Machine Learning for Finance Homework 3

In this homework, we will focus on

1. Sentiment Analysis using NLP
2. Clustering with KMeans++, Inertia, PCA and short EDA
3. Individual project proposal

1. Pre-trained model: FinBERT

Your hedge fund manager wakes up every morning at 4am and wants a concise summary of the news relevant to his portfolio. He/She wants a quick summary showing the headlines of the news as well as classified sentiment scores.

Furthermore, he/she asks you to aggregate this information per stock and summarize the overall tone for this stock

1.1 Install the transformers package

In [1]:
```
pip install transformers
```
executed in 2.43s, finished 23:57:04 2022-03-23

```
Requirement already satisfied: transformers in c:\users\tianj\anaconda3\lib\site-packag
es (4.17.0)
Requirement already satisfied: requests in c:\users\tianj\anaconda3\lib\site-packages
(from transformers) (2.27.1)
Requirement already satisfied: huggingface-hub<1.0,>=0.1.0 in c:\users\tianj\anaconda3
\lib\site-packages (from transformers) (0.4.0)
Requirement already satisfied: packaging>=20.0 in c:\users\tianj\anaconda3\lib\site-pac
kages (from transformers) (21.3)
Requirement already satisfied: tokenizers!=0.11.3,>=0.11.1 in c:\users\tianj\anaconda3
\lib\site-packages (from transformers) (0.11.6)
Requirement already satisfied: sacremoses in c:\users\tianj\anaconda3\lib\site-packages
(from transformers) (0.0.49)
Requirement already satisfied: pyyaml>=5.1 in c:\users\tianj\anaconda3\lib\site-package
s (from transformers) (6.0)
Requirement already satisfied: filelock in c:\users\tianj\anaconda3\lib\site-packages
(from transformers) (3.4.2)
Requirement already satisfied: regex!=2019.12.17 in c:\users\tianj\anaconda3\lib\site-p
ackages (from transformers) (2021.8.3)
Requirement already satisfied: numpy>=1.17 in c:\users\tianj\anaconda3\lib\site-package
s (from transformers) (1.20.3)
Requirement already satisfied: tqdm>=4.27 in c:\users\tianj\anaconda3\lib\site-packages
(from transformers) (4.62.3)
Requirement already satisfied: typing-extensions>=3.7.4.3 in c:\users\tianj\anaconda3\l
ib\site-packages (from huggingface-hub<1.0,>=0.1.0->transformers) (3.10.0.2)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in c:\users\tianj\anaconda3\lib
\site-packages (from packaging>=20.0->transformers) (3.0.4)
Requirement already satisfied: colorama in c:\users\tianj\anaconda3\lib\site-packages
(from tqdm>=4.27->transformers) (0.4.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\tianj\anaconda3\lib\site-packag
es (from requests->transformers) (3.3)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\tianj\anaconda3\lib\si
te-packages (from requests->transformers) (1.26.8)
Requirement already satisfied: charset-normalizer~=2.0.0 in c:\users\tianj\anaconda3\li
b\site-packages (from requests->transformers) (2.0.4)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\tianj\anaconda3\lib\site-
packages (from requests->transformers) (2021.10.8)
Requirement already satisfied: six in c:\users\tianj\anaconda3\lib\site-packages (from
sacremoses->transformers) (1.16.0)
Requirement already satisfied: click in c:\users\tianj\anaconda3\lib\site-packages (fro
m sacremoses->transformers) (8.0.4)
Requirement already satisfied: joblib in c:\users\tianj\anaconda3\lib\site-packages (fr
om sacremoses->transformers) (1.1.0)
Note: you may need to restart the kernel to use updated packages.
```

1.2 import the requests package and other libraries you might need

In [2]:
```
import pandas as pd
import requests
```
executed in 464ms, finished 23:57:05 2022-03-23

The following dictionary represents our portfolio of stocks using the following logic (key, value) =

(holding-name, branch), where
$branch \in \{" business "," entertainment "," general "," health "," science "," sports "," te$
following the NewsAPI documentation.

To save you some time, a typical request is built similar to: "https://newsapi.org/v2/top-headlines?country=us&q=*KEYWORD*&category=*CATEGORY*&sortBy=top&apiKey=*YOURKEYHERE*" (https://newsapi.org/v2/top-headlines?country=us&q=*KEYWORD*&category=*CATEGORY*&sortBy=top&apiKey=*YOURKEYHERE*").

For example:

https://newsapi.org/v2/top-headlines?country=us&q=apple&category=technology&sortBy=top&apiKey=123ab45c687d (https://newsapi.org/v2/top-headlines?country=us&q=apple&category=technology&sortBy=top&apiKey=123ab45c687d)...

Please see:

1. https://newsapi.org/docs (https://newsapi.org/docs) (To construct your HTTP GET-requests)
2. https://newsapi.org/register (https://newsapi.org/register) (To obtain your API key)

In [3]:
```
portfolio = dict({"apple": 'technology', "tesla": 'business', "amazon": 'technology', "s&p

# we are going to use the S&P500 to get a general idea of the sentiment of news in the US r
```
executed in 14ms, finished 23:57:05 2022-03-23

1.3 Write the function fetch_news() that returns a dictionary that stores the name of the holding as key (analogous to our stock portfolio) and as value an array that holds the strings of news

In [4]:
```
# This is my api_key
api_key = "1fa2671224264342a6af4a9cc19c8eb8"

def fetch_news():
    result = {}
    for keyword in portfolio.keys():
        request_str = "https://newsapi.org/v2/top-headlines?country=us&q=" + \
                        keyword + "&category=" + portfolio.get(keyword) + \
                        "&sortBy=top&apiKey=" + api_key
        result[keyword] = [dic.get("title") for dic in requests.get(request_str).json().ge
    return result
```
executed in 14ms, finished 23:57:05 2022-03-23

```
In [5]: fetch_news()
```

executed in 278ms, finished 23:57:05 2022-03-23

```
Out[5]: {'apple': ["Apple's digital car keys now work with some Hyundai vehicles - Engadget",
  'Larger 15-Inch MacBook Air Expected in 2023 - MacRumors',
  'Deals: 24-inch M1 iMac Amazon lows, $500 off 16-inch M1 Pro MacBook Pro, more - 9to5
Mac',
  'Detailed iPhone 14 Pro/14 Pro Max leak highlights key design revisions - PhoneAren
a',
  "Apple executives say creating Mac Studio was 'overwhelming' - AppleInsider",
  'Apple responds to iOS 15.4 battery drain complaints - AppleInsider',
  'iPhone 15 Pro Rumored to Feature Under-Screen Face ID System From Samsung - MacRumor
s',
  'Apple services including App Store resume after outage for second straight day - Reu
ters.com',
  'New iPhone SE Nearly as Tough as iPhone 13 in Drop Test Thanks to Improved Glass - M
acRumors'],
 'tesla': ['Tesla, Lucid supplier LGES plans to build $1.4 bln battery factory in Arizo
na - Reuters',
  'Subaru Owner Whose Car Was Destroyed by Jumping Tesla Meets GoFundMe Goal - The Driv
e',
  'Tesla Faces Backlash Over Musk Plan for Texas Gigafactory Party - Bloomberg',
  "Tesla's new factory in Germany will reduce reliance on China, says Canaccord's analy
st Dorsheimer - CNBC Television"],
 'amazon': ['Today' s best deals: Amazon Kindles and Fire HD tablets, LG OLED TVs, and
more - Ars Technica',
  'Deals: 24-inch M1 iMac Amazon lows, $500 off 16-inch M1 Pro MacBook Pro, more - 9to5
Mac',
  "Psst...Amazon has a secret coupon page, and today's deals are amazing - Yahoo Lif
e"],
 's&p500': ['Blood pressure medication recalled over cancer risk concerns - CNN',
  "Dow Jones Futures: Stock Market Rally Retreats; 'Monster' Apple In Buy Area - Invest
or's Business Daily",
  'Ethereum price breaks through $3K, but analysts warn that a retest is needed - Coint
elegraph',
  "Cramer's lightning round: I'm not holding my breath for Robinhood - CNBC",
  'First drive, Nissan Ariya: the stable relationship - Electrek.co',
  "Jim Cramer tells investors why they shouldn't despair after Wednesday's market decli
nes - CNBC Television",
  "'Meme stock' rally redux? GameStop, AMC shares rocket higher - Reuters",
  'Hy-Vee employees report layoffs - KCCI Des Moines',
  "US Senate to Consider Bill Examining El Salvador's Bitcoin Experiment - CoinDesk",
  'Russian Stock Market to Partially Reopen on Thursday - The Wall Street Journal',
  'Tesla, Lucid supplier LGES plans to build $1.4 bln battery factory in Arizona - Reut
ers',
  'Moderna to Seek Authorization of Its Coronavirus Vaccine for Young Children - The Ne
w York Times',
  "Renault suspends Russian business after Zelensky's speech • FRANCE 24 English - FRAN
CE 24 English",
  'Electric vehicle start-up Nikola has begun production of its first battery-electric
semitruck - CNBC',
  'Who stands to make and lose money if the SEC climate rule becomes law - CNBC',
  'Oregon Dollar Tree store exposed workers to 'serious physical harm,' OSHA says - O
regonLive',
  'Wall Street bonuses hit new record - Fox Business',
  'Exclusive: Clients plead with top custodian banks to stay in Russia - Reuters.com',
  'Okta says hundreds of companies impacted by security breach - TechCrunch',
  'Oil climbs 5% after Russia warns of prolonged pipeline outage - CNN']}
```

1.4 Following the lecture notes, using the pre-trained Finbert Classifier, classify the news fetched in 1.3 into neutral, positive or negative by modifying the below code:

1.5 Last but not least, find the total tone for each element in our portfolio, where:

- neutral=0
- positive=+1
- negative=-1

In [6]:

```
# BUILD YOUR CODE ON TOP OF THIS EXAMPLE CODE IN THE CELL BELOW
from transformers import BertTokenizer, BertForSequenceClassification
import numpy as np

finbert = BertForSequenceClassification.from_pretrained('yiyanghkust/finbert-tone',num_lab
tokenizer = BertTokenizer.from_pretrained('yiyanghkust/finbert-tone')

sentences = ["there is a shortage of capital, and we need extra financing",
             "growth is strong and we have plenty of liquidity",
             "there are doubts about our finances",
             "profits are flat"]

inputs = tokenizer(sentences, return_tensors="pt", padding=True)
outputs = finbert(**inputs)[0]

labels = {0:'neutral', 1:'positive',2:'negative'}
for idx, sent in enumerate(sentences):
    print(sent, '----', labels[np.argmax(outputs.detach().numpy()[idx])])

'''
there is a shortage of capital, and we need extra financing ---- negative
growth is strong and we have plenty of liquidity ---- positive
there are doubts about our finances ---- negative
profits are flat ---- neutral
'''
```

executed in 2.00s, finished 23:57:07 2022-03-23

```
there is a shortage of capital, and we need extra financing ---- negative
growth is strong and we have plenty of liquidity ---- positive
there are doubts about our finances ---- negative
profits are flat ---- neutral
```

Out[6]:

```
'\nthere is a shortage of capital, and we need extra financing ---- negative\ngrowth is
strong and we have plenty of liquidity ---- positive\nthere are doubts about our financ
es ---- negative\nprofits are flat ---- neutral\n'
```

In [7]:
```
# Your code here
import re

# Define a function to remove the author name at the end of each title.
def parseSentence(sen: str) -> str:
    return re.split("[\s]-[\s]", sen)[0]

classified_news = {}
finbert = BertForSequenceClassification.from_pretrained('yiyanghkust/finbert-tone',
                                                        num_labels=3)
tokenizer = BertTokenizer.from_pretrained('yiyanghkust/finbert-tone')
labels = {0:'neutral', 1:'positive', 2:'negative'}

for company in fetch_news().keys():
    classified_news[company] = []
    sentences = list(map(parseSentence, fetch_news().get(company)))
    if not sentences:
        continue
    else:
        inputs = tokenizer(sentences, return_tensors="pt", padding=True)
        outputs = finbert(**inputs)[0]
        for idx, sent in enumerate(sentences):
            classified_news[company].append(sent + '----' + \
                                labels[np.argmax(outputs.detach().numpy()[idx])])
classified_news
```

executed in 3.66s, finished 23:57:11 2022-03-23

Out[7]:
```
{'apple': ["Apple's digital car keys now work with some Hyundai vehicles----neutra
l",
  'Larger 15-Inch MacBook Air Expected in 2023----neutral',
  'Deals: 24-inch M1 iMac Amazon lows, $500 off 16-inch M1 Pro MacBook Pro, more----
neutral',
  'Detailed iPhone 14 Pro/14 Pro Max leak highlights key design revisions----neutra
l',
  "Apple executives say creating Mac Studio was 'overwhelming'----neutral",
  'Apple responds to iOS 15.4 battery drain complaints----neutral',
  'iPhone 15 Pro Rumored to Feature Under-Screen Face ID System From Samsung----neut
ral',
  'Apple services including App Store resume after outage for second straight day---
-neutral',
  'New iPhone SE Nearly as Tough as iPhone 13 in Drop Test Thanks to Improved Glass-
---positive'],
 'tesla': ['Tesla, Lucid supplier LGES plans to build $1.4 bln battery factory in Ar
izona----neutral',
  'Subaru Owner Whose Car Was Destroyed by Jumping Tesla Meets GoFundMe Goal----neut
ral',
  'Tesla Faces Backlash Over Musk Plan for Texas Gigafactory Party----neutral',
  "Tesla's new factory in Germany will reduce reliance on China, says Canaccord's an
alyst Dorsheimer----neutral"],
 'amazon': ['Today' s best deals: Amazon Kindles and Fire HD tablets, LG OLED TVs, a
nd more----positive',
  'Deals: 24-inch M1 iMac Amazon lows, $500 off 16-inch M1 Pro MacBook Pro, more----
neutral',
  "Psst...Amazon has a secret coupon page, and today's deals are amazing----positiv
e"],
 's&p500': ['Blood pressure medication recalled over cancer risk concerns----negativ
e',
  "Dow Jones Futures: Stock Market Rally Retreats; 'Monster' Apple In Buy Area----po
sitive",
  'Ethereum price breaks through $3K, but analysts warn that a retest is needed----n
egative',
  "Cramer's lightning round: I'm not holding my breath for Robinhood----neutral",
  'First drive, Nissan Ariya: the stable relationship----neutral',
  "Jim Cramer tells investors why they shouldn't despair after Wednesday's market de
clines----neutral",
  "'Meme stock' rally redux? GameStop, AMC shares rocket higher----positive",
  'Hy-Vee employees report layoffs----neutral',
  "US Senate to Consider Bill Examining El Salvador's Bitcoin Experiment----neutra
l",
  'Russian Stock Market to Partially Reopen on Thursday----neutral',
  'Tesla, Lucid supplier LGES plans to build $1.4 bln battery factory in Arizona----
neutral',
  'Moderna to Seek Authorization of Its Coronavirus Vaccine for Young Children----ne
utral',
  "Renault suspends Russian business after Zelensky's speech • FRANCE 24 English----
neutral",
  'Electric vehicle start-up Nikola has begun production of its first battery-electr
ic semitruck----neutral',
  'Who stands to make and lose money if the SEC climate rule becomes law----neutra
l',
  'Oregon Dollar Tree store exposed workers to ‘serious physical harm,’ OSHA says-
---negative',
  'Wall Street bonuses hit new record----positive',
  'Exclusive: Clients plead with top custodian banks to stay in Russia----neutral',
  'Okta says hundreds of companies impacted by security breach----neutral',
  'Oil climbs 5% after Russia warns of prolonged pipeline outage----positive']}
```

In [8]:
```python
# Your code here
tone_dic = {"neutral": 0, "positive": 1, "negative": -1}
total_tone = 0
for company in classified_news.keys():
    for sentence in classified_news.get(company):
        total_tone += tone_dic.get(re.split("----", sentence)[1])
total_tone
```

executed in 14ms, finished 23:57:11 2022-03-23

Out[8]:
```
4
```

2. K-Means++ Clustering with PCA

To make it easier for you, for the analysis important EDA has already been performed. However, it is your task to interpret

2.1 Import necessary libraries...

In [9]:
```
import pandas as pd
import numpy as np
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler

from sklearn.cluster import KMeans
import seaborn as sns

import matplotlib.pyplot as plt
from matplotlib.pyplot import figure
```

executed in 1.16s, finished 23:57:12 2022-03-23

2.2 Read the credit card data into a dataframe

In [10]:
```
cc_holders = pd.read_csv("credit_card.csv")
```

executed in 28ms, finished 23:57:12 2022-03-23

2.3 Show a snippet of the dataframe

In [11]:
```
cc_holders
```

executed in 29ms, finished 23:57:12 2022-03-23

Out[11]:

	CUST_ID	BALANCE	BALANCE_FREQUENCY	PURCHASES	ONEOFF_PURCHASES
0	C10001	40.900749	0.818182	95.40	0.00
1	C10002	3202.467416	0.909091	0.00	0.00
2	C10003	2495.148862	1.000000	773.17	773.17
3	C10004	1666.670542	0.636364	1499.00	1499.00
4	C10005	817.714335	1.000000	16.00	16.00
...	...	...	...	...	...
8945	C19186	28.493517	1.000000	291.12	0.00
8946	C19187	19.183215	1.000000	300.00	0.00
8947	C19188	23.398673	0.833333	144.40	0.00
8948	C19189	13.457564	0.833333	0.00	0.00
8949	C19190	372.708075	0.666667	1093.25	1093.25

8950 rows × 18 columns

2.4 Gather information about missing data, data type and the dimensions of the dataset

In [12]:
```
cc_holders.info()
```
executed in 14ms, finished 23:57:12 2022-03-23

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8950 entries, 0 to 8949
Data columns (total 18 columns):
 #   Column                            Non-Null Count  Dtype  
---  ------                            --------------  -----  
 0   CUST_ID                           8950 non-null   object 
 1   BALANCE                           8950 non-null   float64
 2   BALANCE_FREQUENCY                 8950 non-null   float64
 3   PURCHASES                         8950 non-null   float64
 4   ONEOFF_PURCHASES                  8950 non-null   float64
 5   INSTALLMENTS_PURCHASES            8950 non-null   float64
 6   CASH_ADVANCE                      8950 non-null   float64
 7   PURCHASES_FREQUENCY               8950 non-null   float64
 8   ONEOFF_PURCHASES_FREQUENCY        8950 non-null   float64
 9   PURCHASES_INSTALLMENTS_FREQUENCY  8950 non-null   float64
 10  CASH_ADVANCE_FREQUENCY            8950 non-null   float64
 11  CASH_ADVANCE_TRX                  8950 non-null   int64  
 12  PURCHASES_TRX                     8950 non-null   int64  
 13  CREDIT_LIMIT                      8949 non-null   float64
 14  PAYMENTS                          8950 non-null   float64
 15  MINIMUM_PAYMENTS                  8637 non-null   float64
 16  PRC_FULL_PAYMENT                  8950 non-null   float64
 17  TENURE                            8950 non-null   int64  
dtypes: float64(14), int64(3), object(1)
memory usage: 1.2+ MB
```

2.5 Gather statistical measures about each column

In [13]:
```
cc_holders.describe().style.background_gradient(cmap = 'twilight')
```
executed in 92ms, finished 23:57:12 2022-03-23

Out[13]:

	BALANCE	BALANCE_FREQUENCY	PURCHASES	ONEOFF_PURCHASES	INSTALLMI
count	8950.000000	8950.000000	8950.000000	8950.000000	
mean	1564.474828	0.877271	1003.204834	592.437371	
std	2081.531879	0.236904	2136.634782	1659.887917	
min	0.000000	0.000000	0.000000	0.000000	
25%	128.281915	0.888889	39.635000	0.000000	
50%	873.385231	1.000000	361.280000	38.000000	
75%	2054.140036	1.000000	1110.130000	577.405000	
max	19043.138560	1.000000	49039.570000	40761.250000	

2.6 Examine the correlation value for each column with each other column

```
In  [14]: cc_holders.corr().style.background_gradient(cmap = 'RdYlGn')
```

executed in 30ms, finished 23:57:12 2022-03-23

Out[14]:

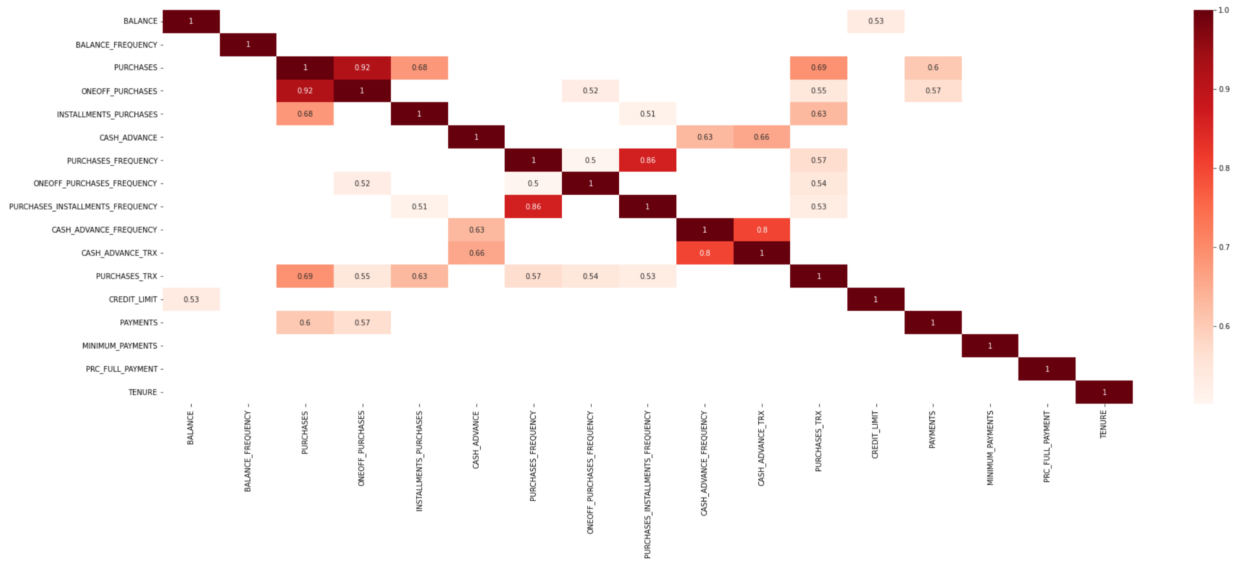
	BALANCE	BALANCE_FREQUENCY	PURCHASES
BALANCE	1.000000	0.322412	0.181261
BALANCE_FREQUENCY	0.322412	1.000000	0.133674
PURCHASES	0.181261	0.133674	1.000000
ONEOFF_PURCHASES	0.164350	0.104323	0.916845
INSTALLMENTS_PURCHASES	0.126469	0.124292	0.679896
CASH_ADVANCE	0.496692	0.099388	-0.051474
PURCHASES_FREQUENCY	-0.077944	0.229715	0.393017
ONEOFF_PURCHASES_FREQUENCY	0.073166	0.202415	0.498430
PURCHASES_INSTALLMENTS_FREQUENCY	-0.063186	0.176079	0.315567
CASH_ADVANCE_FREQUENCY	0.449218	0.191873	-0.120143
CASH_ADVANCE_TRX	0.385152	0.141555	-0.067175
PURCHASES_TRX	0.154338	0.189626	0.689561
CREDIT_LIMIT	0.531283	0.095843	0.356963
PAYMENTS	0.322802	0.065008	0.603264
MINIMUM_PAYMENTS	0.398684	0.132569	0.093860
PRC_FULL_PAYMENT	-0.318959	-0.095082	0.180379
TENURE	0.072692	0.119776	0.086288

2.7 Create a heatmap showing all positive or negative correlations higher than .5

In [15]:

```python
cc_corr = cc_holders.corr()

cc_corr_filt = cc_corr[((cc_corr >= .5) | (cc_corr <= -.5))]
plt.figure(figsize=(30,10))
sns.heatmap(cc_corr_filt, annot=True, cmap="Reds")
plt.show()
```

```
executed in 386ms, finished 23:57:13 2022-03-23
```



What do you observe? What conclusions can you draw from looking at the correlations, and how would that influence a principal components analysis routine? Why do we do Principal Components Analysis?

Answer

The heatmap directly shows us the pairs of variables which have a high correlation (with absolute value over 0.5) with each other. The darker cells means higher correlations, and indeed we should ignore the correlations with value 1 on the diagonal, since they are just correlations with themselves.

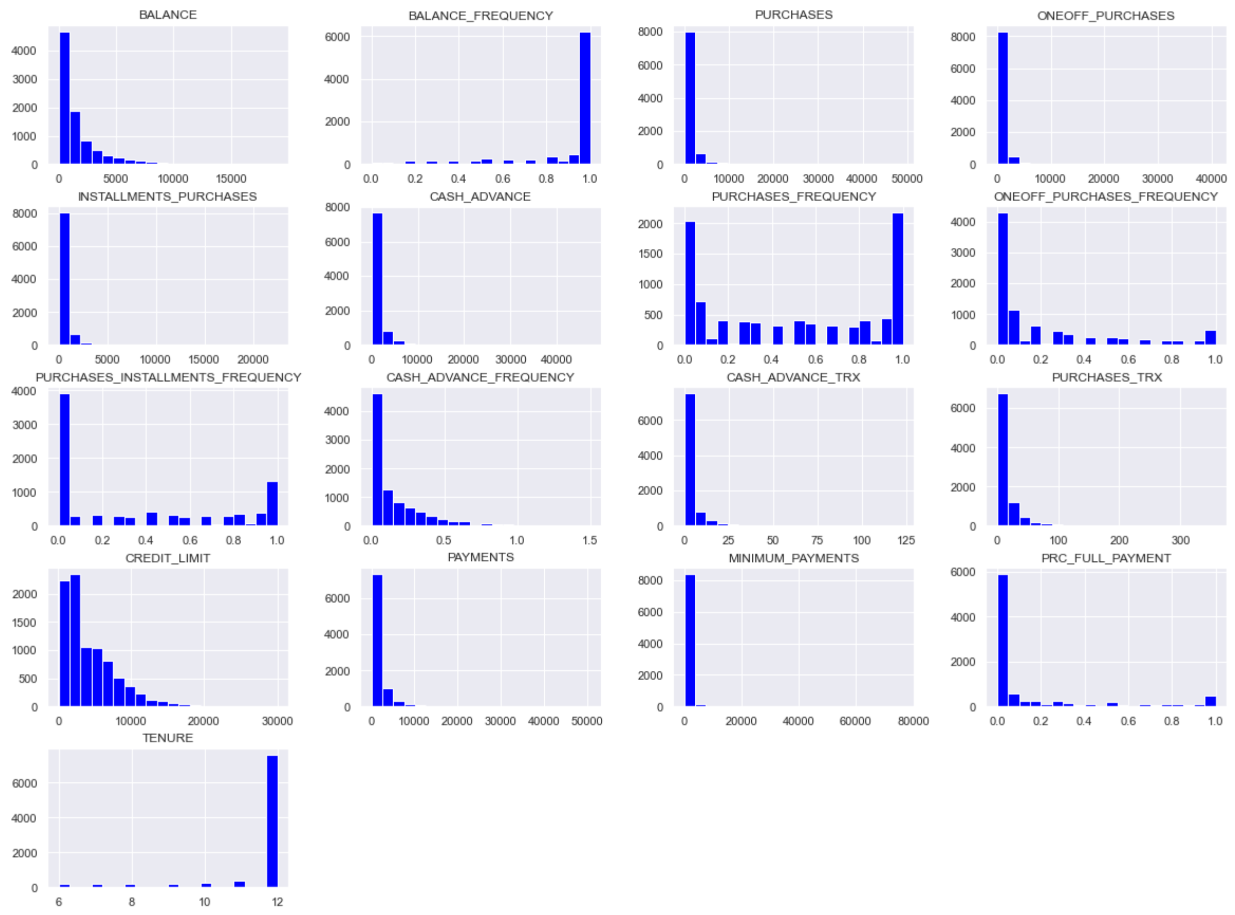
However, we can also observe some extremely high correlations, for instance, variable **PURCHASES** and **ONEOFF_PURCHASES** have a 0.92 correlation, and variable **PURCHASES_INSTALLMENTS_FREQUENCY** and **PURCHASES_FREQUENCY**. Besides, pairs such as **PURCHASES** and **PURCHASES_TRX**, **CASH_ADVANCE** and **CASH_ADVANCE_TRX** also have relatively high correlation over 0.65. Therefore, one important information we can get is that, for this data set, high correlations were more likely been found between the variables who have similar names, such as **PURCHASES** and **PURCHASES_TRX**. This characteristic may influence the number of components when performing PCA, since some variables are similar to each other hence may be regarded as exactly one component.

We do the PCA for reducing the dimensions in our model, or making our model simplier to reduce the possibility of overfitting and hence producing a high test error, since a high correlation represents that same information are covered by multiple variables simultaneously.

2.8 Examine the discrete distributions of each feature column

```
In [16]: sns.set(rc={'figure.figsize':(20, 15)})
         dists = cc_holders.hist(bins=20, color='blue')
```

executed in 1.63s, finished 23:57:14 2022-03-23



What can you conclude about the distributions of the variables in the data set? In 3-4 sentences, describe methods of handling outliers following the lecture notes.

2.9 *Your answer here*

For most of the variables shown in the tables above, their distributions are not balanced and symmetric, which means that they skew to one side.

We can take logarithms on the skewed data which makes the gap smaller. We can also binning or bucketing the data into some groups, that it, tranforming continuous data into categorical.

2.10 Examine the data set for missing values

```
In  [17]: cc_holders.isna().sum().sort_values(ascending=False)
```

executed in 14ms, finished 23:57:14 2022-03-23

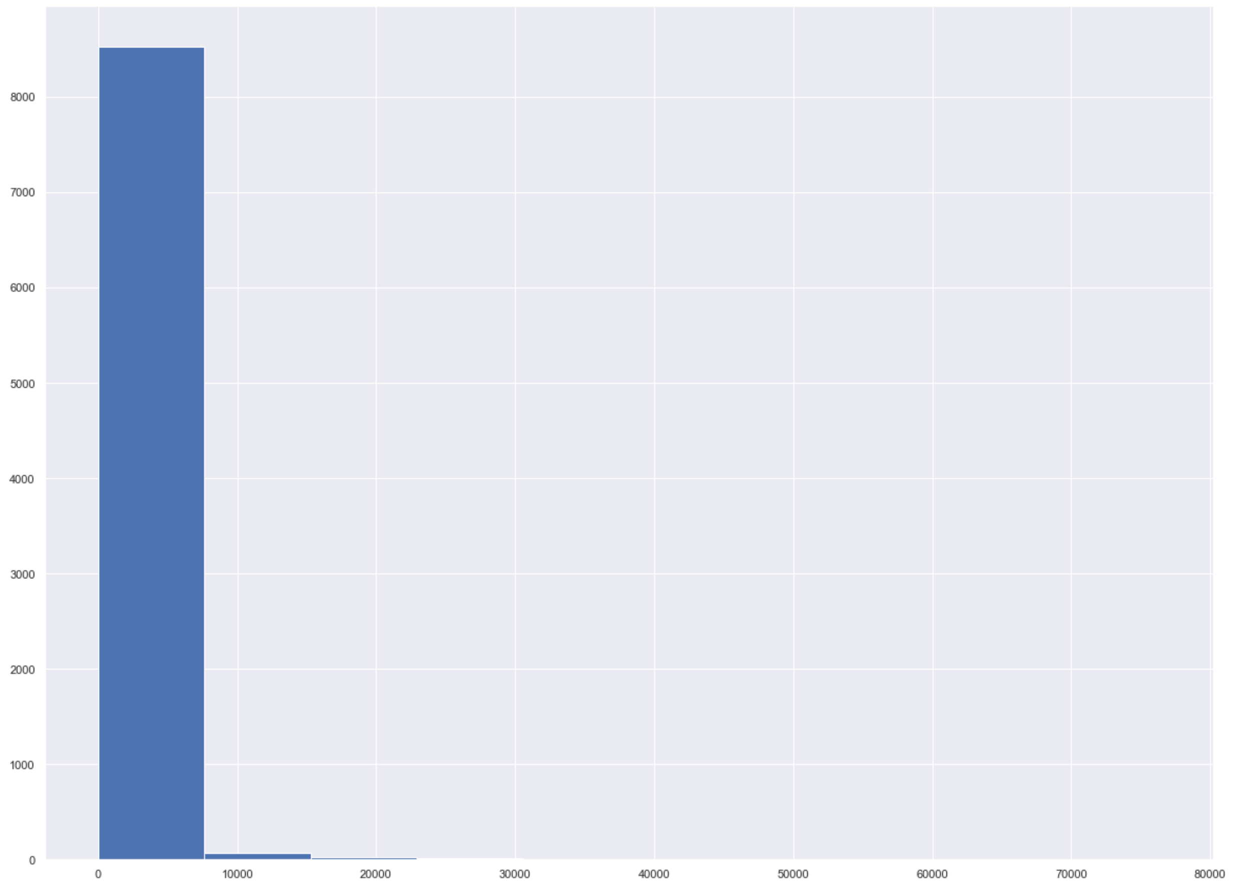
```
Out[17]: MINIMUM_PAYMENTS                    313
         CREDIT_LIMIT                          1
         CUST_ID                               0
         BALANCE                               0
         PRC_FULL_PAYMENT                      0
         PAYMENTS                              0
         PURCHASES_TRX                         0
         CASH_ADVANCE_TRX                      0
         CASH_ADVANCE_FREQUENCY                0
         PURCHASES_INSTALLMENTS_FREQUENCY      0
         ONEOFF_PURCHASES_FREQUENCY            0
         PURCHASES_FREQUENCY                   0
         CASH_ADVANCE                          0
         INSTALLMENTS_PURCHASES                0
         ONEOFF_PURCHASES                      0
         PURCHASES                             0
         BALANCE_FREQUENCY                     0
         TENURE                                0
         dtype: int64
```

2.11 Is it reasonable to fill missing values with the median? If so, fill the missing values with the median.

```
In  [18]: # Your code here
          print(cc_holders["MINIMUM_PAYMENTS"].isna().sum() / len(cc_holders["MINIMUM_PAYMENTS"]))
          plt.hist(cc_holders["MINIMUM_PAYMENTS"])
```

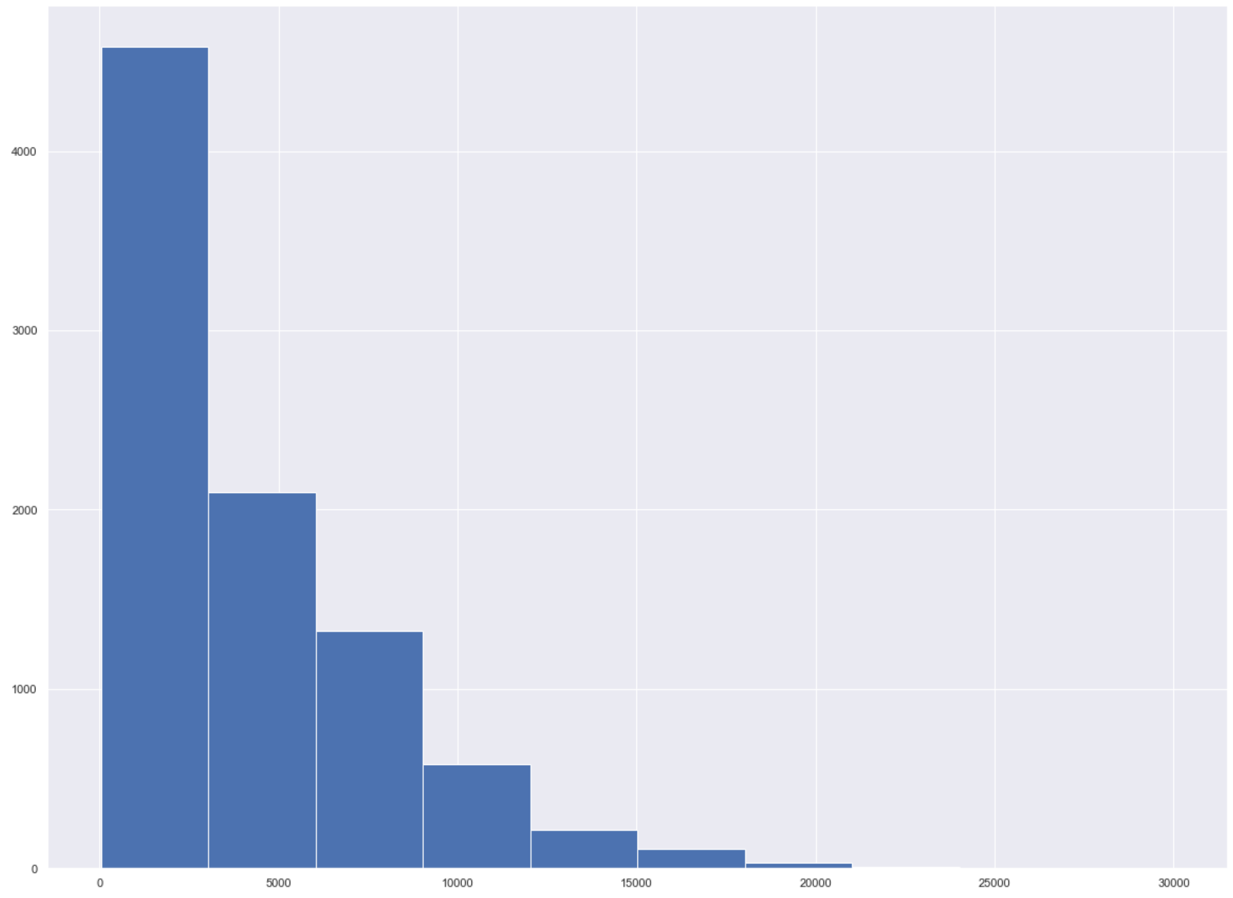
executed in 153ms, finished 23:57:14 2022-03-23

```
          0.034972067039106144

Out[18]: (array([8.532e+03, 6.500e+01, 2.100e+01, 1.100e+01, 1.000e+00, 3.000e+00,
                 1.000e+00, 2.000e+00, 0.000e+00, 1.000e+00]),
          array([1.91630000e-02, 7.64063800e+03, 1.52812568e+04, 2.29218757e+04,
                 3.05624945e+04, 3.82031133e+04, 4.58437322e+04, 5.34843510e+04,
                 6.11249698e+04, 6.87655887e+04, 7.64062075e+04]),
          <BarContainer object of 10 artists>)
```



In [19]:
```
plt.hist(cc_holders["CREDIT_LIMIT"])
```
executed in 138ms, finished 23:57:15 2022-03-23

Out[19]:
```
(array([4.583e+03, 2.095e+03, 1.326e+03, 5.780e+02, 2.150e+02, 1.090e+02,
        3.300e+01, 6.000e+00, 1.000e+00, 3.000e+00]),
 array([   50.,   3045.,   6040.,   9035.,  12030.,  15025.,  18020.,  21015.,
        24010.,  27005.,  30000.]),
 <BarContainer object of 10 artists>)
```



I think it is reasonable to replace the missing values by the medium, since the missing values are only taking a very small component in our sample, and the distribution of data seems to be concentrated as shown in the histogram above.

In [20]:
```
# Your code here
cc_holders["MINIMUM_PAYMENTS"].fillna(cc_holders["MINIMUM_PAYMENTS"].median(), inplace=Tru
cc_holders["CREDIT_LIMIT"].fillna(cc_holders["CREDIT_LIMIT"].median(), inplace=True)
```
executed in 14ms, finished 23:57:15 2022-03-23

2.12 Check if there are any missing values left

In [21]:
```
# Your code here
cc_holders.isna().sum().sum()
```
executed in 14ms, finished 23:57:15 2022-03-23

Out[21]:
```
0
```

2.13 Drop the CUST_ID column. And why do we do so?

Your answer here

Because the variable **CUST_ID** are full of strings which are irrelavant to our task and purpose.

In [22]:
```
# Your code here
del cc_holders["CUST_ID"]
```
executed in 14ms, finished 23:57:15 2022-03-23

2.14 Using the standard scaler, scale the data. Again, why do we do so?

Your answer here

Using the standard scaler to convert the data to a standard range from their natural range. It also helps the model converge faster, and help the model learn appropriate weights for each features.

```
In  [23]: # Your code here
          scaled_data = pd.DataFrame(StandardScaler().fit_transform(cc_holders))
          scaled_data.columns = cc_holders.columns
          scaled_data
```

executed in 30ms, finished 23:57:15 2022-03-23

Out[23]:

	BALANCE	BALANCE_FREQUENCY	PURCHASES	ONEOFF_PURCHASES	INSTALLME
0	-0.731989	-0.249434	-0.424900	-0.356934	
1	0.786961	0.134325	-0.469552	-0.356934	
2	0.447135	0.518084	-0.107668	0.108889	
3	0.049099	-1.016953	0.232058	0.546189	
4	-0.358775	0.518084	-0.462063	-0.347294	
...	...	...	...	...	
8945	-0.737950	0.518084	-0.333293	-0.356934	
8946	-0.742423	0.518084	-0.329136	-0.356934	
8947	-0.740398	-0.185477	-0.401965	-0.356934	
8948	-0.745174	-0.185477	-0.469552	-0.356934	
8949	-0.572575	-0.889033	0.042146	0.301732	

8950 rows × 17 columns

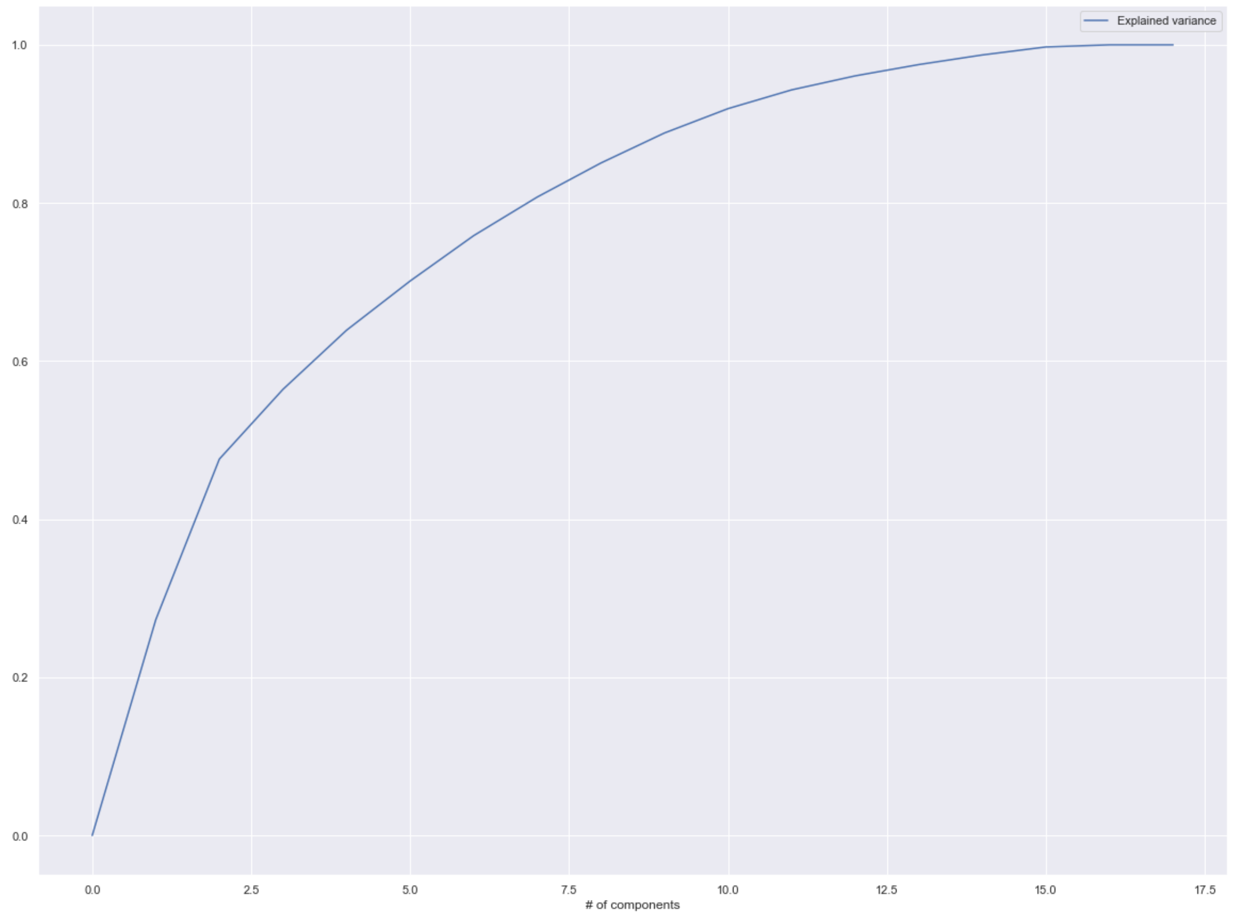
2.15 By doing PCA, naturally, we can explain more variance if we allow more components.

Plot the explained variance over the number of components and decide on a reasonable number of components.

```
In  [24]: # Your code here
explained_var = []
for num in range(cc_holders.shape[1] + 1):
    pca = PCA(num)
    pca.fit(scaled_data)
    explained_var.append([num, sum(pca.explained_variance_ratio_)])

pd.DataFrame(
    explained_var, columns=["# of components", "Explained variance"]).set_index("# of comp
```

executed in 527ms, finished 23:57:15 2022-03-23

```
Out[24]: <AxesSubplot:xlabel='# of components'>
```



The plot above shows that the proportions of variance the model can explain, by taking different number of components.

2.16 Perform PCA with your chosen number of components and print out the explained variances by each component. When looking back at our EDA, why do we need so many components?

Because we actually have many aspects to explain the distribution and statistics of our data, such as purchases and balance, and our model will fail if we cannot explain a sufficient proportion of variances.

In [25]:
```
# Your code here
pca = PCA(10)
pca.fit(scaled_data)
print(f"Variance explained: {pca.explained_variance_ratio_}")
```

executed in 30ms, finished 23:57:15 2022-03-23

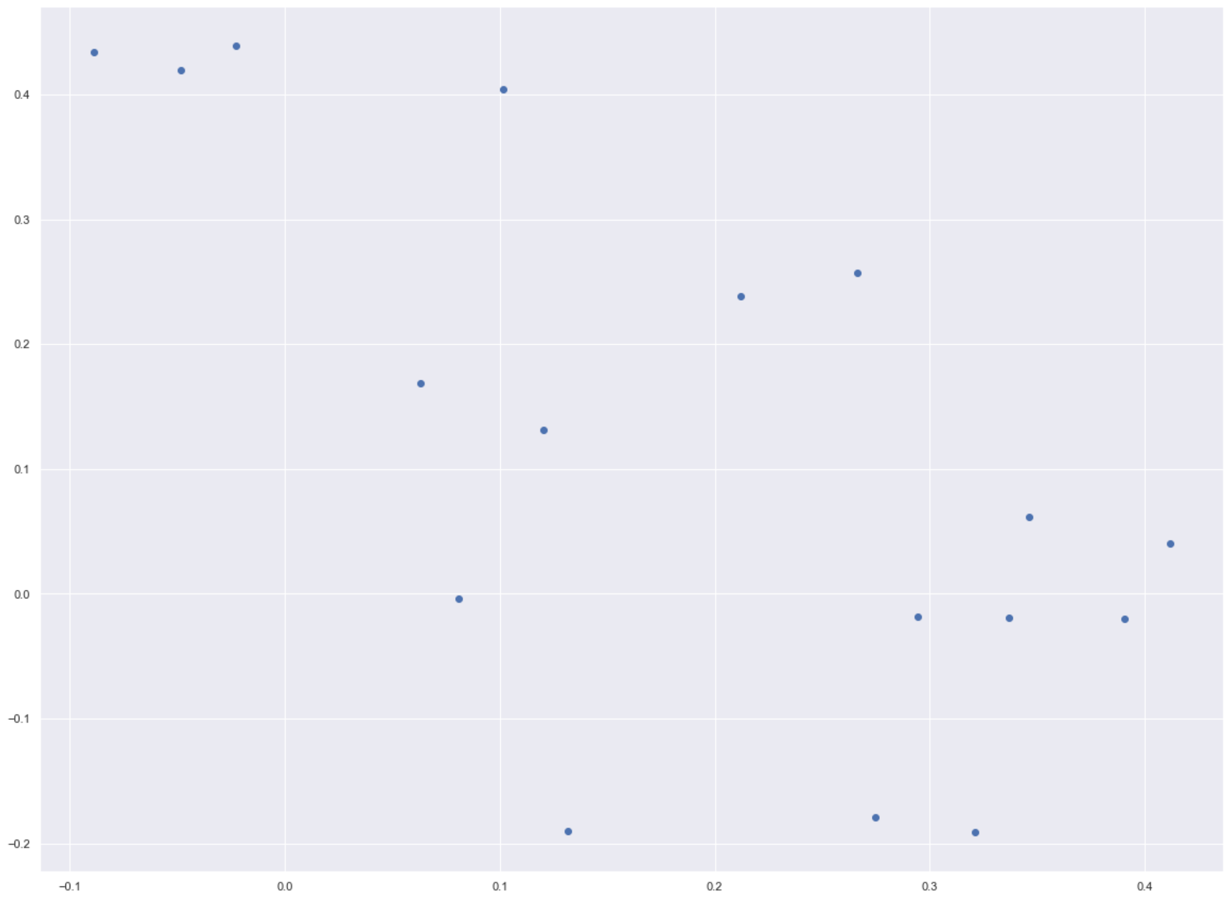
```
Variance explained: [0.27297671 0.2031378  0.08813182 0.07479524 0.06224729 0.05740056
 0.04883426 0.04299203 0.03798259 0.03080002]
```

2.17 Plot the observations following PC1 and PC2

```
In [26]: # Your code here
         pc = pd.DataFrame(pca.components_.T, columns = ["PC"+str(i) for i in range(1,11)])
         plt.scatter(pc["PC1"], pc["PC2"])
```

executed in 168ms, finished 23:57:15 2022-03-23

```
Out[26]: <matplotlib.collections.PathCollection at 0x1db92be3ac0>
```



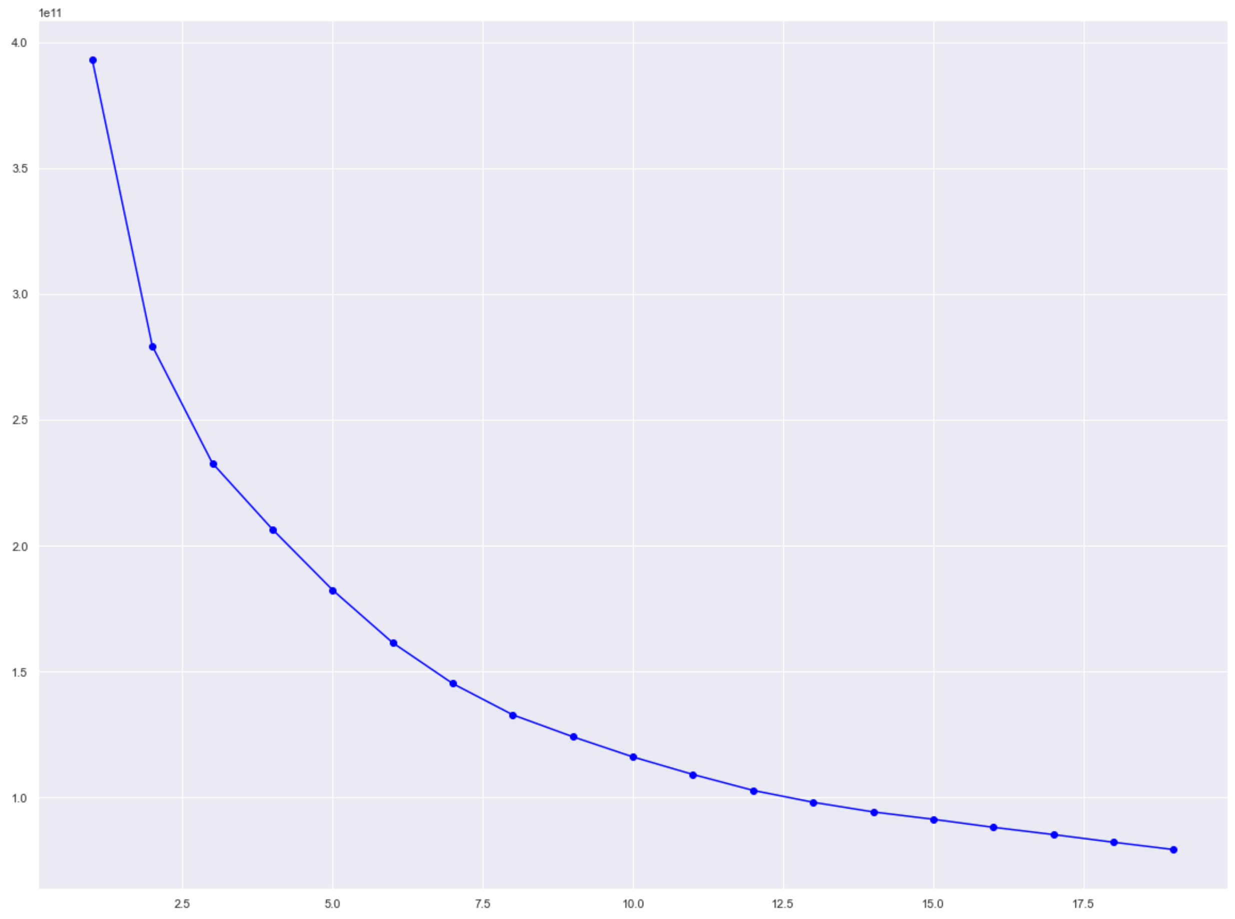
2.18 You have learnt in class, that a well-known measure to find a good number of clusters is Inertia. Following the lecture notes, plot the inertia graph for 1-19 clusters

```
In [27]: # Your code here
         res = []
         for num in range(1, 20):
             clf = KMeans(n_clusters=num)
             clf.fit(cc_holders)
             res.append([num, clf.inertia_])

         inertia_df = pd.DataFrame(res, columns=["# of clusters", "Inertia"]).set_index("# of clust
```

executed in 5.56s, finished 23:57:21 2022-03-23

In [28]:
```
# The code for the plot here
plt.figure()

plt.plot(inertia_df, "o", color = "blue")
plt.plot(inertia_df, color = "blue")
```
executed in 181ms, finished 23:57:21 2022-03-23

Out[28]:
```
[<matplotlib.lines.Line2D at 0x1db94841400>]
```



2.19 Using inertia, we have found the right number of clusters for our use case. Now fit the model and plot the outcome.

In [29]:
```
# Your code here
clf = KMeans(n_clusters=6)
```
executed in 14ms, finished 23:57:21 2022-03-23

In [30]:
```
# Your code here
clf.fit(cc_holders)
```
executed in 200ms, finished 23:57:21 2022-03-23

Out[30]:
```
KMeans(n_clusters=6)
```

In [31]:
```
# Print out the clusters
print(clf.labels_)
```
executed in 14ms, finished 23:57:21 2022-03-23

```
[1 5 5 ... 1 1 1]
```