**CSE 308- Lab #1 -Report**

**Python-Data Manipulation and Visualization**

**Due: Tuesday February 18th 2020, 8:00 am**

1. **Import the proper libraries: Pandas and NumPy and create aliases pd, np respectively.**



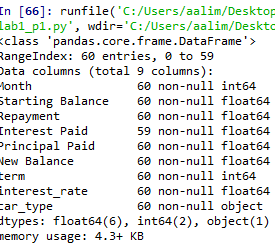
1. **Load sample data (*car\_loan.csv*) into data frame: df**

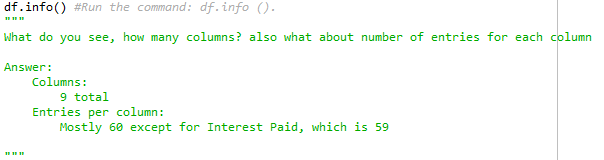


1. **Export Pandas DataFrames to csv. Save file name as *out.csv*. hint: help(df.to\_csv)**



1. **Run the command: df.info (). What do you see, how many columns? also what about number of entries for each column**



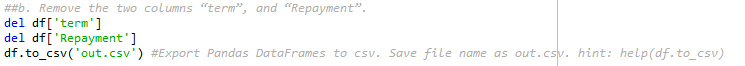


* 1. **5. It is often the case where you change your column names or remove unnecessary columns.**
  2. **a. Change the following columns names:**
  3. **Starting Balance: starting\_balance**

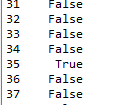
**Interest Paid: interest\_paid**

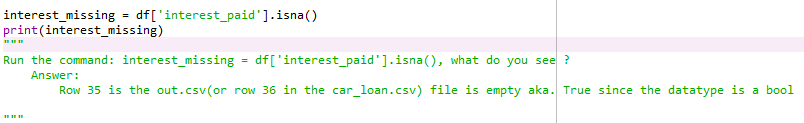
**Principal Paid: principal\_paid**

**New Balance': new\_balance**

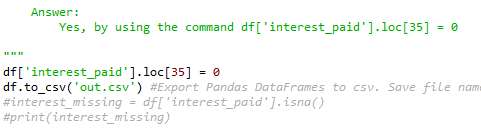
* 1. 
  2. **b. Remove the two columns “term”, and “Repayment”.**
  3. 

1. **Run the command: interest\_missing = df['interest\_paid'].isna(), what do you see ?**

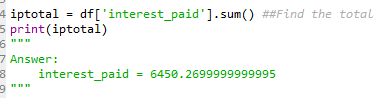




1. **Can you fix the problem in 6 above? hint: use the function df.loc. *property* DataFrame.loc: Access a group of rows and columns by label(s) or a boolean array.**

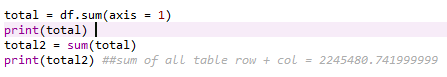


1. **Find the total = amount of interest paid over the course of the loan**





1. **Find the sum of all values across all columns**



0 69583.9202

1 68613.4902

2 67637.3802

3 66655.5502

4 65667.9702

5 64674.6102

6 63675.4302

7 62670.4002

8 61659.4902

9 60642.6602

10 59619.8702

11 58591.0902

12 57556.2902

13 56515.4302

14 55468.4702

15 54415.3802

16 53356.1302

17 52290.6802

18 51218.9902

19 50141.0202

20 49056.7402

21 47966.1102

22 46869.0902

23 45765.6502

24 44655.7502

25 43539.3502

26 42416.4102

27 41286.9002

28 40150.7802

29 39008.0002

30 37858.5302

31 36702.3302

32 35539.3602

33 34369.5802

34 33192.9502

35 31916.1902

36 30818.9802

37 29621.5602

38 28417.1302

39 27205.6502

40 25987.0802

41 24761.3802

42 23528.5002

43 22288.4002

44 21041.0402

45 19786.3802

46 18524.3702

47 17254.9702

48 15978.1402

49 14693.8302

50 13402.0002

51 12102.6102

52 10795.6202

53 9480.9802

54 8158.6402

55 6828.5602

56 5490.6902

57 4144.9802

58 2791.3902

59 1429.8802

dtype: float64

2245480.741999999

1. **Convert Pandas DataFrames to NumPy arrays**



1. **Import the library pyplot from matplotlib and create alias plt**



1. **import seaborn library (wrapper of matplotlib) and create alias: sns**



**13. load data out.csv**

**14. use the loc property to find the values of the followings: month\_numbe, interest\_paid, principal\_paid.**

**For example: month\_number = df.loc[:, 'Month'].values will return:**

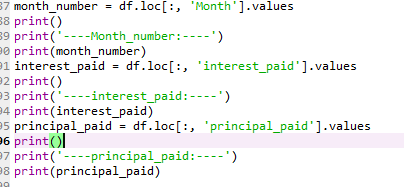
***array([ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,***

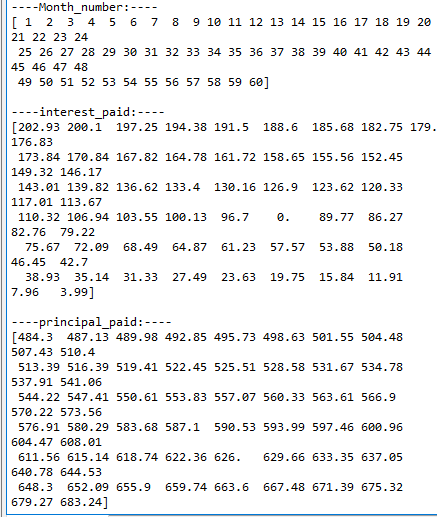
**18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,**

***35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51,***

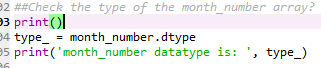
**52, 53, 54, 55, 56, 57, 58, 59, 60])**

***# The values attribute converts a column of values into a numpy array***



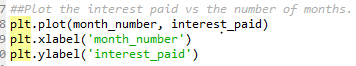


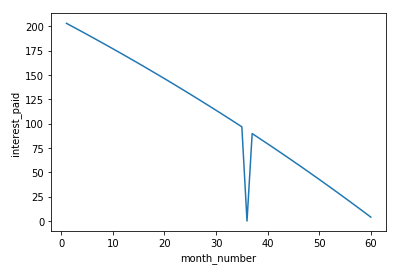
**15. Check the type of the month\_number array*?***



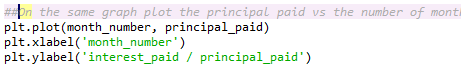


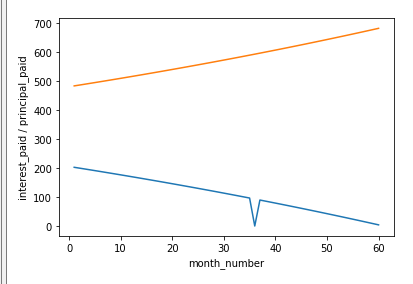
**16. Plot the interest paid vs the number of months.**





**17. On the same graph plot the principal paid vs the number of months**

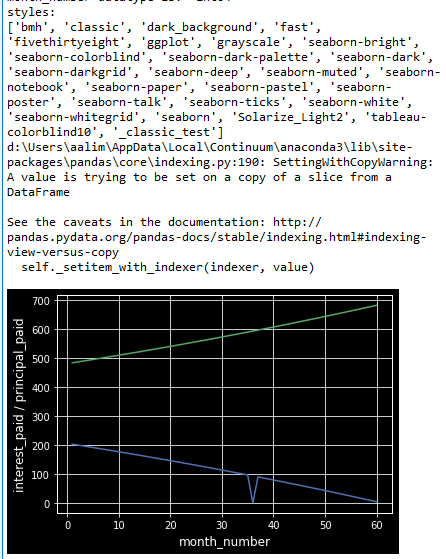


**aalim**

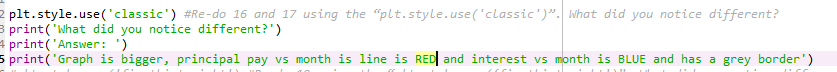
**18. you can use plt.style.available to select an appropriate aesthetic styles for your figures.**

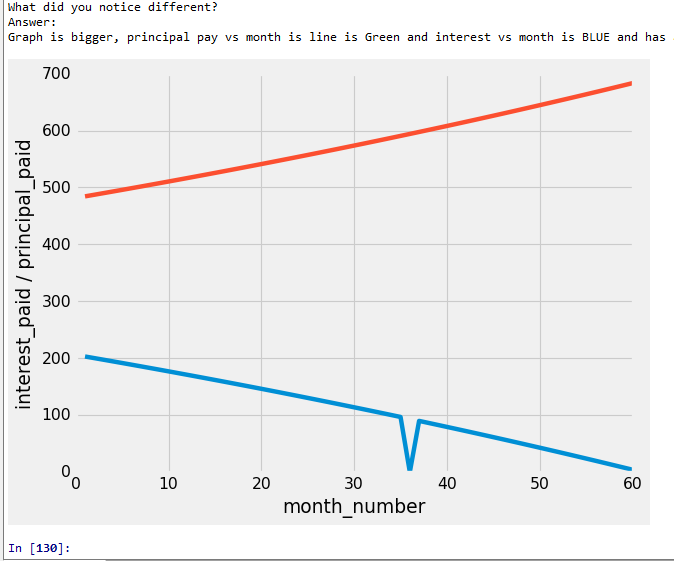
**Run the following command: plt.style.available, you should see a list of different styles.**



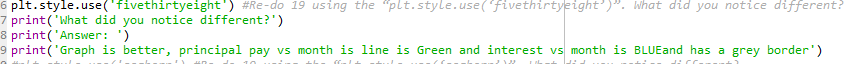


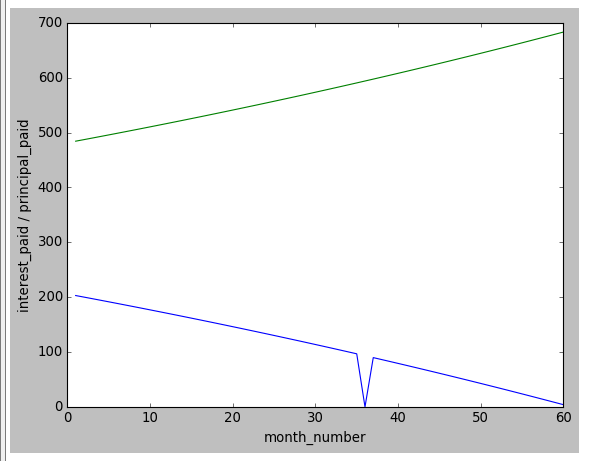
**19. Re-do 16 and 17 using the “plt.style.use('classic')”. What did you notice different?**



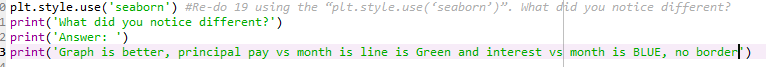


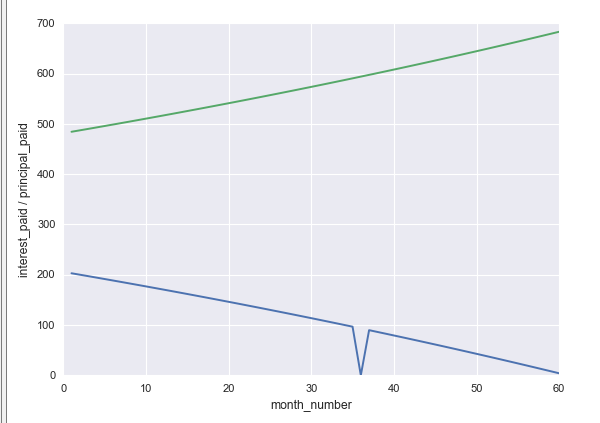
**20. Re-do 19 using the “plt.style.use(‘fivethirtyeight’)”. What did you notice different?**





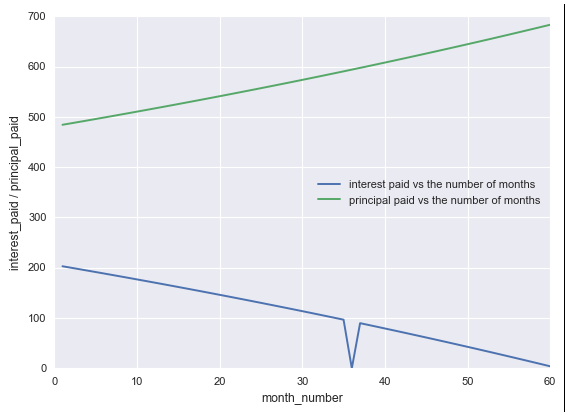
**21. Re-do 19 using the “plt.style.use(‘seaborn’)”. What did you notice different?**



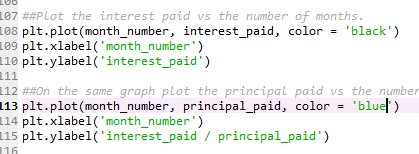


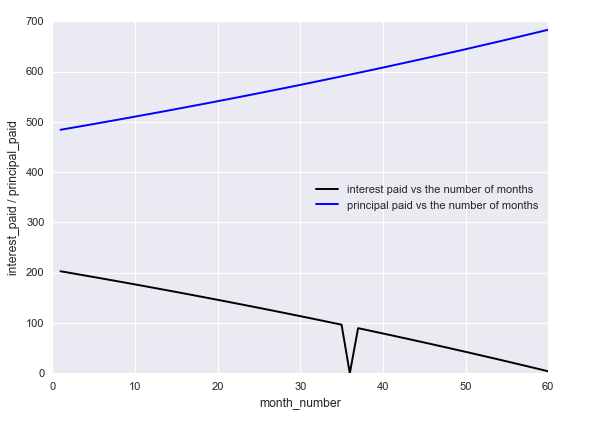
**22. Add legend to your figures. Add it to be "center right”.**

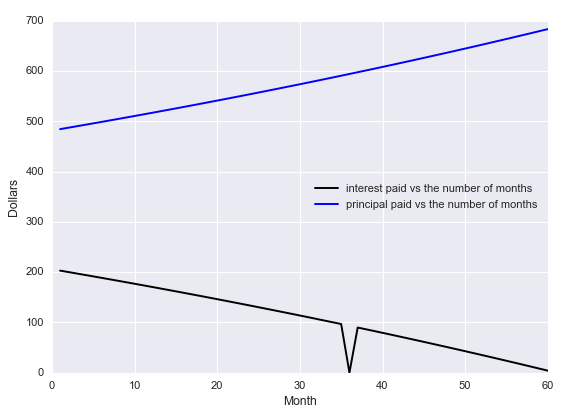
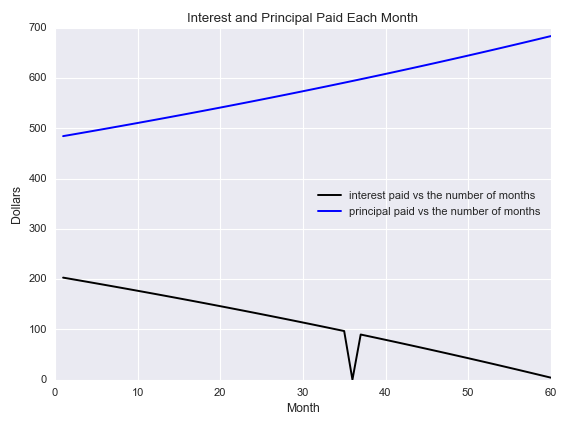




**23. Add markers and colors. The interest\_paid in Black, and principal\_paid in blue**





* 1. **24. Setting plot titles, labels choose font size of 12**
  2. **a. Set xlabel and ylabel : x:Month, y: Dollars**
  3. 
  4. 
  5. **b. Set Title: Interest and Principal Paid Each Month**
  6. 
  7. 

**25. Saving plots to files.**

Plot.png

**Py code**

"""

Aaron Lim

005893468

CSE308 - Lab1

Python-Data Manipulation and Visualization

"""

import pandas as pd # import the entire numpy package, giving it the alias 'np'

import numpy as np # import the entire numpy package, giving it the alias 'np'

df = pd.read\_csv('car\_loan.csv') #Load sample data (car\_loan.csv) into data frame: df

df.to\_csv('out.csv') #Export Pandas DataFrames to csv. Save file name as out.csv. hint: help(df.to\_csv)

df.info() #Run the command: df.info ().

"""

What do you see, how many columns? also what about number of entries for each column

Answer:

Columns:

9 total

Entries per column:

Mostly 60 except for Interest Paid, which is 59

"""

#It is often the case where you change your column names or remove unnecessary columns.

#a. Change the following columns names:

#Starting Balance: starting\_balance

#Interest Paid: interest\_paid

#Principal Paid: principal\_paid

#New Balance': new\_balance

df = pd.read\_csv('car\_loan.csv', skiprows = 1, names = ['Month', 'starting\_balance', 'Repayment', 'interest\_paid', 'principal\_paid', 'new\_balance', 'term', 'interest\_rate', 'cartype']) # Rename all of the columns, keeping them in order

df.to\_csv('out.csv') #Export Pandas DataFrames to csv. Save file name as out.csv. hint: help(df.to\_csv)

##b. Remove the two columns “term”, and “Repayment”.

del df['term']

del df['Repayment']

df.to\_csv('out.csv') #Export Pandas DataFrames to csv. Save file name as out.csv. hint: help(df.to\_csv)

interest\_missing = df['interest\_paid'].isna()

print(interest\_missing)

"""

Run the command: interest\_missing = df['interest\_paid'].isna(), what do you see ?

Answer:

Row 35 is the out.csv(or row 36 in the car\_loan.csv) file is empty aka. True since the datatype is a bool

"""

"""

Can you fix the problem in 6 above?

hint: use the function df.loc. property DataFrame.loc: Access a group of rows and columns by label(s) or a boolean array.

Answer:

Yes, by using the command df['interest\_paid'].loc[35] = 0

"""

df['interest\_paid'].loc[35] = 0

df.to\_csv('out.csv') #Export Pandas DataFrames to csv. Save file name as out.csv. hint: help(df.to\_csv)

#interest\_missing = df['interest\_paid'].isna()

#print(interest\_missing)

iptotal = df['interest\_paid'].sum() ##Find the total = amount of interest paid over the course of the loan

print(iptotal)

"""

Answer:

interest\_paid = 6450.2699999999995

"""

"""

Find the sum of all values across all columns

Answer:

2245480.741999999

"""

total = df.sum(axis = 1)

print(total)

total2 = sum(total)

print(total2) ##sum of all table row + col = 2245480.741999999

df.to\_numpy() ##Convert Pandas DataFrames to NumPy arrays

import matplotlib.pyplot as plt ##Import the library pyplot from matplotlib and create alias plt

import seaborn as sns ##import seaborn library (wrapper of matplotlib) and create alias: sns

##library (wrapper of matplotlib) and create alias: sns

##13. load data out.csv

##use the loc property to find the values of the followings: month\_numbe, interest\_paid, principal\_paid.

month\_number = df.loc[:, 'Month'].values

print()

print('----Month\_number:----')

print(month\_number)

interest\_paid = df.loc[:, 'interest\_paid'].values

print()

print('----interest\_paid:----')

print(interest\_paid)

principal\_paid = df.loc[:, 'principal\_paid'].values

print()

print('----principal\_paid:----')

print(principal\_paid)

##Check the type of the month\_number array?

print()

type\_ = month\_number.dtype

print('month\_number datatype is: ', type\_)

##Plot the interest paid vs the number of months.

plt.plot(month\_number, interest\_paid, color = 'black')

#plt.xlabel('month\_number')

#plt.ylabel('interest\_paid')

##On the same graph plot the principal paid vs the number of months

plt.plot(month\_number, principal\_paid, color = 'blue')

#plt.xlabel('month\_number')

#plt.ylabel('interest\_paid / principal\_paid')

plt.xlabel('Month')

plt.ylabel('Dollars')

plt.title('Interest and Principal Paid Each Month')

styles = plt.style.available ##you can use plt.style.available to select an appropriate aesthetic styles for your figures.

print('styles: ')

print(styles)

#plt.style.use('seaborn-deep')

#plt.style.use('classic') #Re-do 16 and 17 using the “plt.style.use('classic')”. What did you notice different?

#print('What did you notice different?')

#print('Answer: ')

#print('Graph is bigger, principal pay vs month is line is RED and interest vs month is BLUE and has a grey border')

#plt.style.use('fivethirtyeight') #Re-do 19 using the “plt.style.use(‘fivethirtyeight’)”. What did you notice different?

#print('What did you notice different?')

#print('Answer: ')

#print('Graph is better, principal pay vs month is line is Green and interest vs month is BLUEand has a grey border')

plt.style.use('seaborn') #Re-do 19 using the “plt.style.use(‘seaborn’)”. What did you notice different?

print('What did you notice different?')

print('Answer: ')

print('Graph is better, principal pay vs month is line is Green and interest vs month is BLUE, no border')

plt.legend(['interest paid vs the number of months', 'principal paid vs the number of months'], loc = 'center right') ## Add legend to your figures. Add it to be center right.

##plt.show()

plt.savefig('plot.png')