Data Transformation with Python or R

To my peer Data Scientist,

In this, I have done data transformations with Python with my data i.e. us_county.

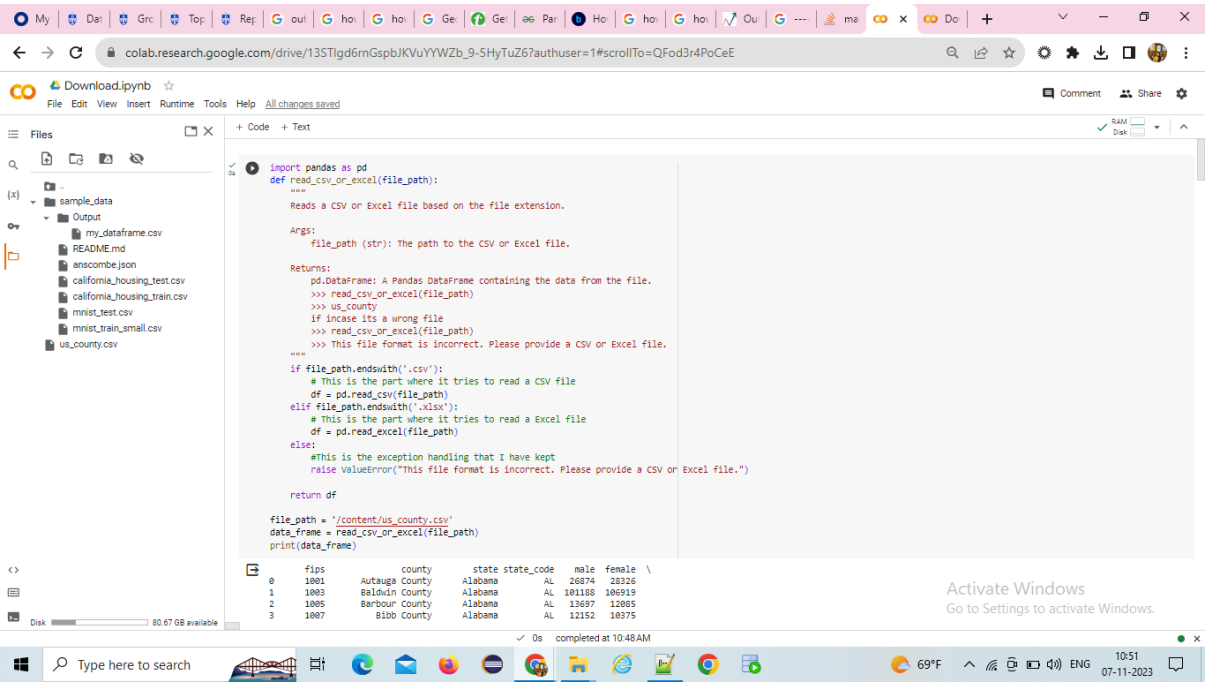
Introduction:

The dataset that I have taken gives detailed information about the country, state, male, female, age group, and demographics information such as latitude and longitude. The datasets clearly mentioned that the United States of America had the most reported COVID-19 cases. **The main objective of this analysis is** to find out the patterns within the dataset to get a further understanding of the data. I also wanted to leverage it to choose a machine algorithm for predicting the survival rate of patients during the period of COVID-19. The dataset consists of demographic information population information (Such as male and female rates) and age information. In my previous dataset, I used a raw dataset so in this I am trying to clean the dataset by deleting unwanted rows and columns and by doing certain transformations. **US_COUNTY.CSV**

DATASET LINK:https://drive.google.com/drive/folders/1RfLhJVOK45x9oGBmOKyZEpBAaHuITYaw

Data attributes: Fips, County, State, State code, male, female, median age, population, female percentage, Lat, long. So, in total my dataset has 3220 rows * 11 columns. The columns have a title/heading, which makes them readable.

Step 1: Loading the data into the data frame.



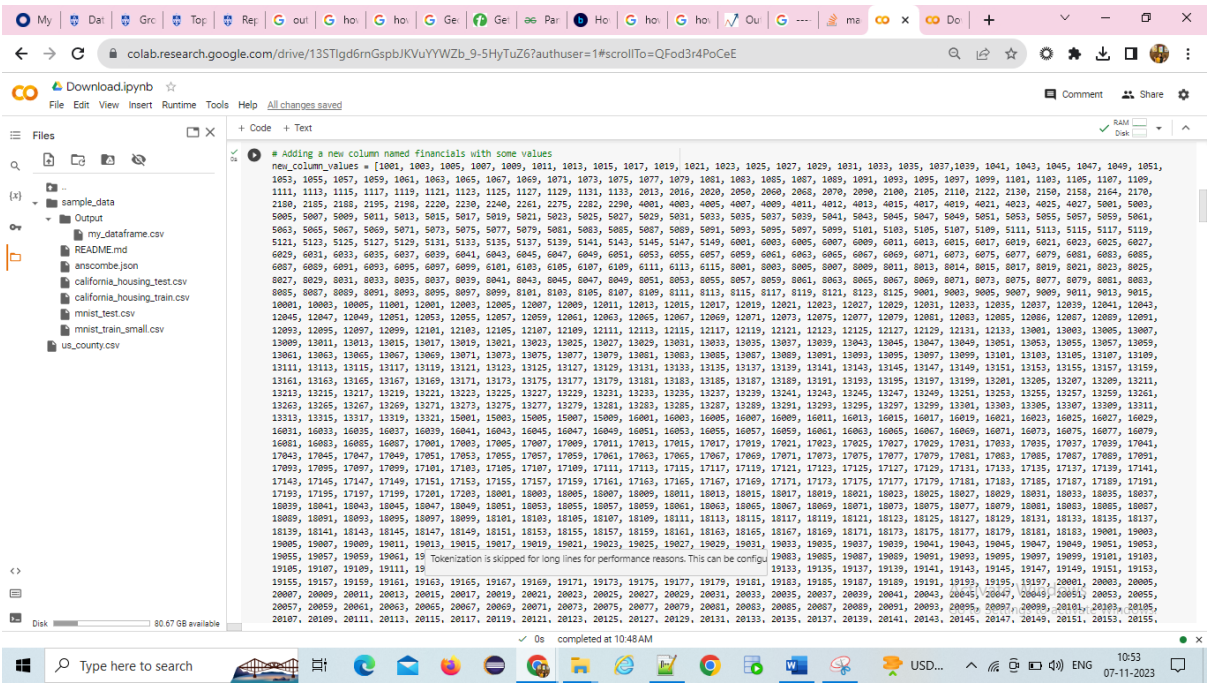
Step 2: After running the code it will display the data that is there in CSV file.

```
       fips              county        state state_code    male  female  \
0      1001       Autauga County      Alabama         AL   26874   28326
1      1003       Baldwin County      Alabama         AL  101188  106919
2      1005       Barbour County      Alabama         AL   13697   12085
3      1007          Bibb County      Alabama         AL   12152   10375
4      1009        Blount County      Alabama         AL   28434   29211
...     ...                  ...          ...        ...     ...     ...
3215  72145  Vega Baja Municipio  Puerto Rico        NaN   25580   27791
3216  72147    Vieques Municipio  Puerto Rico        NaN    4332    4439
3217  72149   Villalba Municipio  Puerto Rico        NaN   11169   11824
3218  72151    Yabucoa Municipio  Puerto Rico        NaN   16541   17608
3219  72153      Yauco Municipio  Puerto Rico        NaN   17475   18964

      median_age  population  female_percentage        lat       long
0           37.8       55200          51.315217  32.534923 -86.642730
1           42.8      208107          51.376936  30.727479 -87.722564
2           39.9       25782          46.873788  31.869581 -85.393210
3           39.9       22527          46.055844  32.998628 -87.126475
4           40.8       57645          50.673953  33.980869 -86.567380
...          ...         ...                ...        ...        ...
3215        40.7       53371          52.071350  18.428461 -66.397926
3216        43.6        8771          50.609965  18.122662 -65.439095
3217        38.8       22993          51.424347  18.128155 -66.472816
3218        42.5       34149          51.562271  18.070468 -65.896311
3219        43.0       36439          52.043141  18.079728 -66.858276

[3220 rows x 11 columns]
```

Step 3: Next I am adding a new column financial data.



Step 4: A new column has been added successfully.

```
New column added successfully!!!!
0        1001.0
1        1003.0
2        1005.0
3        1007.0
4        1009.0
         ...
3215    72145.0
3216    72147.0
3217    72149.0
3218        NaN
3219    72153.0
Name: financialdata, Length: 3220, dtype: float64
```

Step 5: Now I am trying to change the datatype from integer to float.

```
[6] # Changing the data type of column 'financialdata' from integer to float
    data_frame['financialdata'] = data_frame['financialdata'].astype(float)
```

Step 6: After executing this code, we can clearly see that the datatype has been changed. Initially, it was an integer now it is changed to float.

```
#this is the code to know the datatypes of all the columns
data_frame.dtypes
```

```
fips                   int64
county                object
state                 object
state_code            object
male                   int64
female                 int64
median_age           float64
population             int64
female_percentage    float64
lat                  float64
long                 float64
financialdata        float64
dtype: object
```

Step 7: This is the output for the datatype that is been changed.

```
[8] #Output for the datatype that is changed from int64 to float64
    print(data_frame.financialdata)
```

```
0        1001.0
1        1003.0
2        1005.0
3        1007.0
4        1009.0
         ...
3215    72145.0
3216    72147.0
3217    72149.0
3218        NaN
3219    72153.0
Name: financialdata, Length: 3220, dtype: float64
```

Step 8: Now I am trying to insert some values for the NaN values in my data.

```
# Fill missing values with a specific value i.e. with 0 and in the financial data column the only place that i inserted 0 is before the last value
#And I also added zeros in certain state_code column as it was NaN
data_frame.fillna({'financialdata': 0 , 'state_code': 0})
```

	fips	county	state	state_code	male	female	median_age	population	female_percentage	lat	long	financialdata
0	1001	Autauga County	Alabama	AL	26874	28326	37.8	55200	51.315217	32.534923	-86.642730	1001.0
1	1003	Baldwin County	Alabama	AL	101188	106919	42.8	208107	51.376936	30.727479	-87.722564	1003.0
2	1005	Barbour County	Alabama	AL	13697	12085	39.9	25782	46.873788	31.869581	-85.393210	1005.0
3	1007	Bibb County	Alabama	AL	12152	10375	39.9	22527	46.055844	32.998628	-87.126475	1007.0
4	1009	Blount County	Alabama	AL	28434	29211	40.8	57645	50.673953	33.980869	-86.567380	1009.0
...	...	...	...	...	...	...	...	...	...	...	...	...
3215	72145	Vega Baja Municipio	Puerto Rico	0	25580	27791	40.7	53371	52.071350	18.428461	-66.397926	72145.0
3216	72147	Vieques Municipio	Puerto Rico	0	4332	4439	43.6	8771	50.609965	18.122662	-65.439095	72147.0
3217	72149	Villalba Municipio	Puerto Rico	0	11169	11824	38.8	22993	51.424347	18.128155	-66.472816	72149.0
3218	72151	Yabucoa Municipio	Puerto Rico	0	16541	17608	42.5	34149	51.562271	18.070468	-65.896311	0.0
3219	72153	Yauco Municipio	Puerto Rico	0	17475	18964	43.0	36439	52.043141	18.079728	-66.858276	72153.0

3220 rows × 12 columns

Step 9: Now I am trying to rename my columns male and female into male count and female count respectively.

```
[10] # Here I am trying to Rename the 'male' to 'male_count' and 'female' to 'female_count'
     data_frame = data_frame.rename(columns={'male': 'male_count','female': 'female_count'})
```

Step 10: This is the output for changing the column name.

```
[11] #This is the output for rename columns
     column_headings = data_frame.columns
     print(column_headings)

     Index(['fips', 'county', 'state', 'state_code', 'male_count', 'female_count',
            'median_age', 'population', 'female_percentage', 'lat', 'long',
            'financialdata'],
           dtype='object')
```

Step 11: Here I am trying to do some operations with two columns of my data

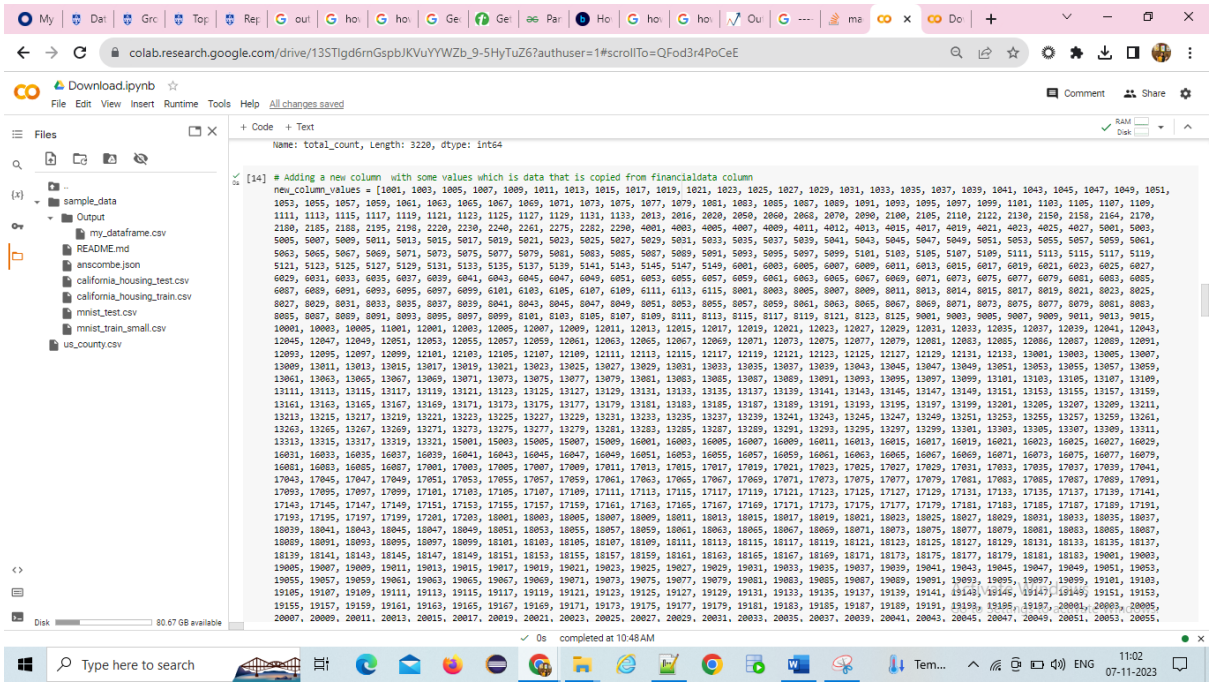
```
[12] # here i am trying to create a new column 'total_count' as the sum of columns 'male_count' and 'female_count'
     data_frame['total_count'] = data_frame['male_count'] + data_frame['female_count']
```

Step 12: This is the output of creating a calculated field based on two other columns

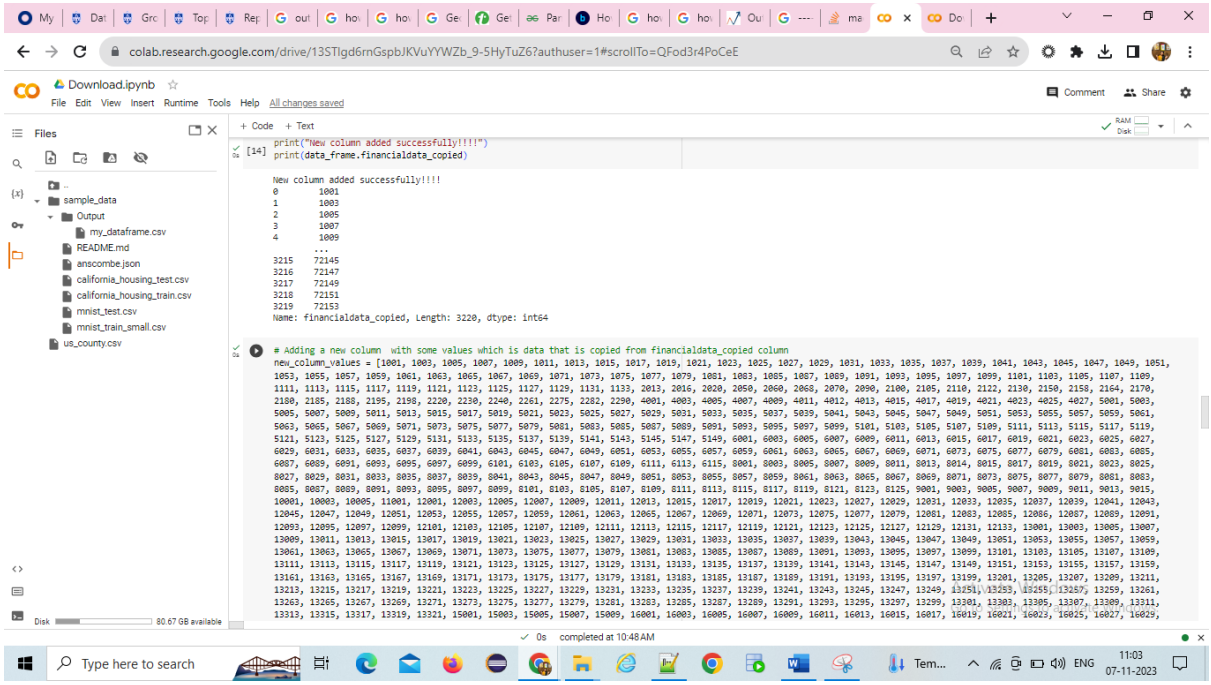
```
[13] #this is the output of creating a calculated field based on two other columns
     print(data_frame.total_count)

     0        55200
     1       208107
     2        25782
     3        22527
     4        57645
              ...
     3215     53371
     3216      8771
     3217     22993
     3218     34149
     3219     36439
     Name: total_count, Length: 3220, dtype: int64
```

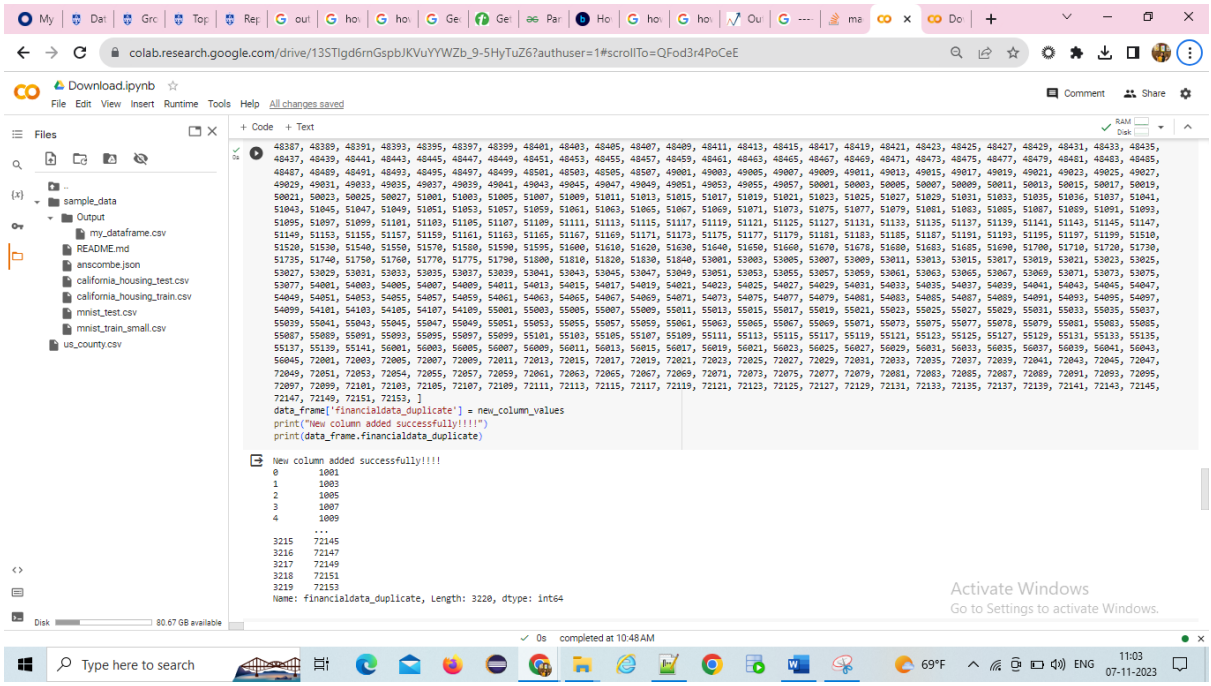
Step 13: Now I trying to duplicate a column that I have created



Step 14: For safety purposes, I am adding one more duplicate column so that I can delete it



Step 15: The reason why I do this is I don't want to delete my essential columns so I created a replica of the dummy column that I have created so that I could be able to drop the column without affecting my original data.



Step 16: Now two duplicate columns, has been inserted and I also performed delete function.

```
[16] column_headings = data_frame.columns
     print(column_headings)
```

```
Index(['fips', 'county', 'state', 'state_code', 'male_count', 'female_count',
       'median_age', 'population', 'female_percentage', 'lat', 'long',
       'financialdata', 'total_count', 'financialdata_copied',
       'financialdata_duplicate'],
      dtype='object')
```

```
[17] #Here I am trying to drop the 'financialdata_duplicate' and 'financialdata_copied' column using the del statement
     del data_frame['financialdata_duplicate']
```

```
[18] #Here I am trying to drop the 'financialdata_duplicate' and 'financialdata_copied' column using the del statement
     del data_frame[ 'financialdata_copied']
```

```
#Here we can clear know that the two unwanted columns such as financialdata_duplicate and financialdata_copied is deleted successfully!!!
column_headings = data_frame.columns
print(column_headings)
```

```
Index(['fips', 'county', 'state', 'state_code', 'male_count', 'female_count',
       'median_age', 'population', 'female_percentage', 'lat', 'long',
       'financialdata', 'total_count'],
      dtype='object')
```

Step 17: I am trying to save my file in my specified file path where I can have a copy of my data.

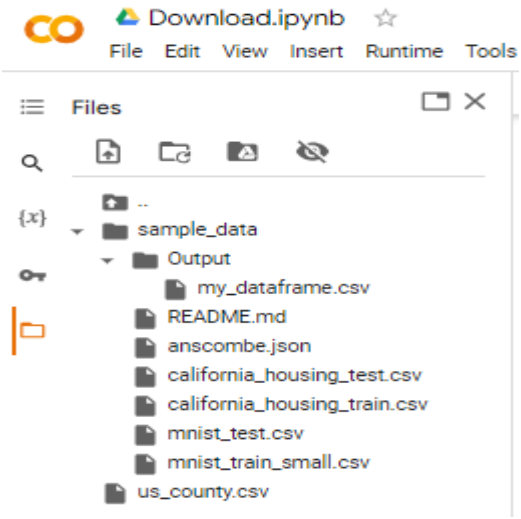
```
import pandas as pd

df = pd.DataFrame(data_frame)

# Specify the path where you want to save the CSV file
file_path = '/content/sample_data/Output/my_dataframe.csv'

# Save the DataFrame to the specified location
df.to_csv(file_path, index=False)
```

Step 18: This is the place where I am trying to save my data.



Step 19: Here I am trying to standardize my data.

```python
import pandas as pd
from sklearn.preprocessing import StandardScaler

# Here i am trying to standardize only two columns that is fips and population column
numeric_columns = data_frame[['fips', 'population']]

# here i am trying to initialize the StandardScaler
scaler = StandardScaler()
numeric_standardized = scaler.fit_transform(numeric_columns)
df_standardized = pd.DataFrame(numeric_standardized, columns=numeric_columns.columns)

print("Original DataFrame:")
print(data_frame)
print("\nStandardized DataFrame (numeric columns only):")
print(df_standardized)
```

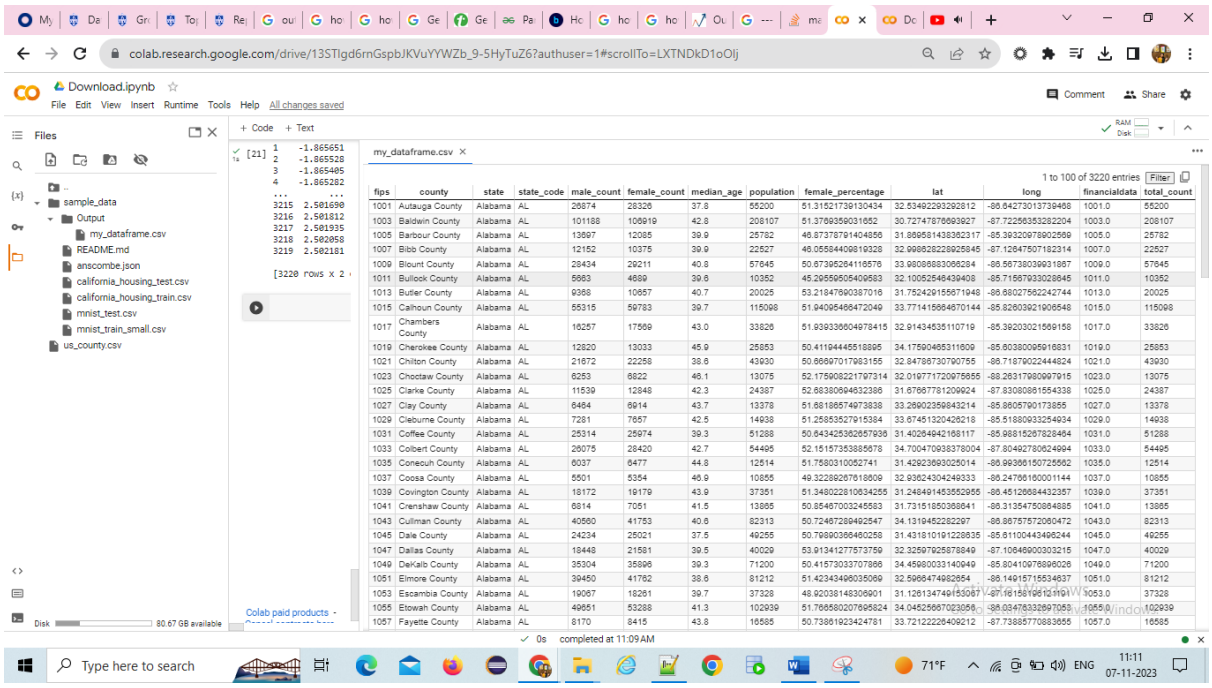
```
Original DataFrame:
       fips              county        state state_code  male_count  \
0      1001       Autauga County      Alabama         AL       26874
1      1003       Baldwin County      Alabama         AL      101188
2      1005       Barbour County      Alabama         AL       13697
3      1007          Bibb County      Alabama         AL       12152
4      1009        Blount County      Alabama         AL       28434
...     ...                  ...          ...        ...         ...
3215  72145  Vega Baja Municipio  Puerto Rico        NaN       25580
3216  72147    Vieques Municipio  Puerto Rico        NaN        4332
3217  72149   Villalba Municipio  Puerto Rico        NaN       11169
3218  72151    Yabucoa Municipio  Puerto Rico        NaN       16541
3219  72153      Yauco Municipio  Puerto Rico        NaN       17475

      female_count  median_age  population  female_percentage        lat  \
0            28326        37.8       55200          51.315217  32.534923
1           106919        42.8      208107          51.376936  30.727479
2            12085        39.9       25782          46.873788  31.869581
3            10375        39.9       22527          46.055844  32.998628
4            29211        40.8       57645          50.673953  33.980869
...            ...         ...         ...                ...        ...
3215         27791        40.7       53371          52.071350  18.428461
3216          4439        43.6        8771          50.609965  18.122662
3217         11824        38.8       22993          51.424347  18.128155
3218         17608        42.5       34149          51.562271  18.070468
3219         18964        43.0       36439          52.043141  18.079728

            long  financialdata  total_count
0    -86.642730         1001.0        55200
1    -87.722564         1003.0       208107
2    -85.393210         1005.0        25782
3    -87.126475         1007.0        22527
4    -86.567380         1009.0        57645
...         ...            ...          ...
3215 -66.397926        72145.0        53371
3216 -65.439095        72147.0         8771
3217 -66.472816        72149.0        22993
3218 -65.896311            NaN        34149
3219 -66.858276        72153.0        36439

[3220 rows x 13 columns]
```

Step 20: This is the dataset that I have received after all the transformation and clean-up of data.



fips	county	state	state_code	male_count	female_count	median_age	population	female_percentage	lat	long	financialdata	total_count
1001	Autauga County	Alabama	AL	26874	28326	37.8	55200	51.31521739130434	32.53492293292812	-86.64273013739468	1001.0	55200
1003	Baldwin County	Alabama	AL	101188	106919	42.8	208107	51.37693590316652	30.72747876693927	-87.72256353282204	1003.0	208107
1005	Barbour County	Alabama	AL	13997	12085	39.9	25782	46.87378791404856	31.86958143836237	-85.39320978902569	1005.0	25782
1007	Bibb County	Alabama	AL	12152	10375	39.9	22527	46.05584409819328	32.99862822892585	-87.12647507182314	1007.0	22527
1009	Blount County	Alabama	AL	28434	29211	40.8	57645	50.67395264116576	33.98086883066284	-86.56738039931987	1009.0	57645
1011	Bullock County	Alabama	AL	5663	4689	39.6	10352	45.29559505409583	32.10052546439408	-85.71567933028645	1011.0	10352
1013	Butler County	Alabama	AL	9368	10657	40.7	20025	53.21847690387016	31.75242915571948	-86.68027562242744	1013.0	20025
1015	Calhoun County	Alabama	AL	55315	59783	39.7	115098	51.94095466472049	33.77141566467014	-85.82603921906548	1015.0	115098
1017	Chambers County	Alabama	AL	16257	17569	43.0	33826	51.93933660497841	32.91434535110719	-85.39203021569158	1017.0	33826
1019	Cherokee County	Alabama	AL	12820	13033	45.9	25853	50.41194445518895	34.17590465311609	-85.60380095916631	1019.0	25853
1021	Chilton County	Alabama	AL	21672	22258	38.6	43930	50.66697017983155	32.84786730790755	-86.71879022444824	1021.0	43930
1023	Choctaw County	Alabama	AL	6253	6822	46.1	13075	52.17590822179731 4	32.01977172097565 5	-88.26317980997915	1023.0	13075
1025	Clarke County	Alabama	AL	11539	12848	42.3	24387	52.68380694632386	31.67667812209924	-87.83080861554338	1025.0	24387
1027	Clay County	Alabama	AL	6464	6914	43.7	13378	51.68186574973838	33.26902359843214	-85.8605790173855	1027.0	13378
1029	Cleburne County	Alabama	AL	7281	7657	42.5	14938	51.25853527915384	33.67451320426218	-85.51880933254934	1029.0	14938
1031	Coffee County	Alabama	AL	25314	25974	39.3	51288	50.64342536265793 6	31.40264942168117	-85.98815267828464	1031.0	51288
1033	Colbert County	Alabama	AL	26075	28420	42.7	54495	52.15157353885678	34.70047093837804	-87.80492780624994	1033.0	54495
1035	Conecuh County	Alabama	AL	6037	6477	44.8	12514	51.7580310052741	31.42923693025014	-86.99366150725562	1035.0	12514
1037	Coosa County	Alabama	AL	5501	5354	46.9	10855	49.32289267618609	32.93624304249333	-86.24766160001144	1037.0	10855
1039	Covington County	Alabama	AL	18172	19179	43.9	37351	51.34802281063425 5	31.24849145355295 5	-86.45126684432357	1039.0	37351
1041	Crenshaw County	Alabama	AL	6814	7051	41.5	13865	50.85467003245583	31.73151850368641	-86.31354750864885	1041.0	13865
1043	Cullman County	Alabama	AL	40560	41753	40.6	82313	50.72467289492547	34.1319452282297	-86.86757572060472	1043.0	82313
1045	Dale County	Alabama	AL	24234	25021	37.5	49255	50.79890366460258	31.43181019122835	-85.61100443496244	1045.0	49255
1047	Dallas County	Alabama	AL	18448	21581	39.5	40029	53.91341277573759	32.32597925878849	-87.10846900303215	1047.0	40029
1049	DeKalb County	Alabama	AL	35304	35896	39.3	71200	50.41573033707866	34.45980033140949	-85.80410976896026	1049.0	71200
1051	Elmore County	Alabama	AL	39450	41762	38.6	81212	51.42343496035089	32.5966474982654	-86.14915715534637	1051.0	81212
1053	Escambia County	Alabama	AL	19067	18261	39.7	37328	48.92038148306901	31.12613474915067	-87.61158796121101	1053.0	37328
1055	Etowah County	Alabama	AL	49651	53288	41.3	102939	51.76658020769582 4	34.04525667023056	-86.03478332897053	1055.0	102939
1057	Fayette County	Alabama	AL	8170	8415	43.8	16585	50.73861923424781	33.72122228409212	-87.73885770683555	1057.0	16585

Important Links:

GitHub Link

https://github.com/santhiya-hds5210/ORES--Data-Transformation-with-Python-or-R#ores--data-transformation-with-python-or-r

Drive Link:

https://drive.google.com/drive/u/1/folders/1LKyjbVuUJ01DsA5uYHnUCKJIhjQXOQDP

Dataset Link: **[US_COUNTY.CSV]**

https://drive.google.com/drive/folders/1RfLhJVOK45x9oGBmOKyZEpBAaHuITYaw

Appendix:

1. https://www.google.com/search?q=how+to+add+comma+to+3000+values+in+notepad%2B%2B&oq=how+to+add+comma+to+3000+values+in+no&gs_lcrp=EgZjaHJvbWUqBwgBECEYoAEyBggAEEUYOTIHCAEQIRigATIHCAIQIRigATIHCAMQIRigAdIBCTE4MTU0ajBqN6gCALACAA&sourceid=chrome&ie=UTF-8
2. https://www.google.com/search?q=how+to+make+3000+line+of+data+in+a+single+horizontal+line+in+notepad%2B%2B&oq=how+to+make+3000+line+of+data+in+a+single+horizontal+line+in+notepad%2B%2B&gs_lcrp=EgZjaHJvbWUyBggAEEUYOdIBCTM0NzAwajBqN6gCALACAA&sourceid=chrome&ie=UTF-8
3. https://www.google.com/search?sca_esv=579833118&sxsrf=AM9HkKnY5z6zssSxCf2HJgZ2L_M4cT03fg:1699287887755&q=Geographic+data+examples&uds=H4sIAAAAAAAA_-My5-JwrUjMLchJLRbiT0_NTy9KLMjITFZISSxJNGAsknBHFVJIhSo2QlcMAIcvU5tKAAAA&sa=X&ved=2ahUKEwiVta7X5K-CAxWbMDQIHYiuDPMQxKsJegQIDRAB&ictx=0&biw=1366&bih=651&dpr=1
4. https://www.tutorialspoint.com/get-the-data-type-of-column-in-pandas-python

5. https://www.geeksforgeeks.org/how-to-create-an-empty-dataframe-and-append-rows-columns-to-it-in-pandas/
6. https://builtin.com/data-science/rename-columns-pandas
7. https://www.google.com/search?q=how+to+create+a+calculated+field+based+on+two+other+columns+dataframe&sca_esv=579833118&sxsrf=AM9HkKnatMkR4mMkAJ-HHGCvRSZuFP_e-g%3A1699292407777&ei=9yRJZdSRL4qIptQPvc-qsA0&ved=0ahUKEwiU6dbC9a-CAxUKhIkEHb2nCtYQ4dUDCBA&uact=5&oq=how+to+create+a+calculated+field+based+on+two+other+columns+dataframe&gs_lp=Egxnd3Mtd2l6LXNlcnAiRWhvdyB0byBjcmVhdGUgYSBjYWxjdWxhdGVkIGZpZWxkIGJhc2VkIG9uIHR3byBvdGhlciBjb2x1bW5zIGRhdGFmcmFtZUjiSFD-BViPRXAEeAGQAQCYAbQCoAHlDKoBBzAuOS4xLjG4AQPIAQD4AQHCAgoQABhHGNYEGLADwgIEECMYJ8ICBxAjGLACGCfiAwQYACBBiAYBkAYI&sclient=gws-wiz-serp
8. https://www.google.com/search?q=how+to+drop+a+column+in+dataframe+in+python&oq=how+to+drop+a+column+in+dataframe+in+pyhto&gs_lcrp=EgZjaHJvbWUqCQgBEAAYDRiABDIGCAAQRRg5MgkIARAAGA0YgAQyCAgCEAAYFhgeMggIAxAAGBYYHjIICAQQABgWGB4yCAgFEAAYFhgeMggIBhAAGBYYHjIICAcQABgWGB4yCAgIEAAYFhgeMgoICRAAGIYDGIoF0gEJMTQwOTdqMGo3qAIAsAIA&sourceid=chrome&ie=UTF-8
9. https://www.analyticsvidhya.com/blog/2021/03/zooming-out-a-look-at-outlier-and-how-to-deal-with-them-indata-science/
10. https://www.google.com/search?q=----------+ValueError+Traceback+(most+recent+call+last)+%3Cipython-input-64-6c056e54992a%3E+in+%3Ccell+line%3A+8%3E()+6+7+%23+Fit+and+transform+the+DataFrame+to+standardize+the+data+----%3E+8+df_standardized+%3D+scaler.fit_transform(data_frame)+9+10+%23+Convert+the+standardized+data+back+to+a+DataFrame+7+frames+%2Fusr%2Flocal%2Flib%2Fpython3.10%2Fdist-packages%2Fpandas%2Fcore%2Fgeneric.py+in+__array__(self%2C+dtype)+2068+2069+def+__array__(self%2C+dtype%3A+npt.DTypeLike+%7C+None+%3D+None)+-%3E+np.ndarray%3A+-%3E+2070+return+np.asarray(self._values%2C+dtype%3Ddtype)+2071+2072+def+__array_wrap__(+ValueError%3A+could+not+convert+string+to+float%3A+%27Autauga+County%27&oq=----------+ValueError++++++++++++++++++++++++++++++++Traceback+(most+recent+call+last)+%3Cipython-input-64-6c056e54992a%3E+in+%3Ccell+line%3A+8%3E()+6+7+%23+Fit+and+transform+the+DataFrame+to+standardize+the+data+----%3E+8+df_standardized+%3D+scaler.fit_transform(data_frame)+9+10+%23+Convert+the+standardized+data+back+to+a+DataFrame+7+frames+%2Fusr%2Flocal%2Flib%2Fpython3.10%2Fdist-packages%2Fpandas%2Fcore%2Fgeneric.py+in+__array__(self%2C+dtype)+2068+2069++++def+__array__(self%2C+dtype%3A+npt.DTypeLike+%7C+None+%3D+None)+-%3E+np.ndarray%3A+-%3E+2070+++++++++return+np.asarray(self._values%2C+dtype%3Ddtype)+2071+2072++++++def+__array_wrap__(+ValueError%3A+could+not+convert+string+to+float%3A+%27Autauga+County%27&gs_lcrp=EgZjaHJvbWUyBggAEEUYOdIBBzk2N2owajeoAgCwAgA&sourceid=chrome&ie=UTF-8
11. https://stackoverflow.com/questions/59807430/google-colab-can-we-restore-all-the-data-even-after-the-runtime-disconnects
12. https://chat.openai.com/c/d583192e-1edc-46f2-831e-8e336adefe07

13. https://www.forefront.ai/
14. https://colab.research.google.com/drive/1VorJR5TFT-58iqxp-eTMcx7h5SAAxCjX#scrollTo=Fj86ZXs1WIJM
15. https://canvas.slu.edu/courses/45377/assignments/343229
16. https://www.google.com/search?q=how+to+standardise+my+data+in+dataframe&oq=how+to+standardise+my+data+in+dataframe&gs_lcrp=EgZjaHJvbWUyBggAEEUYOTIHCAEQIRigATIHCAIQIRigAdIBCDExODlqMGo3qAIAsAIA&sourceid=chrome&ie=UTF-8
17. https://www.google.com/search?q=----------+ValueError+Traceback+(most+recent+call+last)+%3Cipython-input-62-029504c47d07%3E+in+%3Ccell+line%3A+7%3E()+5+df_uscounty+%3D+pd.read_csv(%27%2Fcontent%2Fus_county.csv%27)+6+plt.figure(figsize%3D(5%2C5))+----%3E+7+sns.boxplot(y%3D%27male_count%27%2Cdata%3Ddf_uscounty)+8+plt.show()+2+frames+%2Fusr%2Flocal%2Flib%2Fpython3.10%2Fdist-packages%2Fseaborn%2Fcategorical.py+in+establish_variables(self%2C+x%2C+y%2C+hue%2C+data%2C+orient%2C+order%2C+hue_order%2C+units)+539+if+isinstance(var%2C+str)%3A+540+err+%3D+f%22Could+not+interpret+input+%27%7Bvar%7D%27%22+--%3E+541+raise+ValueError(err)+542+543+%23+Figure+out+the+plotting+orientation+ValueError%3A+Could+not+interpret+input+%27male_count%27+%3CFigure+size+500x500+with+0+Axes%3E&oq=----------+ValueError++++++++++++++++++++++++++++++++Traceback+(most+recent+call+last)+%3Cipython-input-62-029504c47d07%3E+in+%3Ccell+line%3A+7%3E()+5+df_uscounty+%3D+pd.read_csv(%27%2Fcontent%2Fus_county.csv%27)+6+plt.figure(figsize%3D(5%2C5))+----%3E+7+sns.boxplot(y%3D%27male_count%27%2Cdata%3Ddf_uscounty)+8+plt.show()+2+frames+%2Fusr%2Flocal%2Flib%2Fpython3.10%2Fdist-packages%2Fseaborn%2Fcategorical.py+in+establish_variables(self%2C+x%2C+y%2C+hue%2C+data%2C+orient%2C+order%2C+hue_order%2C+units)+539++++++++++++++++++if+isinstance(var%2C+str)%3A+540++++++++++++++++++++err+%3D+f%22Could+not+interpret+input+%27%7Bvar%7D%27%22+--%3E+541++++++++++++++++++++raise+ValueError(err)+542+543++++++++++++%23+Figure+out+the+plotting+orientation+ValueError%3A+Could+not+interpret+input+%27male_count%27+%3CFigure+size+500x500+with+0+Axes%3E&gs_lcrp=EgZjaHJvbWUyBggAEEUYOdIBCDEwNTRqMGo3qAIAsAIA&sourceid=chrome&ie=UTF-8
18. https://www.google.com/search?q=how+to+drop+a+column+in+dataframe+in+python&oq=how+to+drop+a+column+in+dataframe+in+python&gs_lcrp=EgZjaHJvbWUyCwgAEEUYJxg5GIoFMggIARAAGBYYHjIICAIQABgWGB4yCAgDEAAYFhgeMggIBBAAGBYYHjIICAUQABgWGB4yCAgGEAAYFhgeMggIBxAAGBYYHjIICAgQABgWGB4yCAgJEAAYFhge0gEIMTEyMWowajeoAgCwAgA&sourceid=chrome&ie=UTF-8
19. https://www.google.com/search?q=what+is+the+code+to+see+the+only+headings+of+the+table+in+the+dataframe+inpythone&oq=what+is+the+code+to+see+the+only+headings+of+the+table+in+the+dataframe+inpythone&gs_lcrp=EgZjaHJvbWUyBggAEEUYOdIBCDYxMTlqMGo3qAIAsAIA&sourceid=chrome&ie=UTF-8
20. https://www.google.com/search?q=how+to+insert+value+for+the+null+in+dataframe+python&oq=how+to+insert+value+for+the+null+in+dataframe+python&gs_lcrp=EgZjaHJvbWUyBggAEEUYOTIICAEQABgWGB4yCAgCEAAYFhgeMgoIAxAAGIYDGIoFMgoIBBAAGIYDGIoFMgoIBRAAGIYDGIoF0gEHODkxajBqN6gCALACAA&sourceid=chrome&ie=UTF-8