    6542
2010    6038
2019    1804
Name: count, dtype: int64
```

In [58]:
```
# Create subplots with specific dimensions
fig, ax = plt.subplots(figsize=(10, 6))

# Plot a bar chart using the same axes (ax)
merged_df5.sum().plot(kind='bar', ax=ax, color=['blue', 'green'])

# Customize the plot
plt.title('Bar Chart for merged_df5')
plt.xlabel('Columns')
plt.ylabel('Sum')
plt.show()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[58], line 5
      2 fig, ax = plt.subplots(figsize=(10, 6))
      4 # Plot a bar chart using the same axes (ax)
----> 5 merged_df5.sum().plot(kind='bar', ax=ax, color=['blue', 'green'])
      7 # Customize the plot
      8 plt.title('Bar Chart for merged_df5')

File ~\anaconda3\Lib\site-packages\pandas\plotting\_core.py:975, in PlotAccessor.__call__(self, *args, **kwargs)
    972             label_name = label_kw or data.columns
    973             data.columns = label_name
--> 975 return plot_backend.plot(data, kind=kind, **kwargs)

File ~\anaconda3\Lib\site-packages\pandas\plotting\_matplotlib\__init__.py:71, in plot(data, kind, **kwargs)
     69         kwargs["ax"] = getattr(ax, "left_ax", ax)
     70 plot_obj = PLOT_CLASSES[kind](data, **kwargs)
---> 71 plot_obj.generate()
     72 plot_obj.draw()
     73 return plot_obj.result

File ~\anaconda3\Lib\site-packages\pandas\plotting\_matplotlib\core.py:446, in MPLPlot.generate(self)
    444 def generate(self) -> None:
    445     self._args_adjust()
--> 446     self._compute_plot_data()
    447     self._setup_subplots()
    448     self._make_plot()

File ~\anaconda3\Lib\site-packages\pandas\plotting\_matplotlib\core.py:632, in MPLPlot._compute_plot_data(self)
    630 # no non-numeric frames or series allowed
    631 if is_empty:
--> 632     raise TypeError("no numeric data to plot")
    634 self.data = numeric_data.apply(self._convert_to_ndarray)

TypeError: no numeric data to plot
```



```
In [ ]:
```

```
In [ ]:
```