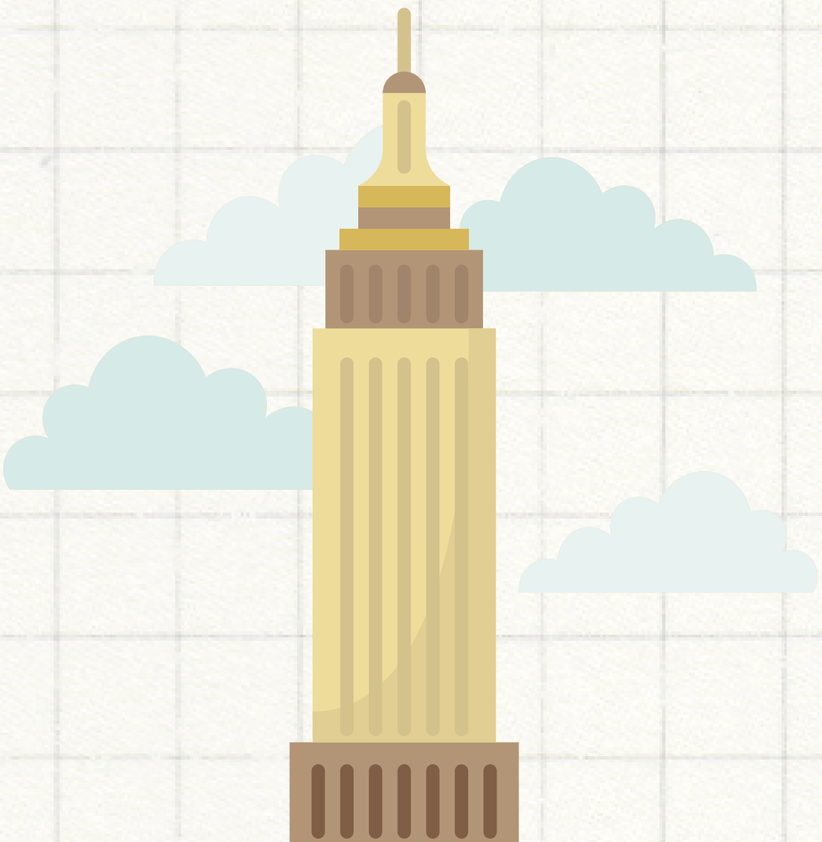
MIDTERM PROJECT

Cost of Living

Carmen Matos & Dominik Koppen

Introduction



Project Focus:
Explore how various factors impact the cost of living.

Geographical Scope:
Analyze cost variations across continents and countries.

Key Objective:
Understand the influence of different elements on living expenses.

Original DataSet

```
# Load dataset from an csv file

cost = pd.read_csv('cost-of-living.csv')
cost
```

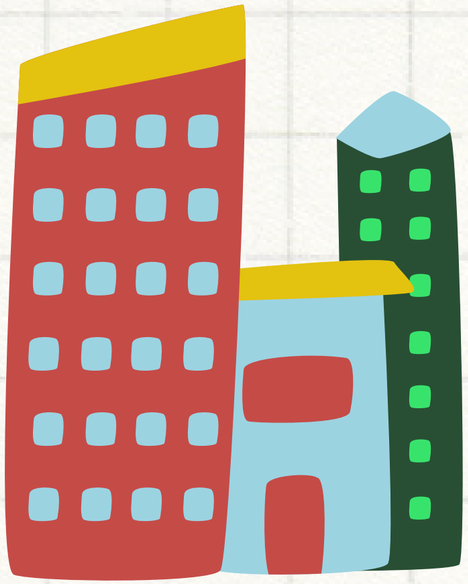
	Unnamed: 0	city	country	x1	x2	x3	x4	x5	x6	x7	...	x47	x48	x49	x50	x51	x52	x53	
0	0	Delhi	India	4.90	22.04	4.28	1.84	3.67	1.78	0.48	...	36.26	223.87	133.38	596.16	325.82	2619.46	1068.90	58
1	1	Shanghai	China	5.59	40.51	5.59	1.12	4.19	3.96	0.52	...	121.19	1080.07	564.30	2972.57	1532.23	17333.09	9174.88	138
2	2	Jakarta	Indonesia	2.54	22.25	3.50	2.02	3.18	2.19	0.59	...	80.32	482.85	270.15	1117.69	584.37	2694.05	1269.44	48
3	3	Manila	Philippines	3.54	27.40	3.54	1.24	1.90	2.91	0.93	...	61.82	559.52	281.78	1754.40	684.81	3536.04	2596.44	41
4	4	Seoul	South Korea	7.16	52.77	6.03	3.02	4.52	3.86	1.46	...	108.30	809.83	583.60	2621.05	1683.74	21847.94	10832.90	267
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
4869	4869	Peterborough	Australia	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
4870	4870	Georgetown	Australia	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
4871	4871	Ixtapa Zihuatanejo	Mexico	5.19	31.13	12.97	0.99	NaN	1.82	0.62	...	103.78	415.11	259.44	518.89	415.11	NaN	NaN	
4872	4872	Iqaluit	Canada	29.78	74.61	13.77	6.70	8.93	3.72	3.54	...	NaN	NaN	NaN	2978.11	2978.11	NaN	NaN	
4873	4873	Neiafu	Tonga	NaN	29.53	10.55	10.55	NaN	NaN	2.11	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

4874 rows × 59 columns

Cleaning Data

Dropping Columns

```
# dropping columns that we consider not relevant such as Cigarettes, Taxi, Mortgage...
cost.drop(['Unnamed: 0','x27', 'x28', 'x30', 'x31', 'x32', 'x33', 'x34', 'x35', 'x37', 'x40', 'x43', 'x55',
```



Cleaning Data

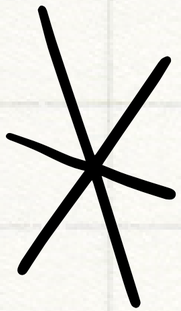
Dropping Columns

```
# dropping columns that we consider not relevant such as Cigarettes, Taxi, Mortgage...
cost.drop(['Unnamed: 0','x27', 'x28', 'x30', 'x31', 'x32', 'x33', 'x34', 'x35', 'x37', 'x40', 'x43', 'x55',
```

Combining Columns

```
### Restaurants 'x1', 'x2', 'x3', 'x4', 'x5', 'x6', 'x7', 'x8'

cost['restaurant'] = cost[['x1', 'x2', 'x3', 'x4', 'x5', 'x6', 'x7', 'x8']].mean(axis=1)
cost.drop(['x1', 'x2', 'x3', 'x4', 'x5', 'x6', 'x7', 'x8'], axis= 1, inplace= True)
```

```
### Groceries x9, x10, x11, x12, x13, x14, x15, x16, x17, x18, x19, x20, x21, x22, x23, x24, x25, x26

cost['groceries'] = cost[['x9', 'x10', 'x11', 'x12', 'x13', 'x14', 'x15', 'x16', 'x17', 'x18', 'x19', 'x20',
                          'x21', 'x22', 'x23', 'x24', 'x25', 'x26']].sum(axis=1)
cost.drop(['x9', 'x10', 'x11', 'x12', 'x13', 'x14', 'x15', 'x16', 'x17', 'x18', 'x19', 'x20',
                          'x21', 'x22', 'x23', 'x24', 'x25', 'x26'], axis= 1, inplace= True)
```



Cleaning Data

Dropping Columns

```
# dropping columns that we consider not relevant such as Cigarettes, Taxi, Mortgage...
cost.drop(['Unnamed: 0','x27', 'x28', 'x30', 'x31', 'x32', 'x33', 'x34', 'x35', 'x37', 'x40', 'x43', 'x55',
```

Combining Columns

```
### Restaurants 'x1', 'x2', 'x3', 'x4', 'x5', 'x6', 'x7', 'x8'

cost['restaurant'] = cost[['x1', 'x2', 'x3', 'x4', 'x5', 'x6', 'x7', 'x8']].mean(axis=1)
cost.drop(['x1', 'x2', 'x3', 'x4', 'x5', 'x6', 'x7', 'x8'], axis= 1, inplace= True)
```

```
### Groceries x9, x10, x11, x12, x13, x14, x15, x16, x17, x18, x19, x20, x21, x22, x23, x24, x25, x26

cost['groceries'] = cost[['x9', 'x10', 'x11', 'x12', 'x13', 'x14', 'x15', 'x16', 'x17', 'x18', 'x19', 'x20',
                          'x21', 'x22', 'x23', 'x24', 'x25', 'x26']].sum(axis=1)
cost.drop(['x9', 'x10', 'x11', 'x12', 'x13', 'x14', 'x15', 'x16', 'x17', 'x18', 'x19', 'x20',
                          'x21', 'x22', 'x23', 'x24', 'x25', 'x26'], axis= 1, inplace= True)
```

Adding Columns

```
# creating a new column for the continents

cc = CountryConverter()
cost['continent'] = cost['country'].apply(cc.convert, to='continent')
```

Clean DataSet

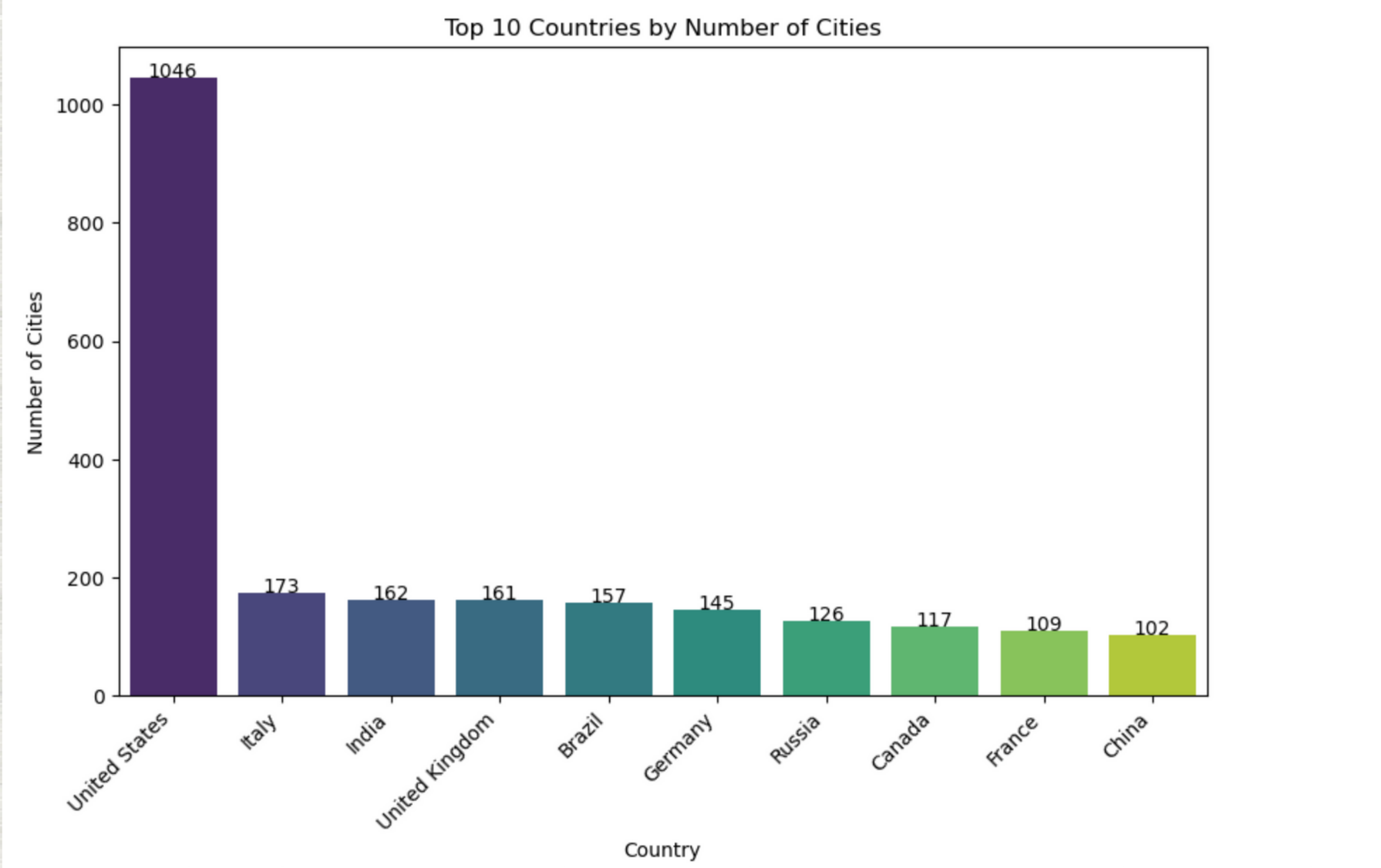
In [62]:
```
1 cost
```

Out[62]:

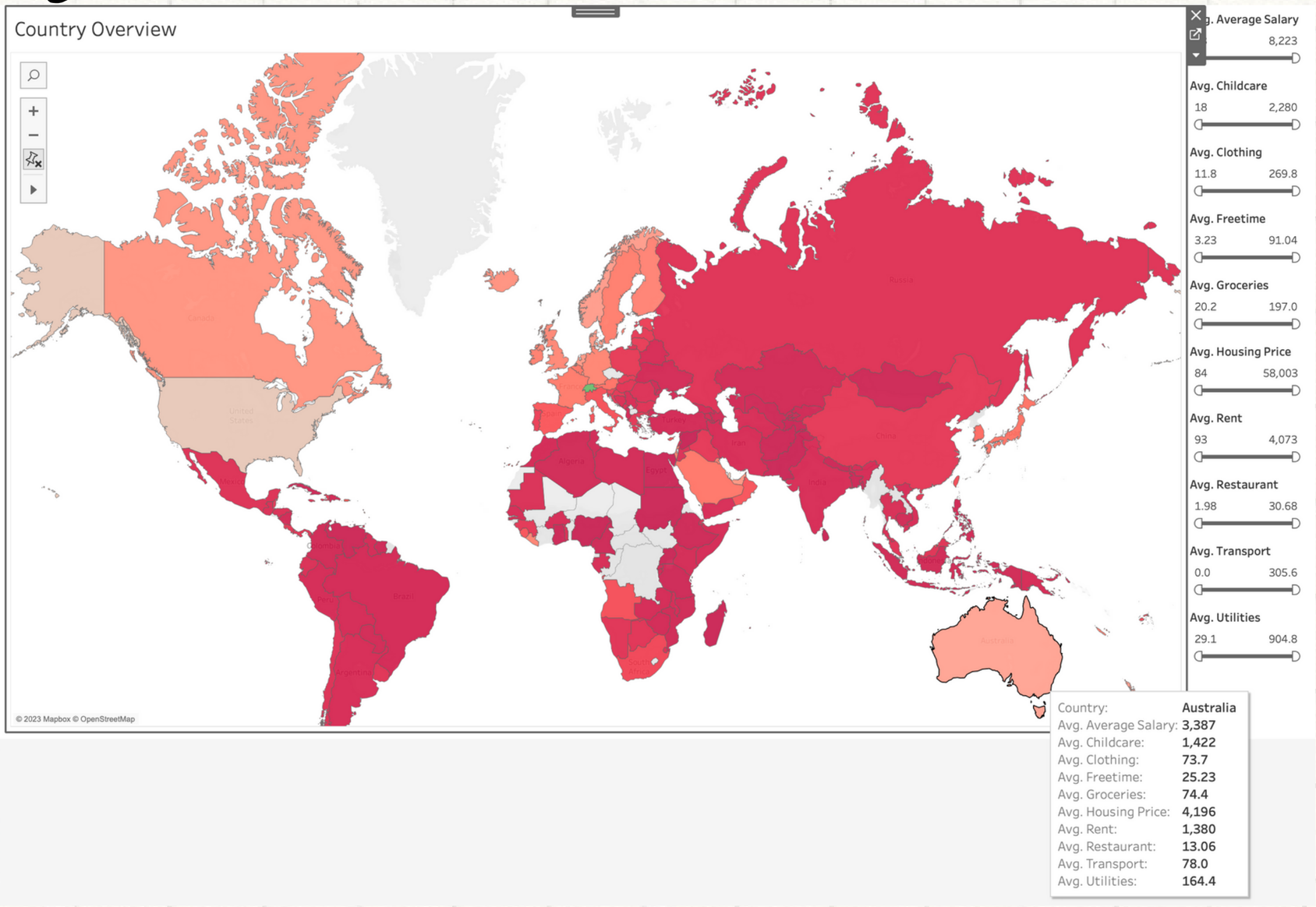
	city	country	continent	restaurant	groceries	transport	freetime	utilities	childcare	clothing	rent	housing_price	average_salary
0	Delhi	India	Asia	4.90	36.63	11.63	12.98	65.33	73.76	37.75	319.81	1844.18	586.35
1	Shanghai	China	Asia	7.72	71.51	27.94	35.80	81.50	1356.63	77.79	1537.29	13253.98	1382.83
2	Jakarta	Indonesia	Asia	4.57	62.00	9.53	17.02	110.57	132.74	54.51	613.76	1981.74	483.19
3	Manila	Philippines	Asia	5.25	46.84	11.86	26.79	135.47	254.89	49.18	820.13	3066.24	419.02
4	Seoul	South Korea	Asia	9.95	127.80	41.47	32.13	198.30	384.01	70.25	1424.56	16340.42	2672.23
...	...	...	...	...	...	...	...	...	...	...	...	...	...
4644	Port Douglas	Australia	Oceania	13.80	100.61	78.01	13.41	179.06	1341.25	57.49	1379.86	4196.00	5867.96
4645	Rockhampton	Australia	Oceania	12.64	68.43	90.53	28.50	147.37	1421.63	68.12	797.78	4498.33	3435.18
4646	Egilsstadhir	Iceland	Europe	13.91	91.46	52.96	30.90	185.94	176.53	115.51	1068.00	1888.86	2471.40
4647	Ixtapa Zihuatanejo	Mexico	America	7.59	4.63	19.72	16.86	105.46	191.87	62.26	402.14	943.56	635.37
4648	Iqaluit	Canada	America	18.14	98.29	66.04	41.01	344.96	778.75	67.13	2978.11	4616.19	2970.08

4649 rows × 13 columns

Top 10 Countries

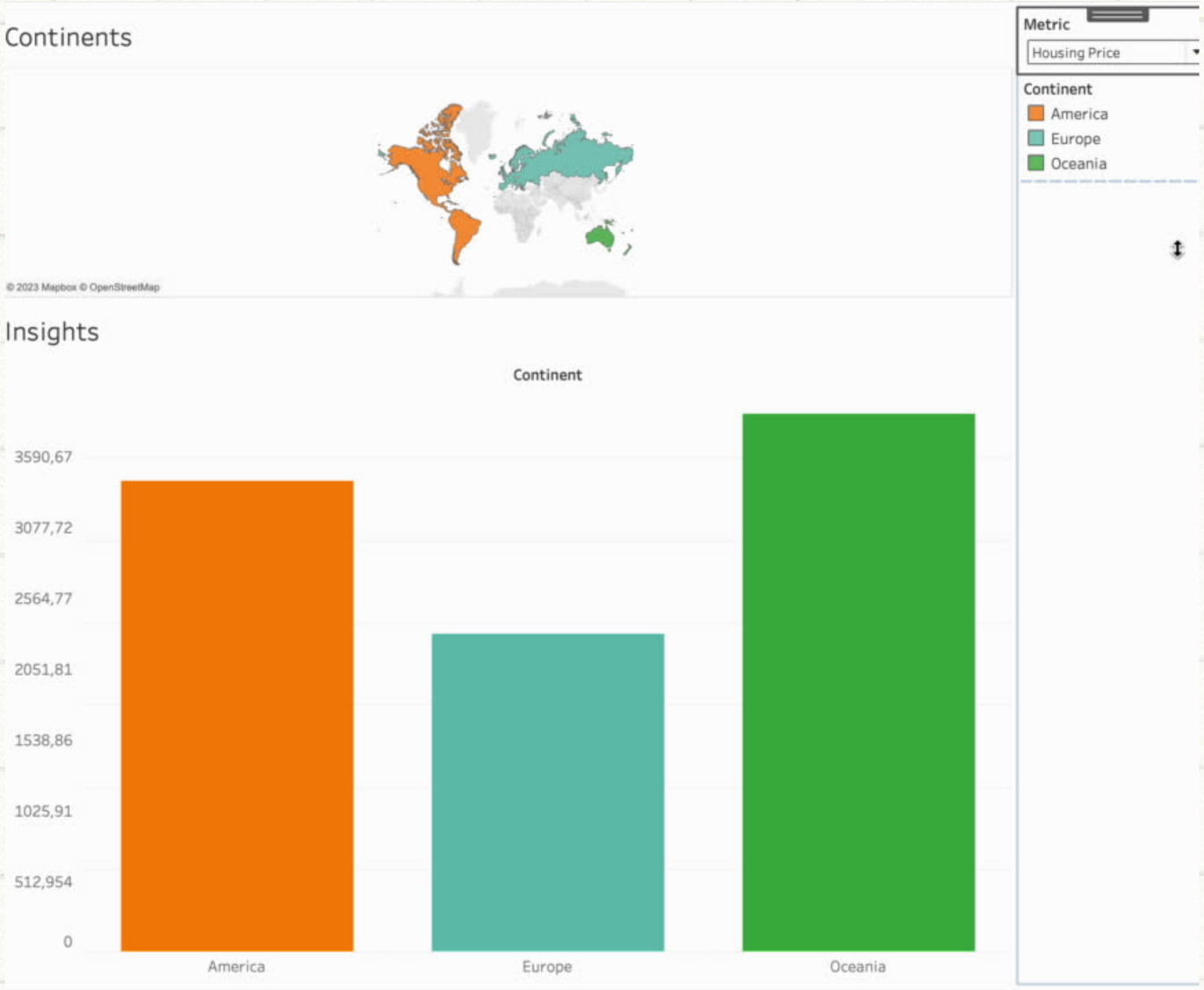


Country Overview

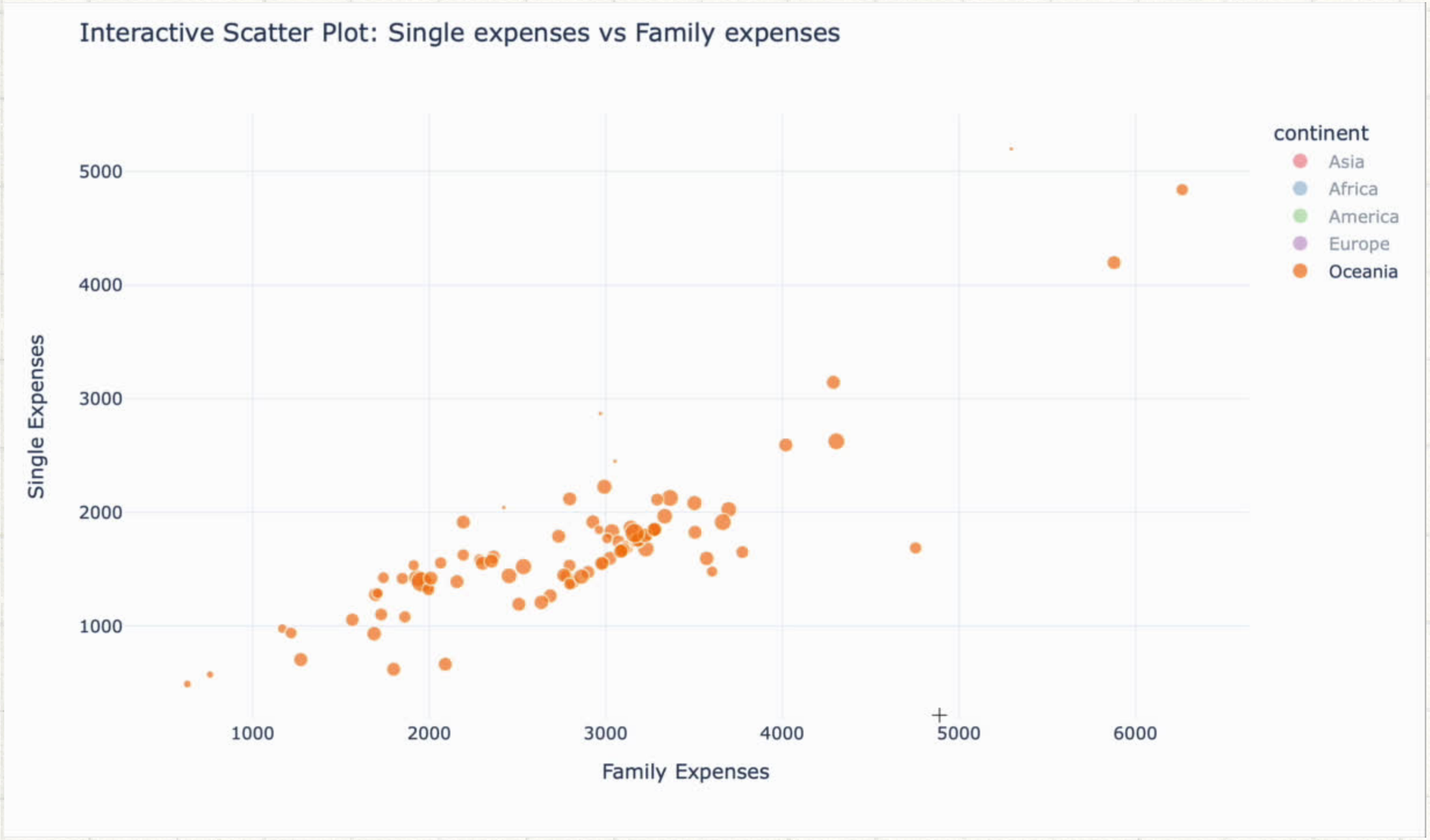


Continent Overview

Values by Continent



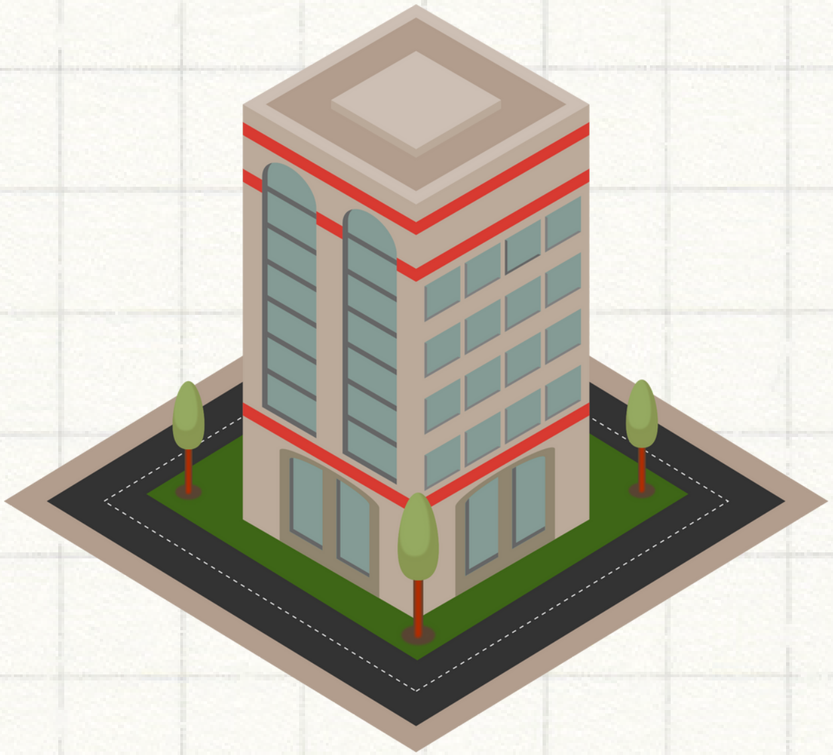
Single expenses vs. family expenses



Linear Regression and Model Validation

Setting up the target value

In [6]:
```
# X-y split (y is the target variable, in this case, "average_salary") ### numerical variables

X = cost[['restaurant', 'groceries', 'transport', 'freetime',
       'utilities', 'childcare', 'clothing', 'rent', 'housing_price']]
y = cost[['average_salary']]
```

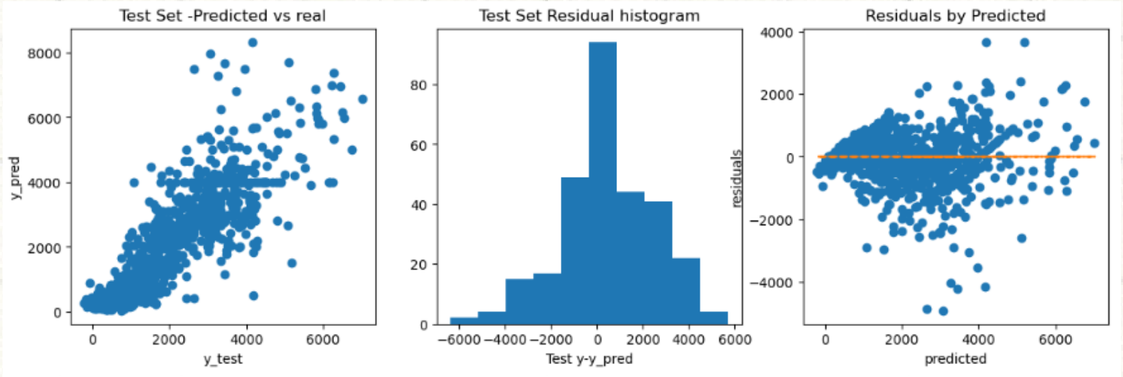


Linear Regression and Model Validation

Setting up the target value

```
In [6]: # X-y split (y is the target variable, in this case, "average_salary") ### numerical variables

        X = cost[['restaurant', 'groceries', 'transport', 'freetime',
               'utilities', 'childcare', 'clothing', 'rent', 'housing_price']]
        y = cost[['average_salary']]
```

Outcome as plots



Linear Regression and Model Validation

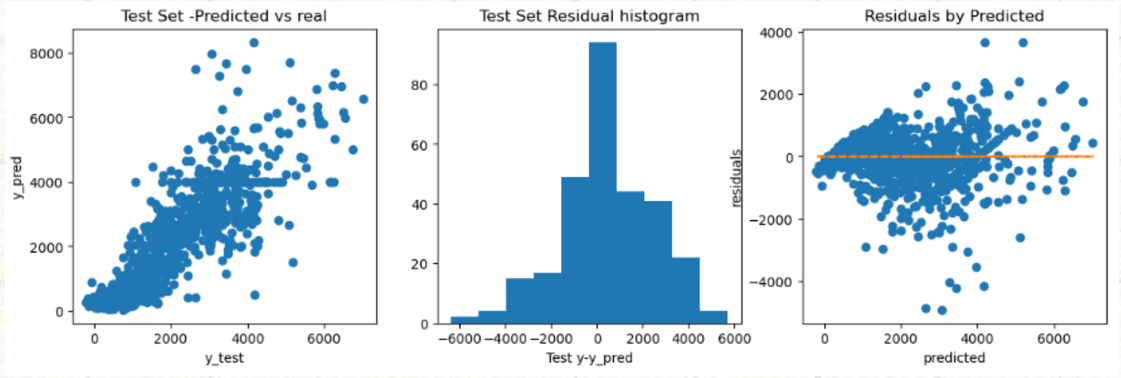
Setting up the target value

```
In [6]: # X-y split (y is the target variable, in this case, "average_salary") ### numerical variables

        X = cost[['restaurant', 'groceries', 'transport', 'freetime',
              'utilities', 'childcare', 'clothing', 'rent', 'housing_price']]
        y = cost[['average_salary']]
```

Outcome in numbers

```
In [24]: # mse: mean squared error
         mse = mse(real_vs_pred['y_test'], real_vs_pred['y_pred'])
         mse

Out[24]: 627376.659071132

In [25]: # rmse: root mean squared error
         rmse = np.sqrt(mse)
         rmse

Out[25]: 792.0711199577548

In [26]: # MAE: Mean Absolute Error
         mae = mean_absolute_error(y_test['average_salary'], y_pred)
         mae

Out[26]: 533.9678957959974

In [27]: # R-squared (R2)
         r2 = r2_score(y_test['average_salary'], y_pred)
         r2

Out[27]: 0.7675970361557893

In [28]: # adjusted R-squared
         n = len(X_test)
         p = X_test.shape[1]
         adjusted_r2 = 1 - (1 - r2) * (n - 1) / (n - p - 1)
         adjusted_r2

Out[28]: 0.765782962717283
```

Outcome as plots

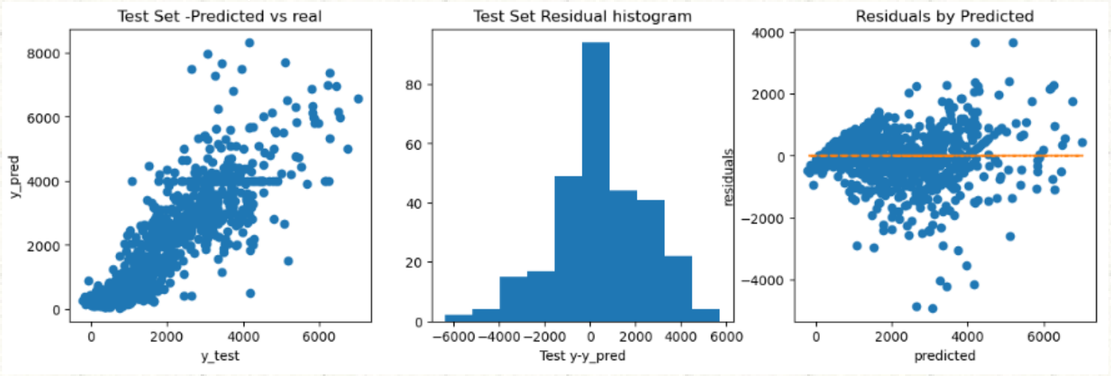


Linear Regression and Model Validation

Setting up the target value

```
In [6]: # X-y split (y is the target variable, in this case, "average_salary") ### numerical variables

        X = cost[['restaurant', 'groceries', 'transport', 'freetime',
                  'utilities', 'childcare', 'clothing', 'rent', 'housing_price']]
        y = cost[['average_salary']]
```

Outcome as plots



Outcome in numbers

```
In [24]: # mse: mean squared error
         mse = mse(real_vs_pred['y_test'], real_vs_pred['y_pred'])
         mse

Out[24]: 627376.659071132

In [25]: # rmse: root mean squared error
         rmse = np.sqrt(mse)
         rmse

Out[25]: 792.0711199577548

In [26]: # MAE: Mean Absolute Error
         mae = mean_absolute_error(y_test['average_salary'], y_pred)
         mae

Out[26]: 533.9678957959974

In [27]: # R-squared (R2)
         r2 = r2_score(y_test['average_salary'], y_pred)
         r2

Out[27]: 0.7675970361557893

In [28]: # adjusted R-squared
         n = len(X_test)
         p = X_test.shape[1]
         adjusted_r2 = 1 - (1 - r2) * (n - 1) / (n - p - 1)
         adjusted_r2

Out[28]: 0.765782962717283
```

Real vs. Predicted

```
In [22]: real_vs_pred
```

Out[22]:

	y_test	y_pred
0	3987.55	3570.256219
1	818.44	822.579474
2	415.77	292.375070
3	2233.58	3064.258408
4	367.28	288.028383
...	...	...
1158	223.61	440.298065
1159	652.99	1168.262972
1160	397.13	732.500909
1161	1644.06	1104.527273
1162	2099.65	2090.724781

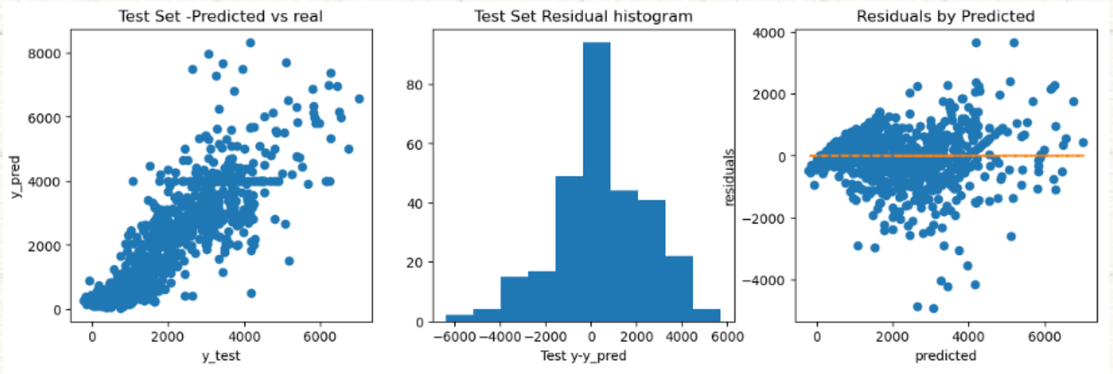
1163 rows × 2 columns

Linear Regression and Model Validation

Setting up the target value

```
In [6]: # X-y split (y is the target variable, in this case, "average_salary") ### numerical variables

        X = cost[['restaurant', 'groceries', 'transport', 'freetime',
                  'utilities', 'childcare', 'clothing', 'rent', 'housing_price']]
        y = cost[['average_salary']]
```

Outcome as plots



Outcome in numbers

```
In [24]: # mse: mean squared error
         mse = mse(real_vs_pred['y_test'], real_vs_pred['y_pred'])
         mse

Out[24]: 627376.659071132

In [25]: # rmse: root mean squared error
         rmse = np.sqrt(mse)
         rmse

Out[25]: 792.0711199577548

In [26]: # MAE: Mean Absolute Error
         mae = mean_absolute_error(y_test['average_salary'], y_pred)
         mae

Out[26]: 533.9678957959974

In [27]: # R-squared (R2)
         r2 = r2_score(y_test['average_salary'], y_pred)
         r2

Out[27]: 0.7675970361557893

In [28]: # adjusted R-squared
         n = len(X_test)
         p = X_test.shape[1]
         adjusted_r2 = 1 - (1 - r2) * (n - 1) / (n - p - 1)
         adjusted_r2

Out[28]: 0.765782962717283
```

Real vs. Predicted

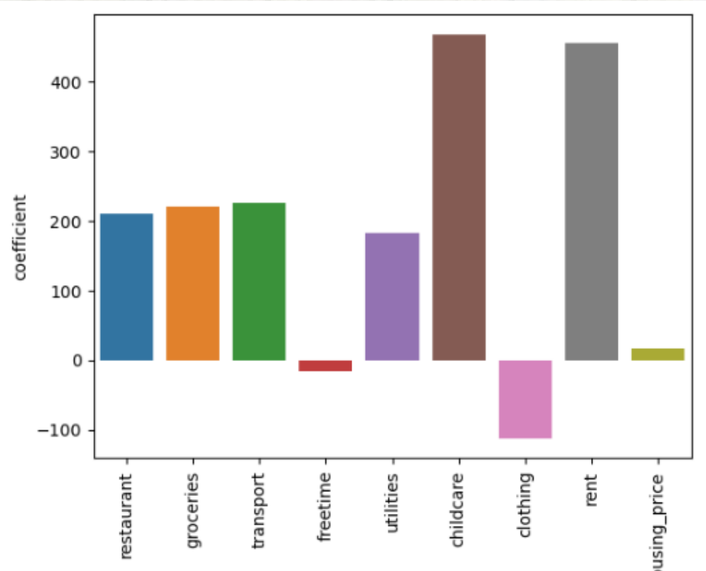
```
In [22]: real_vs_pred
```

Out[22]:

	y_test	y_pred
0	3987.55	3570.256219
1	818.44	822.579474
2	415.77	292.375070
3	2233.58	3064.258408
4	367.28	288.028383
...	...	...
1158	223.61	440.298065
1159	652.99	1168.262972
1160	397.13	732.500909
1161	1644.06	1104.527273
1162	2099.65	2090.724781

1163 rows × 2 columns

Impact of Coefficients



Conclusion

Key Cost Contributors:

Childcare and rent emerge as primary factors influencing the cost of living.

Income and Location Influence:

Cost of living is shaped by average income and local prices, where higher income correlates with a higher cost of living.

Impact on Choices:

The cost of living directly influences decisions on work and residence locations.

Thank you!