


```
from google.colab import drive
drive.mount('/content/drive')
```

 Mounted at /content/drive

```
import pandas as pd
import numpy as np
import math
import warnings
warnings.filterwarnings("ignore")
```

```
# Загрузка данных из CSV файла
data = pd.read_csv("/content/drive/MyDrive/BigData/cost-of-living.csv")
print(data.head()) # Проверяем, что данные загружены корректно
```

```
# Извлекаем данные без первого столбца, содержащего описание показателей
numeric_data = data.drop(columns=['Unnamed: 0'])
print(numeric_data)
```

```
# Количество записей в данных
N = len(data)

# Рассчитываем количество интервалов
k = 1 + 3.222 * math.log10(N)
num_bins = int(round(k))
print("k = " + str(k))
print("Количество интервалов num_bins = " + str(num_bins))
```

```
k = 6.607448585550454
Количество интервалов num_bins = 7
```

```
# Создаем пустой DataFrame для хранения результатов
interval_data = pd.DataFrame(columns=['Интервал', 'Среднее значение', 'Частота', 'Относительная частота', 'Накопительная частота', 'Нак
```

```
# Создаем интервалы и рассчитываем статистики для каждого интервала
for column in numeric_data.columns:
    hist, bins = np.histogram(numeric_data[column], bins=num_bins)
    bin_width = bins[1] - bins[0]

    intervals = []
    mean_values = []
    frequencies = []
    relative_frequencies = []
    cumulative_frequencies = []
    cumulative_relative_frequencies = []

    for i in range(num_bins):
        interval = f"{round(bins[i], 2)} - {round(bins[i] + bin_width, 2)}"
        mean_value = np.mean(numeric_data[(numeric_data[column] >= bins[i]) & (numeric_data[column] < bins[i+1])])
        frequency = hist[i]
        relative_frequency = frequency / len(numeric_data)
        cumulative_frequency = np.sum(hist[:i+1])
        cumulative_relative_frequency = cumulative_frequency / len(numeric_data)

        intervals.append(interval)
        mean_values.append(mean_value)
        frequencies.append(frequency)
        relative_frequencies.append(relative_frequency)
        cumulative_frequencies.append(cumulative_frequency)
        cumulative_relative_frequencies.append(cumulative_relative_frequency)

    # Добавляем результаты в DataFrame
    interval_data[column] = pd.Series({
        'Интервал': intervals,
        'Среднее значение': mean_values,
        'Частота': frequencies,
        'Относительная частота': relative_frequencies,
        'Накопительная частота': cumulative_frequencies,
        'Накопительная относительная частота': cumulative_relative_frequencies
    })

print(interval_data)
```

```
Накопительная частота [51, 52, 52, 52, 52, 53, 55]
Накопительная относительная частота [0.9272727272727272, 0.9454545454545454, 0.945...

Jerusalem, Israel \
Интервал [0.13 - 4988.9, 4988.9 - 9977.66, 9977.66 - 14...
Среднее значение [[212.4153846153846, 193.00596153846158, 141.2...
Частота [52, 1, 0, 0, 0, 0, 2]
Относительная частота [0.9454545454545454, 0.01818181818181818, 0.0,...
Накопительная частота [52, 53, 53, 53, 53, 53, 55]
Накопительная относительная частота [0.9454545454545454, 0.9636363636363636, 0.963...

Melbourne, Australia \
Интервал [0.12 - 2211.51, 2211.51 - 4422.91, 4422.91 - ...
Среднее значение [[71.49183673469388, 41.58836734693878, 30.562...
Частота [49, 1, 2, 0, 0, 1, 2]
Относительная частота [0.8909090909090909, 0.01818181818181818, 0.03...
Накопительная частота [49, 50, 52, 52, 52, 53, 55]
Накопительная относительная частота [0.8909090909090909, 0.9090909090909091, 0.945...

Perth, Australia \
Интервал [0.32 - 2307.41, 2307.41 - 4614.51, 4614.51 - ...
Среднее значение [[71.49183673469388, 41.58836734693878, 30.562...
Частота [49, 3, 0, 0, 1, 0, 2]
Относительная частота [0.8909090909090909, 0.05454545454545454, 0.0,...
Накопительная частота [49, 52, 52, 52, 53, 53, 55]
Накопительная относительная частота [0.8909090909090909, 0.9454545454545454, 0.945...

Sydney, Australia \
Интервал [0.38 - 2263.97, 2263.97 - 4527.56, 4527.56 - ...
Среднее значение [[51.886874999999996, 31.090208333333337, 23.8...
Частота [48, 2, 1, 1, 0, 1, 2]
Относительная частота [0.8727272727272727, 0.03636363636363636, 0.01...
Накопительная частота [48, 50, 51, 52, 52, 53, 55]
Накопительная относительная частота [0.8727272727272727, 0.9090909090909091, 0.927...

Alexandria, Egypt \
Интервал [0.01 - 3375.26, 3375.26 - 6750.5, 6750.5 - 10...
Среднее значение [[255.1254716981132, 214.17905660377357, 154.7...
Частота [53, 0, 0, 0, 0, 0, 2]
Относительная частота [0.9636363636363636, 0.0, 0.0, 0.0, 0.0, 0.0, ...
Накопительная частота [53, 53, 53, 53, 53, 53, 55]
Накопительная относительная частота [0.9636363636363636, 0.9636363636363636, 0.963...

Quito, Ecuador
Интервал [0.11 - 3617.94, 3617.94 - 7235.77, 7235.77 - ...
Среднее значение [[156.3998076923077, 85.50115384615384, 62.582...
Частота [52, 1, 0, 0, 0, 1, 1]
Относительная частота [0.9454545454545454, 0.01818181818181818, 0.0,...
Накопительная частота [52, 53, 53, 53, 53, 54, 55]
Накопительная относительная частота [0.9454545454545454, 0.9636363636363636, 0.963...

[6 rows x 166 columns]
```

```
# Фильтруем данные, чтобы оставить только числовые значения
numeric_data = data.select_dtypes(include=[np.number])

# Рассчитываем среднее, медиану и моду для числовых значений
mean = np.mean(numeric_data)
median = np.median(numeric_data)
mode = numeric_data.mode().iloc[0]

print("Среднее значение:", mean)
print("Медиана:", median)
print("Мода:", mode)
```

```
Среднее значение: Saint Petersburg, Russia    947.569636
Istanbul, Turkey    1030.352000
Izmir, Turkey    973.539818
Helsinki, Finland    1235.551636
Chisinau, Moldova    726.469091
...
Melbourne, Australia    1173.578182
Perth, Australia    1059.191273
Sydney, Australia    1256.769818
Alexandria, Egypt    928.888000
Quito, Ecuador    1067.515636
Length: 160, dtype: float64
Медиана: 5.325
Мода: Saint Petersburg, Russia    2.20
Istanbul, Turkey    3.06
Izmir, Turkey    3.06
Helsinki, Finland    1.54
Chisinau, Moldova    0.59
...
Melbourne, Australia    5.57
Perth, Australia    0.32
Sydney, Australia    4.97
Alexandria, Egypt    0.39
```

```
Quito, Ecuador          1.35  
Name: 0, Length: 160, dtype: float64
```