Analysis of Home Mortgage Disclosure Act (HMDA) Data

Data Source: https://www.consumerfinance.gov/data-research/hmda/historic-data/ (https://www.consumerfinance.gov/data-research/hmda/historic-data/)

Get the data out of S3 using an Athena query

Data is extracted then stored in a CSV. This allowed quick retrieval as I iterated through the analysis.

In [3]:

```
# Setup python to connect to S3
import sys
!{sys.executable} -m pip install PyAthena
from pyathena import connect
```

```
# Get data from S3
athena_storage_location = 's3://aws-athena-query-results-527117955781-us-east-1/
'
aws_region = 'us-east-1'
sql = "SELECT as_of_year, loan_amount_000s, applicant_income_000s, rate_spread F
ROM hmda.hmda_csv;"
conn = connect(s3_staging_dir=athena_storage_location,
               region_name=aws_region)
df = pd.read_sql(sql,conn)
df.to_csv ('./data/hmda_analysis_data.csv', index = None, header=True)
```

```
Collecting PyAthena
  Downloading https://files.pythonhosted.org/packages/34/11/20fbc9d9
c0825e3c1c7ef9ea7e6489b20f582a24ca05ab023003948034cf/PyAthena-1.7.1-
py2.py3-none-any.whl
Collecting future (from PyAthena)
  Downloading https://files.pythonhosted.org/packages/90/52/e20466b8
5000a181e1e144fd8305caf2cf475e2f9674e797b222f8105f5f/future-0.17.1.t
ar.gz (829kB)
    100% |████████████████████████████████| 829kB 23.0MB/s ta 0:00:0
1
Requirement already satisfied: boto3>=1.4.4 in /home/ec2-user/anacon
da3/envs/python3/lib/python3.6/site-packages (from PyAthena) (1.9.18
8)
Requirement already satisfied: botocore>=1.5.52 in /home/ec2-user/an
aconda3/envs/python3/lib/python3.6/site-packages (from PyAthena) (1.
12.188)
Collecting tenacity>=4.1.0 (from PyAthena)
  Downloading https://files.pythonhosted.org/packages/6a/93/dfcf5b1b
46ab29196274b78dcba69fab5e54b6dc303a7eed90a79194d277/tenacity-5.0.4-
py2.py3-none-any.whl
Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /home/ec2-u
ser/anaconda3/envs/python3/lib/python3.6/site-packages (from boto3>=
1.4.4->PyAthena) (0.9.4)
Requirement already satisfied: s3transfer<0.3.0,>=0.2.0 in /home/ec2
-user/anaconda3/envs/python3/lib/python3.6/site-packages (from boto3
>=1.4.4->PyAthena) (0.2.0)
Requirement already satisfied: urllib3<1.26,>=1.20; python_version >
= "3.4" in /home/ec2-user/anaconda3/envs/python3/lib/python3.6/site-
packages (from botocore>=1.5.52->PyAthena) (1.23)
Requirement already satisfied: docutils>=0.10 in /home/ec2-user/anac
onda3/envs/python3/lib/python3.6/site-packages (from botocore>=1.5.5
2->PyAthena) (0.14)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1; python_v
ersion >= "2.7" in /home/ec2-user/anaconda3/envs/python3/lib/python3
.6/site-packages (from botocore>=1.5.52->PyAthena) (2.7.3)
Requirement already satisfied: six>=1.9.0 in /home/ec2-user/anaconda
3/envs/python3/lib/python3.6/site-packages (from tenacity>=4.1.0->Py
Athena) (1.11.0)
Building wheels for collected packages: future
  Running setup.py bdist_wheel for future ... done
  Stored in directory: /home/ec2-user/.cache/pip/wheels/0c/61/d2/d6b
7317325828fbb39ee6ad559dbe4664d0896da4721bf379e
Successfully built future
Installing collected packages: future, tenacity, PyAthena
Successfully installed PyAthena-1.7.1 future-0.17.1 tenacity-5.0.4
You are using pip version 10.0.1, however version 19.2.1 is availabl
e.
You should consider upgrading via the 'pip install --upgrade pip' co
mmand.
```

Setup the analysis environment ead the data from the CSV

In [37]:

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

In [24]:

```
df = pd.read_csv('./data/hmda_analysis_data.csv', dtype = {"as_of_year":int, "loan_amount_000s":float, "applicant_income_000s":float, "rate_spread":float})
```

In [25]:

```
df.head()
```

Out[25]:

	as_of_year	loan_amount_000s	applicant_income_000s	rate_spread
0	2014	126.0	51.0	NaN
1	2014	212.0	135.0	NaN
2	2014	135.0	94.0	1.56
3	2014	361.0	139.0	NaN
4	2014	162.0	81.0	NaN

In [26]:

```
df['loan_to_income_ratio'] = df['loan_amount_000s']/df['applicant_income_000s']
```

In [27]:

```
df.head()
```

Out[27]:

	as_of_year	loan_amount_000s	applicant_income_000s	rate_spread	loan_to_income_ratio
0	2014	126.0	51.0	NaN	2.470588
1	2014	212.0	135.0	NaN	1.570370
2	2014	135.0	94.0	1.56	1.436170
3	2014	361.0	139.0	NaN	2.597122
4	2014	162.0	81.0	NaN	2.000000

Now we have our details in a dataframe. Let's calculate the mean.

In [35]:

```
df_averages = df.groupby('as_of_year').mean()
df_averages
```

Out[35]:

as_of_year	loan_amount_000s	applicant_income_000s	rate_spread	loan_to_income_ratio
2007	229.069417	95.931398	4.675704	2.732374
2008	211.452700	94.634002	4.214035	2.654111
2009	210.212739	99.505074	4.038547	2.616449
2010	215.051597	105.664634	2.499408	2.481347
2011	213.466322	108.402764	2.479931	2.408148
2012	219.481115	108.655808	2.478034	2.457055
2013	219.920301	105.955652	2.145030	2.509665
2014	231.035081	103.415194	1.992665	2.612156
2015	246.576477	107.138255	2.038234	2.670622
2016	257.072973	109.603304	2.013937	2.739110
2017	257.334154	111.901239	2.021350	2.840022

In [33]:

```
df.count()
```

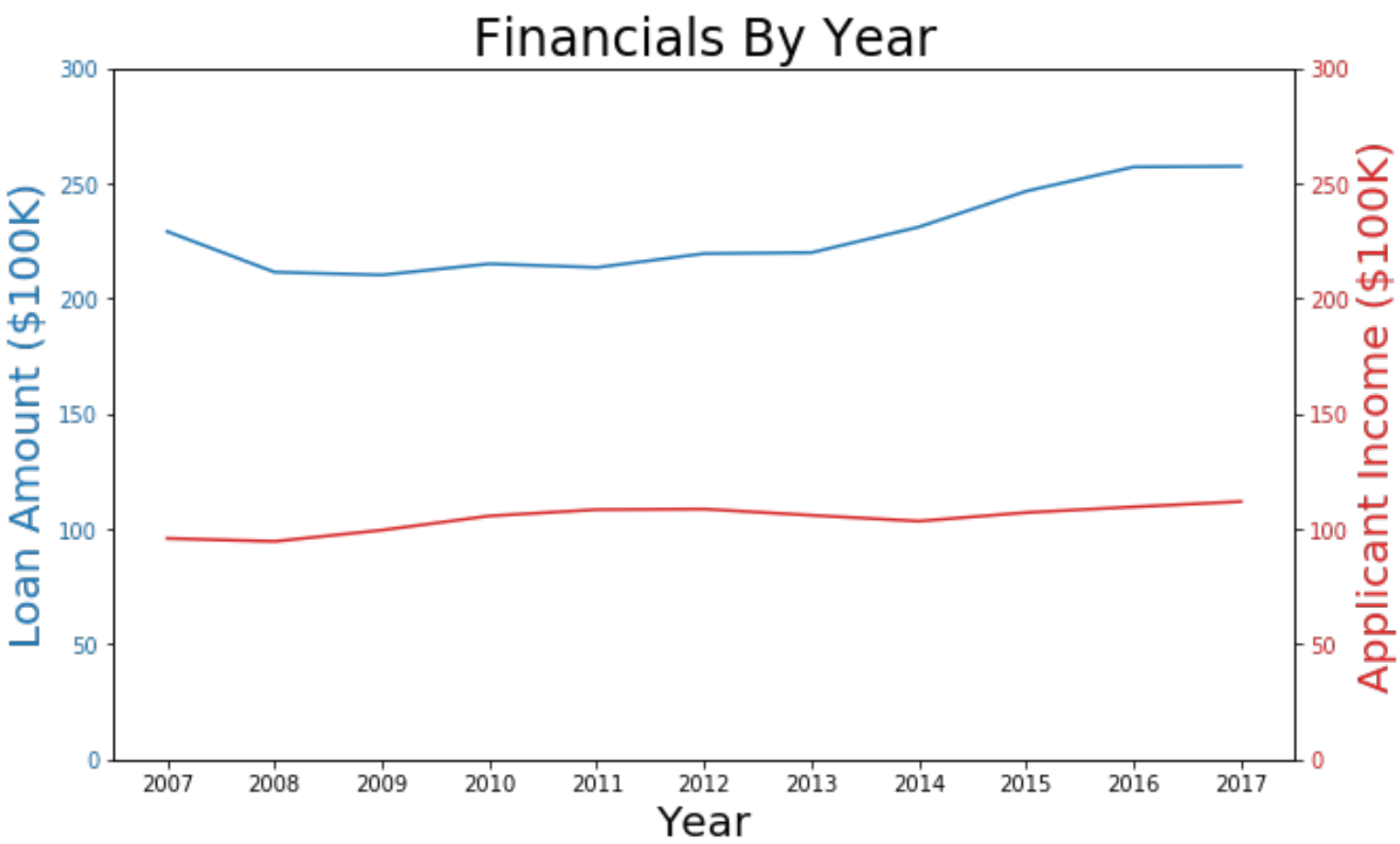
Out[33]:

```
as_of_year               72617573
loan_amount_000s         72616122
applicant_income_000s    68138671
rate_spread               4343760
loan_to_income_ratio     68137221
dtype: int64
```

Create our charts

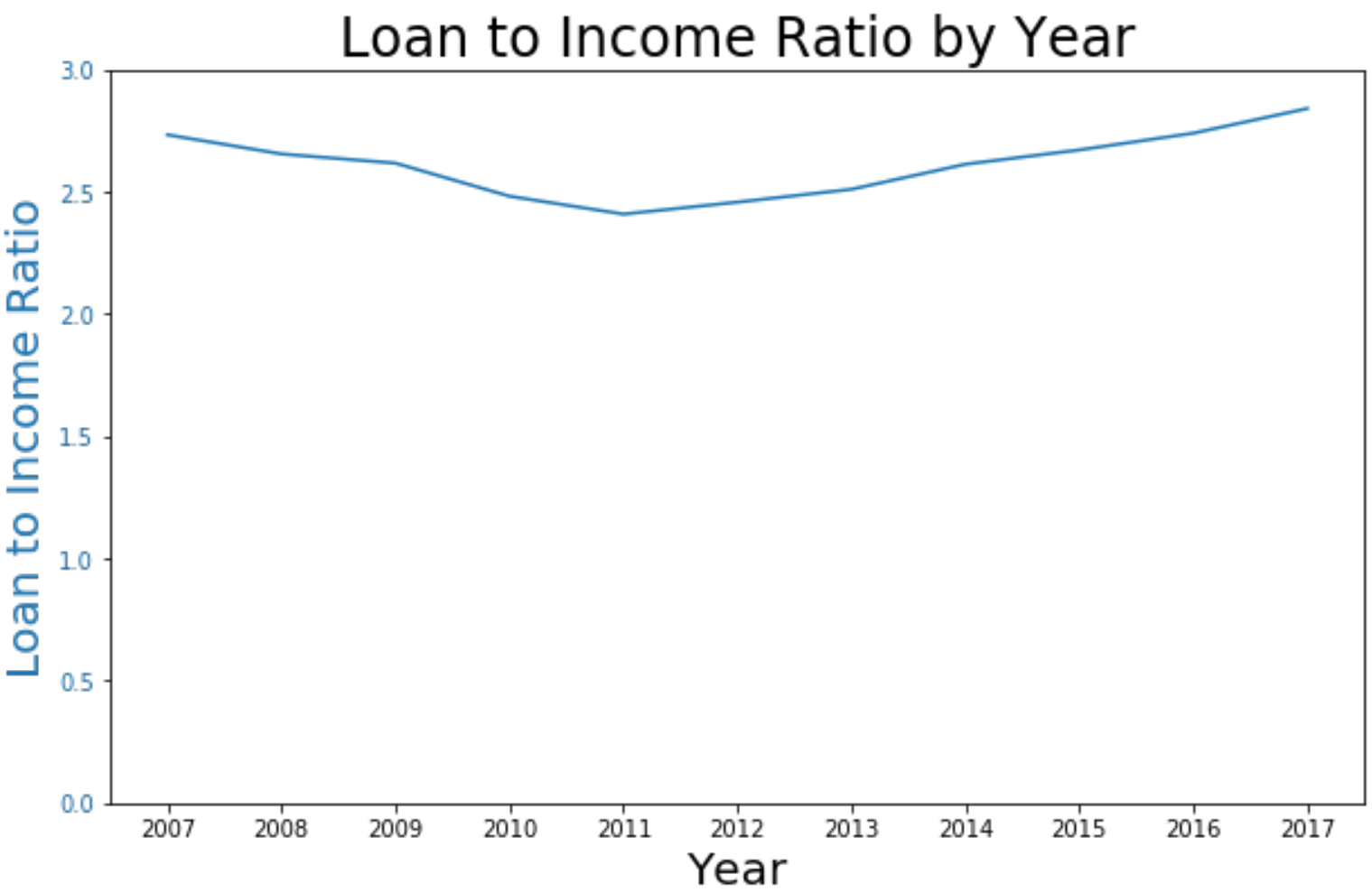
In [74]:

```
fig, ax1 = plt.subplots(figsize=(10,6))

ax1.set_title('Financials By Year',fontsize=24)

color = 'tab:blue'
ax1.set_xlabel('Year', fontsize=20)
ax1.set_ylabel('Loan Amount ($100K)', color=color, fontsize=20)
ax1.plot(df_averages.index, df_averages['loan_amount_000s'], color=color)
ax1.tick_params(axis='y', labelcolor=color)
plt.ylim(0,300)

ax2 = ax1.twinx()  # instantiate a second axes that shares the same x-axis
plt.xticks(df_averages.index)

color = 'tab:red'
ax2.set_ylabel('Applicant Income ($100K)', color=color, fontsize=20)  # we alrea
dy handled the x-label with ax1
ax2.plot(df_averages.index, df_averages['applicant_income_000s'], color=color)
ax2.tick_params(axis='y', labelcolor=color)
plt.ylim(0,300)


#fig.tight_layout()  # otherwise the right y-label is slightly clipped
plt.savefig('Financials.png')
plt.show()
```



In [75]:

```
fig, ax1 = plt.subplots(figsize=(10,6))

ax1.set_title('Loan to Income Ratio by Year',fontsize=24)

color = 'tab:blue'
ax1.set_xlabel('Year', fontsize=20)
ax1.set_ylabel('Loan to Income Ratio', color=color, fontsize=20)
ax1.plot(df_averages.index, df_averages['loan_to_income_ratio'], color=color)
ax1.tick_params(axis='y', labelcolor=color)
plt.ylim(0,3)

plt.xticks(df_averages.index)

#fig.tight_layout()  # otherwise the right y-label is slightly clipped
plt.savefig('LoanToIncome.png')
plt.show()
```



In [ ]: