Investigate_a_Dataset

January 8, 2018

> **Tip**: Welcome to the Investigate a Dataset project! You will find tips in quoted sections like this to help organize your approach to your investigation. Before submitting your project, it will be a good idea to go back through your report and remove these sections to make the presentation of your work as tidy as possible. First things first, you might want to double-click this Markdown cell and change the title so that it reflects your dataset and investigation.

1 Project: Investigate a Dataset (TMDb Movie Data)

1.1 Table of Contents

Introduction

> **Tip**: In this section of the report, provide a brief introduction to the dataset you've selected for analysis. At the end of this section, describe the questions that you plan on exploring over the course of the report. Try to build your report around the analysis of at least one dependent variable and three independent variables. If you're not sure what questions to ask, then make sure you familiarize yourself with the dataset, its variables and the dataset context for ideas of what to explore.
>
> If you haven't yet selected and downloaded your data, make sure you do that first before coming back here. In order to work with the data in this workspace, you also need to upload it to the workspace. To do so, click on the jupyter icon in the upper left to be taken back to the workspace directory. There should be an 'Upload' button in the upper right that will let you add your data file(s) to the workspace. You can then click on the .ipynb file name to come back here.

This is a dataset of movies. It contains various details about movies for our analysis. I want to analyse the given dataset to answer questions about trends and their associations to be explored here are: 1. Popularity over the years 2. Revenue over the years 3. Runtime over the years

```
In [1]: # Use this cell to set up import statements for all of the packages that you
        #   plan to use.
```

```
# Remember to include a 'magic word' so that your visualizations are plotted
#   inline with the notebook. See this page for more:
#   http://ipython.readthedocs.io/en/stable/interactive/magics.html

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

Data Wrangling

Tip: In this section of the report, you will load in the data, check for cleanliness, and then trim and clean your dataset for analysis. Make sure that you document your steps carefully and justify your cleaning decisions.

1.1.1 General Properties

In [2]:
```
# Load your data and print out a few lines. Perform operations to inspect data
#   types and look for instances of missing or possibly errant data.
movie_data = pd.read_csv('tmdb_movies_data.csv')
# Shows full summary of tmdb movie dataset and columns with missing values.
movie_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10866 entries, 0 to 10865
Data columns (total 21 columns):
id                      10866 non-null int64
imdb_id                 10856 non-null object
popularity              10866 non-null float64
budget                  10866 non-null int64
revenue                 10866 non-null int64
original_title          10866 non-null object
cast                    10790 non-null object
homepage                2936 non-null object
director                10822 non-null object
tagline                 8042 non-null object
keywords                9373 non-null object
overview                10862 non-null object
runtime                 10866 non-null int64
genres                  10843 non-null object
production_companies    9836 non-null object
release_date            10866 non-null object
vote_count              10866 non-null int64
vote_average            10866 non-null float64
release_year            10866 non-null int64
budget_adj              10866 non-null float64
revenue_adj             10866 non-null float64
dtypes: float64(4), int64(6), object(11)
```

```
memory usage: 1.7+ MB
```

In [3]:
```
# Shows first few rows of dataset
movie_data.head()
```

Out[3]:

	id	imdb_id	popularity	budget	revenue	original_title	cast	homepage	director	tagline	...	overview	runtime
0	135397	tt0369610	32.985763	150000000	1513528810	Jurassic World	Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...	http://www.jurassicworld.com/	Colin Trevorrow	The park is open.	...	Twenty-two years after the events of Jurassic ...	124
1	76341	tt1392190	28.419936	150000000	378436354	Mad Max: Fury Road	Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic...	http://www.madmaxmovie.com/	George Miller	What a Lovely Day.	...	An apocalyptic story set in the furthest reach...	120
2	262500	tt2908446	13.112507	110000000	295238201	Insurgent	Shailene Woodley|Theo James|Kate Winslet|Ansel...	http://www.thedivergentseries.movie/#insurgent	Robert Schwentke	One Choice Can Destroy You	...	Beatrice Prior must confront her inner demons ...	119
3	140607	tt2488496	11.173104	200000000	2068178225	Star Wars: The Force Awakens	Harrison Ford|Mark Hamill|Carrie Fisher|Adam D...	http://www.starwars.com/films/star-wars-episod...	J.J. Abrams	Every generation has a story.	...	Thirty years after defeating the Galactic Empi...	136
4	168259	tt2820852	9.335014	190000000	1506249360	Furious 7	Vin Diesel|Paul Walker|Jason Statham|Michelle ...	http://www.furious7.com/	James Wan	Vengeance Hits Home	...	Deckard Shaw seeks revenge against Dominic Tor...	137

```
                                        genres  \
0  Action|Adventure|Science Fiction|Thriller
1  Action|Adventure|Science Fiction|Thriller
2         Adventure|Science Fiction|Thriller
3   Action|Adventure|Science Fiction|Fantasy
4                       Action|Crime|Thriller

                                  production_companies release_date vote_count  \
0  Universal Studios|Amblin Entertainment|Legenda...       6/9/15       5562
1  Village Roadshow Pictures|Kennedy Miller Produ...      5/13/15       6185
2  Summit Entertainment|Mandeville Films|Red Wago...      3/18/15       2480
3          Lucasfilm|Truenorth Productions|Bad Robot     12/15/15       5292
4  Universal Pictures|Original Film|Media Rights ...       4/1/15       2947

   vote_average  release_year    budget_adj   revenue_adj
0           6.5          2015  1.379999e+08  1.392446e+09
1           7.1          2015  1.379999e+08  3.481613e+08
2           6.3          2015  1.012000e+08  2.716190e+08
3           7.5          2015  1.839999e+08  1.902723e+09
4           7.3          2015  1.747999e+08  1.385749e+09

[5 rows x 21 columns]
```

In [4]:
```
# Shows last few rows of dataset
movie_data.tail()
```

Out[4]:
```
          id    imdb_id  popularity  budget  revenue  \
10861     21  tt0060371    0.080598       0        0
10862  20379  tt0060472    0.065543       0        0
10863  39768  tt0060161    0.065141       0        0
10864  21449  tt0061177    0.064317       0        0
10865  22293  tt0060666    0.035919   19000        0

                original_title  \
10861        The Endless Summer
10862                Grand Prix
10863       Beregis Avtomobilya
10864    What's Up, Tiger Lily?
10865  Manos: The Hands of Fate

                                                cast homepage  \
10861  Michael Hynson|Robert August|Lord 'Tally Ho' B...      NaN
10862  James Garner|Eva Marie Saint|Yves Montand|Tosh...      NaN
10863  Innokentiy Smoktunovskiy|Oleg Efremov|Georgi Z...      NaN
10864  Tatsuya Mihashi|Akiko Wakabayashi|Mie Hama|Joh...      NaN
10865  Harold P. Warren|Tom Neyman|John Reynolds|Dian...      NaN

                director                                        tagline  \
```

```
10861         Bruce Brown                                                NaN
10862  John Frankenheimer  Cinerama sweeps YOU into a drama of speed and ...
10863      Eldar Ryazanov                                                NaN
10864         Woody Allen                          WOODY ALLEN STRIKES BACK!
10865    Harold P. Warren       It's Shocking! It's Beyond Your Imagination!

          ...                                                  overview runtime  \
10861     ...       The Endless Summer, by Bruce Brown, is one of ...      95
10862     ...       Grand Prix driver Pete Aron is fired by his te...     176
10863     ...       An insurance agent who moonlights as a carthie...      94
10864     ...       In comic Woody Allen's film debut, he took the...      80
10865     ...       A family gets lost on the road and stumbles up...      74

                      genres  \
10861             Documentary
10862  Action|Adventure|Drama
10863          Mystery|Comedy
10864           Action|Comedy
10865                  Horror

                                    production_companies release_date  \
10861                                  Bruce Brown Films      6/15/66
10862  Cherokee Productions|Joel Productions|Douglas ...     12/21/66
10863                                          Mosfilm       1/1/66
10864                          Benedict Pictures Corp.      11/2/66
10865                                      Norm-Iris     11/15/66

      vote_count  vote_average  release_year     budget_adj  revenue_adj
10861         11           7.4          1966       0.000000          0.0
10862         20           5.7          1966       0.000000          0.0
10863         11           6.5          1966       0.000000          0.0
10864         22           5.4          1966       0.000000          0.0
10865         15           1.5          1966  127642.279154          0.0

[5 rows x 21 columns]
```

In [5]:
```
# Dimensions of dataset
movie_data.shape
```

Out[5]:
```
(10866, 21)
```

In [6]:
```
# Datatypes of columns
movie_data.dtypes
```

Out[6]:
```
id                      int64
imdb_id                object
popularity            float64
budget                  int64
revenue                 int64
```

```
original_title          object
cast                    object
homepage                object
director                object
tagline                 object
keywords                object
overview                object
runtime                  int64
genres                  object
production_companies    object
release_date            object
vote_count               int64
vote_average           float64
release_year             int64
budget_adj             float64
revenue_adj            float64
dtype: object
```

In [7]:
```
# Unique values of columns
movie_data.nunique()
```

Out[7]:
```
id                     10865
imdb_id                10855
popularity             10814
budget                   557
revenue                 4702
original_title         10571
cast                   10719
homepage                2896
director                5067
tagline                 7997
keywords                8804
overview               10847
runtime                  247
genres                  2039
production_companies    7445
release_date            5909
vote_count              1289
vote_average              72
release_year              56
budget_adj              2614
revenue_adj             4840
dtype: int64
```

In [8]:
```
# Count of missing values in dataset
movie_data.isnull().sum()
```

Out[8]:
```
id                       0
imdb_id                 10
```

```
popularity                0
budget                    0
revenue                   0
original_title            0
cast                     76
homepage               7930
director                 44
tagline                2824
keywords               1493
overview                  4
runtime                   0
genres                   23
production_companies   1030
release_date              0
vote_count                0
vote_average              0
release_year              0
budget_adj                0
revenue_adj               0
dtype: int64
```

In [9]:
```
# Duplicate rows in dataset
sum(movie_data.duplicated(keep=False))
#movie_data[movie_data.duplicated(['id'], keep=False)]
```

Out[9]: 2

In [10]:
```
# Describes budget column in statistics manner.
movie_data['budget'].describe()
```

Out[10]:
```
count    1.086600e+04
mean     1.462570e+07
std      3.091321e+07
min      0.000000e+00
25%      0.000000e+00
50%      0.000000e+00
75%      1.500000e+07
max      4.250000e+08
Name: budget, dtype: float64
```

In [11]:
```
# Describes revenue column in statistics manner.
movie_data['revenue'].describe()
```

Out[11]:
```
count    1.086600e+04
mean     3.982332e+07
std      1.170035e+08
min      0.000000e+00
25%      0.000000e+00
50%      0.000000e+00
```

```
75%       2.400000e+07
max       2.781506e+09
Name: revenue, dtype: float64
```

Tip: You should *not* perform too many operations in each cell. Create cells freely to explore your data. One option that you can take with this project is to do a lot of explorations in an initial notebook. These don't have to be organized, but make sure you use enough comments to understand the purpose of each code cell. Then, after you're done with your analysis, create a duplicate notebook where you will trim the excess and organize your steps so that you have a flowing, cohesive report.

Tip: Make sure that you keep your reader informed on the steps that you are taking in your investigation. Follow every code cell, or every set of related code cells, with a markdown cell to describe to the reader what was found in the preceding cell(s). Try to make it so that the reader can then understand what they will be seeing in the following cell(s).

1.1.2 Data Cleaning (Replace this with more specific notes!)

```
In [12]: # After discussing the structure of the data and any problems that need to be
         #   cleaned, perform those cleaning steps in the second part of this section.

         # release_date column should have their datatype to be datetime instead of string.
         movie_data['release_date'] = pd.to_datetime(movie_data['release_date'])
```

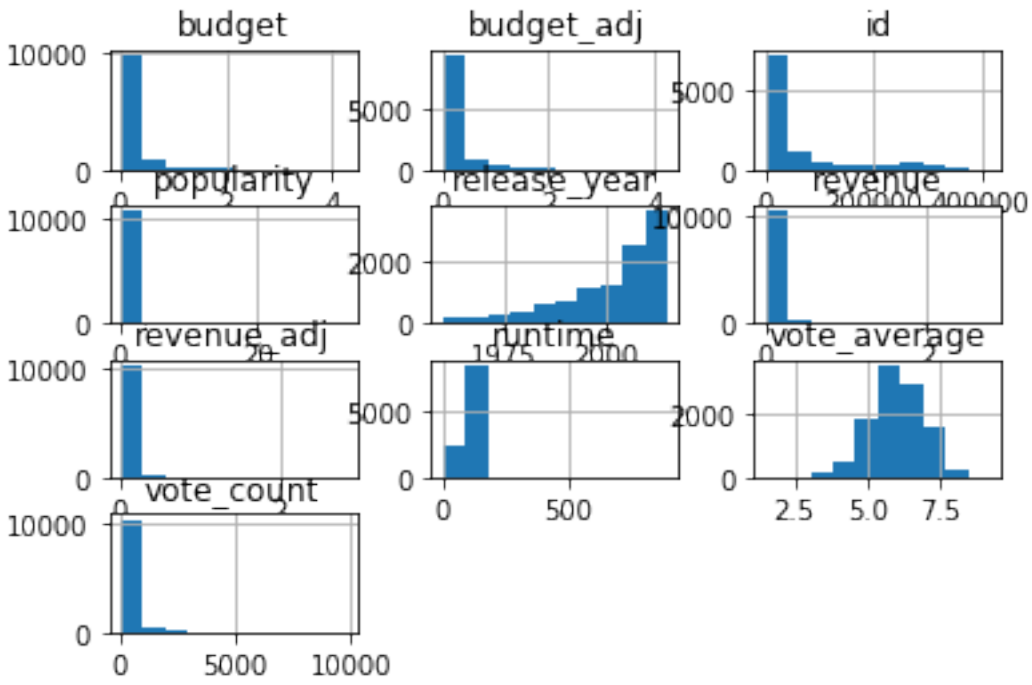
```
In [13]: # Filled all missing values with 0
         movie_data.fillna(0,inplace=True)
```

```
In [14]: # Drop the duplicates in dataset
         movie_data.drop_duplicates(inplace=True)
```

Exploratory Data Analysis

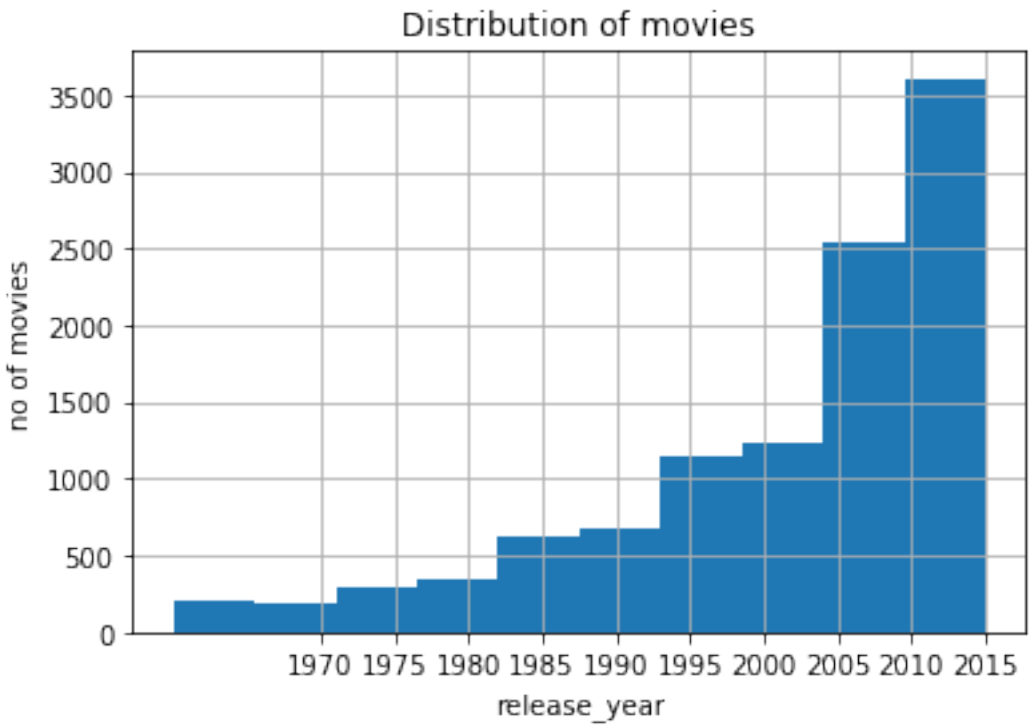
Tip: Now that you've trimmed and cleaned your data, you're ready to move on to exploration. Compute statistics and create visualizations with the goal of addressing the research questions that you posed in the Introduction section. It is recommended that you be systematic with your approach. Look at one variable at a time, and then follow it up by looking at relationships between variables.

```
In [15]: # Histograms of columns in the dataset
         movie_data.hist();
```



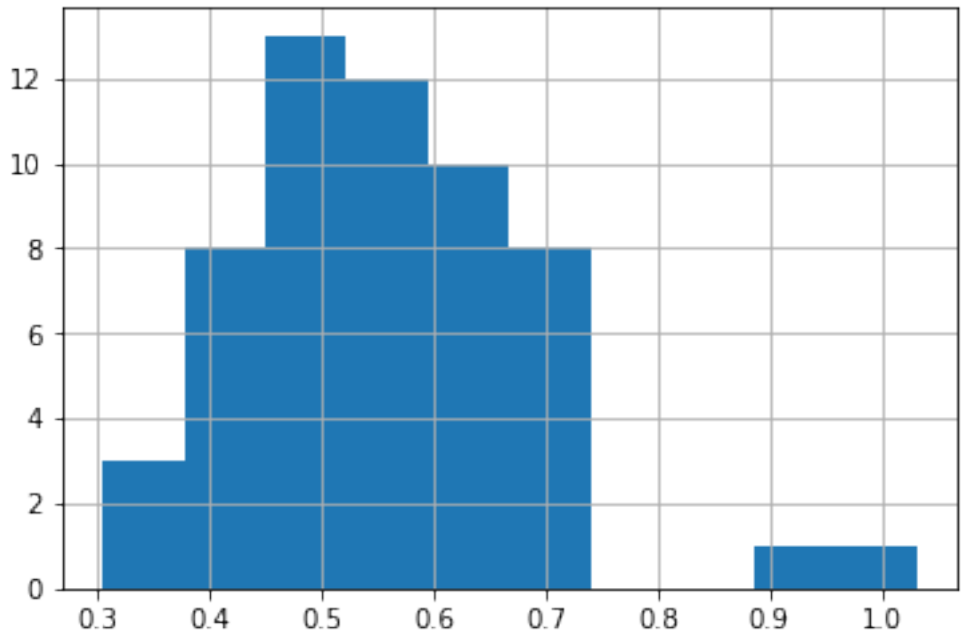
In [16]:
```
# Use this, and more code cells, to explore your data. Don't forget to add
#   Markdown cells to document your observations and findings.

# Plot distribution of movies
plt.hist(movie_data['release_year'])
plt.title('Distribution of movies')
plt.xlabel('release_year')
plt.ylabel('no of movies')
plt.xticks(range(1970,2020, 5))
plt.grid(True)
plt.show()
```

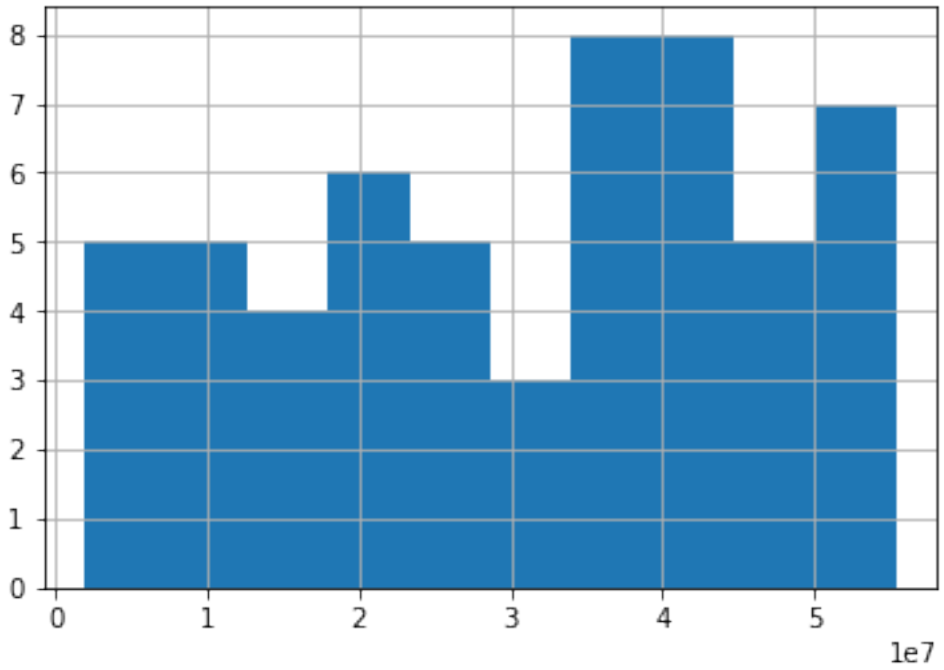


From above histogram, we can say that the shape of histogram is left skewed. Because the left tail is much longer than right tail. The peak period of movie release is between 2009 to 2015.

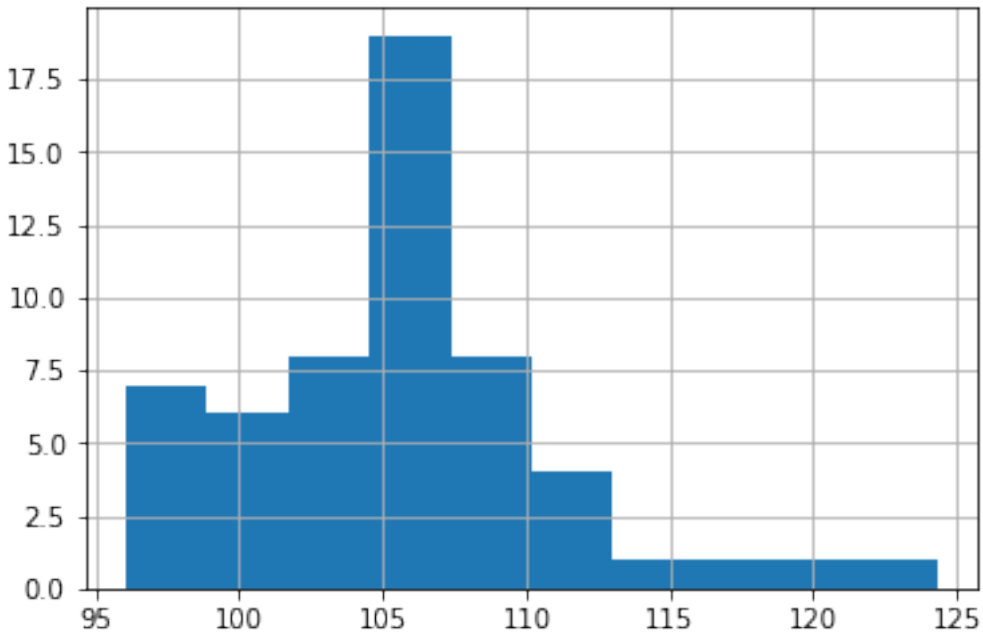
```
In [28]: movie_data.groupby('release_year').mean()['popularity'].hist();
```



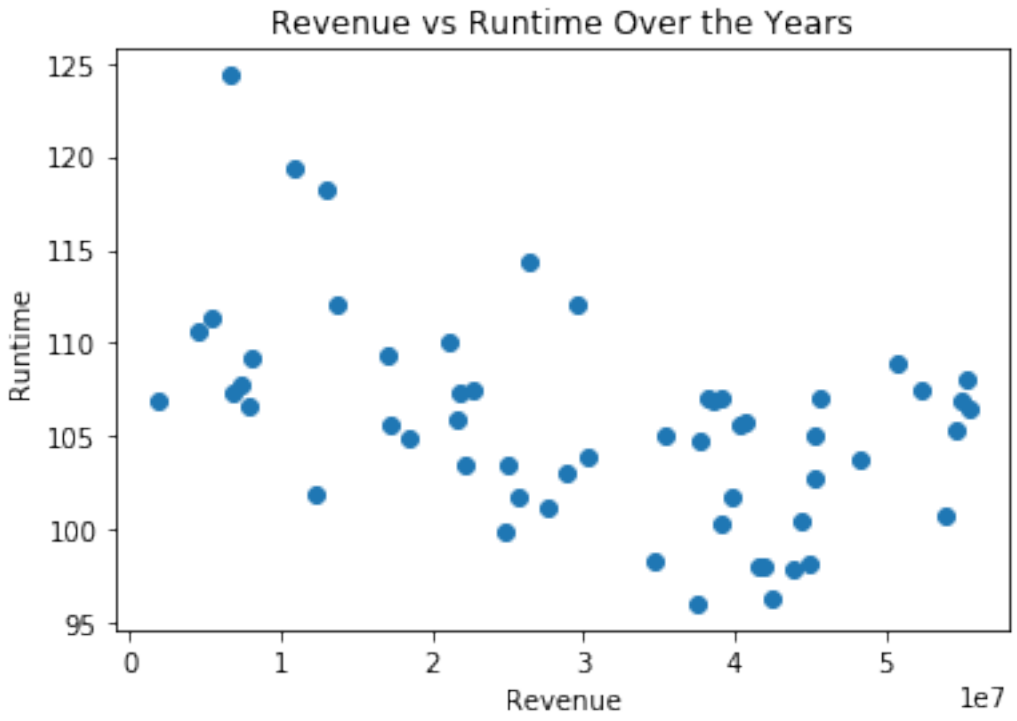
In [24]: movie_data.groupby('release_year').mean()['revenue'].hist();



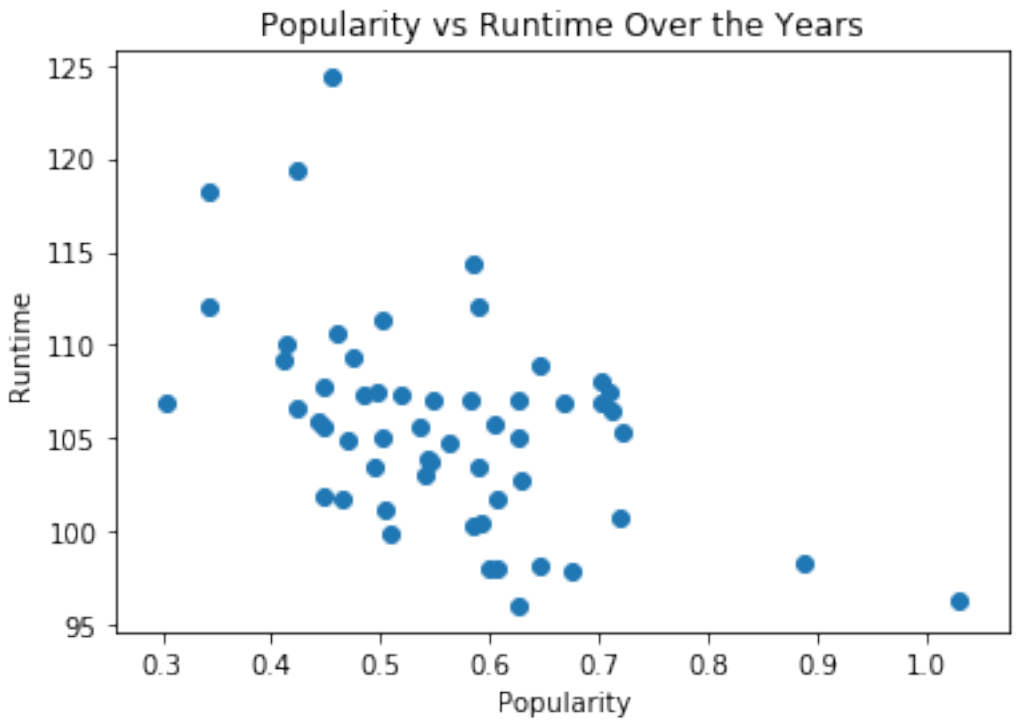
In [25]: movie_data.groupby('release_year').mean()['runtime'].hist();



In [26]:
```
movie_data_yr_mean = movie_data.groupby('release_year').mean()
plt.scatter(x=movie_data_yr_mean['revenue'], y=movie_data_yr_mean['runtime'])
plt.xlabel('Revenue')
plt.ylabel('Runtime')
plt.title('Revenue vs Runtime Over the Years');
```



In [27]:
```
plt.scatter(x=movie_data_yr_mean['popularity'], y=movie_data_yr_mean['runtime'])
plt.xlabel('Popularity')
plt.ylabel('Runtime')
plt.title('Popularity vs Runtime Over the Years');
```



Conclusions

Tip: Finally, summarize your findings and the results that have been performed. Make sure that you are clear with regards to the limitations of your exploration. If you haven't done any statistical tests, do not imply any statistical conclusions. And make sure you avoid implying causation from correlation!

Tip: Once you are satisfied with your work, you should save a copy of the report in HTML or PDF form. Before exporting your report, check over it to make sure that the flow of the report is complete. You should probably remove all of the "Tip" quotes like this one so that the presentation is as tidy as possible. It's also a good idea to look over the project rubric, found on the project submission page at the end of the lesson.

To export the report to the workspace, you should run the code cell below. If it worked correctly, you should get a return code of 0, and you should see the generated .html file in the workspace directory (click on the jupyter icon in the upper left). Alternatively, you can download the html report via the **File** > **Download as** submenu and then manually upload it to the workspace directory. Once you've done this, you can submit your project by clicking on the "Submit Project" button in the lower right. Congratulations!

From above all trends, the features of the movie are working idependently. Through the years, the movie is dependent on budget, revenue and ratings from which we can decide which movie either is blockbuster or flop.

```
In [19]: from subprocess import call
         call(['python', '-m', 'nbconvert', 'Investigate_a_Dataset.ipynb'])
```

```
Out[19]: 0
```