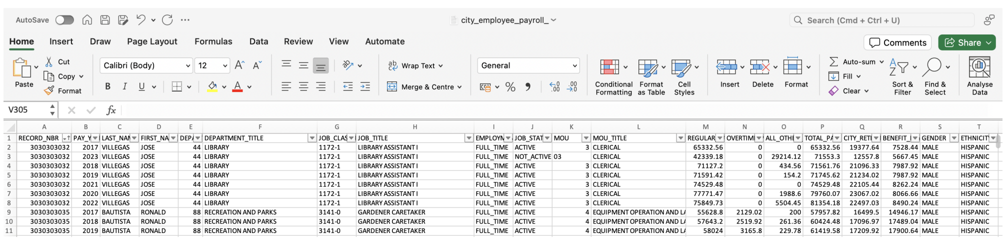
MTH 765P Mini-project

Harshit Saxena

15 Jan 2024

1 Introduction

The example dataset of Los Angeles City Payroll is used in this mini-project. The purpose of this project is to clean, process, analyze, and visualize payroll datasets with the belief that payroll dataset analysis methods can be extended to other similar collections of data as well. This mini-project's comma-separated file was downloaded from Kaggle. Information overview, *see figure* 1.



	RECORD_NBR	PAY_Y	LAST_NAM	FIRST_NA	DEP	DEPARTMENT_TITLE	JOB_CLAS	JOB_TITLE	EMPLOYM	JOB_STAT	MOU	MOU_TITLE	REGULAR	OVERTIM	ALL_OTH	TOTAL_PA	CITY_RET	BENEFIT_	GENDER	ETHNICITY
2	3030303032	2017	VILLEGAS	JOSE	44	LIBRARY	1172-1	LIBRARY ASSISTANT I	FULL_TIME	ACTIVE	3	CLERICAL	65332.56	0	0	65332.56	19377.64	7528.44	MALE	HISPANIC
3	3030303032	2023	VILLEGAS	JOSE	44	LIBRARY	1172-1	LIBRARY ASSISTANT I	FULL_TIME	NOT_ACTIVE	03	CLERICAL	42339.18	0	29214.12	71553.3	12557.8	5667.45	MALE	HISPANIC
4	3030303032	2018	VILLEGAS	JOSE	44	LIBRARY	1172-1	LIBRARY ASSISTANT I	FULL_TIME	ACTIVE	3	CLERICAL	71127.2	0	434.56	71561.76	21096.33	7987.92	MALE	HISPANIC
5	3030303032	2019	VILLEGAS	JOSE	44	LIBRARY	1172-1	LIBRARY ASSISTANT I	FULL_TIME	ACTIVE	3	CLERICAL	71591.42	0	154.2	71745.62	21234.02	7987.92	MALE	HISPANIC
6	3030303032	2021	VILLEGAS	JOSE	44	LIBRARY	1172-1	LIBRARY ASSISTANT I	FULL_TIME	ACTIVE	3	CLERICAL	74529.48	0	0	74529.48	22105.44	8262.24	MALE	HISPANIC
7	3030303032	2020	VILLEGAS	JOSE	44	LIBRARY	1172-1	LIBRARY ASSISTANT I	FULL_TIME	ACTIVE	3	CLERICAL	77771.47	0	1988.6	79760.07	23067.02	8066.66	MALE	HISPANIC
8	3030303032	2022	VILLEGAS	JOSE	44	LIBRARY	1172-1	LIBRARY ASSISTANT I	FULL_TIME	ACTIVE	3	CLERICAL	75849.73	0	5504.45	81354.18	22497.03	8490.24	MALE	HISPANIC
9	3030303035	2017	BAUTISTA	RONALD	88	RECREATION AND PARKS	3141-0	GARDENER CARETAKER	FULL_TIME	ACTIVE	4	EQUIPMENT OPERATION AND LA	55628.8	2129.02	200	57957.82	16499.5	14946.17	MALE	HISPANIC
10	3030303035	2018	BAUTISTA	RONALD	88	RECREATION AND PARKS	3141-0	GARDENER CARETAKER	FULL_TIME	ACTIVE	4	EQUIPMENT OPERATION AND LA	57643.2	2519.92	261.36	60424.48	17096.97	17489.04	MALE	HISPANIC
11	3030303035	2019	BAUTISTA	RONALD	88	RECREATION AND PARKS	3141-0	GARDENER CARETAKER	FULL_TIME	ACTIVE	4	EQUIPMENT OPERATION AND LA	58024	3165.8	229.78	61419.58	17209.92	17900.64	MALE	HISPANIC

Figure 1: *Overview of CSV file*

2 Obtaining/Acquiring the Data

A brief, Kaggle is a collaborative platform for data science and machine learning that provides a varied array of datasets that users can download (as XLS, CSV, and many other forms) or access via APIs. This portal is a wonderful resource for accessing, studying, and utilizing multiple datasets for learning and innovation in the field of data science. Dataset: Las Angeles Payroll.

Payroll information for all Los Angeles City employees, includes the city's three proprietary departments: Water and Power, Airports, and Harbor. The Los Angeles City Controller's Office refreshes data biweekly. Payroll information for Department of Water and Power personnel is updated every three months. The original dataset on Kaggle was obtained from ControllerData.LaCity.Org

3 Description / Pre-Processing

This project's data collection contains 20 columns and 501,552 records. There are 14 dimensional columns and 6 factual columns among the 20 columns. Contents that summarize the dataset's column information are as follow:

1. **Record_NBR:** Unique number to identify an employee
2. **Pay_Year:** Tax year employee was paid. This is not fiscal year
3. **Last_Name:** Employee last name

4. **First_Name:** Employee first name
5. **Department_No:** Department number in city payroll system
6. **Department_Title:** Title of city department
7. **Job_Class_Pgrade:** Job class and pay grade
8. **Job_Title:** Job Title
9. **Employment_Type:** Employment Type Full Time, Par Time or Per Event
10. **Job_Status:** Employeeś job status at the time the data was uploaded
11. **MOU:** Memorandum of Understanding
12. **MOU_Title:** Title of Memorandum of Understanding
13. **Regular_Pay:** Regular work hours payment
14. **Overtime_Pay:** Payments attribute to hours worked beyond regular work schedule
15. **All_Other_Pay:** Any payments other than Regular and Overtime
16. **Total_Pay:** Sum of regular, overtime and all other payments
17. **City_Retirement_Contributions:** Estimated payments made by the city towards employee's retirement
18. **Benefit_Pay:** City contribution for the employeeś health care, dental care, vision care and life insurance
19. **Gender:** Gender as self-reported by employee
20. **Ethnicity:** Ethnicity as self-reported by employee

Effective exploratory data analysis relies significantly on the meticulous processes of data cleaning, transformation and pre-processing. Initially, data cleaning, is imperative to identify and rectify errors, inconsistencies and missing values, safeguarding the reliability of subsequent analyses. Subsequently, data transformation ensures that raw data is appropriately formatted for analysis, involving actions such as scaling variables and encoding categorical data. Collectively, these steps contribute to the quality of exploratory data analysis by establishing a foundation of accuracy, proper formatting and alignment with chosen analytical methods for meaningful insights and informed decision-making.

Los Angeles City Payroll Dataset had various inconsistencies which required data cleaning and transformation before the same could be utilised for analysis and visualisation. Initially, commands such as head(), tail() *see figure* 2, describe() and info() *see figure* 3 of pandas dataframe were executed for understanding errors in the loaded CSV file.

Based on preliminary observations column All_Other_Pay & Total_Pay displayed values in scientific notation. Furthermore individual values present in OVERTIME_PAY, City_Retirement_Contributions, Benefit_Pay, Regular_Pay, All_Other_Pay and Total_Pay had to be modified to 0.2 decimals for ease of visual interpretation. *see figure* 4

Another data cleaning check of null values is executed as null values should be eliminated from a dataset to prevent distortion of statistical measures, ensuring compatibility with algorithms that cannot handle missing values and maintain the overall accuracy and reliability of insights by reducing noise and uncertainty. *see figure* 5. Also, checking for duplicate records in exploratory data analysis (EDA) is vital to prevent biases, ensuring accurate representation of data, and enhance the performance of statistical models by eliminating redundancy. *see figure* 6

```
In [2]: LACity_Payroll_df = pd.read_csv('city_employee_payroll_.csv',low_memory=False)
```

```
In [3]: LACity_Payroll_df.head()
```

Out[3]:

MOU	MOU_TITLE	REGULAR_PAY	OVERTIME_PAY	ALL_OTHER_PAY	TOTAL_PAY	CITY_RETIREMENT_CONTRIBUTIONS	BENEFIT_PAY	GENDER	ETHNICITY
24	POLICE OFFICERS, LIEUTENANT AND BELOW	91626.85	5835.35	2241.75	99703.95	27042.10	16819.00	MALE	HISPANIC
3	CLERICAL	65332.56	0.00	0.00	65332.56	19377.64	7528.44	MALE	HISPANIC
27	LOS ANGELES PORT POLICE COMMAND OFFICERS	184110.40	0.00	1750.00	185860.40	86255.72	6518.28	MALE	CAUCASIAN
4	EQUIPMENT OPERATION AND LABOR	55628.80	2129.02	200.00	57957.82	16499.50	14946.17	MALE	HISPANIC
21	TECHNICAL	132912.00	0.00	350.00	133262.00	39421.70	6446.75	FEMALE	HISPANIC

```
In [4]: LACity_Payroll_df.tail()
```

Out[4]:

MOU	MOU_TITLE	REGULAR_PAY	OVERTIME_PAY	ALL_OTHER_PAY	TOTAL_PAY	CITY_RETIREMENT_CONTRIBUTIONS	BENEFIT_PAY	GENDER	ETHNICITY
23	FIREFIGHTERS AND FIRE CAPTAINS	3318.40	0.00	0.0	3318.40	984.24	0.00	MALE	TWO OR MORE RACES
15	SERVICE EMPLOYEES	1618.55	0.00	35.0	1653.55	480.06	0.00	MALE	NaN
	EQUIPMENT								

Figure 2: *Data Frame - Head / Tail*

```
In [5]: LACity_Payroll_df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 501552 entries, 0 to 501551
Data columns (total 20 columns):
 #   Column                         Non-Null Count   Dtype  
---  ------                         --------------   -----  
 0   RECORD_NBR                     501552 non-null  object 
 1   PAY_YEAR                       501552 non-null  int64  
 2   LAST_NAME                      501542 non-null  object 
 3   FIRST_NAME                     501552 non-null  object 
 4   DEPARTMENT_NO                  501552 non-null  int64  
 5   DEPARTMENT_TITLE               501552 non-null  object 
 6   JOB_CLASS_PGRADE               501322 non-null  object 
 7   JOB_TITLE                      501322 non-null  object 
 8   EMPLOYMENT_TYPE                501552 non-null  object 
 9   JOB_STATUS                     501552 non-null  object 
 10  MOU                            501322 non-null  object 
 11  MOU_TITLE                      501206 non-null  object 
 12  REGULAR_PAY                    501552 non-null  float64
 13  OVERTIME_PAY                   501552 non-null  float64
 14  ALL_OTHER_PAY                  501552 non-null  float64
 15  TOTAL_PAY                      501552 non-null  float64
 16  CITY_RETIREMENT_CONTRIBUTIONS  501552 non-null  float64
 17  BENEFIT_PAY                    501552 non-null  float64
 18  GENDER                         501551 non-null  object 
 19  ETHNICITY                      494856 non-null  object 
dtypes: float64(6), int64(2), object(12)
memory usage: 76.5+ MB
```

```
In [6]: LACity_Payroll_df.describe()
```

Out[6]:

	PAY_YEAR	DEPARTMENT_NO	REGULAR_PAY	OVERTIME_PAY	ALL_OTHER_PAY	TOTAL_PAY	CITY_RETIREMENT_CONTRIBUTIONS	BENEFIT_PAY
count	501552.000000	501552.000000	501552.000000	501552.000000	5.015520e+05	5.015520e+05	501552.000000	501552.000000
mean	2019.930476	65.691486	67781.198128	10836.744109	5.520667e+03	8.413861e+04	19075.823101	10647.492580
std	2.001931	29.496858	50288.436506	23398.729245	1.260215e+04	6.679212e+04	20588.304831	8679.692954

Figure 3: *Data Frame - Describe / Info*

In [7]:
```
'''
    Based on dataframe description.
    We observe that following column need to be transformed in order to modify scientific notation.
        1. ALL_OTHER_PAY
        2. TOTAL_PAY
'''
#Convert column to numeric
LACity_Payroll_df['ALL_OTHER_PAY'] = pd.to_numeric(LACity_Payroll_df['ALL_OTHER_PAY'], errors='coerce')
#Transforming each numeric value to string two decimal
LACity_Payroll_df['ALL_OTHER_PAY'] = LACity_Payroll_df['ALL_OTHER_PAY'].apply(lambda x: '{:.2f}'.format(x))
#Modifying the column back to float
LACity_Payroll_df['ALL_OTHER_PAY'] = LACity_Payroll_df['ALL_OTHER_PAY'].astype(float)

LACity_Payroll_df['TOTAL_PAY'] = pd.to_numeric(LACity_Payroll_df['TOTAL_PAY'], errors='coerce')
LACity_Payroll_df['TOTAL_PAY'] = LACity_Payroll_df['TOTAL_PAY'].apply(lambda x: '{:.2f}'.format(x))
LACity_Payroll_df['TOTAL_PAY'] = LACity_Payroll_df['TOTAL_PAY'].astype(float)

#Transforming each numeric value to string two decimal & modifying column dtype back to float
LACity_Payroll_df['OVERTIME_PAY'] = LACity_Payroll_df['OVERTIME_PAY'].apply(lambda x: '{:.2f}'.format(x))
LACity_Payroll_df['OVERTIME_PAY'] = LACity_Payroll_df['OVERTIME_PAY'].astype(float)

#Transforming each numeric value to string two decimal & modifying column dtype back to float
LACity_Payroll_df['REGULAR_PAY'] = LACity_Payroll_df['REGULAR_PAY'].apply(lambda x: '{:.2f}'.format(x))
LACity_Payroll_df['REGULAR_PAY'] = LACity_Payroll_df['REGULAR_PAY'].astype(float)

#Transforming each numeric value to string two decimal & modifying column dtype back to float
LACity_Payroll_df['CITY_RETIREMENT_CONTRIBUTIONS'] = LACity_Payroll_df['CITY_RETIREMENT_CONTRIBUTIONS'].apply(la
LACity_Payroll_df['CITY_RETIREMENT_CONTRIBUTIONS'] = LACity_Payroll_df['CITY_RETIREMENT_CONTRIBUTIONS'].astype(f

#Transforming each numeric value to string two decimal & modifying column dtype back to float
LACity_Payroll_df['BENEFIT_PAY'] = LACity_Payroll_df['BENEFIT_PAY'].apply(lambda x: '{:.2f}'.format(x))
LACity_Payroll_df['BENEFIT_PAY'] = LACity_Payroll_df['BENEFIT_PAY'].astype(float)
```

Figure 4: *Datatype modification*

```
#Checking for null values in the dataframe
LACity_Payroll_df.isnull().sum()
```

```
RECORD_NBR                          0
PAY_YEAR                            0
LAST_NAME                          10
FIRST_NAME                          0
DEPARTMENT_NO                       0
DEPARTMENT_TITLE                    0
JOB_CLASS_PGRADE                  230
JOB_TITLE                         230
EMPLOYMENT_TYPE                     0
JOB_STATUS                          0
MOU                               230
MOU_TITLE                         346
REGULAR_PAY                         0
OVERTIME_PAY                        0
ALL_OTHER_PAY                       0
TOTAL_PAY                           0
CITY_RETIREMENT_CONTRIBUTIONS       0
BENEFIT_PAY                         0
GENDER                              1
ETHNICITY                        6696
dtype: int64
```

Figure 5: *Nullvalues DataFrame*

```
duplicate_rows_df = LACity_Payroll_df[LACity_Payroll_df.duplicated()]
print("Number of duplicate rows: ", duplicate_rows_df.shape)

#No duplicate records to remove
```

```
Number of duplicate rows:  (0, 20)
```

Figure 6: *Duplicate Records*

4 Analysis

Analyzing and visualization are critical components of exploratory data analysis (EDA), as they provide crucial insights into the underlying patterns and trends in a dataset. The use of statistical measures and procedures to comprehend the central patterns, distributions, and relationships between variables is referred to as analysis. This is supplemented by visualization, which presents data in graphical representations such as charts, plots and graphs, making complex patterns more accessible and assisting in the detection of outliers or trends. Analysis and visualization in EDA work together to identify essential data properties, supporting informed decision-making and leading to more in-depth investigations as needed.

Payroll Datasets could be analysed and interpreted using various mythologies. The scope of this project is to observe patterns in the dataset chosen and convey a storey-line / interpretation from the set of information with the aid of visuals.

Firstly, the observations / records of the dataset are plotted using histogram to display the spread of data.*see figure* 7. This allows identification of any outlier information present in the dataset such as columns - OVERTIME_PAY, City_Retirement_Contributions, Benefit_Pay, Regular_Pay, All_Other_Pay and Total_Pay contain significant number of **0 (zero) values**.

Secondly, a high-level analysis is implemented by observing the relationship between Regular_Pay vs Pay_Year. This allows us to detect if there has been any increase, decrease or stagnation of salary paid to city employees over the course of years (20172023). Based on the scatterplot, we may interpret that regular pay has **increased marginally** over the years; however, this is a general interpretation as further filters / detailed analysis is required to generate a conclusion.*see figure* 8

Thirdly, more accurate method of analysing different components of salary over the years is to plot using a line graph. Before implementing the same, I created a new dataframe with filtering information such as EmploymentType = FullTime, JobStatus = Active and Department = Police. We may observe that **Benefit Pay and has not changed in the last 6 years** and **Regular Pay has also not changed post 2021**. However, **Total Pay has increased due to increase in Overtime Pay from 2021**.*see figure* 9

In addition, with dataframe created in the last analysis I drill down further by comparing salary of Police personnel on active payroll based on gender. Pay vs Gender distribution analysis provides crucial information about the disparity in salary between Male and Female. As observed salary of Male employees in the Los Angeles police department has increased over the years, salary of female employees has not seen significant increase. Also, there is a huge pay gap between both genders that has not been bridged till the end of the dataset.*see figure* 10

Libraries Used:

1. Pandas
2. Matplotlib.Pyplot
3. Seaborn
4. Matplotlib.ticker (Only ScalarFormatter)

```
list(set(LACity_Payroll_df.dtypes.tolist())) #Extracting unique datatypes present in the dataframe
Payroll_Df_num = LACity_Payroll_df.select_dtypes(include = ['float64', 'int64']) #Select only float & int dtype
Payroll_Df_num.head()

#Creating histogram of dimnesion vs count of observation in each bin. Analysis of the same allows findining out
Payroll_Df_num.hist(figsize=(15, 8), bins=100, xlabelsize=8, ylabelsize=8);

#';' used to avoid
```

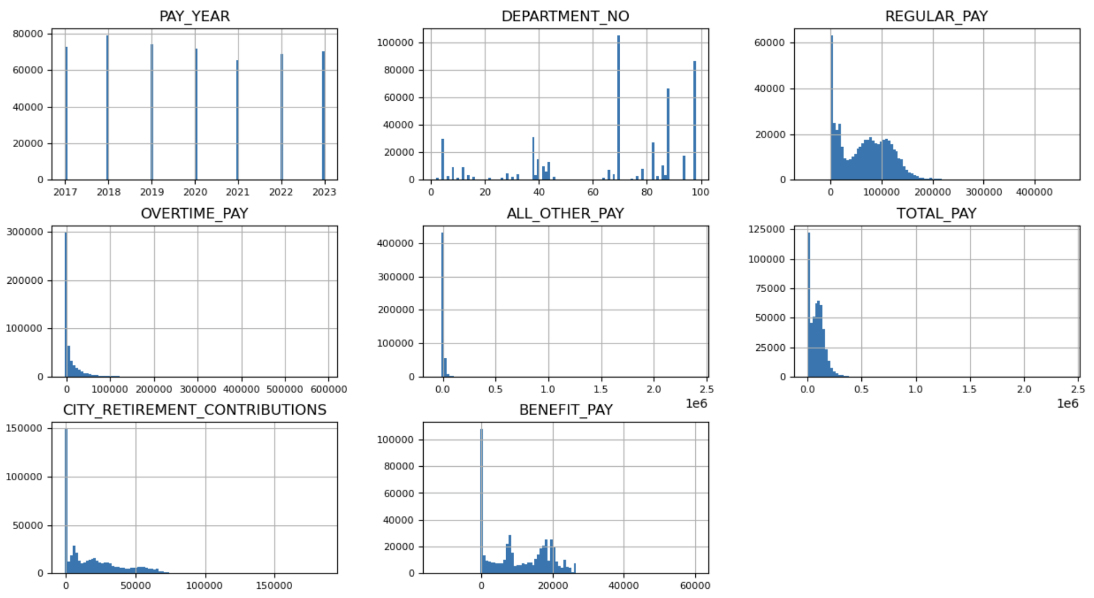


Figure 7: *Observations vs Dimensions*

In [14]:
```
'''
    Implemnting scatter plot to analyse relationship between PAY_YEAR vs REGULAR_PAY & TOTAL_PAY
'''
plt.figure(figsize = (8,3))
plt.scatter(LACity_Payroll_df['PAY_YEAR'], LACity_Payroll_df['REGULAR_PAY'])
plt.xlabel('PAY_YEAR')
plt.ylabel('REGULAR_PAY')
plt.show()

plt.figure(figsize = (8,3))
plt.scatter(LACity_Payroll_df['PAY_YEAR'], LACity_Payroll_df['TOTAL_PAY'])
plt.xlabel('PAY_YEAR')
plt.ylabel('TOTAL_PAY')
plt.show()
```

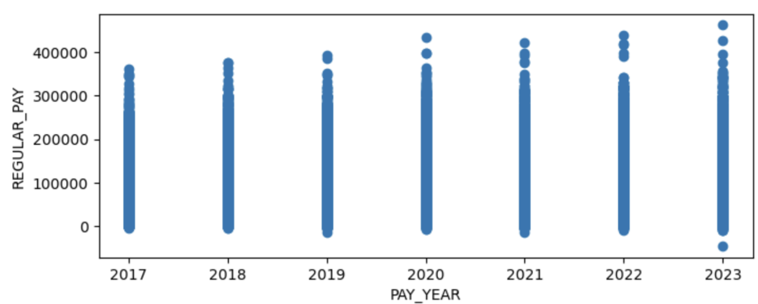


Figure 8: *RegularPay vs PerYear*

In [17]:

```
#police_active_data = LACity_Payroll_df[(LACity_Payroll_df['DEPARTMENT_TITLE'] == 'POLICE') & (LACity_Payroll_df
sns.lineplot(data = Police_Active_df, x = 'PAY_YEAR', y = 'TOTAL_PAY', label = 'Total Pay',markers = True)
sns.lineplot(data = Police_Active_df, x = 'PAY_YEAR', y = 'REGULAR_PAY', label = 'Regular Pay',markers = True)
sns.lineplot(data = Police_Active_df, x = 'PAY_YEAR', y = 'OVERTIME_PAY', label = 'Overtime Pay',markers = True)
sns.lineplot(data = Police_Active_df, x = 'PAY_YEAR', y = 'BENEFIT_PAY', label = 'Benefit Pay',markers = True)

# Adding legend, axis labels, and title
plt.legend()
plt.xlabel('Pay Year')
plt.ylabel('Pay Amount')
plt.title('Pay Components Over the Years for Active Police Department')
plt.tight_layout()
plt.show()
```

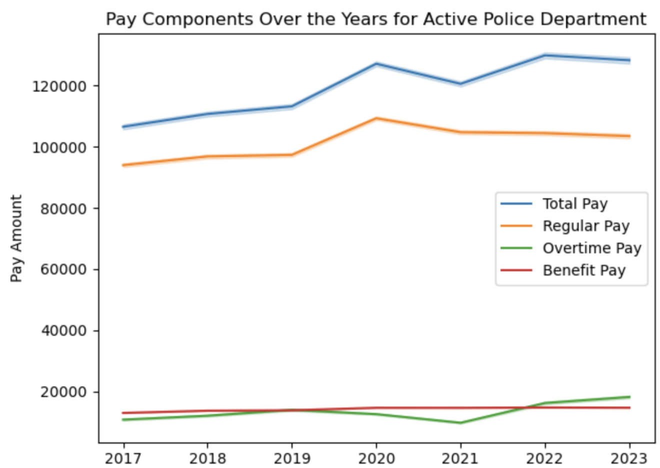


Figure 9: *Police Employee Pay Per Year*

In [15]:

```
'''
    Creating a new dataframe for the below mentioned filters, to analysis salary accordingly.
    Filters:
        1. Department Title = Police
        2. Job Status = Active
        3. Employment Type = Full Time
        4. Gender = Male / Female / INVALID INPUT
        5. Pay Year = 2019 - 2022
'''

Police_Active_df = LACity_Payroll_df[(LACity_Payroll_df['DEPARTMENT_TITLE']=='POLICE') & (LACity_Payroll_df['JOB

gender_order = ['MALE', 'FEMALE', 'INVALID INPUT']
fig, axes = plt.subplots(1, 6, figsize=(20, 5), sharey=True)
for i, year in enumerate([2018,2019,2020,2021,2022, 2023]):
    ax = sns.barplot(x='GENDER', y='TOTAL_PAY',data=Police_Active_df[Police_Active_df['PAY_YEAR']==year],estimat
    ax.set_title(f'Total Pay by Gender - {year}')
    ax.set_xlabel('Gender')
    ax.set_ylabel('Total Pay')
plt.tight_layout()
plt.show()
#print(Police_Active_df.shape)
```

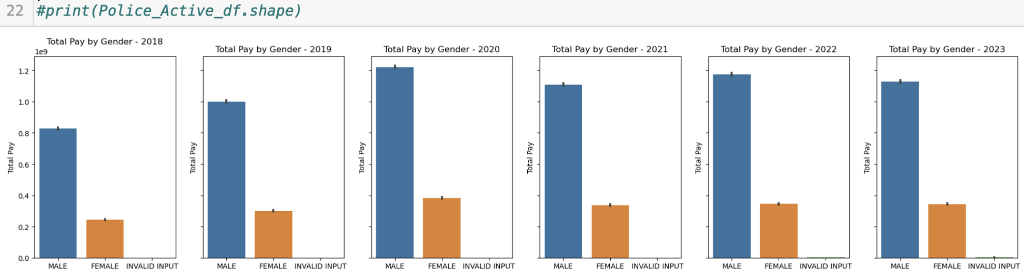


Figure 10: *Police Gender Pay Per Year*

5 Questions

The adaptability of exploratory data analysis (EDA) is clear because it provides a plethora of functions and approaches for extracting critical insights from varied datasets, including the Payroll dataset in question. One can obtain a full grasp of the dataset by performing a variety of activities such as descriptive statistics, correlation analysis, outlier detection, and even utilizing machine learning models. EDA is an effective technique for identifying patterns, correlations, and anomalies, setting the groundwork for informed decision-making and further in-depth investigations when needed. Because of the dynamic nature of EDA, it is adaptable to specific issue statements, making it an essential element in the data analysis process.

With respect to Payroll dataset, following approach could be considered for a detailed analysis:

1. Larger dataset with additional dimensions (such as: Joining date / tenure period of employees, work experience of employees)

2. Perform mathematical / statistical calculations to infer any relationship between Pay and Other Dimensions (such as: Work experience, Gender, Ethnicity)

3. Detailed time-series analysis based on various other parameters (including other departments on Los Angeles Payroll) can be implemented for time-dependent data, analyze trends, seasonality, and cyclical patterns to gain insights.