Analyzing Individual Quantities

This lesson focuses on how to analyze different quantities to look for skewedness and bias in the data.

WE'LL COVER THE FOLLOWING

Analyzing individual variables is usually the way to start with EDA after figuring out data types. Summarizing a variable or looking at its distribution can be very helpful.

We will be using the Default of Credit Card Clients Dataset. This dataset contains information on default payments, demographic factors, credit data, history of payment, and bill statements of credit card clients in Taiwan from April 2005 to September 2005. However, 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_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)
```

Details of all columns in the dataset

Summary stats

Summarizing a variable can give us useful information which can be used to draw conclusions or make decisions. Some common summarizing statistics are:

- mean
- median
- quartiles

We can use the `describe` function on our dataframe which summarizes individual columns for us, or we can select the columns that we want and use functions like `mean`, `std`, and `max` on them.

```
import pandas as pd
df = pd.read_csv('credit_card_cleaned.csv')
# Get summary stats
print(df[['EDUCATION','AGE']].describe())
```

We have selected two variables and then called the function `describe` on them in **line 4**. The output of **line 4** gives us the count, mean, standard deviation, quartiles, minimum, and maximum.

By looking at the output, we find out that

- The average age is 35 and the 75^{th} percentile of `AGE` is at 41 which means

75 percent of the people in the dataset are below 41. This is a useful insight.

- The 75^{th} percentile of `EDUCATION` is 2 which implies that at least 75 percent of the people in this dataset have been to university.

Categorical variables

When we look at the dataset for exploration we see three important categorical variables in this dataset i.e. `EDUCATION`, `GENDER`, and `MARRIAGE`. We might be interested in how many males and females are in the dataset or what is the level of education of the majority. Let's see how the data is distributed among these categorical variables.

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

# Counts for each value of GENDER
print('Value counts of GENDER :')
print(df['GENDER'].value_counts())

# Counts for each value of MARRIAGE
print('Value counts of MARRIAGE :')
print(df['MARRIAGE'].value_counts())

# Counts for each value of EDUCATION
print('Value counts of EDUCATION :')
print(df['EDUCATION'].value_counts())

# Make figure
fig,sub_plots = plt.subplots(1,3,figsize=(13,10))

# Plots
df['GENDER'].value_counts().plot(kind='bar',ax=sub_plots[0],title= 'GENDER')
df['MARRIAGE'].value_counts().plot(kind='bar',ax=sub_plots[1],title = 'MARRIAGE')
df['EDUCATION'].value_counts().plot(kind='bar',ax=sub_plots[2],title = 'EDUCATION')

plt.show()
```

We use the function `value_counts` to retrieve the number of unique values for a variable on **lines 7,11, and 15** and then we print them. The function `value_counts` can only be called on a series object, therefore, if we want to plot the value counts of all three variables separately on a single figure, we have to do some extra work.

have to do some extra work.

We make the customized figure with subplots using `plt.subplots` on **line 18**. We give the layout of the grid to be `1,3` since we want three plots to be drawn side by side. We also provide the figure size as `figsize`. We store the figure as `fig` and the array of subplots as `ax`.

On **lines 21-23**, we plot the value counts by using the function `plot`. We specify the kind of the plot as `bar`. The `plot` function, when used with a *series*, can be provided a subplot as `ax`. Hence, we provide the subplots (`sub_plots`) as `ax`.

`sub_plots[0]` means we want this plot to be drawn as the first subplot in the figure, while `sub_plots[1]` means we want the plot to be drawn at the second position, and so on. Moreover, we provide the titles of the plots as `title` to the `plot` function.

Scroll the graph output to view the text output. By looking at the outputs of **lines 7,11,15**, and the plots, we conclude that:

- There are more females in the dataset than males.
- We have an almost balanced set of married and unmarried people overall.
- The greatest number of people have gone to University and then Graduate School and then High School in the dataset.

Distributions

Looking at the distributions of the variables can be very helpful and can give us key insights.

AGE

We will be drawing a histogram of the ages of the people in the dataset as an example.

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

# Make bins and plot
custom_bins = [20,25,30,35,40,45,50,55,60,65,70,75,80]
```

```
df['AGE'].plot(kind = 'hist',bins = custom_bins ,rwidth = 0.8)
plt.show()

# Print counts in each bin
exact_counts = df['AGE'].value_counts(bins = custom_bins)
print(exact_counts)
```

In **line 6**, we create a list that has values that will be used as customized bins for the histogram. In the next line, we select the `AGE` column and use the `plot` function with it. We specify the kind of plot as `hist` and give our custom bins to the function. Moreover, we specify the width between the bars as `rwidth`.

But what if we want the exact number of people in each bin? To do that, we use the `value_counts` function and provide our `custom_bins` to it in **line 11**. It gives us the exact number of people in each bin. So, we have some useful insights from this distribution:

- The greatest number of people lie in the $25 - 30$ age group.
- The majority of the people lie in the $20 - 40$ age group.

Skeweness

Skewness is the measure of the asymmetry of a distribution. In a *normal* distribution, the mean divides the density curve symmetrically into two equal parts at the median and the value of skewness is zero. When a distribution is asymmetrical, the tail of the distribution is skewed to one side either to the right or to the left.

When the value of the skewness is *negative*, the tail of the distribution is longer towards the left-hand side of the curve. This simply means there are more values towards the left side of the distribution.

When the value of the skewness is *positive*, the tail of the distribution is longer towards the right-hand side of the curve. This simply means there are more values towards the right side of the distribution.

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

```
# Plot desnity
df.plot(kind='density',subplots=True,sharex = False,sharey=False,layout = (5,5), figsize = (1
print(df.skew())
```

We can check skewness both graphically and mathematically. We plot density plots of all the variables in **line 6**. We also use the `skew` function on the dataframe that gives us a measure of skewness for all variables.

From the output of **line 6**, we see that:

- Most variables have a positive skew.
- The payment variables (`PAY_AMT1`, `PAY_AMT2`...) are the most skewed variables. This can also be verified from the density plots.

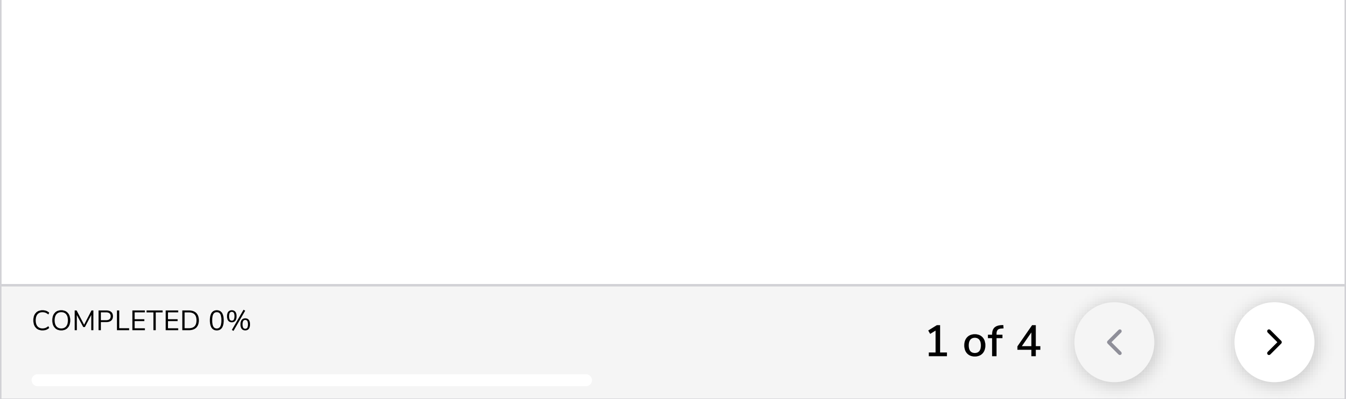
Quiz

You have been given a quiz on `LIMIT_BAL` (the amount of credit that is given to a person) column 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 What is the mean of `LIMIT_BAL`?



This was how we can explore individual variables in the dataset. In the next lesson, we will see some techniques for exploring relationships between categorical variables.