Data description

1. This dataset chosen is “Mexican Federal Government Salaries” from the Kaggle website: https://www.kaggle.com/datasets/ivansabik/mexican-federal-government-salaries
2. This dataset has 1978155 rows × 11 columns

df

	entidadfederativa	sujetoobligado	nombre	denominacion	montoneto	cargo	area	montobruto	idInformacion	periodoreportainici
0	Hidalgo	Jaltocán	Adolfo Hernandez Hernandez	Fontanero	4000.00	Fontanero	OBRAS PUBLICAS	4254.00	16311845	01/01/201
1	Ciudad de México	Secretaría de Salud	ARELY SAMANTA CLEOFAS VELASCO	AUXILIAR DE ENFERMERIA "A"	12177.86	AUXILIAR DE ENFERMERIA "A"	H.G. ENRIQUE CABRERA	16092.00	16480190	01/01/201
2	Ciudad de México	Secretaría de Seguridad Ciudadana	MELODY OLIMPIC GONZALEZ MONTES	POLICIA PRIMERO	11652.00	POLICIA PRIMERO	SUBSECRETARIA DE OPERACION POLICIAL	16030.00	17599078	01/01/202
3	Federación	Autoridad Educativa Federal en la Ciudad de Mé...	ANGEL ALLENDE PULIDO	APOYO Y ASISTENCIA A LA EDUCACION	10180.57	APOYO Y ASISTENCIA A LA EDUCACION	DIRECCIÓN GENERAL DE OPERACIONES DE SERVICIOS ...	2910.65	6514612	01/07/201
4	Aguascalientes	MUNICIPIO DE RINCÓN DE ROMOS	Yolanda Reyes Gonzalez	DIRECTOR	17004.40	DIRECTOR	ACCION CIVICA	6188.40	11927166	01/07/201
...	...	...	...	...	...	...	...	...	...	
1978150	Ciudad de México	Policía Auxiliar	Nancy Fanny Rivera Martinez	Operativo	12119.57	Operativo	Sector 52	13766.68	18698843	01/04/201
1978151	Guerrero	Secretaría de Educación Guerrero (SEG)	MONICA IGNACIA AGUILAR RODRIGUEZ	JEFE DE OFICINA ...	NaN	JEFE DE OFICINA ...	DIRECCION GENERAL DE LOS SERVICIOS ESTATALES D...	9882.12	16658324	01/01/201
1978152	Chiapas	Secretaría/Instituto de Salud	SEBASTIAN GOMEZ SANTIZ	TECNICO EN ATENCION PRIMARIA A LA SALUD	7978.94	TECNICO EN ATENCION PRIMARIA A LA SALUD	OFICINA JURISDICCIONAL (PICHUCALCO)	20300.52	11695594	01/01/202
1978153	Chiapas	Secretaría de Seguridad y Protección Ciudadana	Santana Reyes Gómez	Policía Segundo	6190.95	Policía Segundo	División de la Policía de Servicios	6190.95	12038916	01/04/201
1978154	Federación	Pemex Transformación Industrial (TRI)	JOSE ALFREDO GONZALEZ MORENO	OPERARIO DE PRIMERA (DIVERSOS OFICIOS)	14344.02	OPERARIO DE PRIMERA (DIVERSOS OFICIOS)	GERENCIA DE REFINERÍA DE CADEREYTA	16595.33	1511822	01/01/201

1978155 rows × 11 columns

3. Data types of all the columns are below

```
#to get the data type
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1978155 entries, 0 to 1978154
Data columns (total 11 columns):
 #   Column                Dtype
---  ------                -----
 0   entidadfederativa     object
 1   sujetoobligado        object
 2   nombre                object
 3   denominacion          object
 4   montoneto             float64
 5   cargo                 object
 6   area                  object
 7   montobruto            float64
 8   idInformacion         int64
 9   periodoreportainicio  object
 10  periodoreportafin     object
dtypes: float64(2), int64(1), object(8)
memory usage: 166.0+ MB
```

4. Checked the missing value in the dataset

```
# to check for missing null values
df.isnull().sum()
```

```
entidadfederativa           0
sujetoobligado              0
nombre                  31618
denominacion            52799
montoneto               89176
cargo                   52799
area                    33218
montobruto              39422
idInformacion               0
periodoreportainicio        0
periodoreportafin           0
dtype: int64
```

5. Percentage of missing data in the dataset

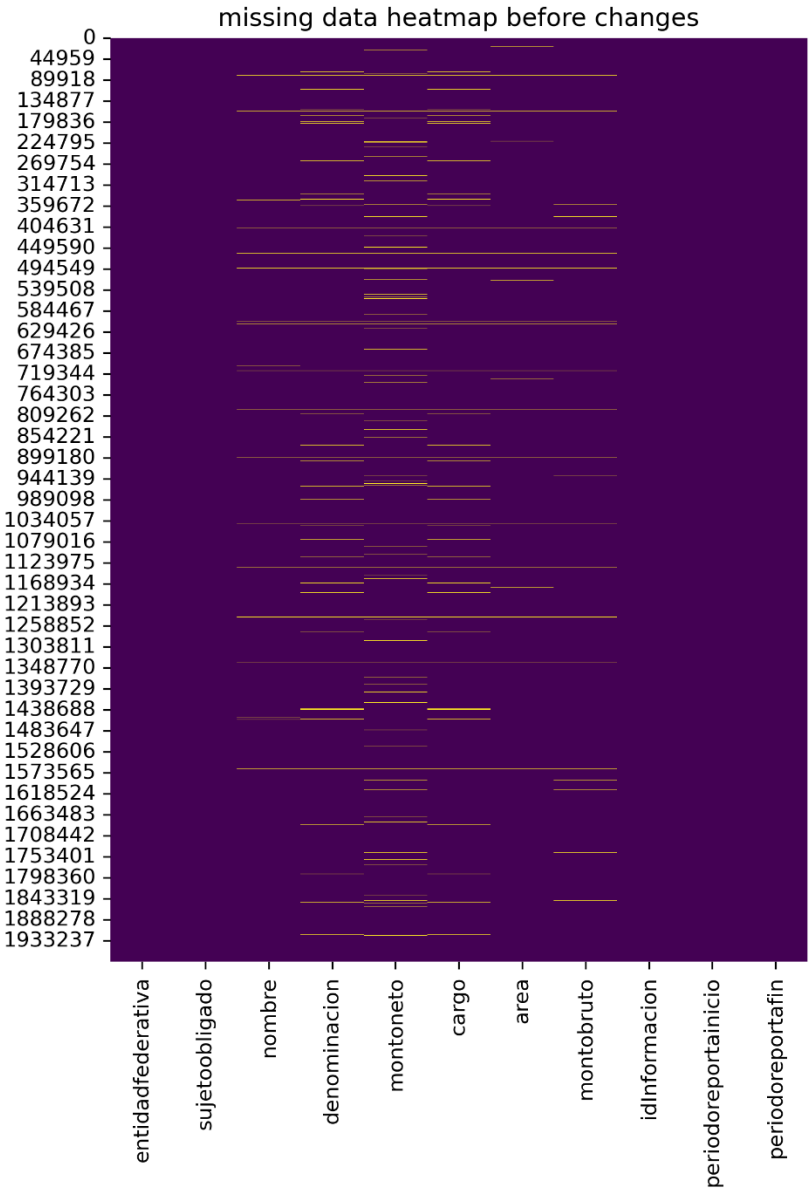
```
#checking the percentage of missing data of the columns
(df.isnull().sum() / len(df)) * 100
```

```
entidadfederativa       0.000000
sujetoobligado          0.000000
nombre                  1.598358
denominacion            2.669103
montoneto               4.508039
cargo                   2.669103
area                    1.679242
montobruto              1.992867
idInformacion           0.000000
periodoreportainicio    0.000000
periodoreportafin       0.000000
dtype: float64
```

6. Heatmap of missing data of original dataset

```
#visualising the missing data on heat map
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(6,8))
sns.heatmap(df.isnull(),cbar=False,cmap='viridis')
plt.title("missing data heatmap before changes")
plt.savefig('heatmap_with_missing_data.png', dpi=300, bbox_inches='tight')
plt.show()
```

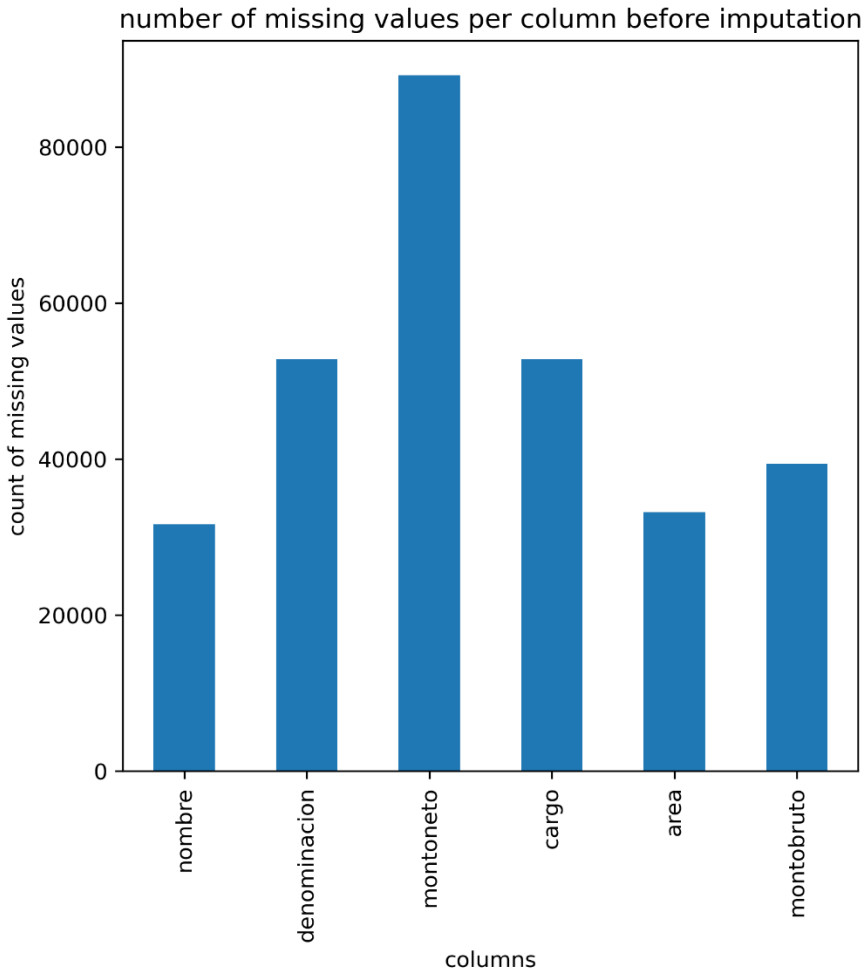


7. Bar plot of missing data of original data set

```
missing_values_in_data = df.isnull().sum()
```

```
missing_values_in_data = missing_values_in_data[missing_values_in_data > 0]  #selecting columns who have missing values
```

```
missing_values_in_data.plot(kind='bar', figsize=(6, 6))
plt.title('number of missing values per column before imputation')
plt.ylabel('count of missing values')
plt.xlabel('columns')
plt.savefig('bar_plot_of_original_dataset.png', dpi=300, bbox_inches='tight')
plt.show()
```



8. As per analysis there's missing data in six columns however only columns **montoneto** and **montobruto** have data type float64 where imputation can be done for data wrangling
9. Imputation in montoneto column

```
#montoneto is float64 data type we can use mean of this
```

```
#calculate the average of the column
avg_montoneto = df['montoneto'].astype("float").mean(axis=0)
```

```
print("Average of montoneto:",avg_montoneto)
```

```
Average of montoneto: 13533.628811659637
```

Replace "NaN" by mean value by the mean value in "montoneto" column

```
df['montoneto'].replace(np.nan,avg_montoneto,inplace = True)
```

10. Imputation in **montobruto** column

```
avg_montobruto = df['montobruto'].astype("float").mean(axis=0)
```

```
print("Average of montobruto:",avg_montobruto)
```

```
Average of montobruto: 16779.699533329236
```

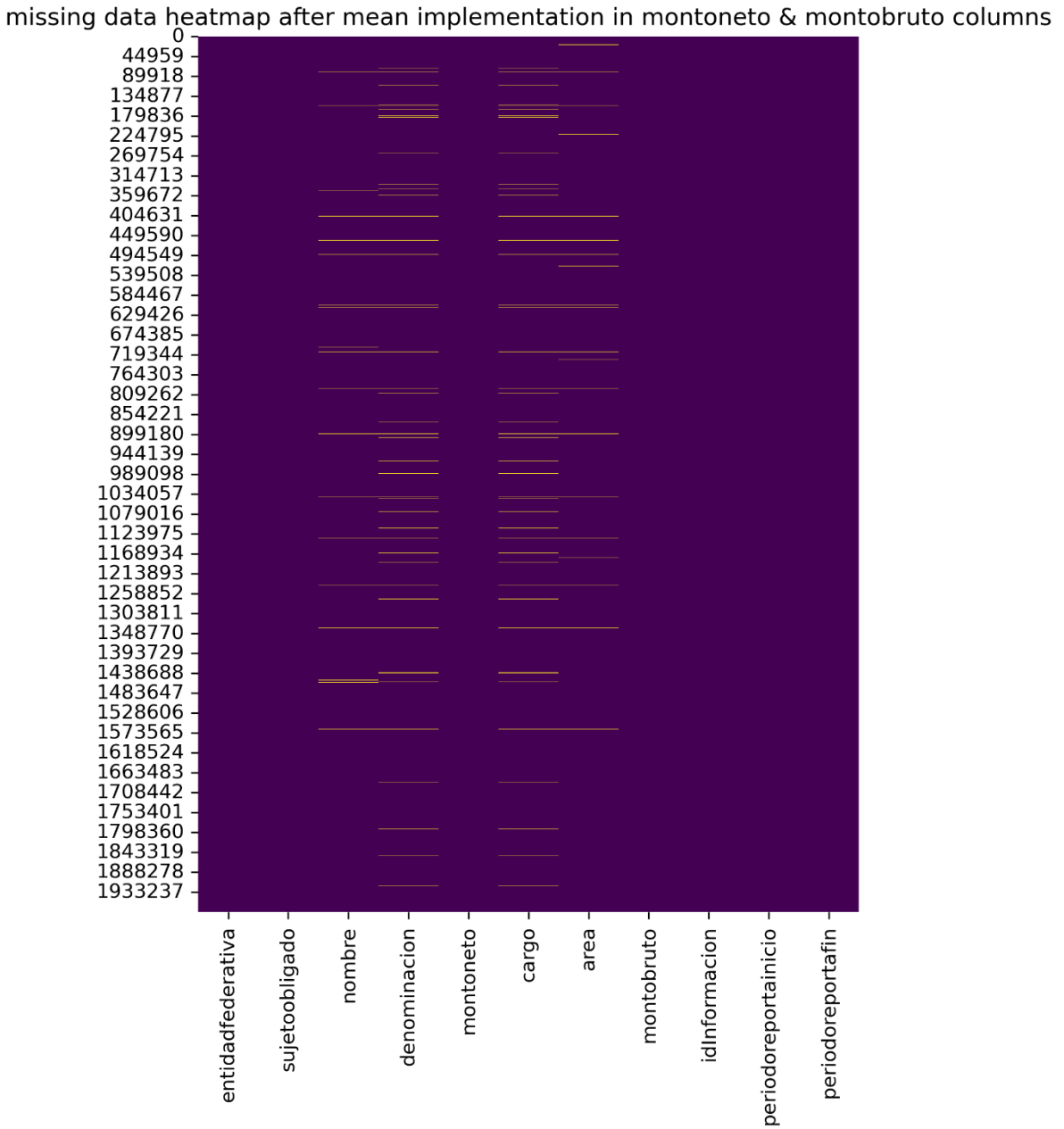
Replace "NaN" by mean value by the mean value in "montobruto" column

```
df['montobruto'].replace(np.nan,avg_montobruto,inplace = True)
```

11. Heat map after data imputataion

```
#visualising the missing data on heat map
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(6,8))
sns.heatmap(df.isnull(),cbar=False,cmap='viridis')
plt.title("missing data heatmap after mean implementation in montoneto & montobruto columns")
plt.savefig('heatmap_after_imputation_of_data.png', dpi=300, bbox_inches='tight')
plt.show()
```

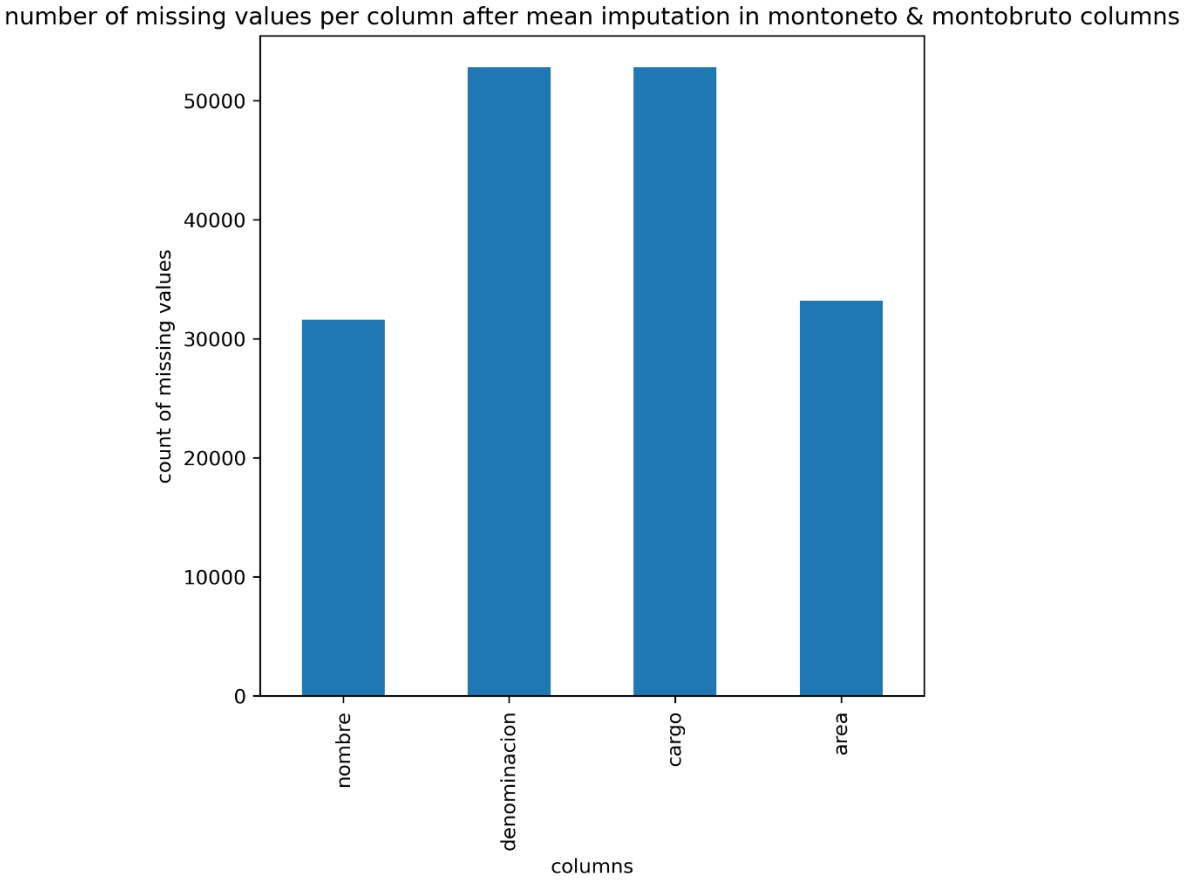


12. Bar Plot after imputation of data set

```
missing_values_in_data = df.isnull().sum()
```

```
missing_values_in_data = missing_values_in_data[missing_values_in_data > 0]  #selecting columns who have missing values
```

```
missing_values_in_data.plot(kind='bar', figsize=(6, 6))
plt.title('number of missing values per column after mean imputation in montoneto & montobruto columns')
plt.ylabel('count of missing values')
plt.xlabel('columns')
plt.savefig('bar_plot_after_imputation_of_dataset.png', dpi=300, bbox_inches='tight')
plt.show()
```



13. Dataset description after Data Wrangling

```
df.describe()
```

	montoneto	montobruto	idInformacion
count	1.978155e+06	1.978155e+06	1.978155e+06
mean	1.353363e+04	1.677970e+04	1.564438e+07
std	1.413076e+05	2.418874e+04	1.185418e+07
min	-2.285230e+05	-2.586643e+05	6.600000e+01
25%	6.269365e+03	7.873810e+03	8.781188e+06
50%	1.008599e+04	1.238634e+04	1.425915e+07
75%	1.493299e+04	1.862635e+04	1.833432e+07
max	1.947336e+08	7.559543e+06	7.753209e+07

```
#percentage of data after imputation of the columns
(df.isnull().sum() / len(df)) * 100
```

```
entidadfederativa       0.000000
sujetoobligado          0.000000
nombre                  1.598358
denominacion            2.669103
montoneto               0.000000
cargo                   2.669103
area                    1.679242
montobruto              0.000000
idInformacion           0.000000
periodoreportainicio    0.000000
periodoreportafin       0.000000
dtype: float64
```

14. Why mean imputation? This amount of dataset preserves central tendency that's why mean represents the best option for missing numeric values.
15. For categorical data with data type "object" we can change the missing values to "unknown" to make similarities for estimation.

```
#Impute missing values in columns which has data type as object
categorical_columns = ['nombre', 'denominacion', 'cargo', 'area']
for col in categorical_columns:
    df[col].fillna('Unknown', inplace=True)
```

```
(df.isnull().sum() / len(df)) * 100
```

```
entidadfederativa       0.0
sujetoobligado          0.0
nombre                  0.0
denominacion            0.0
montoneto               0.0
cargo                   0.0
area                    0.0
montobruto              0.0
idInformacion           0.0
periodoreportainicio    0.0
periodoreportafin       0.0
dtype: float64
```