Exploring Numerical Quantities

This lesson will focus on exploring the numerical quantities and finding out general trends from these quantities.

WE'LL COVER THE FOLLOWING

- Scatter plots
- Binning numerical data
  - AGE
- Quiz

A very important part of exploratory data analysis is finding out general trends or patterns in the data. We can find out different relationships between two quantities that can be very helpful in making decisions at the end. We will use the cleaned version of the dataset from the lesson Inconsistent Data. The details of individual columns are mentioned below.

```
# Default of Credit Card Clients Dataset
# There are 25 variables:

# ID: ID of each client
# LIMIT_BAL: Amount of given credit in NT dollars (includes individual and family/supplementa
# GENDER: Gender (male,female)
# EDUCATION: (1=graduate school, 2=university, 3=high school, 4=others)
# MARRIAGE: Marital status (married, single, others)
# AGE: Age in years
# PAY_1: Repayment status in September, 2005 (0=pay duly, 1=payment delay for one month, 2=pa
# PAY_2: Repayment status in August, 2005 (scale same as above)
# PAY_3: Repayment status in July, 2005 (scale same as above)
# PAY_4: Repayment status in June, 2005 (scale same as above)
# PAY_5: Repayment status in May, 2005 (scale same as above)
# PAY_6: Repayment status in April, 2005 (scale same as above)
# BILL_AMT1: Amount of bill statement in September, 2005 (NT dollar)
# BILL_AMT2: Amount of bill statement in August, 2005 (NT dollar)
# BILL_AMT3: Amount of bill statement in July, 2005 (NT dollar)
# BILL_AMT4: Amount of bill statement in June, 2005 (NT dollar)
# BILL_AMT5: Amount of bill statement in May, 2005 (NT dollar)
# BILL_AMT6: Amount of bill statement in April, 2005 (NT dollar)
# PAY_AMT1: Amount of previous payment in September, 2005 (NT dollar)
# PAY_AMT2: Amount of previous payment in August, 2005 (NT dollar)
# PAY_AMT3: Amount of previous payment in July, 2005 (NT dollar)
# PAY_AMT4: Amount of previous payment in June, 2005 (NT dollar)
```

```
# PAY_AMT5: Amount of previous payment in May, 2005 (NT dollar)
# PAY_AMT6: Amount of previous payment in April, 2005 (NT dollar)
# default.payment.next.month: Default payment (yes,no)
```

Scatter plots

Scatter Plots are a very useful way of visualizing the *inverse* and *direct* relationships between two variables. In a **direct** relationship between two quantities, an *increase/decrease* in one quantity leads to a corresponding *increase/decrease* in the other quantity, whereas in an **inverse** relationship, an *increase/decrease* in one quantity leads to a corresponding *increase/decrease* in the other quantity.

However, in real data, we do not observe strict direct or inverse relationships, rather we observe relationships or patterns that look like direct or linear relationships because there are many external factors that affect a quantity.

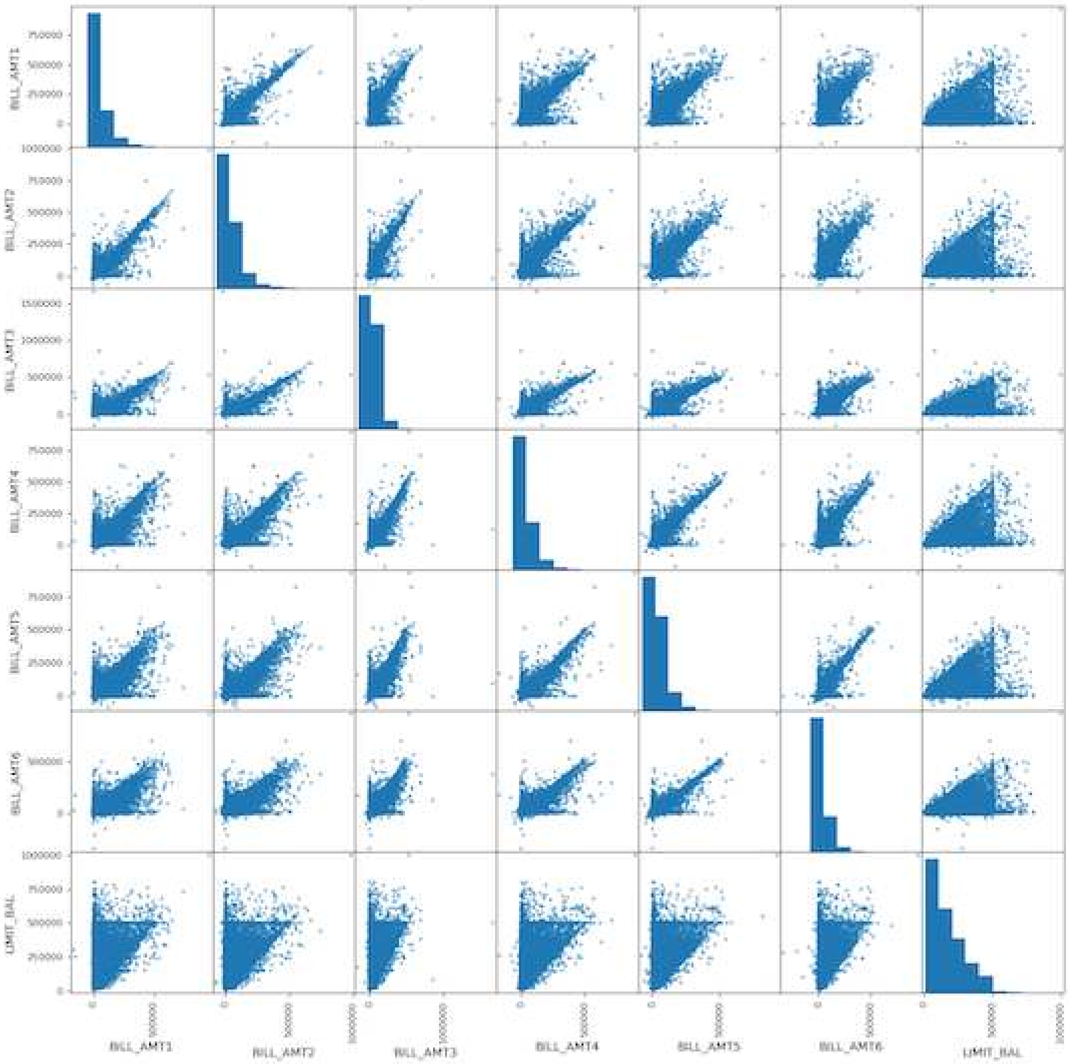
```
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_csv('credit_card_cleaned.csv')

# Filter dataset
cols_to_select = ['BILL_AMT1','BILL_AMT2','BILL_AMT3','BILL_AMT4','BILL_AMT5','BILL_AMT6','LI
df = df[cols_to_select]

# Scatter plots
pd.scatter_matrix(df,figsize = (15,15))
plt.show()
```

We are interested in finding out the spending habits of people from month to month, whether they generally spend the same every month, or if there are some months in which they spend extra, and how the amount of credit given (`LIMIT_BAL`) varies with the bills.

We write the billed amount columns (`BILL_AMT1`, `BILL_AMT2`,...) and `LIMIT_BAL` in **line 6** in a list, and filter using this list in **line** 7. Then we use the pandas function `scatter_matrix` in **line 8** which draws the scatter plots between all the variables in the dataframe.

> Keep in mind that `scatter_matrix` is not a function that is called on a dataframe. Rather it is provided a dataframe as an argument

> dataframe. Rather it is provided a dataframe as an argument.



By looking at the last row of the scatter matrix, we can see that there is a kind of linear relationship between `LIMIT_BAL` and all other bill amount variables. As we increase the bill amounts, the credit given is increased. This means that the bank gives more credit to people who spend more usually. But there a few exceptions that can be seen from the plots. There are a few people who spend very little yet are given high credits.

Another observation that we can make from the scatter matrix is that as we increase the amount of bills in a month, we are likely to see an increase in the amount of bills the next month. For instance, look at the plot where `BILL_AMT6` (The amount of Bills in April 2005) is at the x-axis and `BILL_AMT5`(The amount of Bills in May 2005) is at the y-axis. We see a pattern similar to a direct relationship. We can see the same pattern between `BILL_AMT5` and `BILL_AMT4`, `BILL_AMT4` and `BILL_AMT3` and so on. This tells us that people usually spend

similar amounts of money in these months and if they spend a certain amount in one month, they are expected to spend similar amounts in the next month.

Binning numerical data

We can divide our numerical data in bins and see how many people in each bin default.

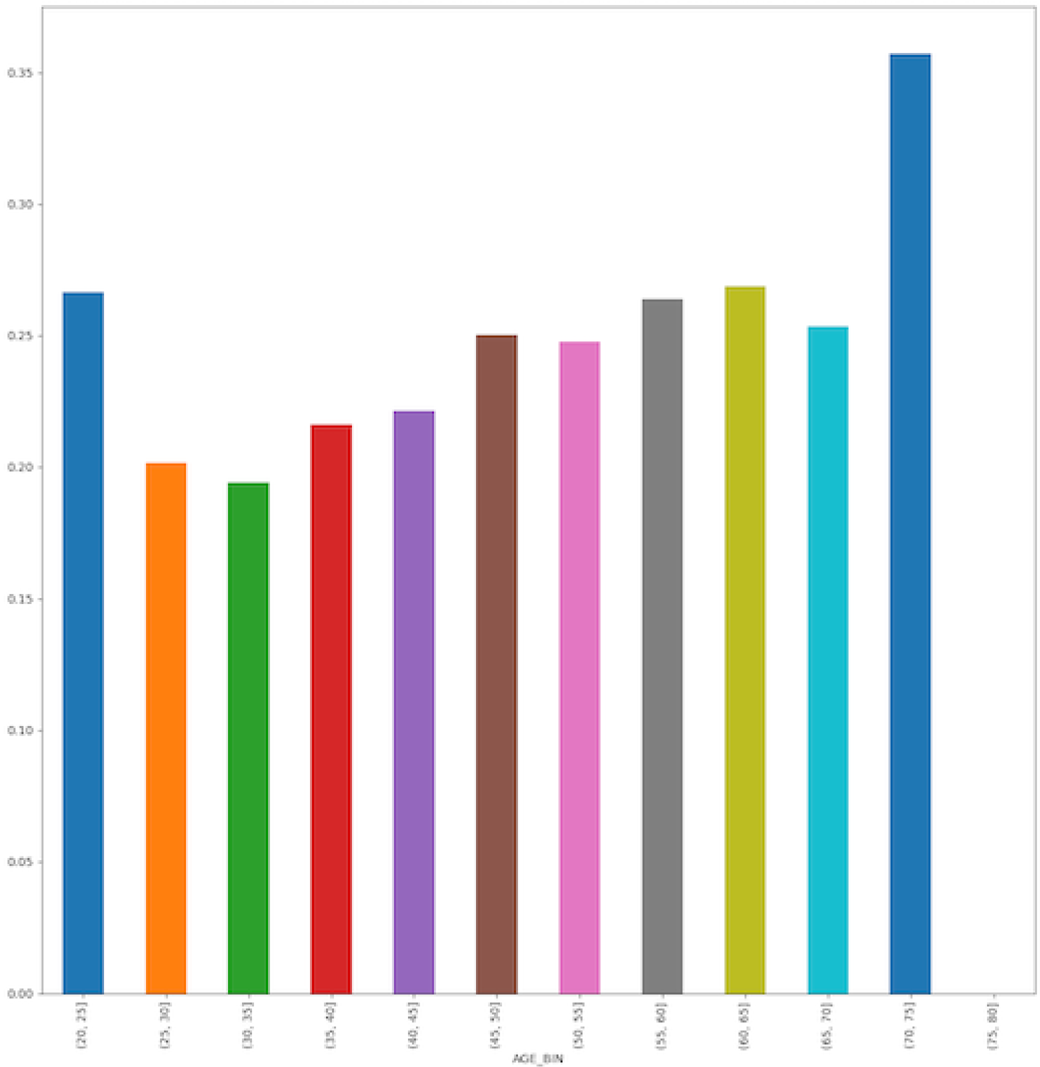
`AGE`

```
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_csv('credit_card_cleaned.csv')

# Create bins and data in bins
custom_bins = [20,25,30,35,40,45,50,55,60,65,70,75,80]
binned = pd.cut(df['AGE'],bins = custom_bins)
print(binned.head())
df['AGE_BIN'] = binned

# Group using bins
grouped_df = df.groupby(['AGE_BIN','default.payment.next.month']).size()
grouped_df = grouped_df.unstack()

print(grouped_df)

# Calculate probabilities
grouped_df['prob_default'] = grouped_df['yes'] / (grouped_df['no'] + grouped_df['yes'])

print('\n\n',grouped_df[['prob_default']])

# Plot probabilities
grouped_df['prob_default'].plot(kind = 'bar',figsize = (15,15))

plt.show()
```

We divide the ages into bins. We use the pandas function `cut` and give it the column that we want to divide as the first argument, and give the bins we made as `bins = custom_bins` in **line** 7. This gives us a series in which we have the age bin against every row. We then add this to our dataframe as a new column (`AGE_BIN`) in **line 9**. Now we have both `AGE` and `AGE_BIN` for every row in the dataset.

After this, we group the data by `AGE_BIN` and `default.payment.next.month`, and call `size()` on the groups to obtain the number of default and non-defaults in each age bin in **line 12**. After this, we use the function `unstack` to change the groups into a dataframe and name the columns as `yes` and `no`.

Then, we calculate the probability of defaulting for each age bin using the simple formula we used in the last lesson in **line 18** and save these as a new column in the dataset named `prob_default`. We plot the probabilities of each age group defaulting in **line 23**.



From the plot of the probabilities for defaulting in each age group, it is visible that very young people and very old people are more likely to default.

Quiz

You have a quiz below. You are also provided an empty code window. You have to answer the quiz questions by writing code and finding answers to the questions.

```
import pandas as pd
df = pd.read_csv('credit_card_cleaned.csv')

# Write your code here
```

1

How many people lie in the interval (0, 100000] of `LIMIT_BAL` who have defaulted?

COMPLETED 0%

1 of 3

These were some techniques to explore numerical data. There is another mathematical way of exploring the relationships between quantities known as *correlation* which we will study in the next lesson.