Report of HW1
B09602017 白宗民

Dataset 1: Housing Data

Q1. What steps will you take upon receiving this dataset before commencing data analysis?

1. **Search** the indicators about the housing decision.

2. **Cleaning** the data by handling the useless data or data missing (by **df.info()**).

3. **Deleting** the [‘No. of Bathroom’] since it’s all the same (by **df[‘…’].unique()**).

4. Change the numeric part from **string** to **float** the further calculation (by **df.astype()**).

5. **Adding** the additional information into the Dataframe.

    - rental yield, sell and rent price per area

6. **One-hot encoding** the “Location” before ML training process.

- It’s all about adding and removing the information and checking the data completeness.

Q2. If you are to inquire about Q1 from ChatGPT or Bing, what responses will you receive? Do you find them reasonable? If not, how will you rectify it?

Prompt for **GPT-4**:
What steps will you take upon receiving this dataset before commencing data analysis?

data format:

Area,No. of Rooms,No. of Bathrooms,Location,Miles (dist. between school and house),Rent Price per Month,Sell Price

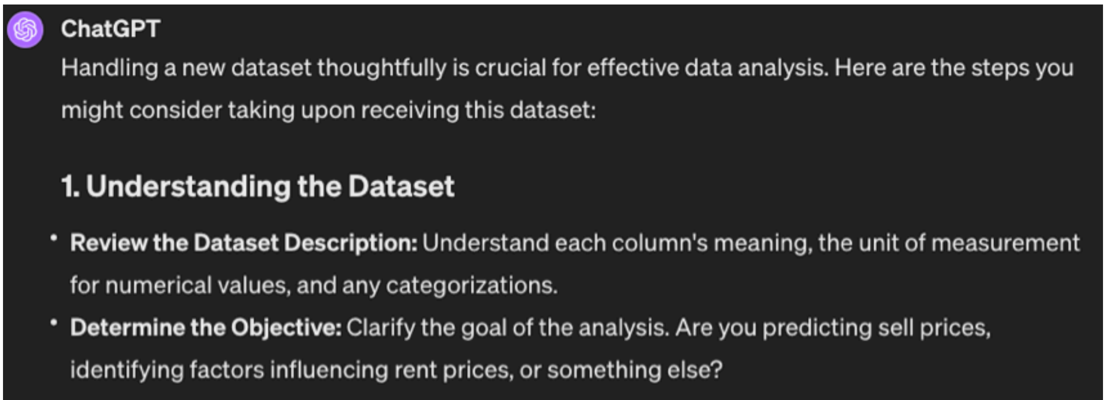
1360,1,1,Rural,463,7401,74446632

1794,3,1,Suburb,210,9259,76199794

1630,2,1,Suburb,157,16469,16249579

...

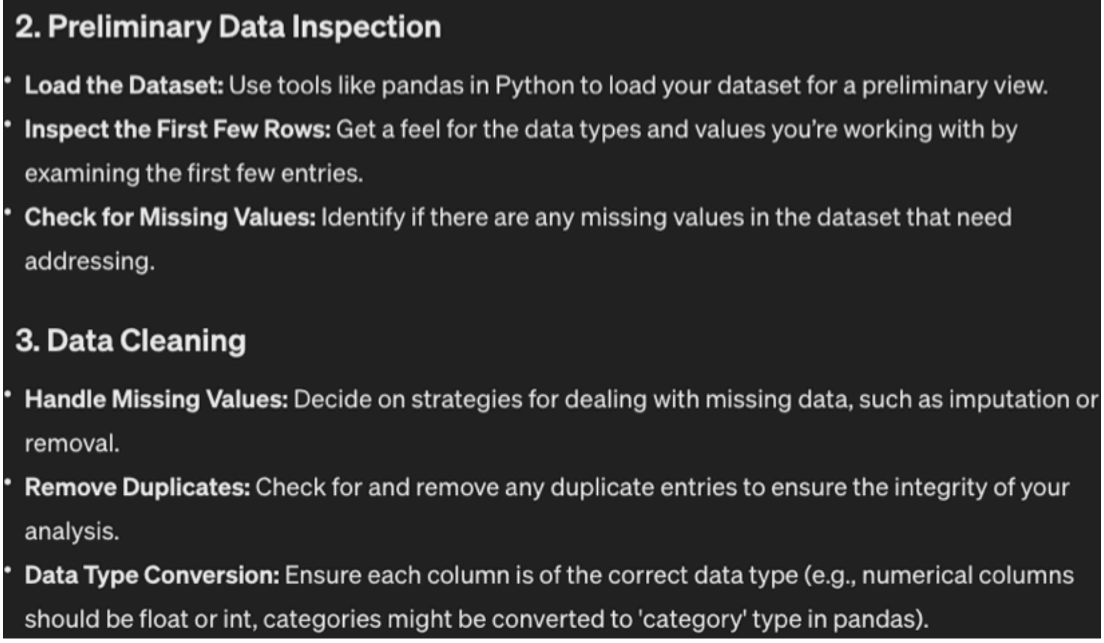
please think step by step (from zero-shot-CoT) and this is very important to my career (from LLM Stimuli).



ChatGPT
Handling a new dataset thoughtfully is crucial for effective data analysis. Here are the steps you might consider taking upon receiving this dataset:

1. Understanding the Dataset

- **Review the Dataset Description:** Understand each column's meaning, the unit of measurement for numerical values, and any categorizations.
- **Determine the Objective:** Clarify the goal of the analysis. Are you predicting sell prices, identifying factors influencing rent prices, or something else?

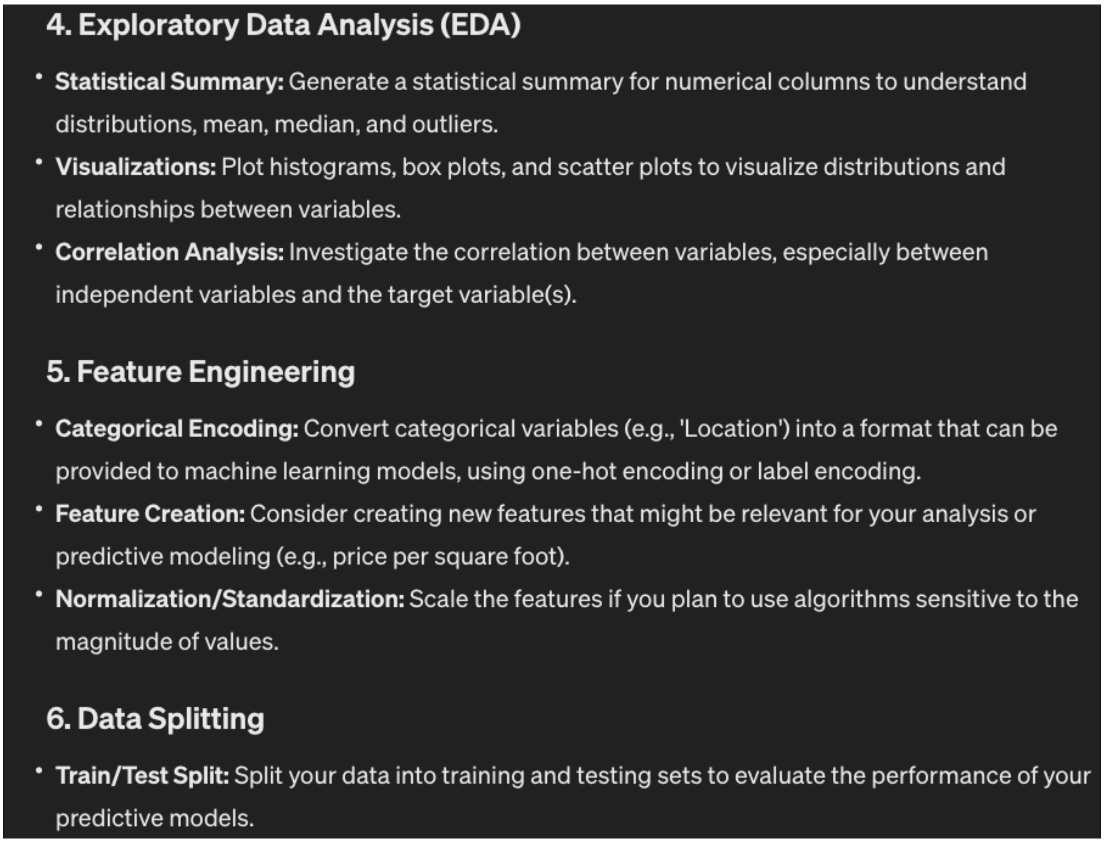


2. Preliminary Data Inspection

- **Load the Dataset:** Use tools like pandas in Python to load your dataset for a preliminary view.
- **Inspect the First Few Rows:** Get a feel for the data types and values you're working with by examining the first few entries.
- **Check for Missing Values:** Identify if there are any missing values in the dataset that need addressing.

3. Data Cleaning

- **Handle Missing Values:** Decide on strategies for dealing with missing data, such as imputation or removal.
- **Remove Duplicates:** Check for and remove any duplicate entries to ensure the integrity of your analysis.
- **Data Type Conversion:** Ensure each column is of the correct data type (e.g., numerical columns should be float or int, categories might be converted to 'category' type in pandas).



4. Exploratory Data Analysis (EDA)

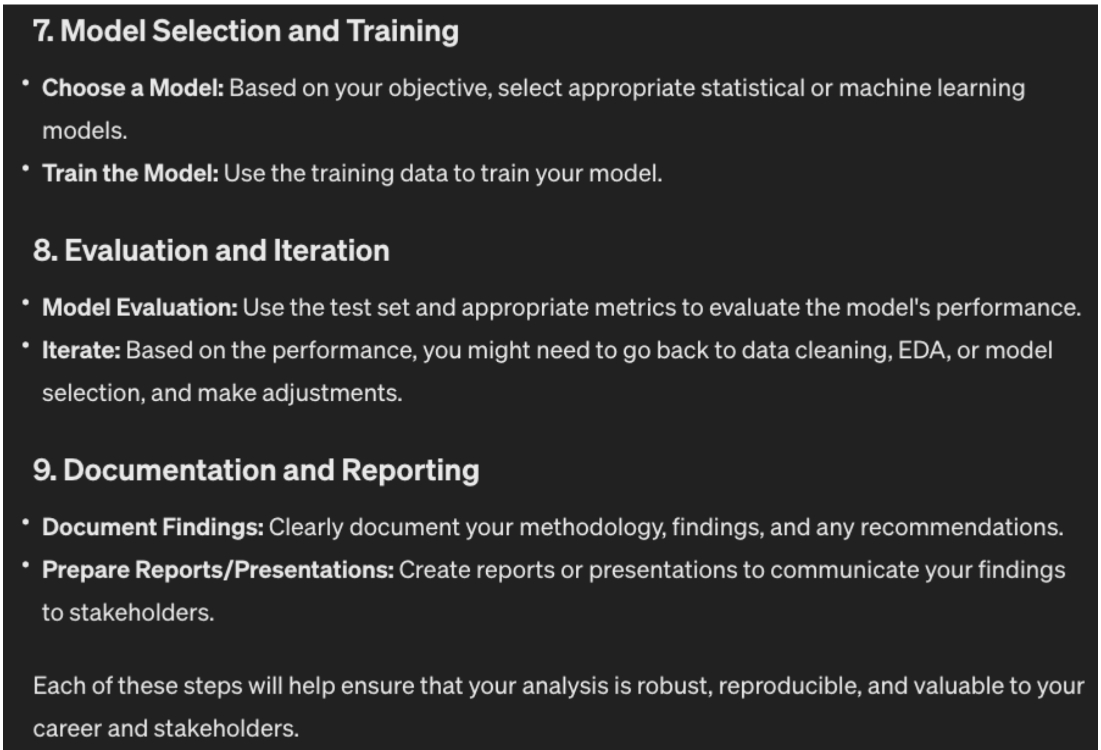
- **Statistical Summary:** Generate a statistical summary for numerical columns to understand distributions, mean, median, and outliers.
- **Visualizations:** Plot histograms, box plots, and scatter plots to visualize distributions and relationships between variables.
- **Correlation Analysis:** Investigate the correlation between variables, especially between independent variables and the target variable(s).

5. Feature Engineering

- **Categorical Encoding:** Convert categorical variables (e.g., 'Location') into a format that can be provided to machine learning models, using one-hot encoding or label encoding.
- **Feature Creation:** Consider creating new features that might be relevant for your analysis or predictive modeling (e.g., price per square foot).
- **Normalization/Standardization:** Scale the features if you plan to use algorithms sensitive to the magnitude of values.

6. Data Splitting

- **Train/Test Split:** Split your data into training and testing sets to evaluate the performance of your predictive models.



7. Model Selection and Training

- **Choose a Model:** Based on your objective, select appropriate statistical or machine learning models.
- **Train the Model:** Use the training data to train your model.

8. Evaluation and Iteration

- **Model Evaluation:** Use the test set and appropriate metrics to evaluate the model's performance.
- **Iterate:** Based on the performance, you might need to go back to data cleaning, EDA, or model selection, and make adjustments.

9. Documentation and Reporting

- **Document Findings:** Clearly document your methodology, findings, and any recommendations.
- **Prepare Reports/Presentations:** Create reports or presentations to communicate your findings to stakeholders.

Each of these steps will help ensure that your analysis is robust, reproducible, and valuable to your career and stakeholders.

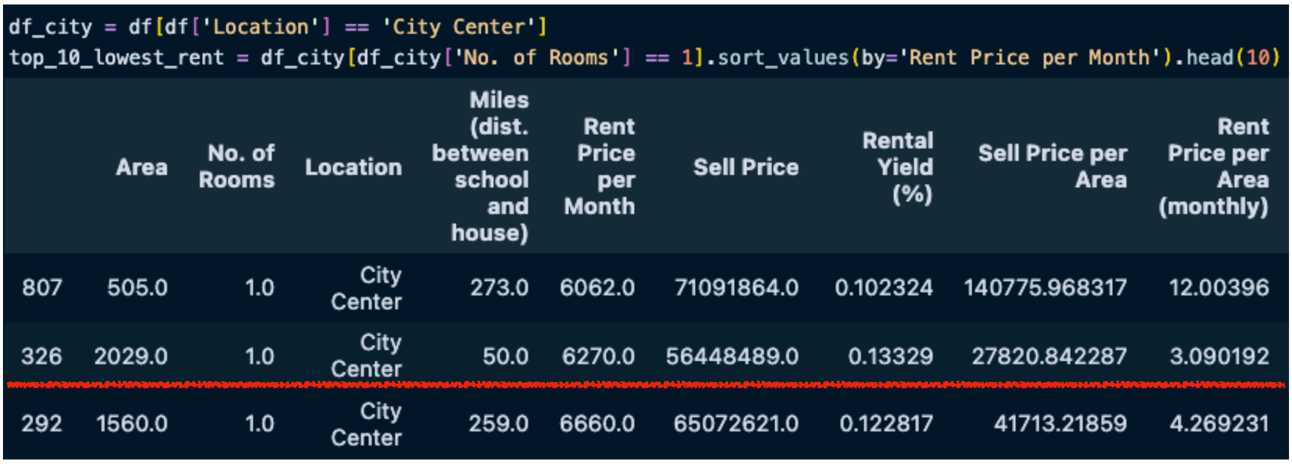
Q3. If you are restricted to renting a house, which one or ones will you select, and why?

Actually I think this depends on the financial condition and the needs at the moment.

Assume that I am a person who likes to live at a convenient area, which is **city center**.

And the **average rent price** at city center is **13272.215909** (kind of a criteria).

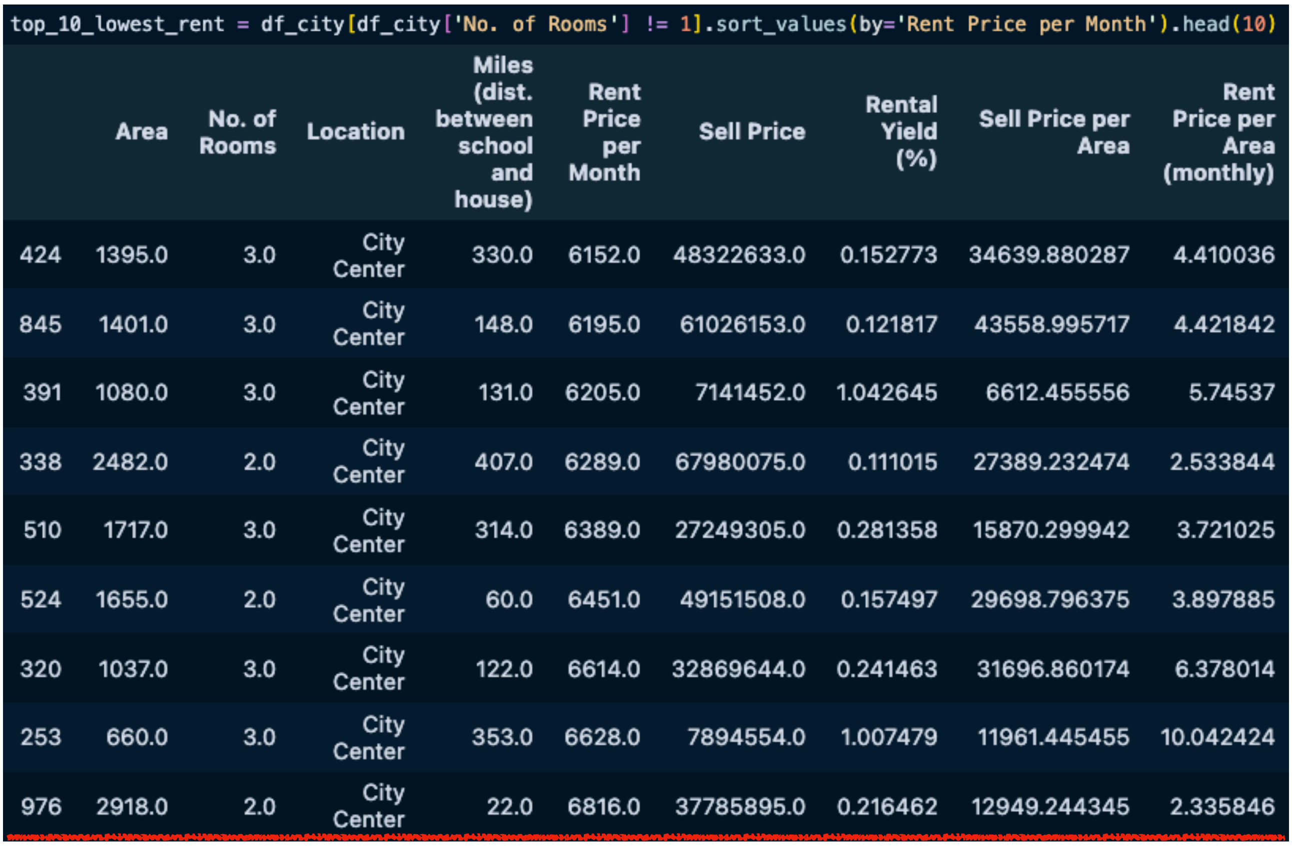
1. If I am single and have few salary: **House 326;** since it has the 2nd lowest rent price per month, big area with only one room(single situation) and only 50 miles distance.



```
df_city = df[df['Location'] == 'City Center']
top_10_lowest_rent = df_city[df_city['No. of Rooms'] == 1].sort_values(by='Rent Price per Month').head(10)
```

	Area	No. of Rooms	Location	Miles (dist. between school and house)	Rent Price per Month	Sell Price	Rental Yield (%)	Sell Price per Area	Rent Price per Area (monthly)
807	505.0	1.0	City Center	273.0	6062.0	71091864.0	0.102324	140775.968317	12.00396
326	2029.0	1.0	City Center	50.0	6270.0	56448489.0	0.13329	27820.842287	3.090192
292	1560.0	1.0	City Center	259.0	6660.0	65072621.0	0.122817	41713.21859	4.269231

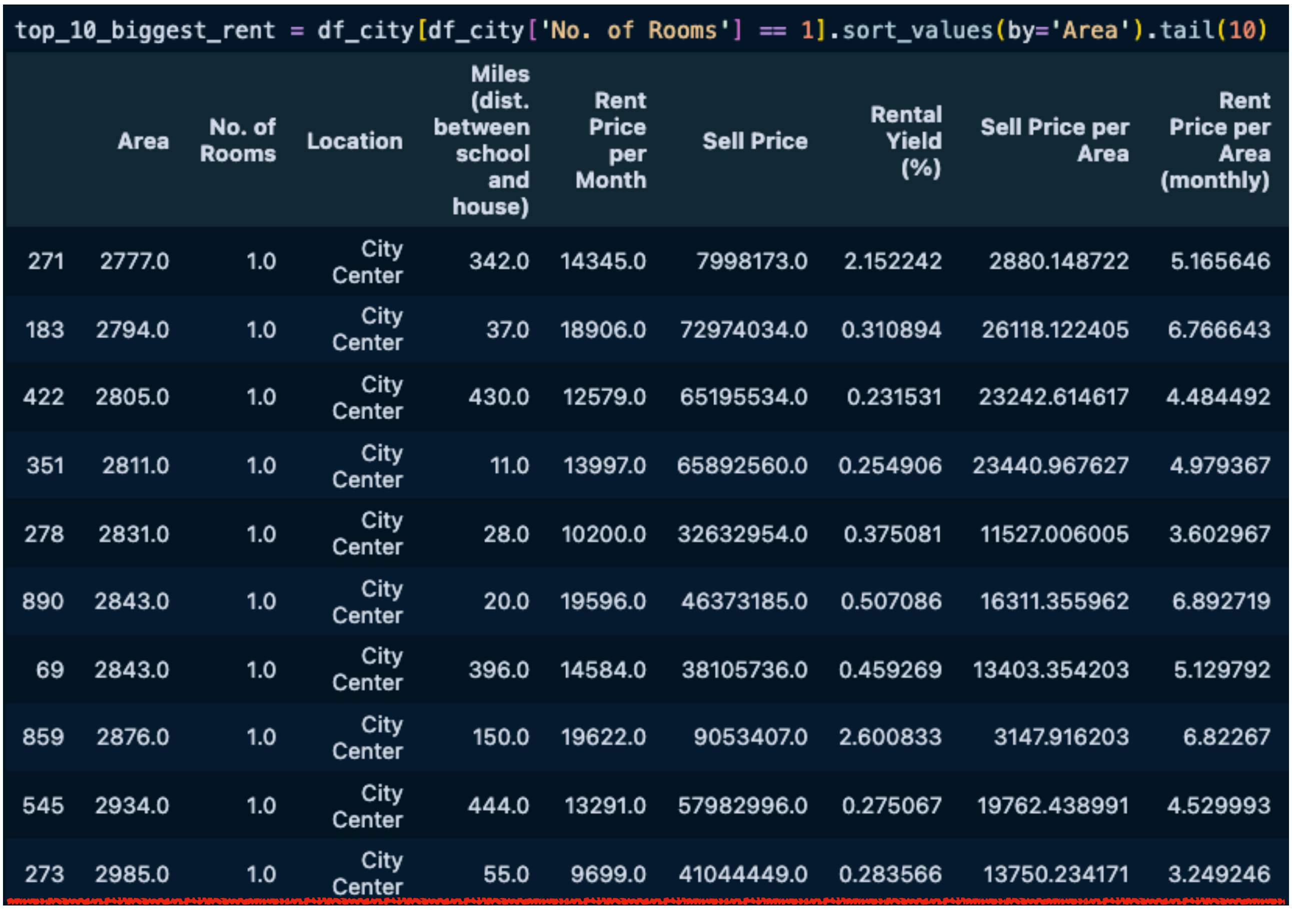
2. If I am not single and have few salary: **House 976**; since it has the 9th lowest rent price and has 2 rooms with 2918 area for lovely family or couple and it's also 22 miles distance.



```
top_10_lowest_rent = df_city[df_city['No. of Rooms'] != 1].sort_values(by='Rent Price per Month').head(10)
```

	Area	No. of Rooms	Location	Miles (dist. between school and house)	Rent Price per Month	Sell Price	Rental Yield (%)	Sell Price per Area	Rent Price per Area (monthly)
424	1395.0	3.0	City Center	330.0	6152.0	48322633.0	0.152773	34639.880287	4.410036
845	1401.0	3.0	City Center	148.0	6195.0	61026153.0	0.121817	43558.995717	4.421842
391	1080.0	3.0	City Center	131.0	6205.0	7141452.0	1.042645	6612.455556	5.74537
338	2482.0	2.0	City Center	407.0	6289.0	67980075.0	0.111015	27389.232474	2.533844
510	1717.0	3.0	City Center	314.0	6389.0	27249305.0	0.281358	15870.299942	3.721025
524	1655.0	2.0	City Center	60.0	6451.0	49151508.0	0.157497	29698.796375	3.897885
320	1037.0	3.0	City Center	122.0	6614.0	32869644.0	0.241463	31696.860174	6.378014
253	660.0	3.0	City Center	353.0	6628.0	7894554.0	1.007479	11961.445455	10.042424
976	2918.0	2.0	City Center	22.0	6816.0	37785895.0	0.216462	12949.244345	2.335846

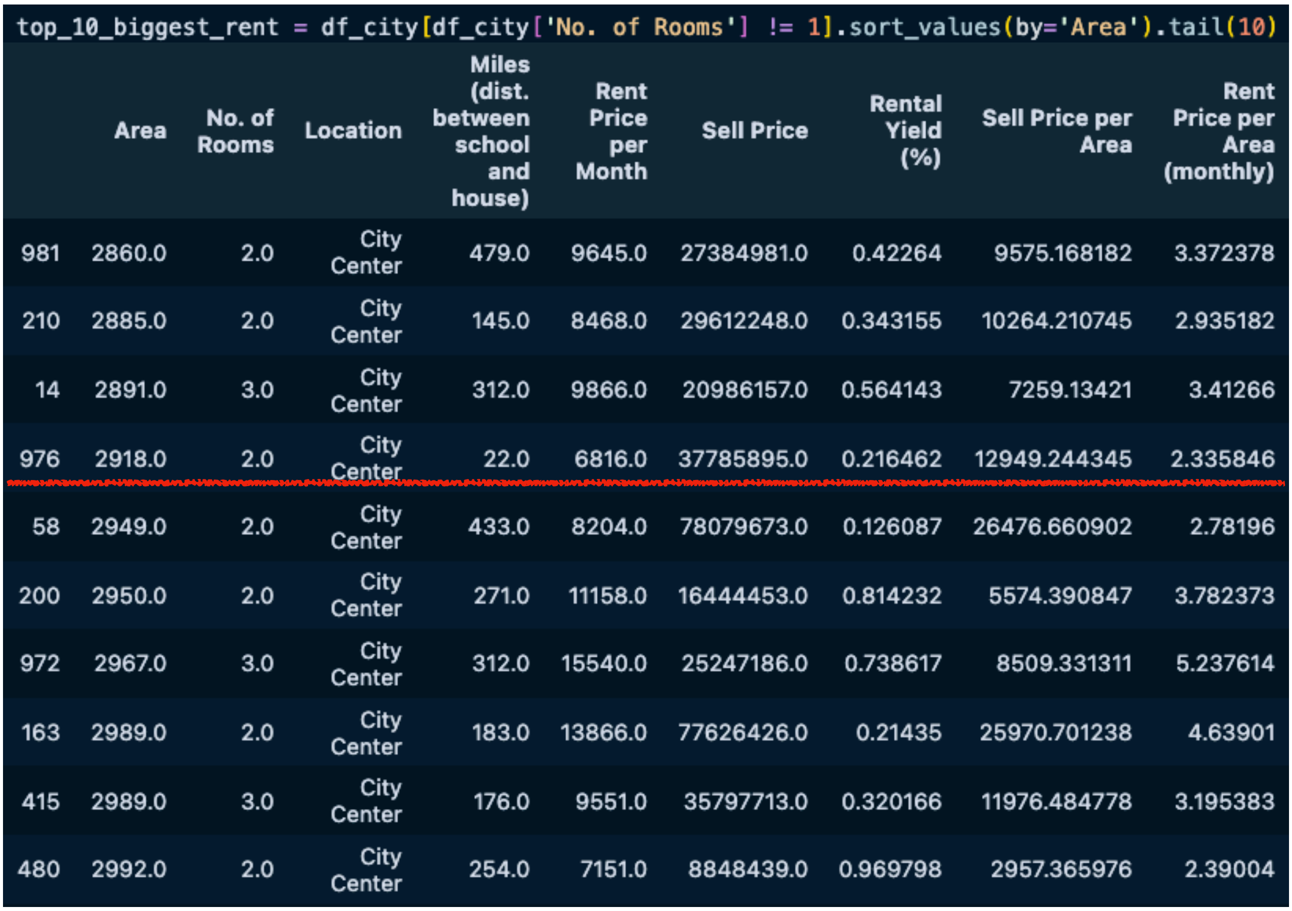
3. If I am single and have a lot of salary: **House 273;** since it has the largest area with only 55 miles distance and only one room for single Vincent.



```
top_10_biggest_rent = df_city[df_city['No. of Rooms'] == 1].sort_values(by='Area').tail(10)
```

	Area	No. of Rooms	Location	Miles (dist. between school and house)	Rent Price per Month	Sell Price	Rental Yield (%)	Sell Price per Area	Rent Price per Area (monthly)
271	2777.0	1.0	City Center	342.0	14345.0	7998173.0	2.152242	2880.148722	5.165646
183	2794.0	1.0	City Center	37.0	18906.0	72974034.0	0.310894	26118.122405	6.766643
422	2805.0	1.0	City Center	430.0	12579.0	65195534.0	0.231531	23242.614617	4.484492
351	2811.0	1.0	City Center	11.0	13997.0	65892560.0	0.254906	23440.967627	4.979367
278	2831.0	1.0	City Center	28.0	10200.0	32632954.0	0.375081	11527.006005	3.602967
890	2843.0	1.0	City Center	20.0	19596.0	46373185.0	0.507086	16311.355962	6.892719
69	2843.0	1.0	City Center	396.0	14584.0	38105736.0	0.459269	13403.354203	5.129792
859	2876.0	1.0	City Center	150.0	19622.0	9053407.0	2.600833	3147.916203	6.82267
545	2934.0	1.0	City Center	444.0	13291.0	57982996.0	0.275067	19762.438991	4.529993
273	2985.0	1.0	City Center	55.0	9699.0	41044449.0	0.283566	13750.234171	3.249246

4. If I am not single and have a lot of salary: **House 976**; since it has the 7th largest area and has 2 rooms for lovely family or couple and it's also 22 miles distance. Not to mention that it has the 9th lowest rent price per month.



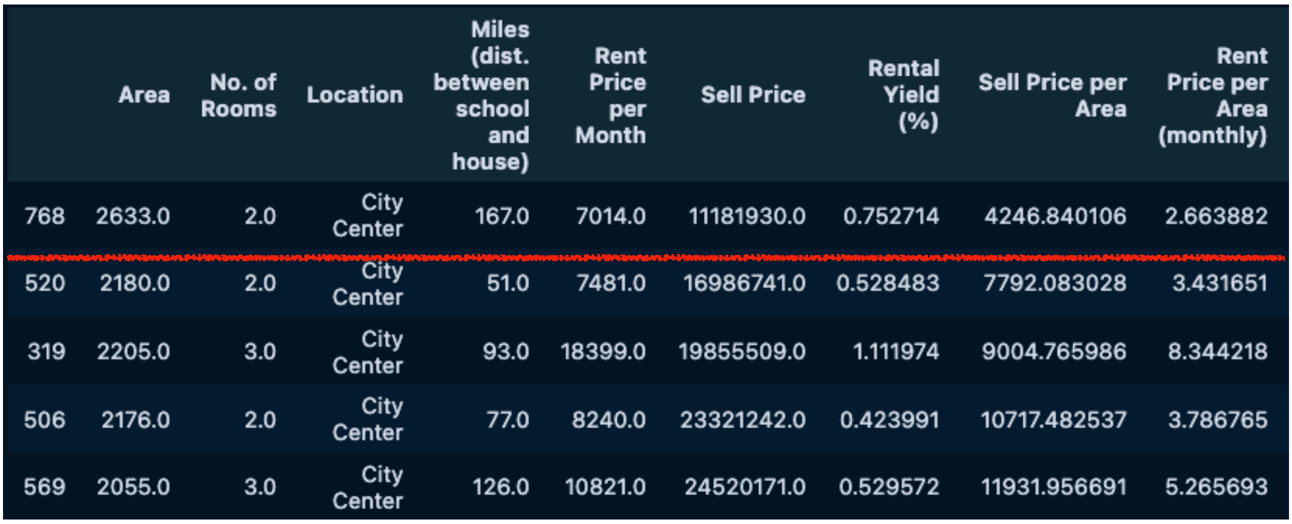
```
top_10_biggest_rent = df_city[df_city['No. of Rooms'] != 1].sort_values(by='Area').tail(10)
```

	Area	No. of Rooms	Location	Miles (dist. between school and house)	Rent Price per Month	Sell Price	Rental Yield (%)	Sell Price per Area	Rent Price per Area (monthly)
981	2860.0	2.0	City Center	479.0	9645.0	27384981.0	0.42264	9575.168182	3.372378
210	2885.0	2.0	City Center	145.0	8468.0	29612248.0	0.343155	10264.210745	2.935182
14	2891.0	3.0	City Center	312.0	9866.0	20986157.0	0.564143	7259.13421	3.41266
976	2918.0	2.0	City Center	22.0	6816.0	37785895.0	0.216462	12949.244345	2.335846
58	2949.0	2.0	City Center	433.0	8204.0	78079673.0	0.126087	26476.660902	2.78196
200	2950.0	2.0	City Center	271.0	11158.0	16444453.0	0.814232	5574.390847	3.782373
972	2967.0	3.0	City Center	312.0	15540.0	25247186.0	0.738617	8509.331311	5.237614
163	2989.0	2.0	City Center	183.0	13866.0	77626426.0	0.21435	25970.701238	4.63901
415	2989.0	3.0	City Center	176.0	9551.0	35797713.0	0.320166	11976.484778	3.195383
480	2992.0	2.0	City Center	254.0	7151.0	8848439.0	0.969798	2957.365976	2.39004

Q4. Assuming you have enough funds to purchase a house, will you opt to continue renting or proceed with a purchase? If renting, which one will you choose? If buying, which one will you select? Why?

Definitely **buy** the house. And I would choose the one with (# of room > 1), (Area>2000) and (Distance < 200) since buying a house is a investment for future family.

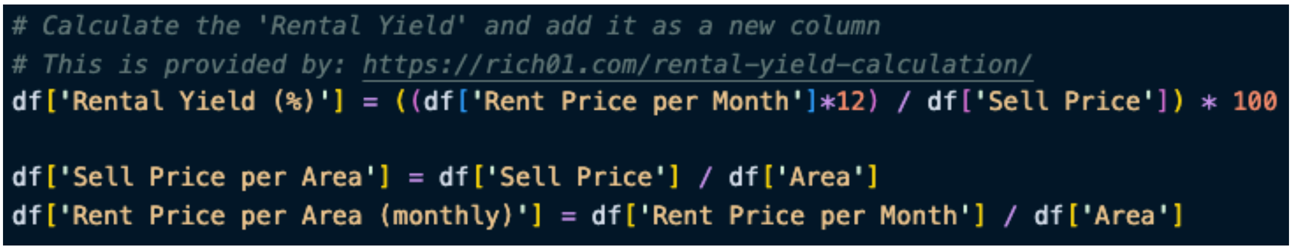
Code: df_buy = df_city[(df_city['No. of Rooms'] != 1) & (df_city['Miles (dist. between school and house)'] < 200) & (df_city['Area'] > 2000)].sort_values(by='Sell Price ').head(10)



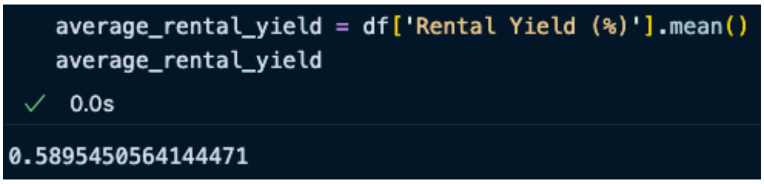
	Area	No. of Rooms	Location	Miles (dist. between school and house)	Rent Price per Month	Sell Price	Rental Yield (%)	Sell Price per Area	Rent Price per Area (monthly)
768	2633.0	2.0	City Center	167.0	7014.0	11181930.0	0.752714	4246.840106	2.663882
520	2180.0	2.0	City Center	51.0	7481.0	16986741.0	0.528483	7792.083028	3.431651
319	2205.0	3.0	City Center	93.0	18399.0	19855509.0	1.111974	9004.765986	8.344218
506	2176.0	2.0	City Center	77.0	8240.0	23321242.0	0.423991	10717.482537	3.786765
569	2055.0	3.0	City Center	126.0	10821.0	24520171.0	0.529572	11931.956691	5.265693

So I would choose **House 768** since it's the one that satisfied all the condition and has the lowest sell price. And it's rental yield is 0.75%, which is higher than average (0.589%) so maybe I can rent it someday in the future if I earn more money and can buy a new fancy house.

(How I calculate the rental yield and average rental yield):



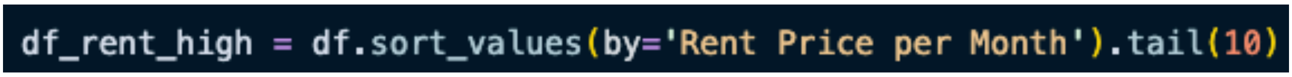
```
# Calculate the 'Rental Yield' and add it as a new column
# This is provided by: https://rich01.com/rental-yield-calculation/
df['Rental Yield (%)'] = ((df['Rent Price per Month']*12) / df['Sell Price']) * 100

df['Sell Price per Area'] = df['Sell Price'] / df['Area']
df['Rent Price per Area (monthly)'] = df['Rent Price per Month'] / df['Area']
```



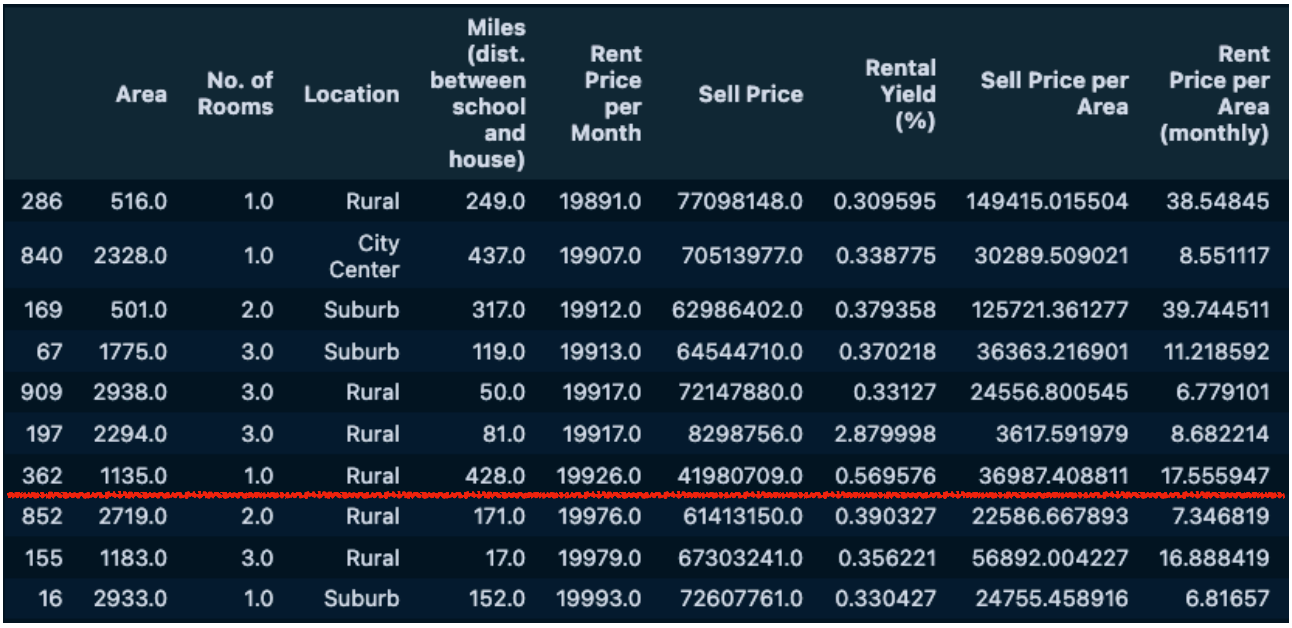
```
average_rental_yield = df['Rental Yield (%)'].mean()
average_rental_yield
✓ 0.0s
0.5895450564144471
```

Q5. Are there any properties with rent or selling prices that seem unusually high or low? Why?

For Rent: I think that **House 362** is unusually high: since it has only 1 room with 1135 area and it's located in rural with also 428 miles distance. But it's price is the 4th highest for the rent price per month

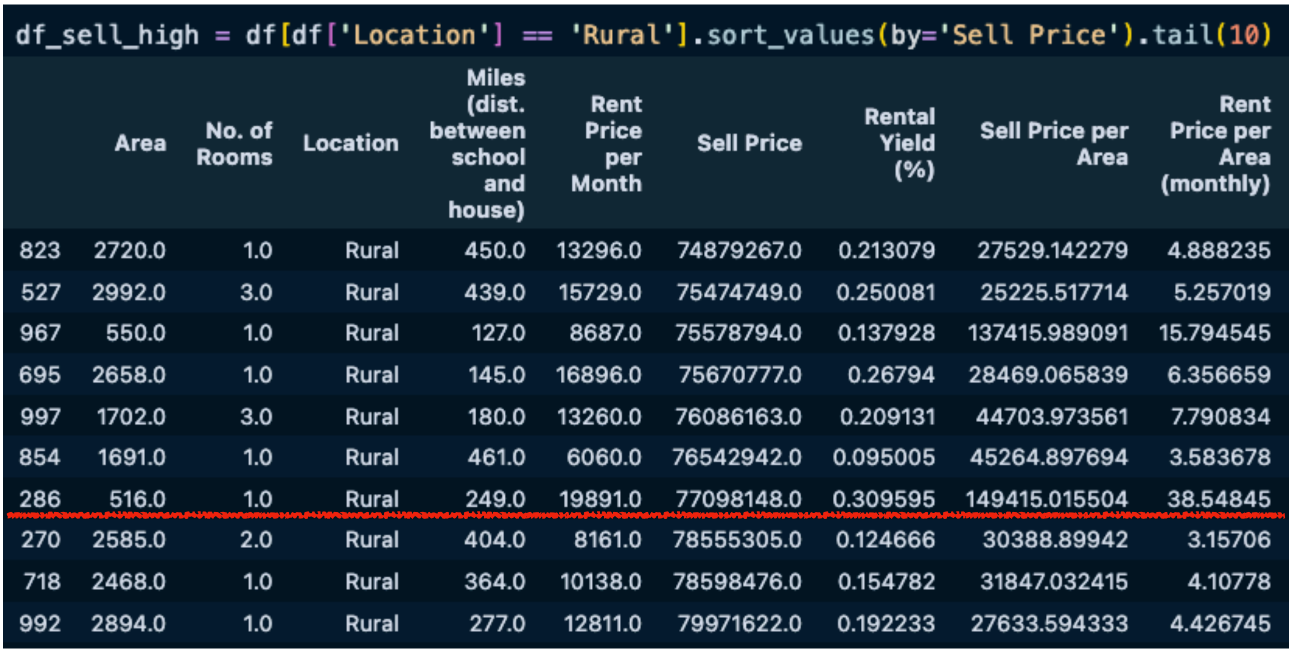


```
df_rent_high = df.sort_values(by='Rent Price per Month').tail(10)
```



	Area	No. of Rooms	Location	Miles (dist. between school and house)	Rent Price per Month	Sell Price	Rental Yield (%)	Sell Price per Area	Rent Price per Area (monthly)
286	516.0	1.0	Rural	249.0	19891.0	77098148.0	0.309595	149415.015504	38.54845
840	2328.0	1.0	City Center	437.0	19907.0	70513977.0	0.338775	30289.509021	8.551117
169	501.0	2.0	Suburb	317.0	19912.0	62986402.0	0.379358	125721.361277	39.744511
67	1775.0	3.0	Suburb	119.0	19913.0	64544710.0	0.370218	36363.216901	11.218592
909	2938.0	3.0	Rural	50.0	19917.0	72147880.0	0.33127	24556.800545	6.779101
197	2294.0	3.0	Rural	81.0	19917.0	8298756.0	2.879998	3617.591979	8.682214
362	1135.0	1.0	Rural	428.0	19926.0	41980709.0	0.569576	36987.408811	17.555947
852	2719.0	2.0	Rural	171.0	19976.0	61413150.0	0.390327	22586.667893	7.346819
155	1183.0	3.0	Rural	17.0	19979.0	67303241.0	0.356221	56892.004227	16.888419
16	2933.0	1.0	Suburb	152.0	19993.0	72607761.0	0.330427	24755.458916	6.81657

For Sell: I think that **House 286** is unusually high: since it has only 1 room with 516 area and it's located in rural with also 249 distance. But it cost 77098148 which is the 4th highest set price for rural house and it's only 2887629 lower than the house with highest sell price.



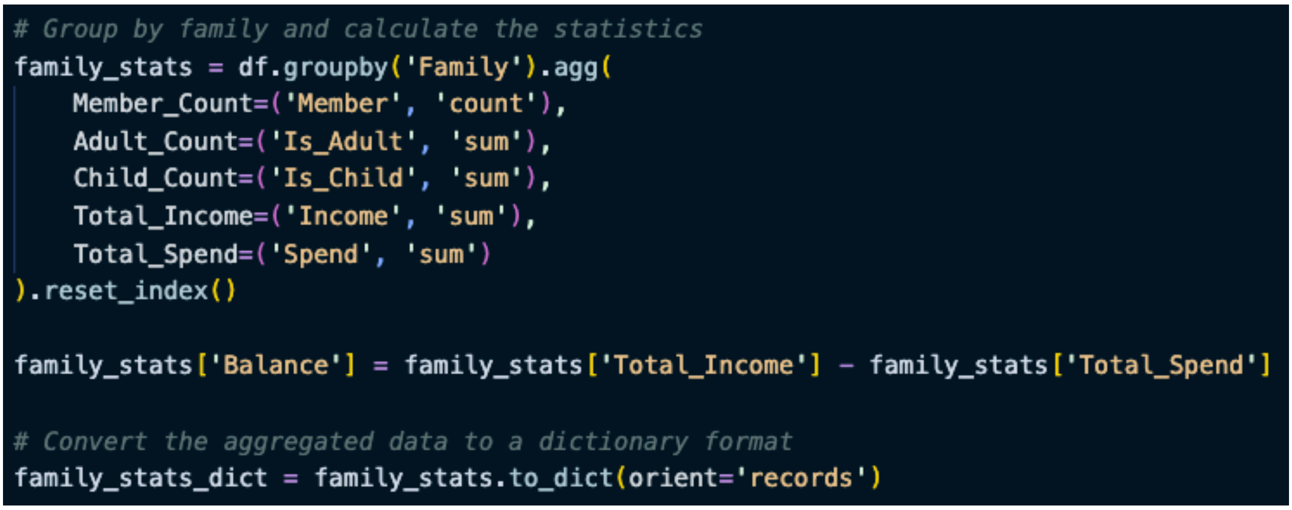
```
df_sell_high = df[df['Location'] == 'Rural'].sort_values(by='Sell Price').tail(10)
```

	Area	No. of Rooms	Location	Miles (dist. between school and house)	Rent Price per Month	Sell Price	Rental Yield (%)	Sell Price per Area	Rent Price per Area (monthly)
823	2720.0	1.0	Rural	450.0	13296.0	74879267.0	0.213079	27529.142279	4.888235
527	2992.0	3.0	Rural	439.0	15729.0	75474749.0	0.250081	25225.517714	5.257019
967	550.0	1.0	Rural	127.0	8687.0	75578794.0	0.137928	137415.989091	15.794545
695	2658.0	1.0	Rural	145.0	16896.0	75670777.0	0.26794	28469.065839	6.356659
997	1702.0	3.0	Rural	180.0	13260.0	76086163.0	0.209131	44703.973561	7.790834
854	1691.0	1.0	Rural	461.0	6060.0	76542942.0	0.095005	45264.897694	3.583678
286	516.0	1.0	Rural	249.0	19891.0	77098148.0	0.309595	149415.015504	38.54845
270	2585.0	2.0	Rural	404.0	8161.0	78555305.0	0.124666	30388.89942	3.15706
718	2468.0	1.0	Rural	364.0	10138.0	78598476.0	0.154782	31847.032415	4.10778
992	2894.0	1.0	Rural	277.0	12811.0	79971622.0	0.192233	27633.594333	4.426745

Dataset 2: Family Data

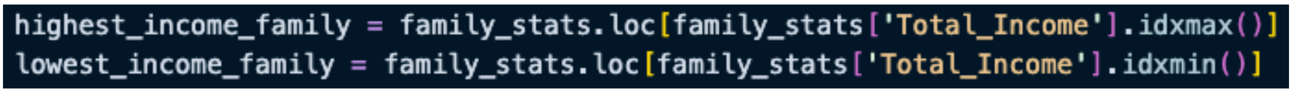
At first, I construct a **list** where each element is a **dictionary** with each family, as a result, this list has 100 elements.

Each dictionary has keys: **Family ID, Member_count, Adult_count, Child_count, Total_income, Total_spend, Balance.**



```python
# Group by family and calculate the statistics
family_stats = df.groupby('Family').agg(
    Member_Count=('Member', 'count'),
    Adult_Count=('Is_Adult', 'sum'),
    Child_Count=('Is_Child', 'sum'),
    Total_Income=('Income', 'sum'),
    Total_Spend=('Spend', 'sum')
).reset_index()

family_stats['Balance'] = family_stats['Total_Income'] - family_stats['Total_Spend']

# Convert the aggregated data to a dictionary format
family_stats_dict = family_stats.to_dict(orient='records')
```

Q1. Which family boasts the highest annual income, and which has the lowest? How do you ascertain this?



```python
highest_income_family = family_stats.loc[family_stats['Total_Income'].idxmax()]
lowest_income_family = family_stats.loc[family_stats['Total_Income'].idxmin()]
```

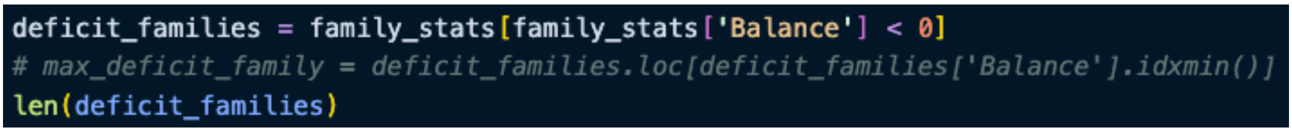
Highest income: **family 6** with 7804425

Lowest income: **family 94** with 46790

By the code I can ascertain it.

Q2. Which families do not possess adequate annual income to cover all members' spending? What is the maximum shortfall? How do you determine this?

There is **no** family does not possess adequate annual income to cover all members' spending. Determine by the code:

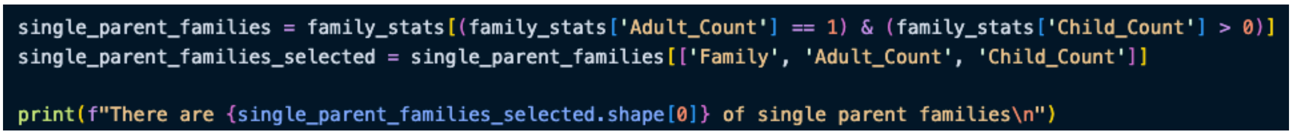


```python
deficit_families = family_stats[family_stats['Balance'] < 0]
# max_deficit_family = deficit_families.loc[deficit_families['Balance'].idxmin()]
len(deficit_families)
```

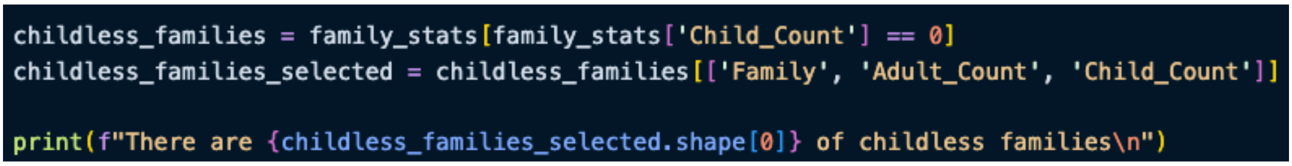
With **len(deficit_family) = 0**

Q3. Are there any single-parent families, where only one Adult is present? Are there any childless families? How do you discern this?

Since I construct the **bool** column: **is_adult** and **is_child**, by the code below I can discern.



```
single_parent_families = family_stats[(family_stats['Adult_Count'] == 1) & (family_stats['Child_Count'] > 0)]
single_parent_families_selected = single_parent_families[['Family', 'Adult_Count', 'Child_Count']]

print(f"There are {single_parent_families_selected.shape[0]} of single parent families\n")
```



```
childless_families = family_stats[family_stats['Child_Count'] == 0]
childless_families_selected = childless_families[['Family', 'Adult_Count', 'Child_Count']]

print(f"There are {childless_families_selected.shape[0]} of childless families\n")
```

Where the **single-parent** families: 23

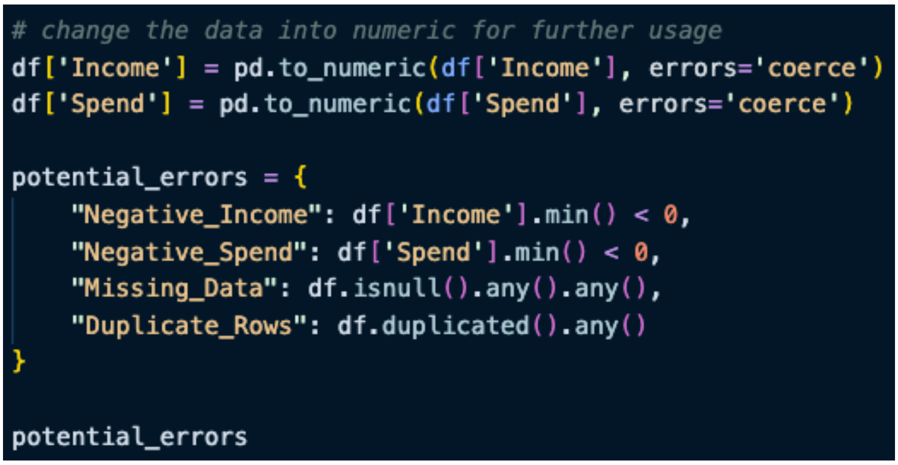
And the **childless** families: 35

Q4. Do you suspect any errors within this dataset? Examples may include negative figures, missing or duplicate data, etc. Why?

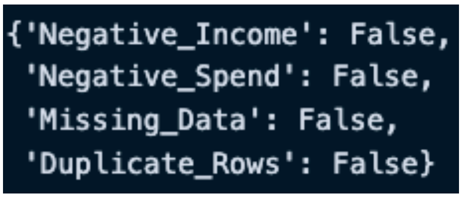
Actually nah by the code:

I checked the numeric number should be positive and check the string should not be null and the duplicated rows also.

So the data would not be **negative** (not null for sure),not **null** and not **duplicated**.



```
# change the data into numeric for further usage
df['Income'] = pd.to_numeric(df['Income'], errors='coerce')
df['Spend'] = pd.to_numeric(df['Spend'], errors='coerce')

potential_errors = {
    "Negative_Income": df['Income'].min() < 0,
    "Negative_Spend": df['Spend'].min() < 0,
    "Missing_Data": df.isnull().any().any(),
    "Duplicate_Rows": df.duplicated().any()
}

potential_errors
```



```
{'Negative_Income': False,
 'Negative_Spend': False,
 'Missing_Data': False,
 'Duplicate_Rows': False}
```

Q5. Can ChatGPT or Bing assist with the aforementioned four questions? If so, to what extent? How do you issue commands to the AI tool? If not, why not?

This is generated by chatGPT-3.5

Yes, but I like the code I wrought more.

To answer your questions, I'll first analyze the provided dataset:

- **Highest and Lowest Annual Income:**
    - To find the family with the highest and lowest annual income, I'll sum up the incomes for each family and then compare them.

- **Families with Inadequate Annual Income:**
    - I'll calculate the total spend for each family and compare it with their total income to identify families with inadequate income. The maximum shortfall would be the largest deficit among these families.
- **Single-parent Families and Childless Families:**
    - I'll identify single-parent families by checking for families with only one member listed as an adult. Childless families would be those without any child members.
- **Detecting Errors in the Dataset:**
    - I'll look for negative incomes or spending, missing data, or duplicate entries.

My prompt:
Q1. Which family boasts the highest annual income, and which has the lowest? How do you ascertain this?
Q2. Which families do not possess adequate annual income to cover all members' spending? What is the maximum shortfall? How do you determine this?
Q3. Are there any single-parent families, where only one Adult is present? Are there any childless families? How do you discern this?
Q4. Do you suspect any errors within this dataset? Examples may include negative figures, missing or duplicate data, etc. Why?

data format:

```
Family,Member,Income,Spend
family1,Adult1,2376330,1119433
family1,Adult2,130268,37337
family1,Adult3,2254489,972327
family2,Adult1,2292355,649806
family2,Adult2,298167,100723
family2,Adult3,349365,134693
family2,Child1,0,1769
family2,Child2,0,3433
family3,Adult1,2301931,807835
…
```

This is very important to my career (from *LLM Stimuli*), please think step by step (from *CoT*).

All the analysis can be seen in my script, as this part is not I need to presented on the stage, I did not present all the figure on the report. But still can see all of this in my code.

Reference

https://arxiv.org/abs/2307.11760
https://arxiv.org/abs/2201.11903
https://chat.openai.com/c/d5d78661-725d-4be6-9306-991d86d7d944
https://rich01.com/rental-yield-calculation/