DATA ANALYTICS PORTFOLIO

SARAH ALDAWOOD

Project 1 : Data Observation for the SF Salaries dataset.

source : https://www.kaggle.com/datasets/kaggle/sf-salaries

this data set represents the Salaries of various Jobs in San Francisco .
It contains 13 columns and 148655 rows.
here is a preview of it :

```
df = pd.read_csv('./Salaries.csv')
df.head(5)
```

1.1s Python

```
/tmp/ipykernel_12672/2477835102.py:1: DtypeWarning: Columns (3,4,5,6,12) have mixed types. Specify dtype option on import or set low_memory=False.
  df = pd.read_csv('./Salaries.csv')
```

	Id	EmployeeName	JobTitle	BasePay	OvertimePay	OtherPay	Benefits	TotalPay	TotalPayBenefits	Year	Notes	Agency	Status
0	1	NATHANIEL FORD	GENERAL MANAGER-METROPOLITAN TRANSIT AUTHORITY	167411.18	0.0	400184.25	NaN	567595.43	567595.43	2011	NaN	San Francisco	NaN
1	2	GARY JIMENEZ	CAPTAIN III (POLICE DEPARTMENT)	155966.02	245131.88	137811.38	NaN	538909.28	538909.28	2011	NaN	San Francisco	NaN
2	3	ALBERT PARDINI	CAPTAIN III (POLICE DEPARTMENT)	212739.13	106088.18	16452.6	NaN	335279.91	335279.91	2011	NaN	San Francisco	NaN
3	4	CHRISTOPHER CHONG	WIRE ROPE CABLE MAINTENANCE MECHANIC	77916.0	56120.71	198306.9	NaN	332343.61	332343.61	2011	NaN	San Francisco	NaN
4	5	PATRICK GARDNER	DEPUTY CHIEF OF DEPARTMENT,(FIRE DEPARTMENT)	134401.6	9737.0	182234.59	NaN	326373.19	326373.19	2011	NaN	San Francisco	NaN

I started exploring inconsistencies and faults in the data

First : messing values

```
# Count the number of null values in each column
null_counts = df.isnull().sum()
print(null_counts)
```

```
Id                    0
EmployeeName          0
JobTitle              0
BasePay             605
OvertimePay           0
OtherPay              0
Benefits          36159
TotalPay              0
TotalPayBenefits      0
Year                  0
Notes            148654
Agency                0
Status           110535
dtype: int64
```

Second : inappropriate data types

```
df.describe()
```

	Id	TotalPay	TotalPayBenefits	Year	Notes
count	148654.000000	148654.000000	148654.000000	148654.000000	0.0
mean	74327.500000	74768.321972	93692.554811	2012.522643	NaN
std	42912.857795	50517.005274	62793.533483	1.117538	NaN
min	1.000000	-618.130000	-618.130000	2011.000000	NaN
25%	37164.250000	36168.995000	44065.650000	2012.000000	NaN
50%	74327.500000	71426.610000	92404.090000	2013.000000	NaN
75%	111490.750000	105839.135000	132876.450000	2014.000000	NaN
max	148654.000000	567595.430000	567595.430000	2014.000000	NaN

```
print(df.dtypes)
```

```
Id                  int64
EmployeeName       object
JobTitle           object
BasePay            object
OvertimePay        object
OtherPay           object
Benefits           object
TotalPay          float64
TotalPayBenefits  float64
Year                int64
Notes             float64
Agency             object
Status             object
dtype: object
```

Third : mistakes and causes of the inconsistencies

```
# to check mistakes in the data I wanted to inspects why some numeric columns
# had Object type and what are the incorrect values


temp_series = pd.to_numeric(df['BasePay'], errors='coerce')
non_numeric_entries = df['BasePay'][temp_series.isna()]

print("Non-numeric entries in the column:")
print(non_numeric_entries.unique())
#print(non_numeric_entries.to_string())
#non_numeric_entries.__len__()
```

```
Non-numeric entries in the column:
[nan 'Not Provided']
```

```
temp_series = pd.to_numeric(df['OvertimePay'], errors='coerce')

non_numeric_entries = df['OvertimePay'][temp_series.isna()]

print("Non-numeric entries in the column:")
print(non_numeric_entries.unique())
```

```
Non-numeric entries in the column:
['Not Provided']
```

The previous mistakes are common and easy to handle . But before that I wanted to explore some more in the JobTitle column because of its important role in the data.
Given that any observation I might draw about a job salary will rely on how many times this job appeared in the data set, I started by counting the numbers of time each unique Job appeared. To my surprise, more than half the jobs appeared less than 50 times and most of those appeared less than 20 times.
I can’t generalize information based on the jobs that rarely appeared and I can’t ignore them either or I’ll lose most of the data!

```
df['JobTitle'] = df['JobTitle'].str.lower()
Jobs = pd.Series( df['JobTitle'].unique())
Jobs_numbers =  df['JobTitle'].value_counts()
print(Jobs_numbers.to_string())
Jobs_numbers.to_json('job_titles.json')
```

```
transit operator                          9424
special nurse                             5791
registered nurse                          4955
custodian                                 3214
firefighter                               3153
recreation leader                         2663
deputy sheriff                            2618
public svc aide-public works              2518
police officer 3                          2421
patient care assistant                    1945
public service trainee                    1656
attorney (civil/criminal)                 1503
police officer                            1476
porter                                    1465
general laborer                           1410
gardener                                  1187
police officer 2                          1141
parking control officer                   1140
library page                              1107
senior clerk                              1064
senior clerk typist                       1055
sergeant 3                                1047
clerk                                      983
eligibility worker                         980
emt/paramedic/firefighter                  918
...
animal care assistant supervisor             1
media/security systems supervisor            1
```

As a solution to the previous problem , I decided to use gpt-4-turbo to combine similar jobs together.
gpt is an excellent large language model from openAI and more importantly it can handle the amount of data that I'm going to give it.
I put the rare jobs (appeared less than 20 times) in a json file . it was over 900 lines and 9939 tokens.
Then , I wrote the following code (notice the prompt) to make my request to the API.

```python
with open('rare_jobs.json', 'r') as file:
    data = json.load(file)

json_str = json.dumps(data)

prompt = f'''this is part of a dataset that containd information about jobs and salaries. I will give you a list of the
rare jobs that appeared less than 20 time and how many times they appeared. I want you to find a way to group these roles based
on similarity in the role and the responsibility . since the data is about salary it is important that only
the jobs similar in responsibility are grouped together and not by the department .
for example you can't put nurses with doctors in a catagory.
I need you to give me back a list of the Job titles and the group name they should be under.
make the new group name clear like : CEO roles . or something similar.
Here are the data {json_str}'''

print(json_str)
```

Python

```python
client = OpenAI(
  api_key=os.getenv('OPENAI_KEY')
)

stream = client.chat.completions.create(
    model="gpt-4-turbo",
    messages=[{"role": "user", "content": prompt}],
    stream=True,
)
for chunk in stream:
    if chunk.choices[0].delta.content is not None:
        print(chunk.choices[0].delta.content, end="")
```

Python

The result I got was good but it needed manual editing.
the details are in the next page

GPT results (No editing) and some mistakes

Senior Management and Executive Roles .1
Chief Surveyor -
Chief Deputy Sheriff -
Mayor -
City Attorney -
Chief of Police -
Chief, Fire Department -
Port Director -
Manager VII, MTA -
Dept Head V -
Deputy Dir I, MTA -
Assistant Chief Attorney 2 -
Assistant Deputy Chief 2 -
Asst Chf of Dept (Fire Dept) -
Benefits Supervisor -
Admin Analyst 3 -

Engineering and Technical Specialist Roles .2
Architectural Assistant II -
Structural Engineer -
Chemist III - the Chemist seemed in the wrong group
Electrical Inspector -
Urban Forestry Inspector -
Transit Power Line Worker -
Senior Power House Operator -
Electrical Transit Mechanic, Assistant Supervisor -
Engineering Assistant -
Utility Analyst -
Electrical Transit Shop Sprv 1 -
Marine Engineer of Fire Boats -
Fire Protection Engineer -
Transmission and Distribution Line Worker -
Landscape Architect -

Healthcare and Medical Roles .3
the healthcare and medical group included very distinct roles that vary in responsibility and level.
Dentist -
Nurse Midwife -
Medical Records Tech Sprv -
Dental Aide -
Microbiologist -
Dental Hygienist -
Medical Examiner's Investigator I -
Senior Medical Social Worker -
Medical Social Work Supervisor -
Medical Transcriber Typist -
Nurse Supervisor Psychiatric -
Veterinary Laboratory Technologist -
Emergency Medical Services Chief -
Forensic Toxicologist -
Medical Staff Services Department Specialist -

Public Safety and Emergency Services Roles .4
Sergeant II (Police Department) -
Park Patrol Officer -
Assistant Inspector (Police Department) -
Mayoral Staff XVI (related to public safety planning) -
Transit Manager II -
Emergency Services Coordinator -
Airport Emergency Planning Coordinator -
Emergency Planning Coordinator -
Institutional Police Officer -

Legal and Court-Related Roles .5 ###
Legal Secretary II -
Legal Process Clerk -
Court Interpreter -
Assistant Chief Victim/Witness Investigator -
Mayoral Staff IV (potentially related to legal services) -
Court Manager -
Court Supervisor II -
Deputy Chief Attorney (Civil & Criminal) -

Transportation and Urban Planning Roles .6 ###
Transit Operations Specialist -
Transportation Planner II -
Traffic Signal Electrician -
Transit Traffic Checker -
Transportation Planner IV -
Parking Meter Repairer Sprv 1 -
Transit Planner IV -

Administrative and Human Resources Roles .7 ###
Mayoral Staff XIII -
Personnel Clerk -
Public Relations Officer -
Human Rights Comm Representative -
Employee Relations Representative -
Personnel/Payroll Representative -
Human Resources Director -
Employee Assistance Counselor -

Environmental and Public Utilities Roles .8 ###
Environment Specialist -
Water Meter Repairer -
Senior Stationary Engineer, Water Treatment Plant -
Environmental Health Technician I -
Water Quality Technician III -
Industrial Hygienist -
Environmental Planner -

Science and Laboratory Roles .9 ###
Criminalist II -
Asst Forensic Toxicologist 1 -
Biologist III -
Chemist III -
Supervising Chemist -
Senior Microbiologist -

Educational and Community Services Roles .10 ###
Recreation Specialist -
Child Care Specialist -
Social Work Supervisor -
Community Development Assistant -
Library Commissioner -
Education Program Specialist -

Groups after manual editing

I do not claim that this grouping is the most accurate however my goal to showcase the techniques used to clean this type of data was accomplished.

:"Senior Management and Executive Roles"
,"Chief Surveyor"
,"Chief Deputy Sheriff"
,"Mayor"
,"City Attorney"
,"Chief of Police"
,"Chief, Fire Department"
,"Port Director"
,"Manager VII, MTA"
,"Dept Head V"
,"Deputy Dir I, MTA"
,"Assistant Chief Attorney 2"
,"Assistant Deputy Chief 2"
,"Asst Chf of Dept (Fire Dept)"
"Admin Analyst 3"
:"Engineering and Technical Specialist Roles"
,"Architectural Assistant II"
,"Structural Engineer"
,"Electrical Inspector"
,"Urban Forestry Inspector"
,"Transit Power Line Worker"
,"Senior Power House Operator"
,"Electrical Transit Mechanic, Assistant Supervisor"
,"Engineering Assistant"
,"Utility Analyst"
,"Electrical Transit Shop Sprv 1"
,"Marine Engineer of Fire Boats"
,"Fire Protection Engineer"
,"Transmission and Distribution Line Worker"
"Landscape Architect"
:"Senior Medical Roles"
,"Dentist"
,"Nurse Midwife"
,"Senior Medical Social Worker"
"Nurse Supervisor Psychiatric"

:"Medical Specialist Roles"
,"Dental Hygienist"
,"Microbiologist"
,"Forensic Toxicologist"
"Veterinary Laboratory Technologist"

:"Entry-Level Medical Roles"
"Dental Aide"
:"Public Safety and Emergency Services Roles"
,"Sergeant II (Police Department)"
,"Park Patrol Officer"
,"Assistant Inspector (Police Department)"
,"Mayoral Staff XVI (related to public safety planning)"
,"Transit Manager II"
"Emergency Medical Services Chief"
,"Emergency Services Coordinator"
,"Airport Emergency Planning Coordinator"
,"Emergency Planning Coordinator"
,"Institutional Police Officer"
,"Medical Examiner's Investigator I"
:"Legal and Court-Related Roles"
,"Legal Secretary II"
,"Legal Process Clerk"
,"Court Interpreter"
,"Assistant Chief Victim/Witness Investigator"
,"Mayoral Staff IV (potentially related to legal services)"
,"Court Manager"
,"Court Supervisor II"
"Deputy Chief Attorney (Civil & Criminal)"
:"Transportation and Urban Planning Roles"
,"Transit Operations Specialist"
,"Transportation Planner II"
,"Traffic Signal Electrician"
,"Transit Traffic Checker"
,"Transportation Planner IV"
,"Parking Meter Repairer Sprv 1"
"Transit Planner IV"
:"Administrative and Human Resources Roles"
,"Mayoral Staff XIII"
,"Personnel Clerk"
,"Public Relations Officer"
,"Human Rights Comm Representative"
,"Employee Relations Representative"
,"Personnel/Payroll Representative"
,"Human Resources Director"
,"Employee Assistance Counselor"
,"Benefits Supervisor"
,"Medical Records Tech Sprv"
,"Medical Transcriber Typist"
"Medical Staff Services Department Specialist"
:"Environmental and Public Utilities Roles"
,"Environment Specialist"
,"Water Meter Repairer"
,"Senior Stationary Engineer, Water Treatment Plant"
,"Environmental Health Technician I"
,"Water Quality Technician III"
,"Industrial Hygienist"
"Environmental Planner"
:"Science and Laboratory Roles"
,"Chemist III"
,"Criminalist II"
,"Asst Forensic Toxicologist 1"
,"Biologist III"
,"Supervising Chemist"
"Senior Microbiologist"
:"Educational and Community Services Roles"
,"Recreation Specialist"
,"Child Care Specialist"
,"Social Work Supervisor"
,"Community Development Assistant"
,"Library Commissioner"
"Education Program Specialist"

checking other columns

```
df['Agency'] = df['Agency'].str.lower()
Agency = pd.Series( df['Agency'].unique())
Agency =  df['Agency'].value_counts()
print(Agency.to_string())
```

```
san francisco    143453
```

```
Year = pd.Series( df['Year'].unique())
Year =  df['Year'].value_counts()
print(Year.to_string())
```

```
2014    37216
2013    36646
2012    35766
2011    33825
```

```
Status = pd.Series( df['Status'].unique())
Status =  df['Status'].value_counts()
print(Status.to_string())
```

```
FT    21655
PT    15561
```

started cleaning

replacing invalid and null values in the data

```
df = df[df['JobTitle'] != 'Not Provided']
df = df[df['BasePay'] != 'Not Provided']
```

```
df = df.drop(['Notes','EmployeeName','Agency'], axis=1)
```

```
# we will assume that the status of the Job is full time if not provided
df['Status'] = df['Status'].fillna('FT')
```

validating numerical values and filling missing information in it . from the first check I noticed that BasePay and Benefits had missing values.
however , TotalPay and TotalPayBenefits doesn't.
so to complete the data I decided to calculate them .
the missing values are probably zeros but I had to make sure.

```
Id                    0
EmployeeName          0
JobTitle              0
BasePay             605
OvertimePay           0
OtherPay              0
Benefits          36159
TotalPay              0
TotalPayBenefits      0
```

Formulas:
Calculating Missing 'BasePay'
BasePay = TotalPay - (OvertimePay + OtherPay)

'Calculating Missing 'Benefits
Benefits = TotalPayBenefits - TotalPay

```
df.replace('Not Provided', 0, inplace=True)
```

First, I ran this code for all columns to check

```
df.fillna(0 , inplace=True)
temp_series = pd.to_numeric(df['OtherPay'], errors='coerce')
non_numeric_entries = df['OtherPay'][temp_series.isna()]
print("Non-numeric entries in the column:")
print(non_numeric_entries.unique())
```

```
Non-numeric entries in the column:
[]
```

Finally, I ran this code to calculate the values and update them.

```
# Calculate missing BasePay
df['BasePay'] = df.apply(lambda row: row['TotalPay'] - (row['OvertimePay'] + row['OtherPay']) if pd.isna(row['BasePay']) else row['BasePay'], axis=1)

# Calculate missing Benefits
df['Benefits'] = df.apply(lambda row: row['TotalPayBenefits'] - row['TotalPay'] if pd.isna(row['Benefits']) else row['Benefits'], axis=1)

# Update TotalPayBenefits if necessary
df['TotalPayBenefits'] = df.apply(lambda row: row['TotalPay'] + row['Benefits'] if pd.isna(row['TotalPayBenefits']) else row['TotalPayBenefits'], axis=1)
```

checking the results

```
df['BasePay'] = df['BasePay'].astype(float)
df['OvertimePay'] = df['OvertimePay'].astype(float)
df['OtherPay'] = df['OtherPay'].astype(float)
df['Benefits'] = df['Benefits'].astype(float)
```

```
df.describe()
```

	Id	BasePay	OvertimePay	OtherPay	Benefits	TotalPay	TotalPayBenefits	Year
count	143453.000000	143453.000000	143453.000000	143453.000000	143453.000000	143453.000000	143453.000000	143453.000000
mean	75059.847413	65363.273113	5122.714955	3637.093877	18976.595266	74123.081944	93099.677210	2012.538525
std	42705.937678	42622.808144	11501.315303	7857.173972	17078.360167	50383.865627	62749.219438	1.113089
min	2.000000	-166.010000	-0.010000	-7058.590000	-33.890000	-618.130000	-618.130000	2011.000000
25%	38280.000000	31428.860000	0.000000	0.000000	2.850000	34735.050000	42360.830000	2012.000000
50%	75150.000000	64419.730000	0.000000	826.670000	23471.980000	70785.820000	91974.600000	2013.000000
75%	111959.000000	94168.640000	4772.770000	4264.900000	33451.190000	105357.220000	132470.030000	2014.000000
max	148654.000000	319275.010000	245131.880000	342802.630000	91302.460000	538909.280000	538909.280000	2014.000000

now that I can get the description of all the numerical columns I noticed the negative values !

```
condition = (
    (df['BasePay'] < 0) |
    (df['OvertimePay'] < 0) |
    (df['OtherPay'] < 0) |
    (df['Benefits'] < 0) |
    (df['TotalPay'] < 0) |
    (df['TotalPayBenefits'] < 0)
)
negative_values = df[condition]
print(negative_values)
```

```
df = df[~condition]
```

```
df.describe()
```

	Id	BasePay	OvertimePay	OtherPay	Benefits	TotalPay	TotalPayBenefits	Year
count	143432.000000	143432.000000	143432.000000	143432.000000	143432.000000	143432.000000	143432.000000	143432.000000
mean	75057.601923	65370.690374	5123.439011	3637.600252	18978.560703	74131.729637	93110.290340	2012.538520
std	42707.744880	42619.931473	11502.000173	7857.570361	17078.271638	50381.207804	62745.712167	1.113142
min	2.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	2011.000000
25%	38274.750000	31446.665000	0.000000	0.000000	2.877500	34777.455000	42405.997500	2012.000000
50%	75150.500000	64428.000000	0.000000	827.180000	23476.385000	70790.960000	91984.350000	2013.000000
75%	111962.250000	94181.667500	4775.600000	4265.195000	33452.072500	105365.467500	132477.957500	2014.000000
max	148650.000000	319275.010000	245131.880000	342802.630000	91302.460000	538909.280000	538909.280000	2014.000000

+ Code + Markdown

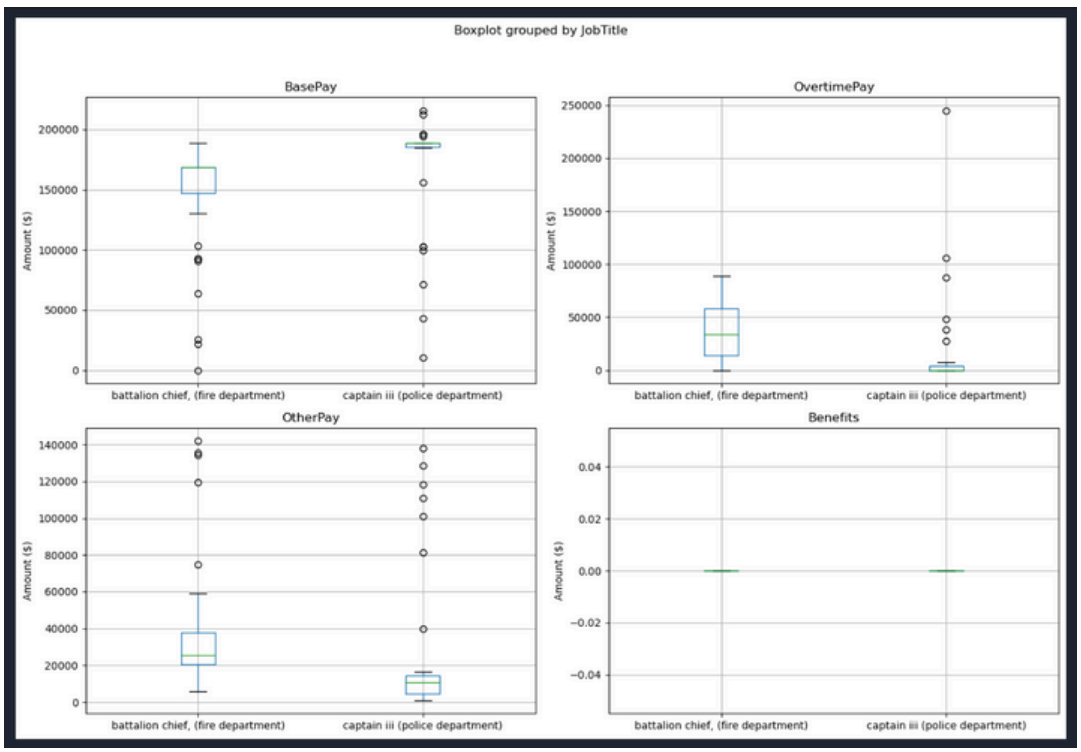
Finally cleaning the data is complete
I saved it in a new csv file then started visualizing the data

To observe the spread and outliers in base pay, overtime pay, other pay, and benefits for the most .common job titles
for a few selected job titles to keep the plot readable.

```
job_titles = ['captain iii (police department)', 'battalion chief, (fire department)']
filtered_df = df[df['JobTitle'].isin(job_titles)]

fig, axes = plt.subplots(2, 2, figsize=(14, 10))
fig.suptitle('Pay Component Distribution by Job Title')

components = ['BasePay', 'OvertimePay', 'OtherPay', 'Benefits']
for i, comp in enumerate(components):
    ax = axes[i//2, i%2]
    filtered_df.boxplot(column=comp, by='JobTitle', ax=ax)
    ax.set_title(comp)
    ax.set_xlabel('')
    ax.set_ylabel('Amount ($)')
plt.tight_layout(rect=[0, 0.03, 1, 0.95])
plt.show()
```

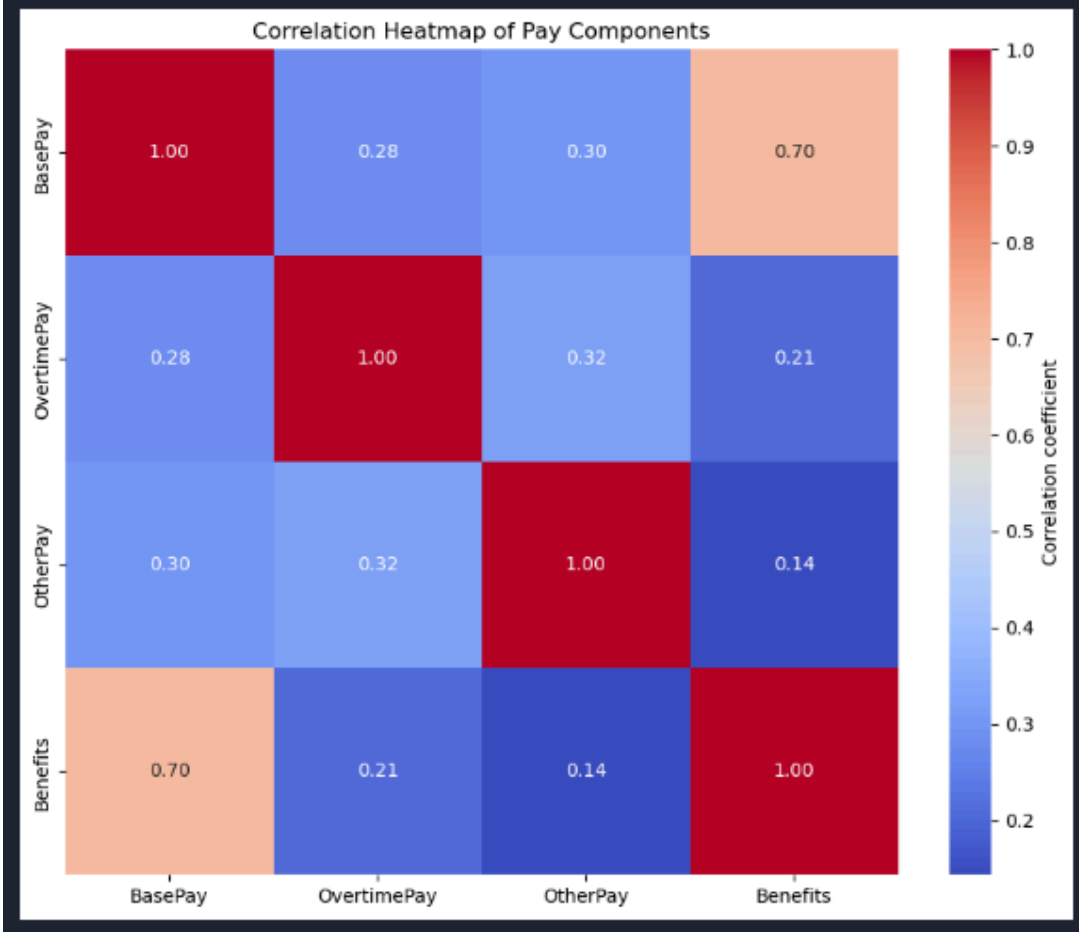


a heat map for the correlation between pay types

```python
# Calculating correlations
correlation_matrix = df[['BasePay', 'OvertimePay', 'Othe

# Plotting heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, fmt=".2f", c
plt.title('Correlation Heatmap of Pay Components')
plt.show()
```

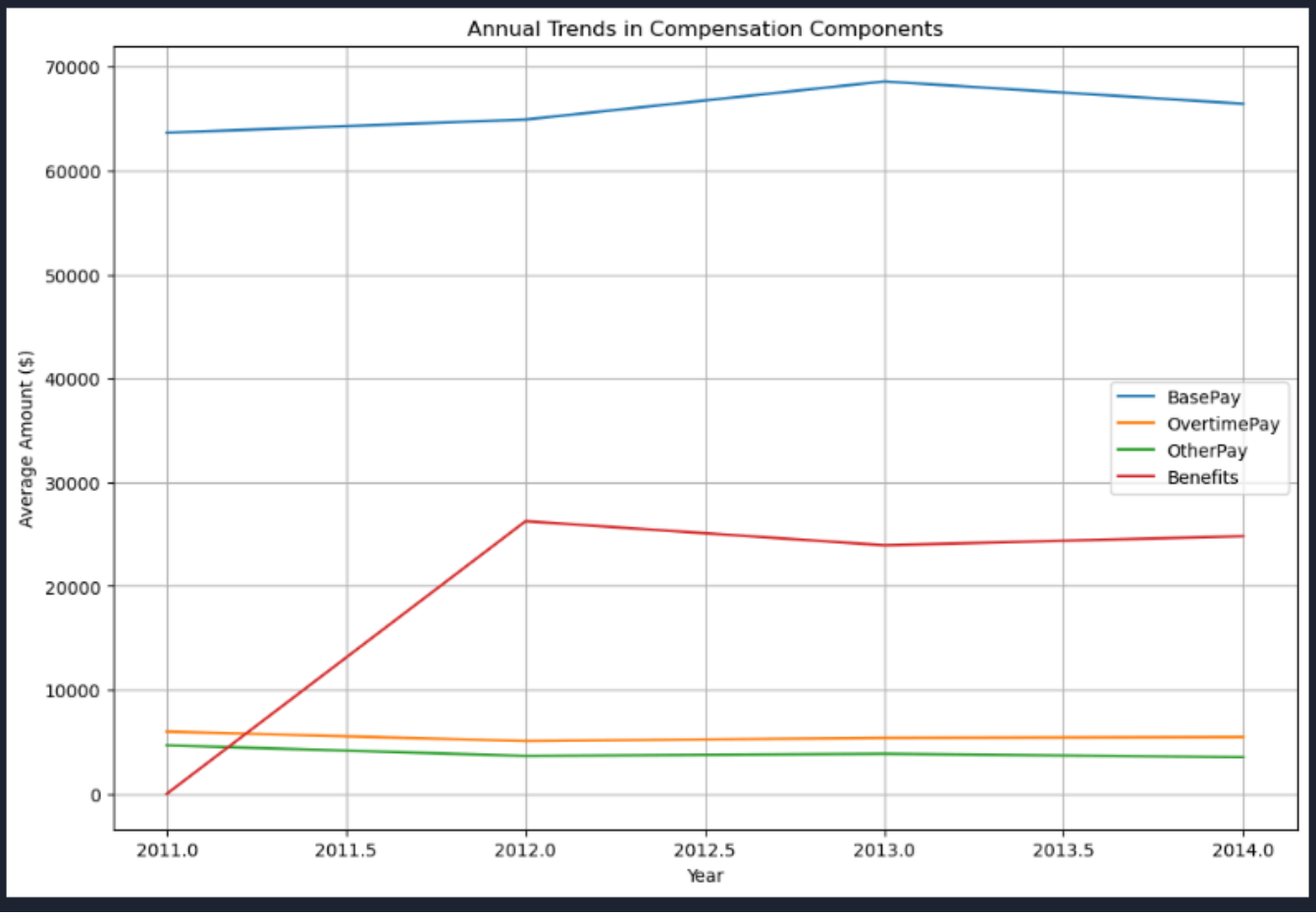
Python



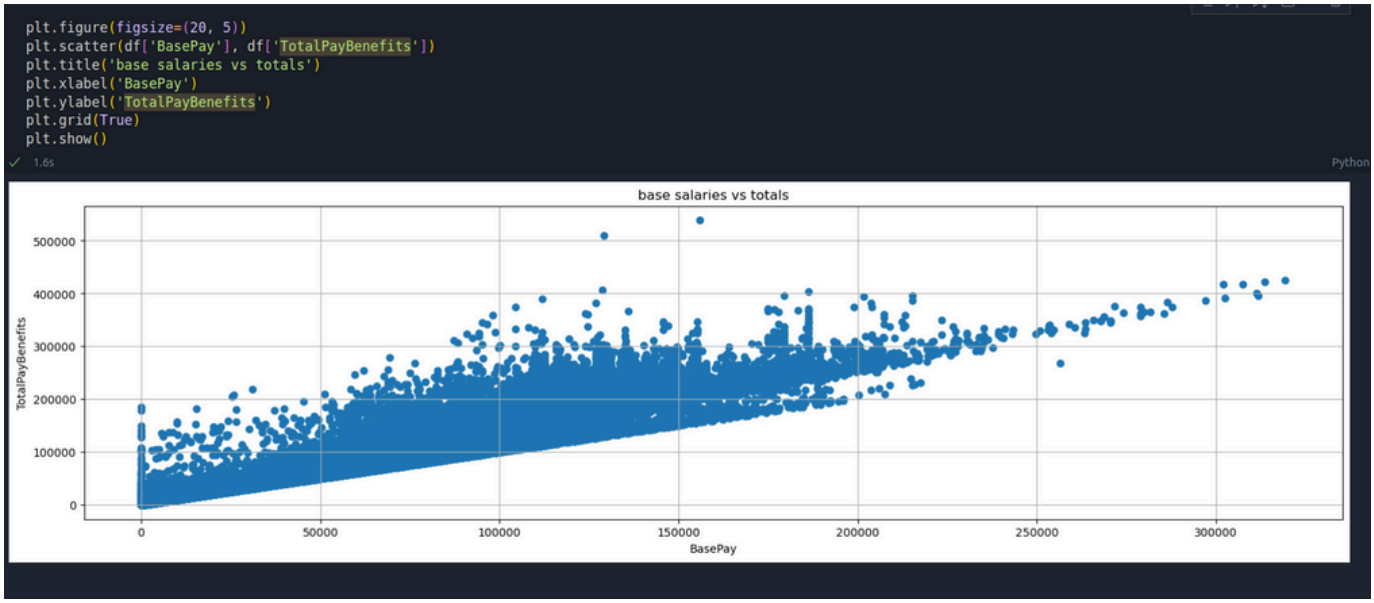
Analyze how the total compensation trends for different job titles have changed over the available years, especially focusing on the .change in benefits

```python
df_grouped_by_year = df.groupby('Year').mean()
plt.figure(figsize=(12, 8))
for column in ['BasePay', 'OvertimePay', 'OtherPay', 'Benefits']:
    plt.plot(df_grouped_by_year.index, df_grouped_by_year[column], label=column
plt.title('Annual Trends in Compensation Components')
plt.xlabel('Year')
plt.ylabel('Average Amount ($)')
plt.legend()
plt.grid(True)
plt.show()
```

Python



this plot shows the regression relation ship between the BasePay and Total



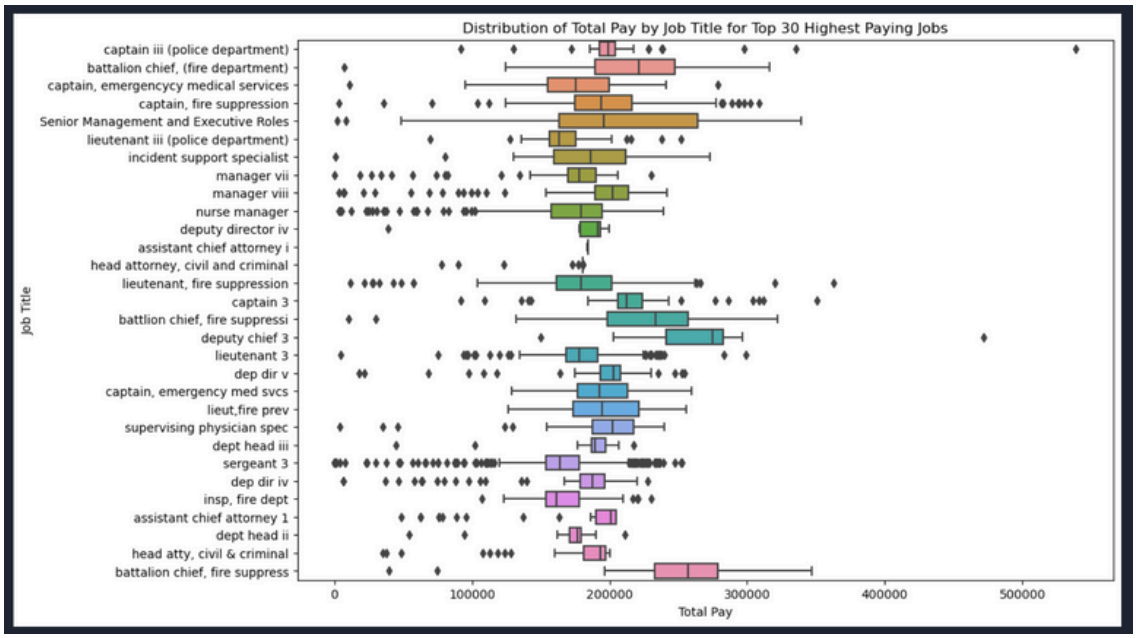
to perform a numerical analysis, you could use techniques to convert "Job Title" into a numerical form, like label encoding or one-hot encoding, and then use these in a model to predict Total Pay. However, this wouldn't give you a correlation coefficient but could indicate the importance of Job Title in predicting Total Pay.
to perform a numerical analysis, you could use techniques to convert "Job Title" into a numerical form, like label encoding or one-hot encoding, and then use these in a model to predict Total Pay. However, this wouldn't give you a correlation coefficient but could indicate the importance of Job Title in predicting Total Pay.

```
# Calculating mean Total Pay for each Job Title
mean_total_pay_by_job_title = df.groupby('JobTitle')['TotalPay'].mean().sort_values(ascending=False)

#limit the number of job titles displayed for readability
top_job_titles = mean_total_pay_by_job_title.head(30).index
filtered_df = df[df['JobTitle'].isin(top_job_titles)]

plt.figure(figsize=(12, 8))
sns.boxplot(x='TotalPay', y='JobTitle', data=filtered_df)
plt.title('Distribution of Total Pay by Job Title for Top 30 Highest Paying Jobs')
plt.xlabel('Total Pay')
plt.ylabel('Job Title')
plt.show()
```

[18] ✓ 0.3s

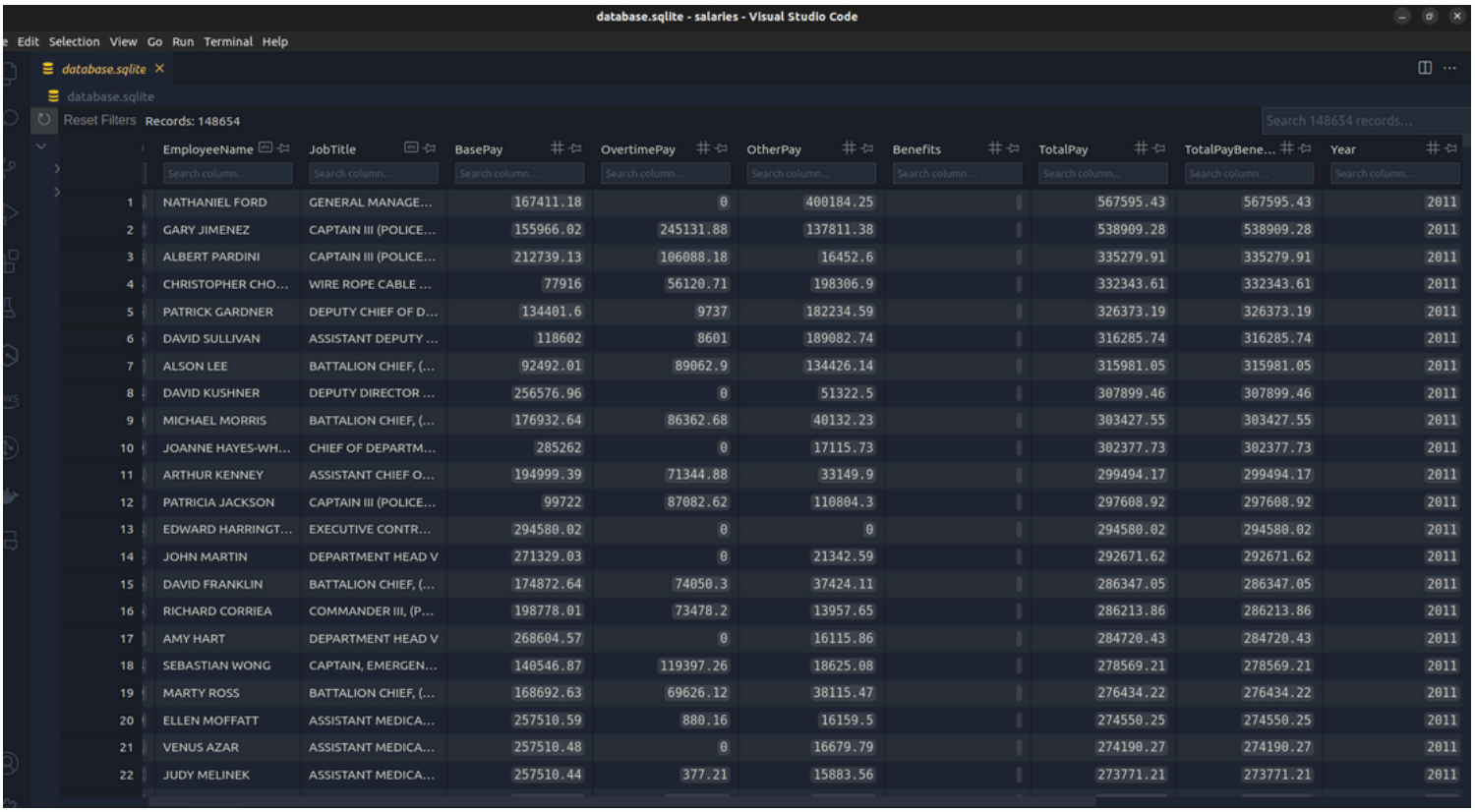


SQL

here I’m going to manipulate the same data in the shape of sqlite file.

I did not clean the data as much but I tried to accomplish similar tasks with sql.

first here is a look of the data in sqlite file



	EmployeeName	JobTitle	BasePay	OvertimePay	OtherPay	Benefits	TotalPay	TotalPayBene...	Year
1	NATHANIEL FORD	GENERAL MANAGE...	167411.18	0	400184.25		567595.43	567595.43	2011
2	GARY JIMENEZ	CAPTAIN III (POLICE...	155966.02	245131.88	137811.38		538909.28	538909.28	2011
3	ALBERT PARDINI	CAPTAIN III (POLICE...	212739.13	106088.18	16452.6		335279.91	335279.91	2011
4	CHRISTOPHER CHO...	WIRE ROPE CABLE ...	77916	56120.71	198306.9		332343.61	332343.61	2011
5	PATRICK GARDNER	DEPUTY CHIEF OF D...	134401.6	9737	182234.59		326373.19	326373.19	2011
6	DAVID SULLIVAN	ASSISTANT DEPUTY ...	118602	8601	189082.74		316285.74	316285.74	2011
7	ALSON LEE	BATTALION CHIEF, (...	92492.01	89062.9	134426.14		315981.05	315981.05	2011
8	DAVID KUSHNER	DEPUTY DIRECTOR ...	256576.96	0	51322.5		307899.46	307899.46	2011
9	MICHAEL MORRIS	BATTALION CHIEF, (...	176932.64	86362.68	40132.23		303427.55	303427.55	2011
10	JOANNE HAYES-WH...	CHIEF OF DEPARTM...	285262	0	17115.73		302377.73	302377.73	2011
11	ARTHUR KENNEY	ASSISTANT CHIEF O...	194999.39	71344.88	33149.9		299494.17	299494.17	2011
12	PATRICIA JACKSON	CAPTAIN III (POLICE...	99722	87082.62	110804.3		297608.92	297608.92	2011
13	EDWARD HARRINGT...	EXECUTIVE CONTR...	294580.02	0	0		294580.02	294580.02	2011
14	JOHN MARTIN	DEPARTMENT HEAD V	271329.03	0	21342.59		292671.62	292671.62	2011
15	DAVID FRANKLIN	BATTALION CHIEF, (...	174872.64	74050.3	37424.11		286347.05	286347.05	2011
16	RICHARD CORRIEA	COMMANDER III, (P...	198778.01	73478.2	13957.65		286213.86	286213.86	2011
17	AMY HART	DEPARTMENT HEAD V	268604.57	0	16115.86		284720.43	284720.43	2011
18	SEBASTIAN WONG	CAPTAIN, EMERGEN...	140546.87	119397.26	18625.08		278569.21	278569.21	2011
19	MARTY ROSS	BATTALION CHIEF, (...	168692.63	69626.12	38115.47		276434.22	276434.22	2011
20	ELLEN MOFFATT	ASSISTANT MEDICA...	257510.59	880.16	16159.5		274550.25	274550.25	2011
21	VENUS AZAR	ASSISTANT MEDICA...	257510.48	0	16679.79		274190.27	274190.27	2011
22	JUDY MELINEK	ASSISTANT MEDICA...	257510.44	377.21	15883.56		273771.21	273771.21	2011

here I’m taking a look at the data and trying to collect information with SELECT statements.

```
(base) si76ra@sira-IdeaPad:~/Desktop/salaries$ sqlite3 database.sqlite
SQLite version 3.41.2 2023-03-22 11:56:21
Enter ".help" for usage hints.
sqlite> .tables
Salaries
sqlite> SELECT * FROM Salaries LIMIT 10 ;
1|NATHANIEL FORD|GENERAL MANAGER-METROPOLITAN TRANSIT AUTHORITY|167411.18|0|400184.25||567595.43|567
595.43|2011||San Francisco|
2|GARY JIMENEZ|CAPTAIN III (POLICE DEPARTMENT)|155966.02|245131.88|137811.38||538909.28|538909.28|20
11||San Francisco|
3|ALBERT PARDINI|CAPTAIN III (POLICE DEPARTMENT)|212739.13|106088.18|16452.6||335279.91|335279.91|20
11||San Francisco|
4|CHRISTOPHER CHONG|WIRE ROPE CABLE MAINTENANCE MECHANIC|77916|56120.71|198306.9||332343.61|332343.6
1|2011||San Francisco|
5|PATRICK GARDNER|DEPUTY CHIEF OF DEPARTMENT,(FIRE DEPARTMENT)|134401.6|9737|182234.59||326373.19|32
6373.19|2011||San Francisco|
6|DAVID SULLIVAN|ASSISTANT DEPUTY CHIEF II|118602|8601|189082.74||316285.74|316285.74|2011||San Fran
cisco|
7|ALSON LEE|BATTALION CHIEF, (FIRE DEPARTMENT)|92492.01|89062.9|134426.14||315981.05|315981.05|2011|
|San Francisco|
8|DAVID KUSHNER|DEPUTY DIRECTOR OF INVESTMENTS|256576.96|0|51322.5||307899.46|307899.46|2011||San Fr
ancisco|
9|MICHAEL MORRIS|BATTALION CHIEF, (FIRE DEPARTMENT)|176932.64|86362.68|40132.23||303427.55|303427.55
|2011||San Francisco|
10|JOANNE HAYES-WHITE|CHIEF OF DEPARTMENT, (FIRE DEPARTMENT)|285262|0|17115.73||302377.73|302377.73|
2011||San Francisco|
sqlite> SELECT COUNT(DISTINCT JobTitle) FROM Salaries ;
2159
sqlite> SELECT AVG(BasePay) FROM Salaries WHERE Status='PT';
31744.644285714
sqlite>
```

here I’m trying to find out inconsistencies in the data and error values.

```
sqlite> SELECT JobTitle ,COUNT(*) AS jobs
   ...> FROM Salaries
   ...> GROUP BY BasePAy HAVING BasePay>500000 LIMIT 10;
Deputy Chief 3|605
Not provided|4
sqlite> SELECT COUNT(*)
FROM Salaries
WHERE BasePay < 0 ;
11
sqlite> SELECT COUNT(*) FROM Salaries WHERE BasePay < 0 ;
11
sqlite> ALTER TABLE Salaries ADD ErrorFlag INTEGER;
sqlite> UPDATE Salaries
   ...> SET ErrorFlag=1 WHERE BasePay <0 ;
sqlite> SELECT * FROM Salaries WHERE BasePay < 0 LIMIT 5 ;
72833|Irwin Sidharta|Junior Clerk|-166.01|249.02|0|6.56|83.01|89.57|2012||San Francisco||1
72866|Robert Scott|Junior Clerk|-121.63|182.7|0|5.44|61.07|66.51|2012||San Francisco||1
72873|Chung Huey Kung|Junior Clerk|-109.22|163.83|0|4.32|54.61|58.93|2012||San Francisco||1
72875|Jordan Li|Junior Clerk|-106.6|159.9|0|4.66|53.3|57.96|2012||San Francisco||1
72879|Richard Jackson|Junior Clerk|-101.88|153.08|0|4.55|51.2|55.75|2012||San Francisco||1
sqlite> SELECT COUNT(*) FROM Salaries
   ...> WHERE Benefits IS NULL OR Benefits ='' OR Benefits = 'Not Provided';
36163
```

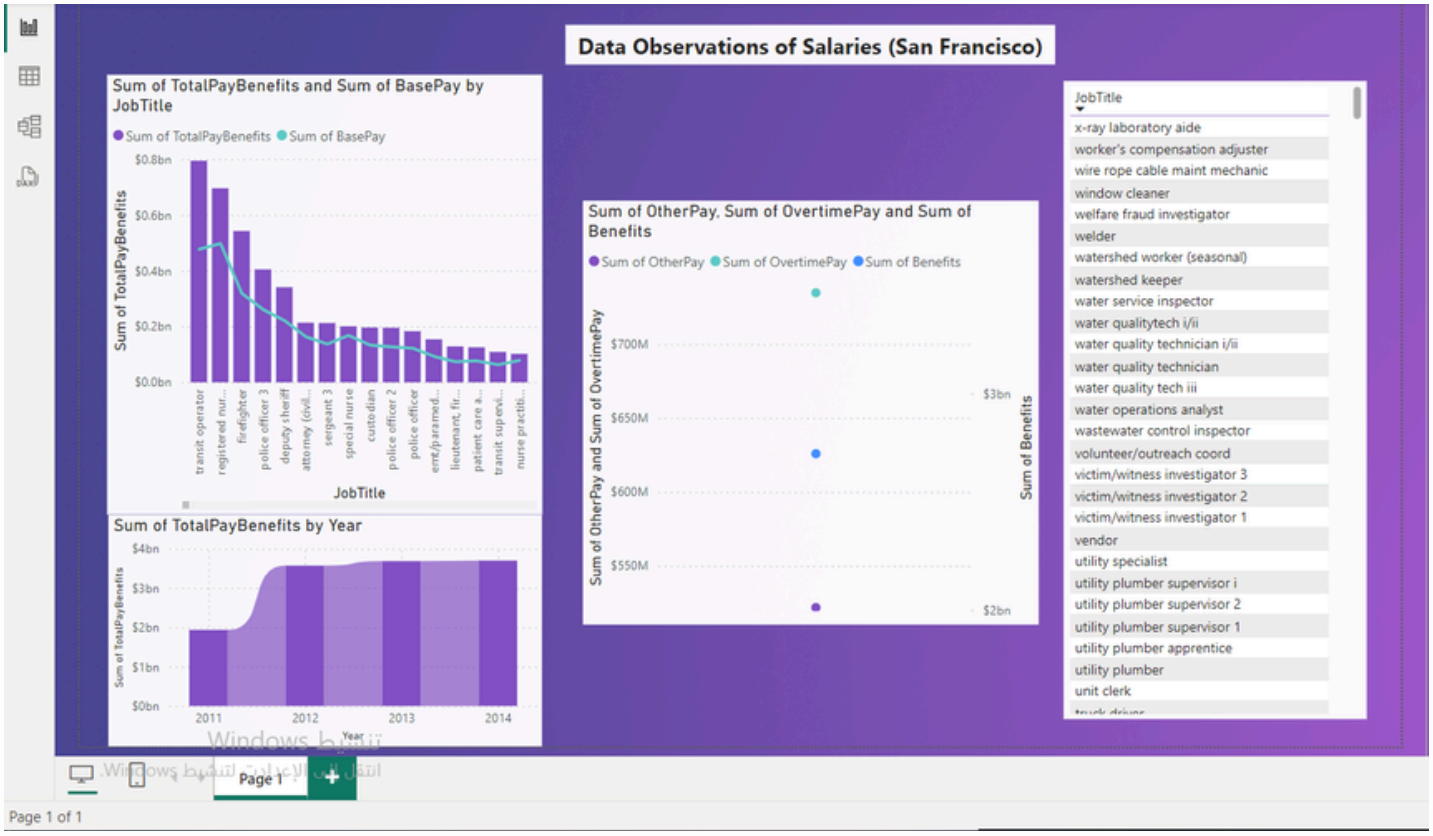
I started a transaction to add a flag to some error values.
changed some of them then committed those changes.

```
sqlite> BEGIN TRANSACTION ;
sqlite> UPDATE Salaries SET BenefitS = 0 WHERE Benefits = '';
sqlite> UPDATE Salaries SET BasePay = ABS(BasePay) WHERE BasePay <0 ;
sqlite> COMMIT ;
sqlite> SELECT * FROM Salaries WHERE BasePay < 0 LIMIT 5 ;
sqlite> SELECT COUNT(*) FROM Salaries WHERE Benefits IS NULL OR Benefits = '' OR Benefits = 'Not Provided';

4
sqlite> SELECT * FROM Salaries WHERE Benefits IS NULL OR Benefits = '' OR Benefits = 'Not Provided';
148647|Not provided|Not provided|Not Provided|Not Provided|Not Provided|Not Provided|0|0|2014||San Francisco||
148651|Not provided|Not provided|Not Provided|Not Provided|Not Provided|Not Provided|0|0|2014||San Francisco||
148652|Not provided|Not provided|Not Provided|Not Provided|Not Provided|Not Provided|0|0|2014||San Francisco||
148653|Not provided|Not provided|Not Provided|Not Provided|Not Provided|Not Provided|0|0|2014||San Francisco||
sqlite>
```

Excel and power bi

To a programmer
excel and power bi are complementary tools
I use excel to speed up a task and power bi to create visualized dashboards of data .
and I often use developers tools with them.
here is a visualization I created with power bi for the same dataset.



manipulating the sqlite dataset with advanced data transform.