In [1]:
```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt
```

In [2]:
```
df = pd.read_csv("Movie.csv")
```

In [3]:
```
df
```

Out[3]:

	Movie_Revenue	production costs	promotional costs	total book sales
0	85.099998	8.5	5.100000	4.700000
1	106.300003	12.9	5.800000	8.800000
2	50.200001	5.2	2.100000	15.100000
3	130.600006	10.7	8.399999	12.200000
4	54.799999	3.1	2.900000	10.600000
5	30.299999	3.5	1.200000	3.500000
6	79.400002	9.2	3.700000	9.700000
7	91.000000	9.0	7.600000	5.900000
8	135.399994	15.1	7.700000	20.799999
9	89.300003	10.2	4.500000	7.900000

In [4]:
```
df.describe()
```

Out[4]:

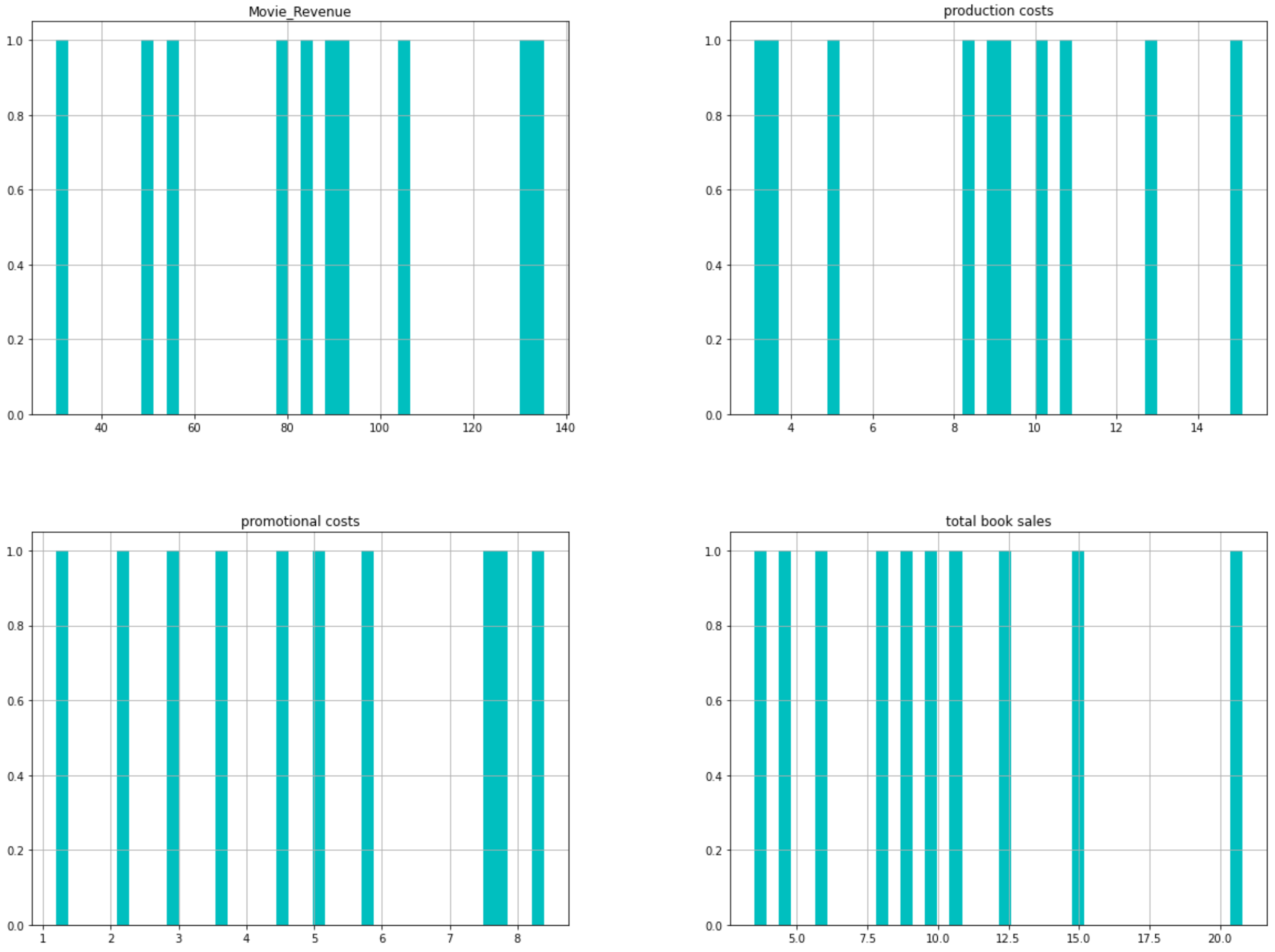
	Movie_Revenue	production costs	promotional costs	total book sales
count	10.000000	10.000000	10.000000	10.000000
mean	85.240001	8.740000	4.900000	9.920000
std	33.786362	3.885357	2.480143	5.173393
min	30.299999	3.100000	1.200000	3.500000
25%	60.950000	6.025000	3.100000	6.400000
50%	87.200001	9.100000	4.800000	9.250000
75%	102.475002	10.575000	7.150000	11.800000
max	135.399994	15.100000	8.399999	20.799999

In [5]:
```
df.isnull().sum()
```

Out[5]:
```
Movie_Revenue        0
production costs     0
promotional costs    0
total book sales     0
dtype: int64
```

In [6]:

```
df.hist(bins=40, figsize=(20,15),color ='c')
plt.show()
```



In [7]:

```
df.dtypes
```

Out[7]:

```
Movie_Revenue         float64
production costs      float64
promotional costs     float64
total book sales      float64
dtype: object
```

In [8]:

```
df
```

Out[8]:

	Movie_Revenue	production costs	promotional costs	total book sales
0	85.099998	8.5	5.100000	4.700000
1	106.300003	12.9	5.800000	8.800000
2	50.200001	5.2	2.100000	15.100000
3	130.600006	10.7	8.399999	12.200000
4	54.799999	3.1	2.900000	10.600000
5	30.299999	3.5	1.200000	3.500000
6	79.400002	9.2	3.700000	9.700000
7	91.000000	9.0	7.600000	5.900000
8	135.399994	15.1	7.700000	20.799999
9	89.300003	10.2	4.500000	7.900000

In [9]:

```
X = df.drop(['Movie_Revenue'],axis=1)
Y = df['Movie_Revenue']
```

In [10]:

```
X
```

Out[10]:

	production costs	promotional costs	total book sales
0	8.5	5.100000	4.700000
1	12.9	5.800000	8.800000
2	5.2	2.100000	15.100000
3	10.7	8.399999	12.200000
4	3.1	2.900000	10.600000
5	3.5	1.200000	3.500000
6	9.2	3.700000	9.700000
7	9.0	7.600000	5.900000
8	15.1	7.700000	20.799999
9	10.2	4.500000	7.900000

In [11]:

```
Y
```

Out[11]:

```
0     85.099998
1    106.300003
2     50.200001
3    130.600006
4     54.799999
5     30.299999
6     79.400002
7     91.000000
8    135.399994
9     89.300003
Name: Movie_Revenue, dtype: float64
```

In [12]:

```
from sklearn.model_selection import train_test_split

X_train, x_test, Y_train, y_test = train_test_split(X, Y, test_size=0.2, random_state=0)
```

In [13]:

```
X_train
```

Out[13]:

	production costs	promotional costs	total book sales
4	3.1	2.900000	10.6
9	10.2	4.500000	7.9
1	12.9	5.800000	8.8
6	9.2	3.700000	9.7
7	9.0	7.600000	5.9
3	10.7	8.399999	12.2
0	8.5	5.100000	4.7
5	3.5	1.200000	3.5

In [14]:

```
x_test
```

Out[14]:

	production costs	promotional costs	total book sales
2	5.2	2.1	15.100000
8	15.1	7.7	20.799999

In [15]:

```
from sklearn.linear_model import LinearRegression
model = LinearRegression()
```

In [16]:

```
model.fit(df[['production costs','promotional costs','total book sales']],df.Movie_Revenue)
```

Out[16]:

```
LinearRegression()
```

In [17]:

```
model.score(df[['production costs','promotional costs','total book sales']],df.Movie_Revenue)
```

Out[17]:

```
0.9667887860584002
```

In [18]:

```
model.predict([['10.7','8.399999','12.2']])
```

```
C:\Users\DELL\anaconda3\lib\site-packages\sklearn\utils\validation.py:63: FutureWarning: Arrays of bytes/strings is being conve
rted to decimal numbers if dtype='numeric'. This behavior is deprecated in 0.24 and will be removed in 1.1 (renaming of 0.26).
Please convert your data to numeric values explicitly instead.
  return f(*args, **kwargs)
```

Out[18]:

```
array([120.97932478])
```