4.9: Intro to Data Visualization with Python

Part 1:

From Step 6: Combine your customer data with the rest of your prepared Instacart data.

```
# Merge of the new customers df to ords prods merge df

df_merged_clean = df_customers_clean.merge(df_ords_prods_merge, on = 'user_id', indicator = True)

df_merged_clean.head()

#dropping the merge column

df_merged_clean = df_merged_clean.drop(columns = ['_merge'])

#checking its dropped

df_merged_clean.head()
```

Export this new dataframe as a pickle file so you can continue to use it in the second part of this task.

```
# Export data to pkl

df_merged_clean.to_pickle(os.path.join(path, '02 Data','Prepared Data', 'orda_prods_all_updated.pkl'))
```

Part 2:

Create a new notebook, import the necessary analysis and visualization libraries, then import your most up-to-date project data (i.e., the data set with your new customer data from the first part of this task).

 # Import Library

```
import pandas as pd

import numpy as np

import os

import matplotlib.pyplot as plt

import seaborn as sns

import scipy
```

#folder shortcut

```
path =  r'C:\Users\Admin\Documents\18-07-2023 Instacart Basket Analysis'
```

#importing newdataset

```
df_ords_prods_all = pd.read_pickle(os.path.join(path, '02 Data', 'Prepared Data', 'orders_products_customers_49_merged.pkl'))
```

#checking rows and columns

```
df_ords_prods_all.shape
```

#taking a look at the df

```
df_ords_prods_all.head()
```

#3 creating a histogram for order hour of day

```
hist_orders_hour_of_day =
df_ords_prods_all['order_hour_of_day'].plot.hist(bins = 24)
```

 # 4 bar chart for loyalty flag customers

```
bar_loyalty_flag =
df_ords_prods_all['loyalty_flag'].value_counts().plot.bar()
```

Check whether there's a difference in expenditure (the "prices" column) depending on the hour of the day. (Hint: To check this, you need to use an accurate sample for your line chart!)

#5 making a accurate subset for making a line chart for prices and order hour of day

```
np.random.seed(4)

dev = np.random.rand(len(df_ords_prods_all)) <= 0.7

big = df_ords_prods_all[dev]

# big is 70% of DF and small is 30%

small = df_ords_prods_all[~dev]

# checking that the big and small data set = the same number

len(big)+len (small)

#making the small df only contain 2 columns and renaming df_2

df_2 = small[['order_hour_of_day','prices']]

#5 making a line plot with the small df

line_hour_price = sns.lineplot(data = df_2, x = 'order_hour_of_day',y
= 'prices')
```

Now that you have information about customers, you need to conduct some exploratory analysis of customer demographics to inform the targeted marketing campaigns. First, determine whether there's a

connection between age and family situation by creating a line chart exploring the connections between age and number of dependents:

#looking at the column names

df_ords_prods_all.info()

6 bar chart for marital status

bar_marital =
df_ords_prods_all['marital_status'].value_counts().plot.bar()

#6 making age group column for all ages

df_ords_prods_all.loc[df_ords_prods_all['age'] >= 90, 'age_group'] =
'90+'

df_ords_prods_all.loc[(df_ords_prods_all['age'] <= 89) &
(df_ords_prods_all['age'] >= 80), 'age group'] = '80-89'

df_ords_prods_all.loc[(df_ords_prods_all['age'] <= 79) &
(df_ords_prods_all['age'] >= 70), 'age_bracket'] = '70-79'

…….

till 10-19

#bar chart on customers age group

bar_age_bracket =
df_ords_prods_all['age_group'].value_counts().plot.bar()

By observing Bar charts we will get to know which age group are more Instacart customers.

#making a new df with just 2 columns in the small subset

df_3 = small[['number_of_dependants','age']]

#6 line chart exploring age and number of dependants

line_age_dependants = sns.lineplot(data = df_3, x = 'age',y = 'number_of_dependants')

By seeing the linechart will get to know the relationship age and number of dependants.

#7 scatterplot of age and income

scatter_age_income = sns.scatterplot(x = 'age', y = 'income',data = df_ords_prods_all)

#7 scatterplot of age and income

scatter_age_bracket_income = sns.scatterplot(x = 'age_bracket', y = 'income',data = df_ords_prods_all)

By observing scattered plot shows that which age group is having highest income.

#8 saving all visualizations in Jupyter folder

hist_orders_hour_of_day.figure.savefig(os.path.join(path, '04 Analysis','Visualizations', 'hist_orders_hour_of_day.png'))

Like the above example we need to save all graphs to visualizations folder .