Data Analysis with Python

E-COMMERCE PURCHASE CASE STUDY

Import the file

```
import pandas as pd
import zipfile

# Unzipping and loading the dataset from Kaggle
with zipfile.ZipFile('C:\\Users\\HP\\Downloads\\Ecommerce Purchases (1).zip', 'r') as zip_ref:
    zip_ref.extractall('Downloads//')

# Load the CSV file after extracting
df = pd.read_csv('C:\\Users\\HP\\Downloads\\Ecommerce Purchases (1).zip')

# Previewing the data
df.head()
```

	Address	Lot	AM or PM	Browser Info	Company	Credit Card	CC Exp Date	CC Security Code	CC Provider	Email	Job	IP Addres
0	16629 Pace Camp Apt. 448\nAlexisborough, NE 77...	46 in	PM	Opera/9.56. (X11; Linux x86_64; sl-SI) Presto/2...	Martinez-Herman	6011929061123406	02/20	900	JCB 16 digit	pdunlap@yahoo.com	Scientist, product/process development	149.146.147.20
	9374 Jasmine Spurs			Opera/8.93. (Windows	Fletcher,							

Problem

DIsPLAY TOP 10 ROWS OF THE DATA SET

```
df.head(10)
```

	Address	Lot	AM or PM	Browser Info	Company	Credit Card	CC Exp Date	CC Security Code	CC Provider
0	16629 Pace Camp Apt. 448\nAlexisborough, NE 77...	46 in	PM	Opera/9.56.(X11; Linux x86_64; sl-SI) Presto/2...	Martinez-Herman	6011929061123406	02/20	900	JCB 16 digit
1	9374 Jasmine Spurs Suite 508\nSouth John, TN 8...	28 rn	PM	Opera/8.93.(Windows 98; Win 9x 4.90; en-US) Pr...	Fletcher, Richards and Whitaker	3337758169645356	11/18	561	Mastercard

Problem

check last 10 rows of the data set

```
df.tail(10)
```

	Address	Lot	AM or PM	Browser Info	Company	Credit Card
9990	75731 Molly Springs\nWest Danielle, VT 96934-5102	93 ty	PM	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_4;...	Pace, Vazquez and Richards	869968197049750
9991	PSC 8165, Box 8498\nAPO AP 60327-0346	50 dA	AM	Mozilla/5.0 (compatible; MSIE 8.0; Windows NT	Snyder Inc	4221582137197481

Problem

check data type of each column.

```
df.dtypes
```

```
Address                object
Lot                    object
AM or PM               object
Browser Info           object
Company                object
Credit Card             int64
CC Exp Date            object
CC Security Code        int64
CC Provider            object
Email                  object
Job                    object
IP Address             object
Language               object
Purchase Price        float64
dtype: object
```

Problem

4.check null values in the data set

```
df.isnull().sum()
```

```
Address                0
Lot                    0
AM or PM               0
Browser Info           0
Company                0
Credit Card            0
CC Exp Date            0
CC Security Code       0
```

Problem

how many rows and columns are there in our data set

```
len(df.columns)
```

```
14
```

```
len(df)
```

```
10000
```

Problem

Highest and Lowest purchase price

```
df.columns
```

```
Index(['Address', 'Lot', 'AM or PM', 'Browser Info', 'Company', 'Credit Card',
       'CC Exp Date', 'CC Security Code', 'CC Provider', 'Email', 'Job',
       'IP Address', 'Language', 'Purchase Price'],
      dtype='object')
```

```
df['Purchase Price'].max()
```

```
np.float64(99.99)
```

```
df['Purchase Price'].min()
```

```
np.float64(0.0)
```

Problem

Average purchase price

```
df['Purchase Price'].mean()
```

```
np.float64(50.347302)
```

Problem

How many people have french 'fr' as their language

```
df.columns
```

```
Index(['Address', 'Lot', 'AM or PM', 'Browser Info', 'Company', 'Credit Card',
       'CC Exp Date', 'CC Security Code', 'CC Provider', 'Email', 'Job',
       'IP Address', 'Language', 'Purchase Price'],
      dtype='object')
```

```
len(df[df['Language']=='fr'])
```

```
1097
```

Problem

Job title contains engineer

```
df[df['Job'].str.contains('engineer',case=False)]
```

d	CC Exp Date	CC Security Code	CC Provider	Email	Job	IP Address	La
6	11/18	561	Mastercard	anthony41@reed.com	Drilling engineer	15.160.41.51	

Problem

find email of the person with the following ip address: 132.207.160.22

```
df.columns
```

```
Index(['Address', 'Lot', 'AM or PM', 'Browser Info', 'Company', 'Credit Card',
       'CC Exp Date', 'CC Security Code', 'CC Provider', 'Email', 'Job',
       'IP Address', 'Language', 'Purchase Price'],
      dtype='object')
```

```
df[df['IP Address']=='132.207.160.22']['Email']
```

```
2    amymiller@morales-harrison.com
Name: Email, dtype: object
```

Problem

How many people have mastercard as their credit card provider and a purchase above 50

```
df.columns
```

```
Index(['Address', 'Lot', 'AM or PM', 'Browser Info', 'Company', 'Credit Card',
       'CC Exp Date', 'CC Security Code', 'CC Provider', 'Email', 'Job',
       'IP Address', 'Language', 'Purchase Price'],
      dtype='object')
```

```
len(df[(df['CC Provider']=='Mastercard') & (df['Purchase Price']>50)])
```

```
405
```

Problem

Find email of the person with the following credit number- 4664825258997302

```
df.columns
```

```
Index(['Address', 'Lot', 'AM or PM', 'Browser Info', 'Company', 'Credit Card',
       'CC Exp Date', 'CC Security Code', 'CC Provider', 'Email', 'Job',
       'IP Address', 'Language', 'Purchase Price'],
      dtype='object')
```

```
df[df['Credit Card']==4664825258997302]['Email']
```

```
9992    bberry@wright.net
Name: Email, dtype: object
```

Problem

How many people purchase during 'am' and how many people purchase during 'pm'

```
df.columns
```

```
Index(['Address', 'Lot', 'AM or PM', 'Browser Info', 'Company', 'Credit Card',
       'CC Exp Date', 'CC Security Code', 'CC Provider', 'Email', 'Job',
       'IP Address', 'Language', 'Purchase Price'],
      dtype='object')
```

```
df['AM or PM'].value_counts()
```

```
AM or PM
PM    5068
AM    4932
Name: count, dtype: int64
```

Problem

How many people have credit card that expires in 2020

```
df.columns
```

```
Index(['Address', 'Lot', 'AM or PM', 'Browser Info', 'Company', 'Credit Card',
       'CC Exp Date', 'CC Security Code', 'CC Provider', 'Email', 'Job',
       'IP Address', 'Language', 'Purchase Price'],
      dtype='object')
```

```
len(df[df['CC Exp Date'].apply(lambda x:x[3:]=='20')])
```

```
988
```

Problem

Top 5 most popular in email providers (e.g, gmail.com,yahoo.com etc)

```
list1=[]
for email in df['Email']:
    list1.append(email.split('@')[1])
```

```
df['temp']=list1
```

```
df['temp'].value_counts().head()
```

```
temp
hotmail.com     1638
yahoo.com       1616
gmail.com       1605
smith.com         42
williams.com      37
Name: count, dtype: int64
```

created by

Arpita Chakroborty

arpitachakroborty.official@gmail.com