UNIVERSITY OF SOUTH DAKOTA

Project 01

CSC585- Data Mining Methods

October 30 , 2021

Md Minhazur Rahman

ID: 101085958

MS in Physics-Analytics for Large Data Sets

Department of Physics,

University of South Dakota.

Codes for data extraction and transformation

In [49]:
```
import sqlite3 as sq
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [50]:
```
## Function for creating data frame from sqlite database
def make_df(table_name,*columns):
    conn = sq.connect('db.sqlite3')
    sql = 'select '
    sql += ','.join(columns) + ' from ' + table_name
    df = pd.read_sql_query(sql,conn)
    conn.close()
    df.columns = [i for i in columns]
    return df
```

In [51]:
```
qid_res = make_df('surveys_response','questionID_id','response')
qid_res.head()
```

Out[51]:

	questionID_id	response
0	16	5
1	17	3
2	18	3
3	19	1
4	20	5

In [52]:
```
qstn_txt = make_df('surveys_question_text','questionTextID','factorID_id','positive_p')
qstn_txt.shape
```

Out[52]: (118, 3)

In [53]:
```
surv = make_df( 'surveys_survey','surveyID', 'userID_id', 'creationDate','completionDat
surv.head()
```

Out[53]:

	surveyID	userID_id	creationDate	completionDate
0	33	5	2021-03-09 21:07:01	2021-03-10 03:10:08.933258
1	34	6	2021-03-09 21:07:01	2021-03-10 05:28:51.943376
2	35	7	2021-03-09 21:07:02	None
3	36	8	2021-03-09 21:07:02	None
4	37	9	2021-03-09 21:07:02	None

In [54]:
```
factor = make_df('surveys_factor','factorID', 'factorName', 'studyID_id')
factor.head()
```

Out[54]:

	factorID	factorName	studyID_id
0	1	Factor 1	1
1	2	Factor 2	1
2	3	Factor 3	1
3	4	Factor 4	1
4	5	Factor 5	1

In [55]:
```
qstn = make_df('surveys_question','questionID' , 'questionTextID_id' , 'surveyID_id')
factor.head()
```

Out[55]:

	factorID	factorName	studyID_id
0	1	Factor 1	1
1	2	Factor 2	1
2	3	Factor 3	1
3	4	Factor 4	1
4	5	Factor 5	1

In [56]:
```
user = make_df('surveys_user','userID', 'userGroup', 'age', 'location', 'hireDate')
user.head()
```

Out[56]:

	userID	userGroup	age	location	hireDate
0	1	The Boss	NaN	None	None
1	2	test	NaN	None	None
2	3	test	NaN	None	None
3	4	test	NaN	None	None
4	5	Sophomore	NaN	None	None

In [57]:
```
dw_df = pd.merge(qid_res,qstn,left_on = 'questionID_id',right_on='questionID',how='inne
dw_df = pd.merge(dw_df,qstn_txt,left_on = 'questionTextID_id',right_on='questionTextID'
dw_df = pd.merge(dw_df,factor,left_on = 'factorID_id',right_on='factorID',how='inner')
dw_df = pd.merge(dw_df,surv,left_on = 'surveyID_id',right_on='surveyID',how='inner')
dw_df.shape
```

Out[57]: (8296, 15)

In [58]:
```
dw_df.isnull().any()
```

Out[58]:
```
questionID_id        False
response             False
questionID           False
questionTextID_id    False
surveyID_id          False
questionTextID       False
factorID_id          False
positive_p           False
factorID             False
factorName           False
studyID_id           False
surveyID             False
userID_id            False
creationDate         False
completionDate        True
dtype: bool
```

In [59]:
```
response_values = list(pd.unique(dw_df['response']))
string_response = [r for r in response_values if r not in '123456']
string_response
```

Out[59]:
```
['jh', 'jbj', 'False', 'sd', 'asd']
```

In [60]:
```
b = ~dw_df.response.isin(string_response)
dw_df = dw_df[b]
dw_df['response'] = pd.to_numeric(dw_df.response)
dw_df.shape
```

Out[60]:
```
(8291, 15)
```

In [61]:
```
response = dw_df.response
dw_df.loc[dw_df.positive_p==0,'response'] = (7- response)
dw_df['creationDate'] = pd.to_datetime(dw_df['creationDate'])
```

In [62]:
```
dw_df.to_csv('dw_df.csv')
```

In [ ]:
```

```

Codes for data visualization

In [402…
```
import sqlite3 as sq
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [403…
```
dw_df = pd.read_csv('dw_df.csv')
```

In [404…
```
dw_df['creationDate'] = pd.to_datetime(dw_df['creationDate']).dt.date
```

In [405…
```
studyID3_df = dw_df.loc[((dw_df.studyID_id==3) & (dw_df.factorID != 26))]
studyID3_df.isnull().any()
```

Out[405…
```
Unnamed: 0          False
questionID_id       False
response            False
questionID          False
questionTextID_id   False
surveyID_id         False
questionTextID      False
factorID_id         False
positive_p          False
factorID            False
factorName          False
studyID_id          False
surveyID            False
userID_id           False
creationDate        False
completionDate      False
dtype: bool
```

In [406…
```
studyID3_df.head()
```

Out[406…

	Unnamed: 0	questionID_id	response	questionID	questionTextID_id	surveyID_id	questionTextID
4572	4577	4006	5	4006	37	987	37
4573	4578	4007	3	4007	43	987	43
4574	4579	4008	4	4008	49	987	49
4575	4580	4009	2	4009	54	987	54

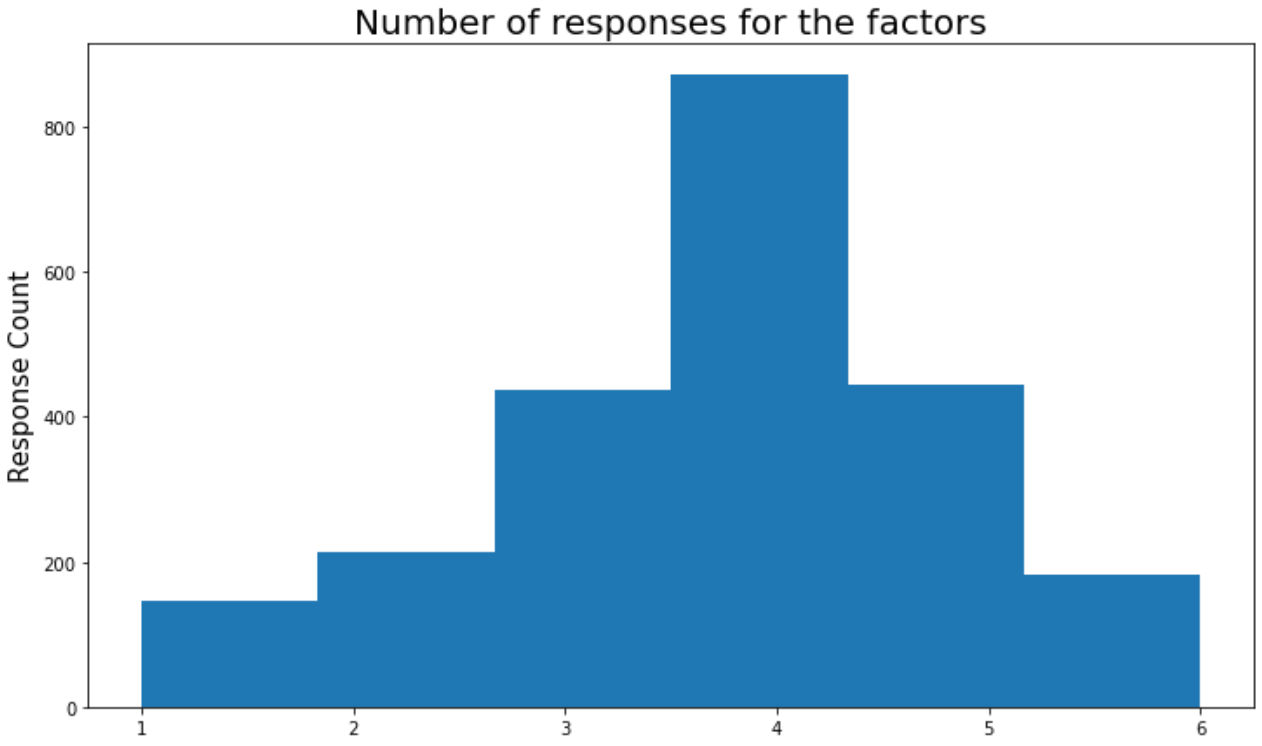
	Unnamed: 0	questionID_id	response	questionID	questionTextID_id	surveyID_id	questionTextID
4576	4581	4010	4	4010	56	987	56

In [407…
```
studyID3_df.factorID_id.value_counts()
```

Out[407…
```
15    641
12    491
13    452
14    380
11    327
Name: factorID_id, dtype: int64
```

In [408…
```
plt.hist(studyID3_df.response,bins=6)
fig = plt.gcf()
fig.set_size_inches(12,7)
plt.ylabel('Response Count',fontsize=15)
plt.title('Number of responses for the factors',fontsize=20)
```

Out[408…
```
Text(0.5, 1.0, 'Number of responses for the factors')
```



In [409…
```
studyID3_df = studyID3_df.groupby(by=['creationDate','factorID_id'],as_index=False).mea
studyID3_df.shape
```

Out[409…
```
(210, 14)
```

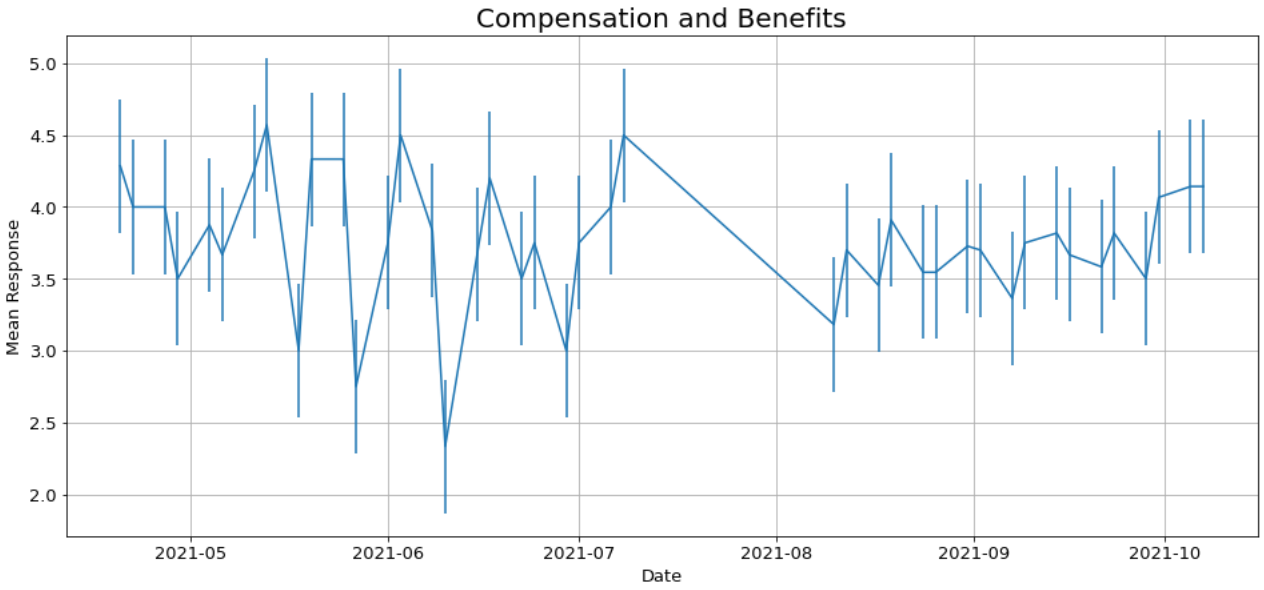
In [ ]:

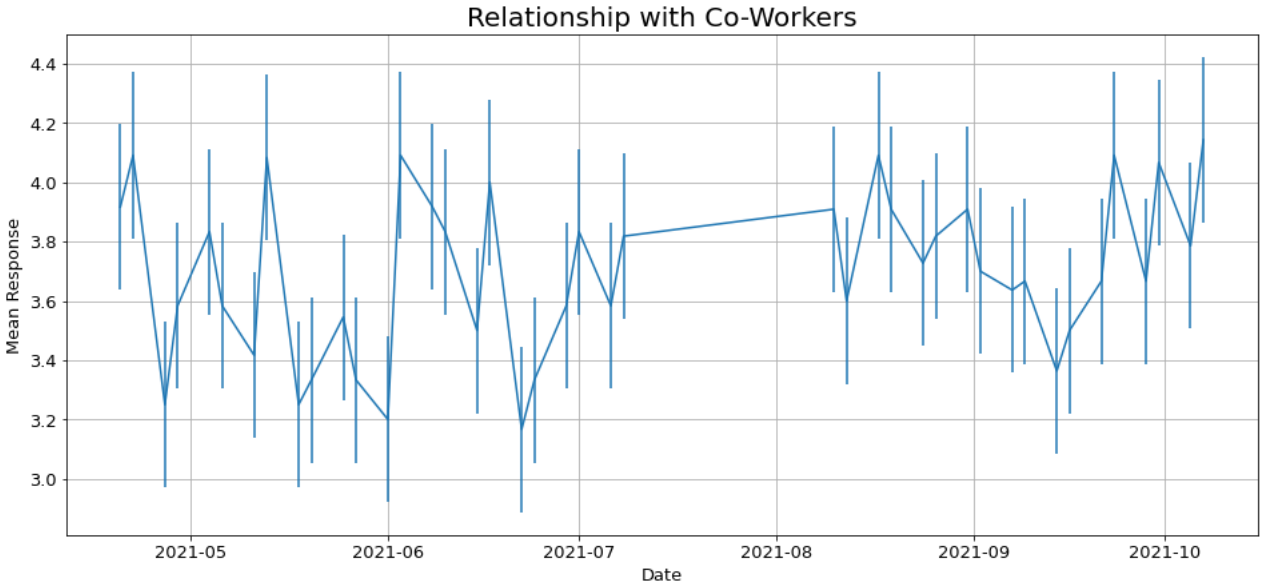
In [410…

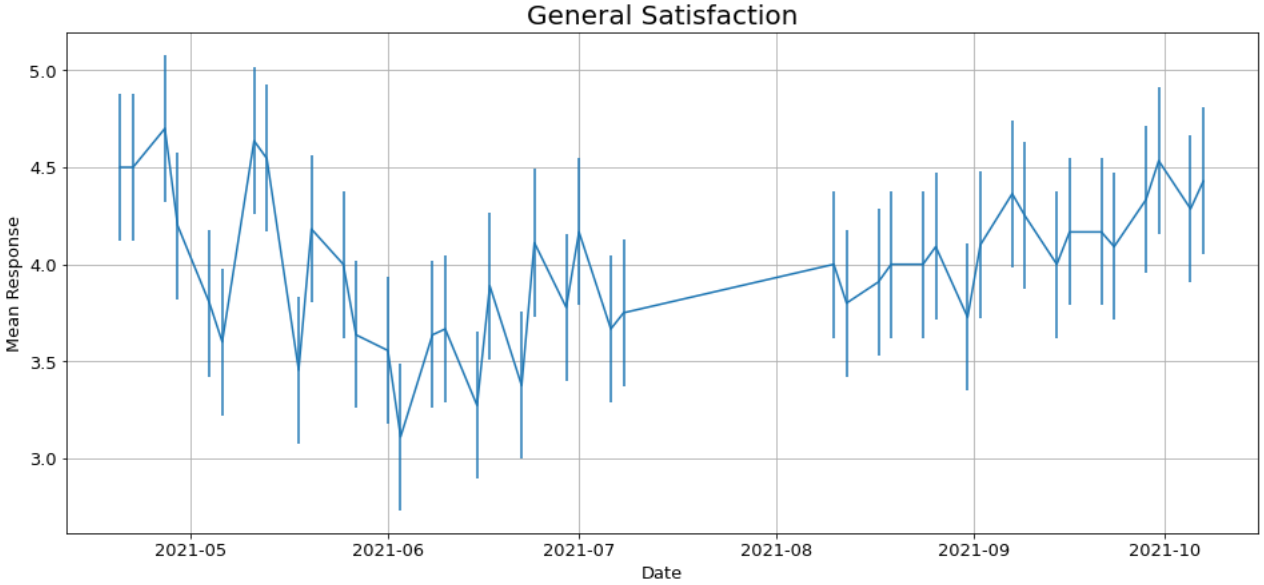
```
factorID_ids = studyID3_df.factorID_id.unique()

## Methods for ploting figures for each factor
def plot_response(factorID_ids):
    i = 0
    for factorID_id in factorID_ids:
        ## this is used to get the figure number from sub plot
        i+=1

        ## This codes retrieves factor name by factor ID
        factorName = (list(dw_df.loc[dw_df['factorID_id']==factorID_id,'factorName'])[0

        ## made data frame for individula factor
        df_by_factor = studyID3_df[studyID3_df['factorID_id']==factorID_id]

        ## Codes for ploting

        row = len(factorID_ids)
        ax = plt.subplot(row,1,i)

        err = df_by_factor.std().response
        plt.errorbar(df_by_factor.creationDate,df_by_factor.response,yerr=err)

        fig = plt.gcf()
        fig.set_size_inches(16,40)

        plt.grid(True)
        plt.xlabel('Date',fontsize=13)
        plt.ylabel('Mean Response',fontsize=13)
        plt.title(factorName,fontsize=20)

        # Adjusting tick sizes
        ax.tick_params(axis='both', which='major', labelsize=13)
        plt.show()
```

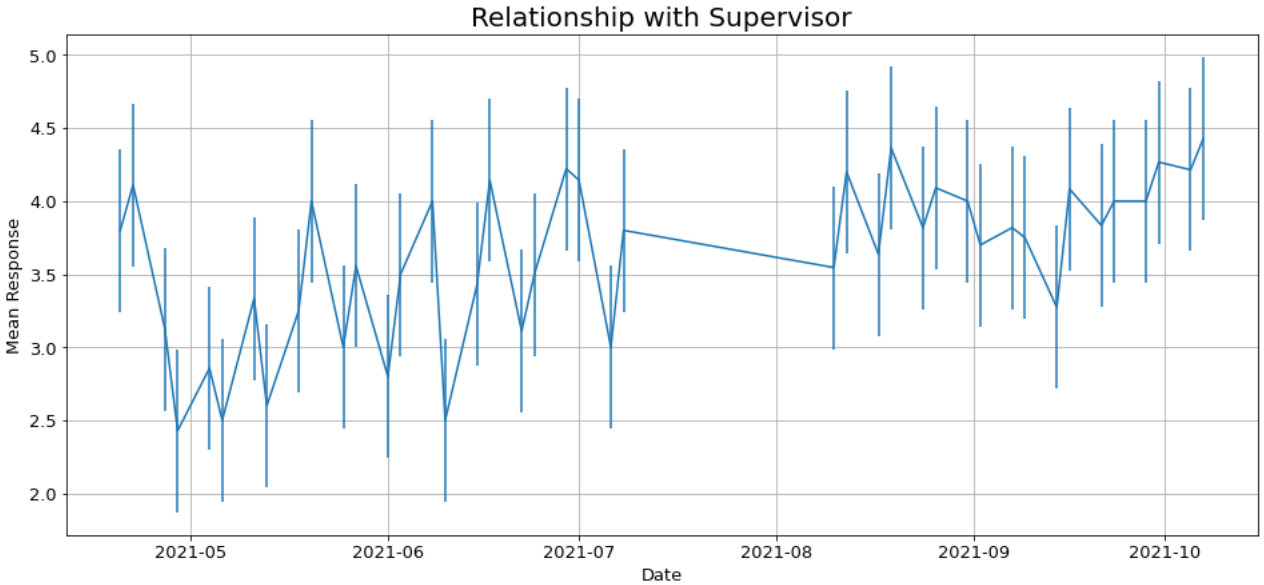
In [411…

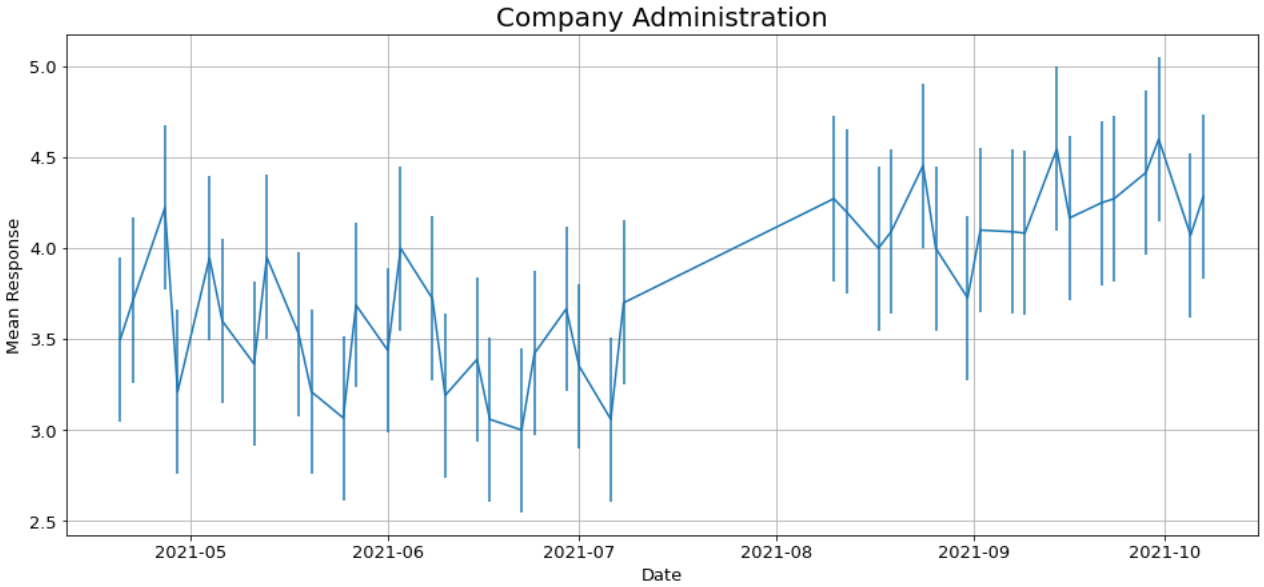
```
plot_response(factorID_ids)
```




Relationship with Co-Workers
Mean Response
4.4
4.2
4.0
3.8
3.6
3.4
3.2
3.0
2021-05
2021-06
2021-07
2021-08
2021-09
2021-10
Date


General Satisfaction
Mean Response
5.0
4.5
4.0
3.5
3.0
2021-05
2021-06
2021-07
2021-08
2021-09
2021-10
Date


Relationship with Supervisor
Mean Response
5.0
4.5
4.0
3.5
3.0
2.5
2.0
2021-05
2021-06
2021-07
2021-08
2021-09
2021-10
Date


Company Administration
Mean Response
5.0
4.5
4.0
3.5
3.0
2.5
2021-05
2021-06
2021-07
2021-08
2021-09
2021-10
Date

In [ ]: