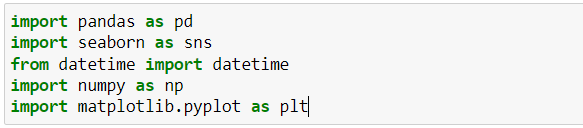
**Case Study**

**Loan Dataset:**

This data set **includes customers who have paid off their loans, who have been past due and put into collection** without paying back their loan and interests,

**Code:**

**Import Required Modules**



pandas - Pandas for Data Frame

datetime - datetime for work with the time zone, months, year

numpy - NumPy for work with numeric

matplotlib.pyplot - Data Visualization

seaborn - Data Visualization

**Step1: Read the given loan.csv file**



- “loandata” is a variable defined for loan data

- Print the top 5 rows from the dataframe

**Step 2: Data Filter and Cleaning**

### 2(a) Remove columns containing either 75 % or more than 75 % Null Values

### Most of the columns contain null values, which contain huge amounts of data. Remove the columns that have greater than 50% or 75% null values from the entire data frame.

### 

### 2(b). List out the columns after dropping them

### List of the columns after dropping them above .

### 2(c). Count the missing values from all the columns.

### Count the null values from all the columns in a DataFrame.

### 

### Output:

### 

### 2(d). Display the bar chart of null values from the dataframe

### Display the bar histogram chart for null values count from the columns

### 

# **Step3: Fill the Null values and manipulate the values in all columns**

### 3(a). Fill the null values in the columns

### Fill the null values with NA

### 

### 3(b). Fill the missing values with the observable/Relevant data

### Fill the null values with relevant data, based on the column data.

### 

### 3(c). Remove left space and extra characters from the column values

### Some of the column data was not formatted properly, applying the lambda function to remove extra spaces and special characters etc.,

### 

### 3(d). Fill the data in Date columns

### Date formats are not properly formed, to avoid the issues/errors converting the date format with a normal format’s like %y-%d-%m or %d-%m-%y or %m-%d-%y

### 

### *Apply the function Date for convertion:*

### Parse the dates from the DataFrame (strptime) and converting them with normal format using strftime. After that applying the function in the date columns using with lambda

### 

### 

### *Updating the 'earliest\_cr\_line' date format, which it have month-year and convert to year-month-date:*

### One of the column days are missing in the entire column. Adding the random days(1-31) value using with np.random.radint. And adding with join in the specific column.

### 

# **Step4: Filter the outlier and Filter it**

### Create box plot and check Outlier for set of the columns and apply the inter-quartile range to detect the outliers and filter it

### 

#### **04(a). loan\_amnt, funded\_amnt, funded\_amnt\_inv**

### 

#### **Filter the data with IQR (Interquartile range)**

Example:

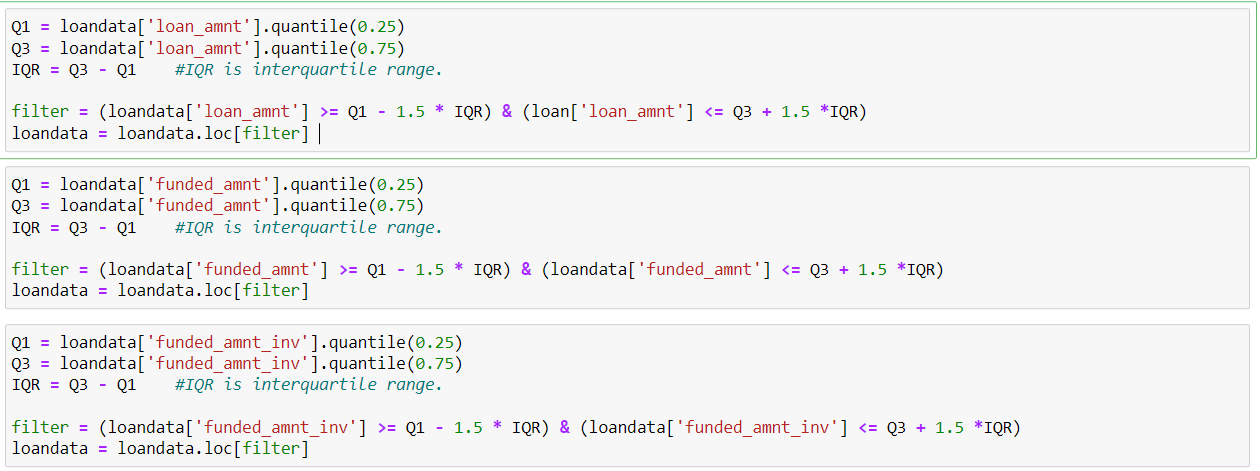
Q1 = data['column'].quantile(0.25)

Q3 = data['column'].quantile(0.75)

IQR = Q3 - Q1 #IQR is interquartile range.

filter = (data['column'] >= Q1 - 1.5 \* IQR) & (data['column'] <= Q3 + 1.5 \*IQR)

data = data.loc[filter]



### 

#### **04(b). total\_pymnt, total\_pymnt\_inv, total\_rec\_prncp, total\_rec\_int, last\_pymnt\_amnt**

### 

### 

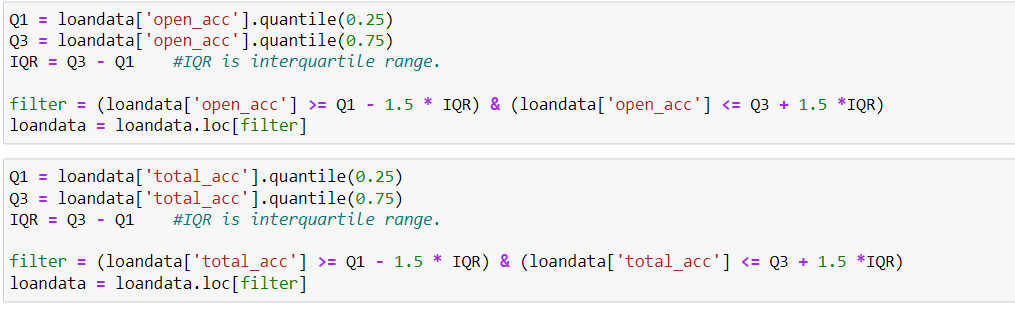
#### **Filter the data with IQR (Interquartile range)**

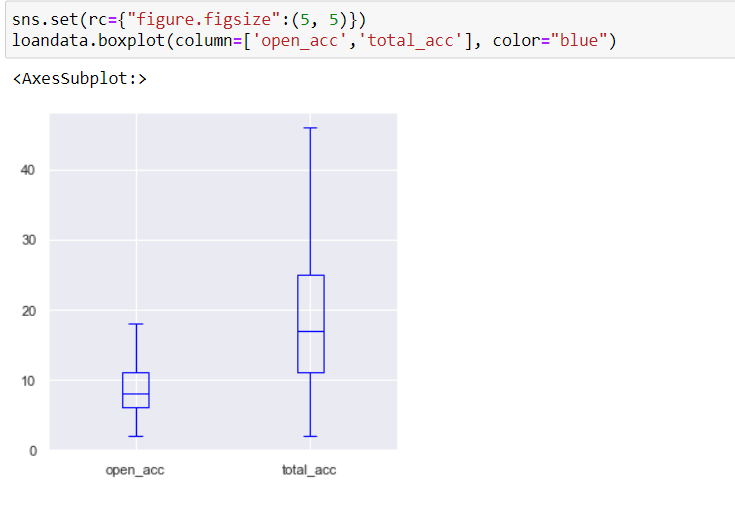
### 

### 

### 4(c). total\_acc and open\_acc

# 





# **Step5: Data Analysis**

### 5(a). Derived Metrics

### 

### 5(b). Categorical Variables

#### **Loan Status**

### 

### 

### *Loan Term with 'Loan Status'*

### 

### 

# **5(c). Correlation from the dataframe**

### *correlation 'loan\_amnt', 'funded\_amnt', 'funded\_amnt\_inv'*

### 

### *correlation from 'total\_pymnt','total\_pymnt\_inv','total\_rec\_prncp','total\_rec\_int', 'last\_pymnt\_amnt'*

### 

### *Loan amount applied by member living in own or rent and analysing with funded amount & annual income (Risk analisys)*

### 

### *Heat map for risk analysis corelation between laon\_amnt, funded\_amnt and anual\_inc*

### 