

# CSE163Team50 (1)

March 12, 2024

## 1 Housing Unaffordability: A Closer Look at Rent Trends and Homelessness

Code for data visualizations, and data cleaning

```
[ ]: """  
Code block to import Google Drive, We initially did this on Google Colab  
"""  
from google.colab import drive  
drive.mount('/content/drive')
```

Mounted at /content/drive

**Just incase the environment doesn't have these libraries installed** Google Colab already has it

```
[1]: !pip install pandas  
!pip install geopandas  
!pip install matplotlib  
!pip install folium  
!pip install plotly  
!pip install seaborn  
!pip install numpy
```

Requirement already satisfied: pandas in /opt/conda/lib/python3.10/site-packages (2.1.4)

Requirement already satisfied: numpy<2,>=1.22.4 in /opt/conda/lib/python3.10/site-packages (from pandas) (1.26.3)

Requirement already satisfied: python-dateutil>=2.8.2 in /opt/conda/lib/python3.10/site-packages (from pandas) (2.8.2)

Requirement already satisfied: pytz>=2020.1 in /opt/conda/lib/python3.10/site-packages (from pandas) (2023.3.post1)

Requirement already satisfied: tzdata>=2022.1 in /opt/conda/lib/python3.10/site-packages (from pandas) (2023.4)

Requirement already satisfied: six>=1.5 in /opt/conda/lib/python3.10/site-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)

Requirement already satisfied: geopandas in /opt/conda/lib/python3.10/site-packages (0.14.1)

Requirement already satisfied: fiona>=1.8.21 in /opt/conda/lib/python3.10/site-packages (from geopandas) (1.9.5)

Requirement already satisfied: packaging in /opt/conda/lib/python3.10/site-packages (from geopandas) (23.2)

Requirement already satisfied: pandas>=1.4.0 in /opt/conda/lib/python3.10/site-packages (from geopandas) (2.1.4)

Requirement already satisfied: pyproj>=3.3.0 in /opt/conda/lib/python3.10/site-packages (from geopandas) (3.6.1)

Requirement already satisfied: shapely>=1.8.0 in /opt/conda/lib/python3.10/site-packages (from geopandas) (2.0.2)

Requirement already satisfied: attrs>=19.2.0 in /opt/conda/lib/python3.10/site-packages (from fiona>=1.8.21->geopandas) (23.2.0)

Requirement already satisfied: certifi in /opt/conda/lib/python3.10/site-packages (from fiona>=1.8.21->geopandas) (2023.11.17)

Requirement already satisfied: click~=8.0 in /opt/conda/lib/python3.10/site-packages (from fiona>=1.8.21->geopandas) (8.1.7)

Requirement already satisfied: click-plugins>=1.0 in /opt/conda/lib/python3.10/site-packages (from fiona>=1.8.21->geopandas) (1.1.1)

Requirement already satisfied: cligj>=0.5 in /opt/conda/lib/python3.10/site-packages (from fiona>=1.8.21->geopandas) (0.7.2)

Requirement already satisfied: six in /opt/conda/lib/python3.10/site-packages (from fiona>=1.8.21->geopandas) (1.16.0)

Requirement already satisfied: setuptools in /opt/conda/lib/python3.10/site-packages (from fiona>=1.8.21->geopandas) (69.0.3)

Requirement already satisfied: numpy<2,>=1.22.4 in /opt/conda/lib/python3.10/site-packages (from pandas>=1.4.0->geopandas) (1.26.3)

Requirement already satisfied: python-dateutil>=2.8.2 in /opt/conda/lib/python3.10/site-packages (from pandas>=1.4.0->geopandas) (2.8.2)

Requirement already satisfied: pytz>=2020.1 in /opt/conda/lib/python3.10/site-packages (from pandas>=1.4.0->geopandas) (2023.3.post1)

Requirement already satisfied: tzdata>=2022.1 in /opt/conda/lib/python3.10/site-packages (from pandas>=1.4.0->geopandas) (2023.4)

Requirement already satisfied: matplotlib in /opt/conda/lib/python3.10/site-packages (3.8.2)

Requirement already satisfied: contourpy>=1.0.1 in /opt/conda/lib/python3.10/site-packages (from matplotlib) (1.2.0)

Requirement already satisfied: cycler>=0.10 in /opt/conda/lib/python3.10/site-packages (from matplotlib) (0.12.1)

Requirement already satisfied: fonttools>=4.22.0 in /opt/conda/lib/python3.10/site-packages (from matplotlib) (4.47.0)

Requirement already satisfied: kiwisolver>=1.3.1 in /opt/conda/lib/python3.10/site-packages (from matplotlib) (1.4.5)

Requirement already satisfied: numpy<2,>=1.21 in /opt/conda/lib/python3.10/site-packages (from matplotlib) (1.26.3)

Requirement already satisfied: packaging>=20.0 in /opt/conda/lib/python3.10/site-packages (from matplotlib) (23.2)

Requirement already satisfied: pillow>=8 in /opt/conda/lib/python3.10/site-packages (from matplotlib) (10.2.0)

Requirement already satisfied: pyparsing>=2.3.1 in /opt/conda/lib/python3.10/site-packages (from matplotlib) (3.1.1)

Requirement already satisfied: python-dateutil>=2.7 in /opt/conda/lib/python3.10/site-packages (from matplotlib) (2.8.2)

Requirement already satisfied: six>=1.5 in /opt/conda/lib/python3.10/site-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)

Requirement already satisfied: folium in /opt/conda/lib/python3.10/site-packages (0.15.1)

Requirement already satisfied: branca>=0.6.0 in /opt/conda/lib/python3.10/site-packages (from folium) (0.7.0)

Requirement already satisfied: jinja2>=2.9 in /opt/conda/lib/python3.10/site-packages (from folium) (3.1.2)

Requirement already satisfied: numpy in /opt/conda/lib/python3.10/site-packages (from folium) (1.26.3)

Requirement already satisfied: requests in /opt/conda/lib/python3.10/site-packages (from folium) (2.31.0)

Requirement already satisfied: xyzservices in /opt/conda/lib/python3.10/site-packages (from folium) (2023.10.1)

Requirement already satisfied: MarkupSafe>=2.0 in /opt/conda/lib/python3.10/site-packages (from jinja2>=2.9->folium) (2.1.3)

Requirement already satisfied: charset-normalizer<4,>=2 in /opt/conda/lib/python3.10/site-packages (from requests->folium) (3.3.2)

Requirement already satisfied: idna<4,>=2.5 in /opt/conda/lib/python3.10/site-packages (from requests->folium) (3.6)

Requirement already satisfied: urllib3<3,>=1.21.1 in /opt/conda/lib/python3.10/site-packages (from requests->folium) (1.26.18)

Requirement already satisfied: certifi>=2017.4.17 in /opt/conda/lib/python3.10/site-packages (from requests->folium) (2023.11.17)

Collecting plotly

Using cached plotly-5.19.0-py3-none-any.whl.metadata (7.0 kB)

Collecting tenacity>=6.2.0 (from plotly)

Using cached tenacity-8.2.3-py3-none-any.whl.metadata (1.0 kB)

Requirement already satisfied: packaging in /opt/conda/lib/python3.10/site-packages (from plotly) (23.2)

Using cached plotly-5.19.0-py3-none-any.whl (15.7 MB)

Using cached tenacity-8.2.3-py3-none-any.whl (24 kB)

Installing collected packages: tenacity, plotly

Successfully installed plotly-5.19.0 tenacity-8.2.3

Requirement already satisfied: seaborn in /opt/conda/lib/python3.10/site-packages (0.13.1)

Requirement already satisfied: numpy!=1.24.0,>=1.20 in /opt/conda/lib/python3.10/site-packages (from seaborn) (1.26.3)

Requirement already satisfied: pandas>=1.2 in /opt/conda/lib/python3.10/site-packages (from seaborn) (2.1.4)

Requirement already satisfied: matplotlib!=3.6.1,>=3.4 in /opt/conda/lib/python3.10/site-packages (from seaborn) (3.8.2)

Requirement already satisfied: contourpy>=1.0.1 in /opt/conda/lib/python3.10/site-packages (from matplotlib!=3.6.1,>=3.4->seaborn)

(1.2.0)  
Requirement already satisfied: cycler>=0.10 in /opt/conda/lib/python3.10/site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (0.12.1)  
Requirement already satisfied: fonttools>=4.22.0 in /opt/conda/lib/python3.10/site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (4.47.0)  
Requirement already satisfied: kiwisolver>=1.3.1 in /opt/conda/lib/python3.10/site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (1.4.5)  
Requirement already satisfied: packaging>=20.0 in /opt/conda/lib/python3.10/site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (23.2)  
Requirement already satisfied: pillow>=8 in /opt/conda/lib/python3.10/site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (10.2.0)  
Requirement already satisfied: pyparsing>=2.3.1 in /opt/conda/lib/python3.10/site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (3.1.1)  
Requirement already satisfied: python-dateutil>=2.7 in /opt/conda/lib/python3.10/site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (2.8.2)  
Requirement already satisfied: pytz>=2020.1 in /opt/conda/lib/python3.10/site-packages (from pandas>=1.2->seaborn) (2023.3.post1)  
Requirement already satisfied: tzdata>=2022.1 in /opt/conda/lib/python3.10/site-packages (from pandas>=1.2->seaborn) (2023.4)  
Requirement already satisfied: six>=1.5 in /opt/conda/lib/python3.10/site-packages (from python-dateutil>=2.7->matplotlib!=3.6.1,>=3.4->seaborn) (1.16.0)  
Requirement already satisfied: numpy in /opt/conda/lib/python3.10/site-packages (1.26.3)

## Libraries used for this project

```
[1]: #imported libraries for this project
from ast import Index
import pandas as pd
import geopandas as gpd
import matplotlib.pyplot as plt
import numpy as np
import folium
import plotly.express as px
import seaborn as sns
import plotly.graph_objects as go
from plotly.subplots import make_subplots
```

This function is meant to convert our excel file extensions to csv format for data processing. We utilized the pandas library to achieve this.

```
[2]: def convert_xlxs_to_csv(xlxs_filename, output_csv):
    """
```

*This function is meant to convert all the datasets obtained as XLSX into a CSV format for data manipulation.*

*Parameters:*

*xlsx\_filename : Excel filename*

*output\_csv : Desired output include .csv extension*

*"""*

```
excel = pd.read_excel(xlsx_filename)
```

```
excel.to_csv(output_csv, index=False)
```

```
print("Successful")
```

## 1.1 Reading All The Datasets

This contains all the datasets used in this code.

```
[3]: rent_income_census = pd.read_csv('datasets/rent_income.csv', header=0)
zillow_rent = pd.read_csv('datasets/zillow_rent_price.csv')
homeless2020 = pd.read_csv('datasets/wa_pit_homeless2020.csv')
homeless2022 = pd.read_csv('datasets/wa_pit_homeless2022.csv')
homeless_shelter2020 = pd.read_csv('datasets/homeless_shelter_data2020.csv')
homeless_shelter2022 = pd.read_csv('datasets/homeless_shelter_data2022.csv')
county_pop = pd.read_csv('datasets/county_pop.csv')
```

```
[4]: rent_income_census
```

```
[4]:
```

	Unnamed: 0	Total:	Unnamed: 2	\
0	Label	NaN	Less than 10.0 percent	
1	Adams County, Washington	NaN	NaN	
2	Estimate	2,212	69	
3	Asotin County, Washington	NaN	NaN	
4	Estimate	2,640	172	
..	...	...	...	
74	Estimate	33,729	851	
75	Whitman County, Washington	NaN	NaN	
76	Estimate	9,840	292	
77	Yakima County, Washington	NaN	NaN	
78	Estimate	32,234	1,149	

	Unnamed: 3	Unnamed: 4	Unnamed: 5	\
0	10.0 to 14.9 percent	15.0 to 19.9 percent	20.0 to 24.9 percent	
1	NaN	NaN	NaN	
2	591	158	130	
3	NaN	NaN	NaN	
4	230	332	234	
..	...	...	...	
74	1,890	3,940	3,971	
75	NaN	NaN	NaN	

76	839	865	795
77	NaN	NaN	NaN
78	3,266	4,132	3,835

	Unnamed: 6	Unnamed: 7	Unnamed: 8 \
0	25.0 to 29.9 percent	30.0 to 34.9 percent	35.0 to 39.9 percent
1	NaN	NaN	NaN
2	226	148	102
3	NaN	NaN	NaN
4	387	273	195
..	...	...	...
74	3,493	2,595	2,808
75	NaN	NaN	NaN
76	612	386	586
77	NaN	NaN	NaN
78	3,894	2,604	2,004

	Unnamed: 9	Unnamed: 10	Unnamed: 11
0	40.0 to 49.9 percent	50.0 percent or more	Not computed
1	NaN	NaN	NaN
2	161	420	207
3	NaN	NaN	NaN
4	179	517	121
..	...	...	...
74	3,335	9,355	1,491
75	NaN	NaN	NaN
76	832	3,785	848
77	NaN	NaN	NaN
78	1,976	6,644	2,730

[79 rows x 12 columns]

## 1.2 rent\_income.csv Data Cleaning

The dataset initially downloaded from the website had incorrect alignment with rows, and counties displaying NaN, and the estimates row below it displaying the values. So we cleaned it by removing the estimates row and giving that rows values to the county row for each county, and also removing/altering strings so we can merge the dataframe later with another dataset.

```
[5]: # Make first row as the header
new_header = rent_income_census.iloc[0] # grab the first row for the header
rent_income_census = rent_income_census[1:] # take the data less the header row
rent_income_census.columns = new_header # set the header row as the df header

# Rename 'Total:' to 'Total'
rent_income_census = rent_income_census.rename(columns={'Total:': 'Total'})
```

```

[6]: #reassign column labels to "county" and "total"
rent_income_census.columns.values[0] = 'County'
rent_income_census.columns.values[1] = 'Total'

[7]: rent_income_census.columns = rent_income_census.columns.values
for index in range(len(rent_income_census) - 1, -1, -1): # Change this line to
    ↪go to 0
    # Check if the current row is an 'Estimate' row
    if "Estimate" in rent_income_census.loc[index, 'County']:
        # Move 'Estimate' values up to the corresponding county row
        for col in rent_income_census.columns[1:]:
            # Only update if it's not the first row (avoid index out of bounds)
            ↪and the value is not NaN
            if index > 0 and pd.notna(rent_income_census.loc[index, col]):
                rent_income_census.loc[index - 1, col] = rent_income_census.
            ↪loc[index, col]
            # Remove the 'Estimate' row and reset the index
            rent_income_census.drop(index, inplace=True)
            rent_income_census.reset_index(drop=True, inplace=True)

# Drop duplicates, excluding the 'County' column
rent_income_census = rent_income_census.
    ↪drop_duplicates(subset=rent_income_census.columns.difference(['County']))

# Remove ', Washington' from the 'County' column
rent_income_census['County'] = rent_income_census['County'].str.replace(', 
    ↪Washington', '')

[8]: string_numbers = []
numeric_cols = ['Total', 'Less than 10.0 percent', '10.0 to 14.9 percent',
    '15.0 to 19.9 percent', '20.0 to 24.9 percent', '25.0 to 29.9
    ↪percent',
    '30.0 to 34.9 percent', '35.0 to 39.9 percent', '40.0 to 49.9
    ↪percent',
    '50.0 percent or more', 'Not computed'
    ]
for col in numeric_cols:
    # Record the string representations of numbers
    string_numbers.extend(rent_income_census[col].loc[rent_income_census[col].
    ↪apply(lambda x: isinstance(x, str))].tolist())
    # Remove commas and convert to numeric, replacing non-numeric values with
    ↪NaNs
    rent_income_census[col] = pd.to_numeric(rent_income_census[col].str.
    ↪replace(',', ''), errors='coerce')
print(rent_income_census.dtypes)

```

```

County          object
Total           float64
Less than 10.0 percent  float64
10.0 to 14.9 percent  float64
15.0 to 19.9 percent  float64
20.0 to 24.9 percent  float64
25.0 to 29.9 percent  float64
30.0 to 34.9 percent  float64
35.0 to 39.9 percent  float64
40.0 to 49.9 percent  float64
50.0 percent or more  float64
Not computed      float64
dtype: object

```

```

[9]: # reorganize the index after cleaning
rent_income_census = rent_income_census.reset_index(drop=True)

```

```

[10]: # to clean the very last row of the dataframe
yakima_index = rent_income_census.index[rent_income_census['County'] == 'Yakima_
↳County'].tolist()

if yakima_index and yakima_index[0] + 1 < len(rent_income_census):
    yakima_index = yakima_index[0]
    estimate_index = yakima_index + 1

    for col in rent_income_census.columns[1:]:
        rent_income_census.at[yakima_index, col] = rent_income_census.
↳at[estimate_index, col]

    rent_income_census.drop(estimate_index, inplace=True)
    rent_income_census.reset_index(drop=True, inplace=True)

```

```

[11]: rent_income_census

```

```

[11]:

```

	County	Total	Less than 10.0 percent \
0	Adams County	2212.0	69.0
1	Asotin County	2640.0	172.0
2	Benton County	24125.0	937.0
3	Chelan County	11116.0	691.0
4	Clallam County	9234.0	280.0
5	Clark County	62839.0	1654.0
6	Columbia County	450.0	54.0
7	Cowlitz County	14503.0	408.0
8	Douglas County	4654.0	342.0
9	Ferry County	721.0	77.0
10	Franklin County	8526.0	521.0
11	Garfield County	227.0	9.0



12	Grant County	11861.0	931.0
13	Grays Harbor County	8497.0	329.0
14	Island County	9291.0	238.0
15	Jefferson County	3112.0	64.0
16	King County	401313.0	15460.0
17	Kitsap County	31981.0	1141.0
18	Kittitas County	7402.0	272.0
19	Klickitat County	2405.0	147.0
20	Lewis County	8525.0	356.0
21	Lincoln County	1004.0	96.0
22	Mason County	5305.0	288.0
23	Okanogan County	5138.0	507.0
24	Pacific County	1910.0	40.0
25	Pend Oreille County	1374.0	168.0
26	Pierce County	121139.0	3077.0
27	San Juan County	1945.0	75.0
28	Skagit County	15078.0	447.0
29	Skamania County	903.0	70.0
30	Snohomish County	96712.0	2276.0
31	Spokane County	77399.0	3078.0
32	Stevens County	3745.0	330.0
33	Thurston County	37865.0	1056.0
34	Wahkiakum County	288.0	0.0
35	Walla Walla County	7778.0	248.0
36	Whatcom County	33729.0	851.0
37	Whitman County	9840.0	292.0
38	Yakima County	32234.0	1149.0

	10.0 to 14.9 percent	15.0 to 19.9 percent	20.0 to 24.9 percent \
0	591.0	158.0	130.0
1	230.0	332.0	234.0
2	2530.0	2841.0	3165.0
3	1151.0	1007.0	1634.0
4	780.0	1328.0	897.0
5	4864.0	7546.0	7752.0
6	21.0	108.0	11.0
7	1067.0	1720.0	1781.0
8	384.0	699.0	553.0
9	54.0	87.0	41.0
10	902.0	903.0	951.0
11	41.0	10.0	16.0
12	1199.0	1750.0	1324.0
13	752.0	1148.0	1073.0
14	842.0	814.0	1350.0
15	316.0	565.0	258.0
16	36924.0	53568.0	54386.0
17	2270.0	3796.0	3782.0

18	420.0	1051.0	823.0
19	78.0	201.0	153.0
20	648.0	1017.0	1051.0
21	132.0	105.0	148.0
22	290.0	551.0	730.0
23	759.0	496.0	494.0
24	163.0	175.0	242.0
25	57.0	134.0	102.0
26	7639.0	14063.0	16708.0
27	174.0	192.0	213.0
28	1304.0	1665.0	1483.0
29	64.0	101.0	85.0
30	6252.0	11127.0	12912.0
31	5972.0	8810.0	9362.0
32	330.0	436.0	362.0
33	2439.0	4385.0	4806.0
34	30.0	0.0	2.0
35	440.0	661.0	1007.0
36	1890.0	3940.0	3971.0
37	839.0	865.0	795.0
38	3266.0	4132.0	3835.0

	25.0 to 29.9 percent	30.0 to 34.9 percent	35.0 to 39.9 percent \
0	226.0	148.0	102.0
1	387.0	273.0	195.0
2	2663.0	2156.0	1449.0
3	1498.0	722.0	442.0
4	847.0	670.0	482.0
5	8050.0	5697.0	4568.0
6	55.0	37.0	11.0
7	1906.0	1396.0	1139.0
8	452.0	292.0	278.0
9	33.0	44.0	47.0
10	862.0	835.0	584.0
11	10.0	30.0	0.0
12	1098.0	1079.0	662.0
13	830.0	560.0	543.0
14	918.0	1135.0	647.0
15	318.0	259.0	203.0
16	48377.0	35643.0	26496.0
17	3925.0	3489.0	2454.0
18	592.0	430.0	428.0
19	245.0	404.0	42.0
20	876.0	875.0	390.0
21	94.0	44.0	16.0
22	527.0	251.0	220.0
23	397.0	430.0	180.0

24	199.0	92.0	169.0
25	68.0	101.0	89.0
26	13715.0	11967.0	9113.0
27	158.0	140.0	102.0
28	1709.0	1765.0	824.0
29	101.0	57.0	33.0
30	12365.0	10858.0	6655.0
31	9072.0	7494.0	5164.0
32	324.0	313.0	356.0
33	4120.0	4239.0	2255.0
34	45.0	8.0	22.0
35	922.0	533.0	715.0
36	3493.0	2595.0	2808.0
37	612.0	386.0	586.0
38	3894.0	2604.0	2004.0

	40.0 to 49.9 percent	50.0 percent or more	Not computed
0	161.0	420.0	207.0
1	179.0	517.0	121.0
2	2502.0	4517.0	1365.0
3	1014.0	1891.0	1066.0
4	1185.0	1866.0	899.0
5	6091.0	13274.0	3343.0
6	15.0	70.0	68.0
7	1293.0	3091.0	702.0
8	95.0	854.0	705.0
9	66.0	101.0	171.0
10	730.0	1533.0	705.0
11	10.0	39.0	62.0
12	639.0	1855.0	1324.0
13	900.0	1521.0	841.0
14	825.0	1885.0	637.0
15	194.0	431.0	504.0
16	32446.0	83511.0	14502.0
17	2653.0	6947.0	1524.0
18	475.0	2284.0	627.0
19	265.0	385.0	485.0
20	781.0	1745.0	786.0
21	42.0	80.0	247.0
22	235.0	1360.0	853.0
23	368.0	698.0	809.0
24	195.0	385.0	250.0
25	124.0	198.0	333.0
26	12071.0	26999.0	5787.0
27	127.0	422.0	342.0
28	1404.0	3320.0	1157.0
29	72.0	186.0	134.0

30	8340.0	22115.0	3812.0
31	7388.0	17766.0	3293.0
32	171.0	540.0	583.0
33	2891.0	10028.0	1646.0
34	39.0	116.0	26.0
35	810.0	1744.0	698.0
36	3335.0	9355.0	1491.0
37	832.0	3785.0	848.0
38	1976.0	6644.0	2730.0

```
[12]: # max values of all columns
rent_income_census.max()
```

```
[12]: County          Yakima County
Total                401313.0
Less than 10.0 percent    15460.0
10.0 to 14.9 percent     36924.0
15.0 to 19.9 percent     53568.0
20.0 to 24.9 percent     54386.0
25.0 to 29.9 percent     48377.0
30.0 to 34.9 percent     35643.0
35.0 to 39.9 percent     26496.0
40.0 to 49.9 percent     32446.0
50.0 percent or more     83511.0
Not computed             14502.0
dtype: object
```

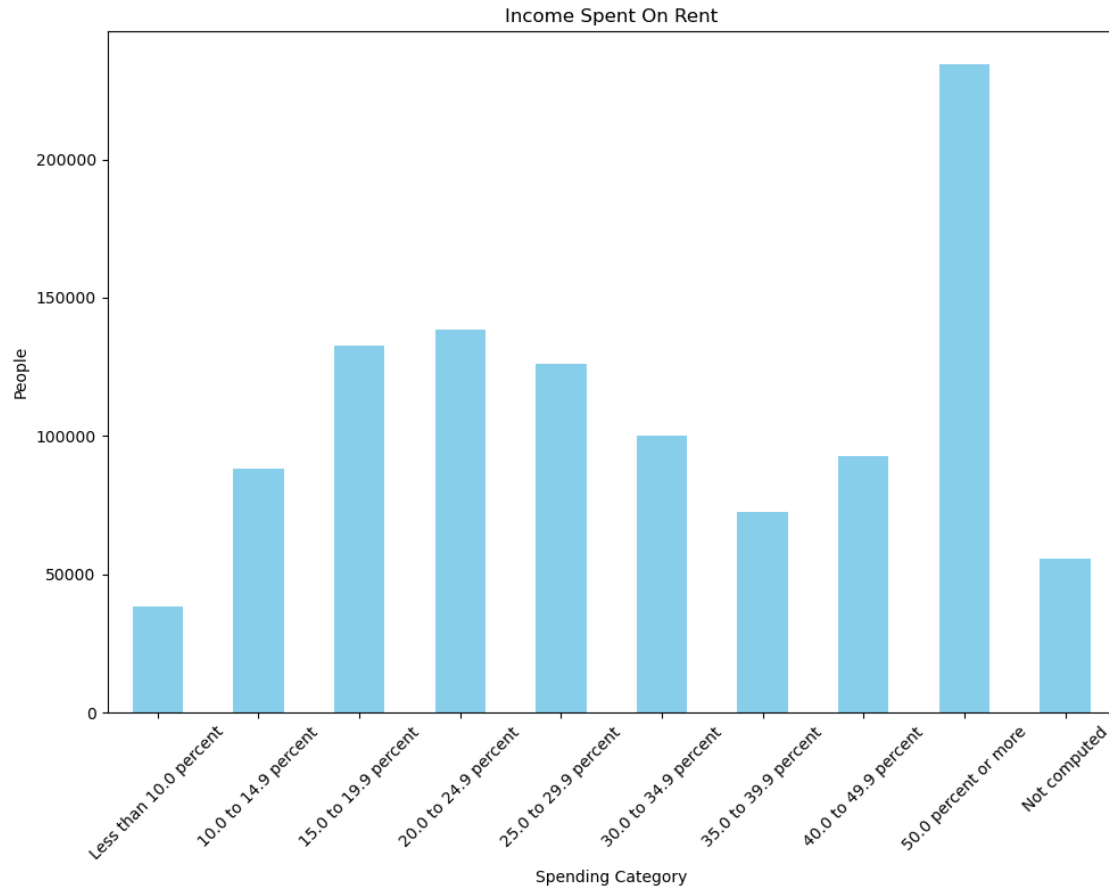
### 1.3 Bar Plot Showing Income Spent on Rent

datasets used : rent\_income.csv

```
[13]: column_sums = rent_income_census.drop(['County', 'Total'], axis=1).sum()

plt.figure(figsize=(10, 8))
column_sums.plot(kind='bar', color='skyblue')
plt.title('Income Spent On Rent')
plt.xlabel('Spending Category')
plt.ylabel('People')
plt.xticks(rotation=45)
plt.tight_layout()

plt.show()
```



## 2 wa\_pit\_homeless2020 & 2022 Data Cleaning

```
[14]: homeless2020 = homeless2020.replace('< 11', 11).replace('*', 0)
```

```
[15]: homeless2020_column_names = [
    'County',
    'Persons of Household with no Minor',
    'Households with no Minors',
    'Persons in Households with Minors',
    'Households with Minors',
    'Persons in Households with only Minors',
    'Households with only Minors',
    'Total Persons',
    'Total Households'
]
homeless2020.columns = homeless2020_column_names
new_num_column_names = [
    'Persons of Household with no Minor',
```

```

    'Households with no Minors',
    'Persons in Households with Minors',
    'Households with Minors',
    'Persons in Households with only Minors',
    'Households with only Minors',
    'Total Persons',
    'Total Households'
]

num_col_values = []

for col in new_num_column_names:
    num_col_values.extend(homeless2020[col].loc[homeless2020[col].apply(lambda x:
↪isinstance(x, str))].tolist())
    homeless2020[col] = pd.to_numeric(homeless2020[col]).astype(float)
homeless2020.dtypes

```

```

[15]: County                                object
Persons of Household with no Minor          float64
Households with no Minors                   float64
Persons in Households with Minors           float64
Households with Minors                     float64
Persons in Households with only Minors       float64
Households with only Minors                 float64
Total Persons                             float64
Total Households                          float64
dtype: object

```

```

[21]: homeless2020['Sum of people'] = homeless2020['Persons of Household with no_
↪Minor'] + homeless2020['Persons in Households with Minors'] +
↪homeless2020['Persons in Households with only Minors']
homeless2020['Total Persons'] = homeless2020['Sum of people']
homeless2020['Sum of Households'] = homeless2020['Households with no Minors'] +
↪homeless2020['Households with Minors'] + homeless2020['Households with only_
↪Minors']
homeless2020['Total Households'] = homeless2020['Sum of Households']
column_drop = ['Sum of people', 'Sum of Households']
homeless2020 = homeless2020.drop(columns = column_drop)
homeless2020 = homeless2020.drop(index=39)
homeless2020

```

```

[21]:
      County  Persons of Household with no Minor \
0      Adams County                                0.0
1      Asotin County                               13.0
2      Benton County                              50.0
3      Chelan County                             229.0
4      Clallam County                             151.0

```

5	Clark County	536.0
6	Columbia County	11.0
7	Cowlitz County	244.0
8	Douglas County	12.0
9	Ferry County	11.0
10	Franklin County	44.0
11	Garfield County	11.0
12	Grant County	104.0
13	Grays Harbor County	92.0
14	Island County	105.0
15	Jefferson County	119.0
16	King County	7707.0
17	Kitsap County	390.0
18	Kittitas County	11.0
19	Klickitat County	28.0
20	Lewis County	97.0
21	Lincoln County	0.0
22	Mason County	90.0
23	Okanogan County	55.0
24	Pacific County	48.0
25	Pend Oreille County	11.0
26	Pierce County	1527.0
27	San Juan County	55.0
28	Skagit County	181.0
29	Skamania County	36.0
30	Snohomish County	818.0
31	Spokane County	1171.0
32	Stevens County	35.0
33	Thurston County	672.0
34	Wahkiakum County	11.0
35	Walla Walla County	123.0
36	Whatcom County	521.0
37	Whitman County	11.0
38	Yakima County	457.0

	Households with no Minors	Persons in Households with Minors \
0	0.0	0.0
1	13.0	11.0
2	50.0	81.0
3	215.0	92.0
4	147.0	46.0
5	491.0	372.0
6	11.0	11.0
7	223.0	81.0
8	12.0	11.0
9	11.0	0.0
10	44.0	11.0

11	11.0	11.0
12	97.0	75.0
13	91.0	15.0
14	94.0	24.0
15	112.0	20.0
16	7222.0	3743.0
17	366.0	133.0
18	11.0	11.0
19	27.0	11.0
20	89.0	45.0
21	0.0	0.0
22	86.0	83.0
23	49.0	11.0
24	44.0	11.0
25	11.0	29.0
26	1445.0	358.0
27	55.0	11.0
28	162.0	130.0
29	35.0	11.0
30	776.0	284.0
31	1118.0	363.0
32	33.0	11.0
33	645.0	310.0
34	11.0	0.0
35	122.0	11.0
36	496.0	165.0
37	11.0	14.0
38	442.0	176.0

	Households with Minors	Persons in Households with only Minors \
0	0.0	0.0
1	11.0	0.0
2	23.0	11.0
3	30.0	16.0
4	16.0	11.0
5	120.0	11.0
6	11.0	0.0
7	28.0	11.0
8	11.0	0.0
9	0.0	0.0
10	11.0	11.0
11	11.0	0.0
12	19.0	11.0
13	11.0	11.0
14	11.0	0.0
15	11.0	0.0
16	1190.0	301.0



17	42.0	11.0
18	11.0	11.0
19	11.0	11.0
20	16.0	0.0
21	0.0	0.0
22	25.0	11.0
23	11.0	11.0
24	11.0	11.0
25	11.0	11.0
26	113.0	12.0
27	11.0	0.0
28	36.0	11.0
29	11.0	0.0
30	92.0	30.0
31	104.0	25.0
32	11.0	0.0
33	95.0	13.0
34	0.0	0.0
35	11.0	11.0
36	55.0	11.0
37	11.0	11.0
38	49.0	0.0

	Households with only Minors	Total Persons	Total Households
0	0.0	0.0	0.0
1	0.0	24.0	24.0
2	11.0	142.0	84.0
3	11.0	337.0	256.0
4	11.0	208.0	174.0
5	11.0	919.0	622.0
6	0.0	22.0	22.0
7	11.0	336.0	262.0
8	0.0	23.0	23.0
9	0.0	11.0	11.0
10	11.0	66.0	66.0
11	0.0	22.0	22.0
12	11.0	190.0	127.0
13	0.0	118.0	102.0
14	0.0	129.0	105.0
15	0.0	139.0	123.0
16	210.0	11751.0	8622.0
17	11.0	534.0	419.0
18	11.0	33.0	33.0
19	11.0	50.0	49.0
20	0.0	142.0	105.0
21	0.0	0.0	0.0
22	11.0	184.0	122.0

23	11.0	77.0	71.0
24	11.0	70.0	66.0
25	11.0	51.0	33.0
26	12.0	1897.0	1570.0
27	0.0	66.0	66.0
28	0.0	322.0	198.0
29	0.0	47.0	46.0
30	29.0	1132.0	897.0
31	22.0	1559.0	1244.0
32	0.0	46.0	44.0
33	0.0	995.0	740.0
34	0.0	11.0	11.0
35	11.0	145.0	144.0
36	11.0	697.0	562.0
37	0.0	36.0	22.0
38	0.0	633.0	491.0

```
[17]: homeless2022 = homeless2022.replace('< 11', 11).replace('*', 0)
```

```
[18]: homeless2022_column_names = [
    'County',
    'Persons of Household with no Minor',
    'Households with no Minors',
    'Persons in Households with Minors',
    'Households with Minors',
    'Persons in Households with only Minors',
    'Households with only Minors',
    'Total Persons',
    'Total Households'
]
homeless2022.columns = homeless2022_column_names
new_num_column_names = [
    'Persons of Household with no Minor',
    'Households with no Minors',
    'Persons in Households with Minors',
    'Households with Minors',
    'Persons in Households with only Minors',
    'Households with only Minors',
    'Total Persons',
    'Total Households'
]

num_col_values = []

for col in new_num_column_names:
    num_col_values.extend(homeless2022[col].loc[homeless2022[col].apply(lambda x:
    isinstance(x, str))].tolist())
```

```
homeless2022[col] = pd.to_numeric(homeless2022[col]).astype(float)
homeless2022.dtypes
```

```
[18]: County                                object
Persons of Household with no Minor        float64
Households with no Minors                 float64
Persons in Households with Minors         float64
Households with Minors                   float64
Persons in Households with only Minors    float64
Households with only Minors              float64
Total Persons                           float64
Total Households                       float64
dtype: object
```

```
[19]: homeless_2022 = homeless2022.drop(index=39)
```

```
[20]: homeless2022['Sum of people'] = homeless2022['Persons of Household with no
↳Minor'] + homeless2022['Persons in Households with Minors'] +
↳homeless2022['Persons in Households with only Minors']
homeless2022['Total Persons'] = homeless2022['Sum of people']
homeless2022['Sum of Households'] = homeless2022['Households with no Minors'] +
↳homeless2022['Households with Minors'] + homeless2022['Households with only
↳Minors']
homeless2022['Total Households'] = homeless2022['Sum of Households']
column_drop = ['Sum of people', 'Sum of Households']
homeless2022 = homeless2022.drop(columns = column_drop)
homeless2022 = homeless2022.drop(index=39)
homeless2022
```

```
[20]:
```

	County	Persons of Household with no Minor \
0	Adams County	0.0
1	Asotin County	71.0
2	Benton County	78.0
3	Chelan County	246.0
4	Clallam County	118.0
5	Clark County	785.0
6	Columbia County	17.0
7	Cowlitz County	174.0
8	Douglas County	11.0
9	Ferry County	11.0
10	Franklin County	13.0
11	Garfield County	0.0
12	Grant County	223.0
13	Grays Harbor County	120.0
14	Island County	110.0
15	Jefferson County	112.0
16	King County	9327.0

17	Kitsap County	444.0
18	Kittitas County	33.0
19	Klickitat County	11.0
20	Lewis County	102.0
21	Lincoln County	0.0
22	Mason County	135.0
23	Okanogan County	41.0
24	Pacific County	108.0
25	Pend Oreille County	11.0
26	Pierce County	1516.0
27	San Juan County	50.0
28	Skagit County	216.0
29	Skamania County	24.0
30	Snohomish County	882.0
31	Spokane County	1457.0
32	Stevens County	31.0
33	Thurston County	484.0
34	Wahkiakum County	11.0
35	Walla Walla County	133.0
36	Whatcom County	573.0
37	Whitman County	11.0
38	Yakima County	529.0

	Households with no Minors	Persons in Households with Minors \
0	0.0	0.0
1	63.0	24.0
2	69.0	111.0
3	226.0	143.0
4	116.0	60.0
5	695.0	632.0
6	17.0	0.0
7	167.0	97.0
8	11.0	11.0
9	11.0	11.0
10	13.0	0.0
11	0.0	0.0
12	217.0	63.0
13	115.0	11.0
14	102.0	35.0
15	104.0	18.0
16	9324.0	3592.0
17	415.0	111.0
18	30.0	23.0
19	11.0	11.0
20	100.0	18.0
21	0.0	0.0
22	132.0	103.0

23	39.0	16.0
24	91.0	11.0
25	11.0	11.0
26	1443.0	331.0
27	41.0	11.0
28	203.0	98.0
29	24.0	0.0
30	846.0	287.0
31	1413.0	290.0
32	27.0	19.0
33	471.0	170.0
34	11.0	0.0
35	132.0	14.0
36	553.0	258.0
37	11.0	11.0
38	524.0	163.0

	Households with Minors	Persons in Households with only Minors \
0	0.0	0.0
1	11.0	0.0
2	32.0	13.0
3	41.0	0.0
4	20.0	0.0
5	198.0	21.0
6	0.0	0.0
7	30.0	0.0
8	11.0	0.0
9	11.0	0.0
10	0.0	0.0
11	0.0	0.0
12	21.0	0.0
13	11.0	11.0
14	12.0	11.0
15	11.0	0.0
16	1121.0	449.0
17	37.0	11.0
18	11.0	0.0
19	11.0	0.0
20	11.0	0.0
21	0.0	0.0
22	35.0	0.0
23	11.0	0.0
24	11.0	0.0
25	11.0	0.0
26	106.0	11.0
27	11.0	0.0
28	31.0	0.0

29	0.0	0.0
30	92.0	15.0
31	90.0	11.0
32	11.0	0.0
33	61.0	11.0
34	0.0	0.0
35	11.0	11.0
36	83.0	11.0
37	11.0	0.0
38	50.0	0.0

	Households with only Minors	Total Persons	Total Households
0	0.0	0.0	0.0
1	0.0	95.0	74.0
2	13.0	202.0	114.0
3	0.0	389.0	267.0
4	0.0	178.0	136.0
5	21.0	1438.0	914.0
6	0.0	17.0	17.0
7	0.0	271.0	197.0
8	0.0	22.0	22.0
9	0.0	22.0	22.0
10	0.0	13.0	13.0
11	0.0	0.0	0.0
12	0.0	286.0	238.0
13	11.0	142.0	137.0
14	11.0	156.0	125.0
15	0.0	130.0	115.0
16	449.0	13368.0	10894.0
17	11.0	566.0	463.0
18	0.0	56.0	41.0
19	0.0	22.0	22.0
20	0.0	120.0	111.0
21	0.0	0.0	0.0
22	0.0	238.0	167.0
23	0.0	57.0	50.0
24	0.0	119.0	102.0
25	0.0	22.0	22.0
26	11.0	1858.0	1560.0
27	0.0	61.0	52.0
28	0.0	314.0	234.0
29	0.0	24.0	24.0
30	15.0	1184.0	953.0
31	11.0	1758.0	1514.0
32	0.0	50.0	38.0
33	11.0	665.0	543.0
34	0.0	11.0	11.0

35	11.0	158.0	154.0
36	11.0	842.0	647.0
37	0.0	22.0	22.0
38	0.0	692.0	574.0

### 3 Zillow Rent Data (By Time)

This is the visualization of `zillow_rent_price.csv`. We melted to the dataframe to represent a single observation of rent per specific region and a specific date. This melting process converts the dataset from one column per time period to having one row per time period, with rent values aligned alongside the correct dates and region.

```
[22]: zillow_rent.head()
```

```
[22]:
```

	RegionID	SizeRank	RegionName	RegionType	StateName	State	\
0	3101	0	Los Angeles County	county	CA	CA	
1	139	1	Cook County	county	IL	IL	
2	1090	2	Harris County	county	TX	TX	
3	2402	3	Maricopa County	county	AZ	AZ	
4	2841	4	San Diego County	county	CA	CA	

		Metro	StateCodeFIPS	MunicipalCodeFIPS	\
0	Los Angeles-Long Beach-Anaheim, CA		6		37
1	Chicago-Naperville-Elgin, IL-IN-WI		17		31
2	Houston-The Woodlands-Sugar Land, TX		48		201
3	Phoenix-Mesa-Chandler, AZ		4		13
4	San Diego-Chula Vista-Carlsbad, CA		6		73

	1/31/2015	...	4/30/2023	5/31/2023	6/30/2023	7/31/2023	\
0	1776.567465	...	2771.995961	2777.228511	2785.536906	2792.929868	
1	1445.120815	...	1925.106443	1950.088286	1970.440023	1982.117579	
2	1206.425013	...	1589.906146	1599.097788	1608.366280	1615.989129	
3	955.578570	...	1848.342327	1856.980671	1857.442299	1861.053096	
4	1700.015970	...	2943.663855	2982.325174	3008.274888	3027.181872	

	8/31/2023	9/30/2023	10/31/2023	11/30/2023	12/31/2023	\
0	2801.705622	2803.596024	2800.984907	2790.870323	2783.335135	
1	1987.074699	1981.658545	1969.576929	1958.412818	1960.957242	
2	1617.336094	1615.528413	1610.872183	1606.715132	1606.000454	
3	1860.631889	1858.492465	1853.904365	1844.183060	1836.573508	
4	3035.209208	3033.045781	3019.007427	2995.634403	2975.039500	

	1/31/2024
0	2785.102892
1	1973.616367
2	1608.716707
3	1833.607640

4 2963.970587

[5 rows x 118 columns]

```
[23]: wa_zillow = zillow_rent[zillow_rent['State'] == 'WA']
time_measure = wa_zillow.columns[9:]
wa_zillow_melted = pd.melt(wa_zillow, id_vars=['RegionName'],
                           value_vars=time_measure, var_name='Date',
                           value_name='Rent')
wa_zillow_melted['Date'] = pd.to_datetime(wa_zillow_melted['Date'], format='%m/%d/%Y')
```

```
[24]: wa_zillow_melted['Year'] = wa_zillow_melted['Date'].dt.year
yearly_avg_rent = wa_zillow_melted.groupby(['RegionName', 'Year'])['Rent'].
    .mean().reset_index()
yearly_avg_rent['Pct_Change'] = yearly_avg_rent.groupby('RegionName')['Rent'].
    .pct_change() * 100
yearly_avg_rent = yearly_avg_rent.dropna()
max_increase_per_county = yearly_avg_rent.groupby('RegionName')['Pct_Change'].
    .max().reset_index()
max_increase_per_county_sorted = max_increase_per_county.
    .sort_values(by='Pct_Change', ascending=False)
```

/tmp/ipykernel\_839/3480381424.py:3: FutureWarning: The default fill\_method='ffill' in SeriesGroupBy.pct\_change is deprecated and will be removed in a future version. Either fill in any non-leading NA values prior to calling pct\_change or specify 'fill\_method=None' to not fill NA values.

```
yearly_avg_rent['Pct_Change'] =
yearly_avg_rent.groupby('RegionName')['Rent'].pct_change() * 100
```

```
[25]: wa_zillow_melted
```

```
[25]:
```

	RegionName	Date	Rent	Year
0	King County	2015-01-31	1487.104228	2015
1	Pierce County	2015-01-31	1027.814040	2015
2	Snohomish County	2015-01-31	1239.144701	2015
3	Spokane County	2015-01-31	749.701925	2015
4	Clark County	2015-01-31	1073.868408	2015
...	...	...	...	...
2720	Walla Walla County	2024-01-31	1561.388889	2024
2721	Whitman County	2024-01-31	1417.639205	2024
2722	Kittitas County	2024-01-31	1452.555556	2024
2723	Douglas County	2024-01-31	1842.500000	2024
2724	Jefferson County	2024-01-31	1993.611111	2024

[2725 rows x 4 columns]



## 4 Chart displaying rent trends across Washington counties

dataset used: zillow\_rent\_price.csv

```
[26]: rent_trend = px.line(wa_zillow_melted, x='Date', y='Rent', color='RegionName',  
    ↪title='Monthly Rent Trends by County in Washington State')  
  
rent_trend.update_layout(  
    xaxis_title='Date/Year',  
    yaxis_title='Average Rent($)',  
    legend_title='County',  
    width=1000, # Set the width as desired  
    height=600, # Set the height as desired  
    # positioning of legend below the plot  
    legend=dict(  
        orientation="h", # Horizontal orientation  
        yanchor="bottom",  
        y=-0.50,  
        xanchor="center",  
        x=0.5  
    ),  
)  
  
rent_trend.show()
```

## 5 Bar plot showing percent change in rent

dataset used: zillow\_rent\_price.csv

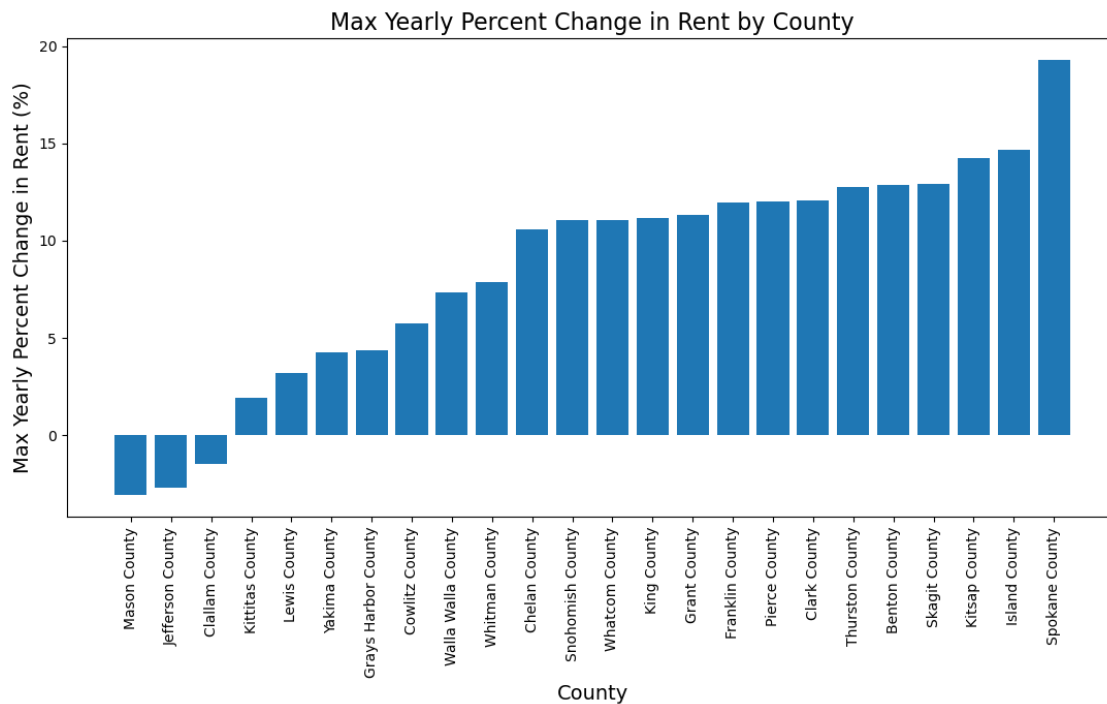
```
[27]: wa_zillow_melted['Year'] = wa_zillow_melted['Date'].dt.year  
  
yearly_avg_rent = wa_zillow_melted.groupby(['RegionName', 'Year'])['Rent'].  
    ↪mean().reset_index()  
yearly_avg_rent['Pct_Change'] = yearly_avg_rent.groupby('RegionName')['Rent'].  
    ↪pct_change() * 100  
  
yearly_avg_rent = yearly_avg_rent.dropna()  
  
max_increase_per_county = yearly_avg_rent.groupby('RegionName')['Pct_Change'].  
    ↪max().reset_index()  
max_increase_per_county_sorted = max_increase_per_county.  
    ↪sort_values(by='Pct_Change')  
  
plt.figure(figsize=(11, 7))  
plt.bar(max_increase_per_county_sorted['RegionName'],  
    ↪max_increase_per_county_sorted['Pct_Change'])  
plt.xticks(rotation=90)
```

```
plt.xlabel('County', fontsize=14)
plt.ylabel('Max Yearly Percent Change in Rent (%)', fontsize=14)
plt.title('Max Yearly Percent Change in Rent by County', fontsize=16)
plt.tight_layout()

plt.show()
```

/tmp/ipykernel\_839/892006287.py:4: FutureWarning:

The default `fill_method='ffill'` in `SeriesGroupBy.pct_change` is deprecated and will be removed in a future version. Either fill in any non-leading NA values prior to calling `pct_change` or specify `'fill_method=None'` to not fill NA values.



## 6 Merging rent\_income\_census & GeoDataFrame

This section performs merging operations from a shapefile and merging by “County” to provide geometry so we can map out our data. We got the WA shapefile from U.S. Census which provides geometry coordinates and county names, we then merged with county names to achieve geometry.

```
[28]: counties_layer = gpd.read_file('datasets/WA_County_Boundaries')
```

```
[29]: counties_layer.head()
```

```
[29]: OBJECTID JURISDICT_ JURISDIC_1 JURISDIC_2 JURISDIC_3 \
0 52002 25 4 Grant Grant County
1 56163 33 4 Garfield Garfield County
2 56525 8 4 Island Island County
3 57153 4699350 4 Kittitas Kittitas County
4 57197 35 4 Walla Walla Walla Walla County
```

```
JURISDIC_4 JURISDIC_5 JURISDIC_6 EDIT_DATE EDIT_STATU EDIT_WHO \
0 13 53025 None 2018-03-15 1 TSTE490
1 12 53023 None 2022-06-23 1 TSTE490
2 15 53029 None 2018-03-15 1 TSTE490
3 19 53037 None 2023-07-27 0 TSTE490
4 36 53071 None 2015-10-14 1 JDUG490
```

```
GLOBALID \
0 {294E70C7-27E7-4699-BA2C-93E5AD37A747}
1 {9B3E2F19-8942-4680-B112-3DFEC7DCB14C}
2 {D9DD1789-7BAC-41F9-B5A3-29E2287189D6}
3 {601FAD56-613F-488B-B85E-28F3BD212CFD}
4 {E6C7AF2E-BBDB-41DD-90A0-C4E49634461D}
```

```
geometry
0 POLYGON ((-13245041.204 6100462.041, -13245049...
1 POLYGON ((-13077215.155 5893282.479, -13076922...
2 POLYGON ((-13645903.473 6175425.382, -13645833...
3 POLYGON ((-13482428.890 6040101.397, -13482417...
4 POLYGON ((-13170470.944 5878093.595, -13170220...
```

```
[30]: county_col_drop = ['EDIT_DATE', 'JURISDIC_2', 'GLOBALID', 'EDIT_STATU',
↳ 'EDIT_WHO', 'JURISDIC_1', 'JURISDIC_6', 'OBJECTID', 'JURISDICT_', 'JURISDIC_4']
counties_layer = counties_layer.drop(columns=county_col_drop)
```

```
[31]: counties_layer.columns.values[0] = 'County'
counties_layer.columns.values[1] = 'FIPS'
```

```
[32]: counties_layer.head()
```

```
[32]: County FIPS \
0 Grant County 53025
1 Garfield County 53023
2 Island County 53029
3 Kittitas County 53037
4 Walla Walla County 53071
```

```
geometry
0 POLYGON ((-13245041.204 6100462.041, -13245049...
1 POLYGON ((-13077215.155 5893282.479, -13076922...
```

```

2 POLYGON ((-13645903.473 6175425.382, -13645833...
3 POLYGON ((-13482428.890 6040101.397, -13482417...
4 POLYGON ((-13170470.944 5878093.595, -13170220...

```

```

[33]: print(counties_layer.columns)
      print(rent_income_census.columns)
      counties_layer.columns = counties_layer.columns.str.strip()
      rent_income_census.columns = rent_income_census.columns.str.strip()

```

```

Index(['County', 'FIPS', 'geometry'], dtype='object')
Index(['County', 'Total', 'Less than 10.0 percent', '10.0 to 14.9 percent',
      '15.0 to 19.9 percent', '20.0 to 24.9 percent', '25.0 to 29.9 percent',
      '30.0 to 34.9 percent', '35.0 to 39.9 percent', '40.0 to 49.9 percent',
      '50.0 percent or more', 'Not computed'],
      dtype='object')

```

```

[34]: merged_rent_data = counties_layer.merge(rent_income_census, on='County',
      ↪how='left')

```

```

[35]: merged_rent_data

```

```

[35]:
      County  FIPS \
0      Grant County  53025
1    Garfield County  53023
2      Island County  53029
3    Kittitas County  53037
4  Walla Walla County  53071
5    Columbia County  53013
6    San Juan County  53055
7      Asotin County  53003
8    Franklin County  53021
9  Pend Oreille County  53051
10   Clallam County  53009
11   Pacific County  53049
12   Stevens County  53065
13   Douglas County  53017
14      King County  53033
15  Snohomish County  53061
16      Mason County  53045
17      Adams County  53001
18      Lewis County  53041
19   Lincoln County  53043
20   Spokane County  53063
21     Chelan County  53007
22  Wahkiakum County  53069
23    Whitman County  53075
24    Cowlitz County  53015

```

25	Thurston County	53067
26	Kitsap County	53035
27	Whatcom County	53073
28	Skagit County	53057
29	Grays Harbor County	53027
30	Skamania County	53059
31	Clark County	53011
32	Jefferson County	53031
33	Klickitat County	53039
34	Okanogan County	53047
35	Ferry County	53019
36	Yakima County	53077
37	Benton County	53005
38	Pierce County	53053

		geometry	Total \
0	POLYGON	((-13245041.204 6100462.041, -13245049...	11861.0
1	POLYGON	((-13077215.155 5893282.479, -13076922...	227.0
2	POLYGON	((-13645903.473 6175425.382, -13645833...	9291.0
3	POLYGON	((-13482428.890 6040101.397, -13482417...	7402.0
4	POLYGON	((-13170470.944 5878093.595, -13170220...	7778.0
5	POLYGON	((-13119277.932 5875313.561, -13119349...	450.0
6	POLYGON	((-13678738.782 6239384.542, -13678710...	1945.0
7	POLYGON	((-13046628.720 5847940.517, -13046278...	2640.0
8	POLYGON	((-13159766.598 5899385.299, -13159880...	8526.0
9	POLYGON	((-13080281.031 6138101.060, -13080277...	1374.0
10	POLYGON	((-13762933.712 6146044.720, -13752545...	9234.0
11	POLYGON	((-13794176.032 5908293.996, -13792142...	1910.0
12	POLYGON	((-13080281.031 6138101.060, -13080284...	3745.0
13	POLYGON	((-13257560.068 6129730.973, -13257263...	4654.0
14	POLYGON	((-13593254.964 6069657.588, -13592053...	401313.0
15	POLYGON	((-13601415.957 6156455.175, -13601292...	96712.0
16	POLYGON	((-13736648.080 6041327.062, -13734265...	5305.0
17	POLYGON	((-13131350.696 5984709.557, -13131339...	2212.0
18	POLYGON	((-13726667.316 5908398.973, -13724294...	8525.0
19	POLYGON	((-13228357.287 6097273.339, -13228274...	1004.0
20	POLYGON	((-13029073.626 6071491.429, -13029068...	77399.0
21	POLYGON	((-13451695.289 6198988.419, -13451524...	11116.0
22	POLYGON	((-13736747.915 5842104.061, -13735718...	288.0
23	POLYGON	((-13028819.631 5984496.998, -13028817...	9840.0
24	POLYGON	((-13702664.169 5842750.924, -13700419...	14503.0
25	POLYGON	((-13668140.191 5968532.266, -13667828...	37865.0
26	POLYGON	((-13632070.339 6053935.446, -13632151...	31981.0
27	POLYGON	((-13661948.246 6275217.332, -13660730...	33729.0
28	POLYGON	((-13646976.232 6211587.326, -13646829...	15078.0
29	POLYGON	((-13818358.748 6029600.620, -13817157...	8497.0
30	POLYGON	((-13593638.701 5842796.230, -13591537...	903.0

```

31 POLYGON ((-13609304.472 5789347.279, -13609297... 62839.0
32 POLYGON ((-13653405.352 6126390.566, -13653235... 3112.0
33 POLYGON ((-13527802.842 5787404.070, -13526218... 2405.0
34 POLYGON ((-13228831.248 6274874.740, -13228837... 5138.0
35 POLYGON ((-13157667.072 6274864.610, -13157440... 721.0
36 POLYGON ((-13500270.861 5956159.912, -13500209... 32234.0
37 POLYGON ((-13292341.455 5883579.005, -13292340... 24125.0
38 POLYGON ((-13651653.941 6008143.837, -13651640... 121139.0

```

	Less than 10.0 percent	10.0 to 14.9 percent	15.0 to 19.9 percent \
0	931.0	1199.0	1750.0
1	9.0	41.0	10.0
2	238.0	842.0	814.0
3	272.0	420.0	1051.0
4	248.0	440.0	661.0
5	54.0	21.0	108.0
6	75.0	174.0	192.0
7	172.0	230.0	332.0
8	521.0	902.0	903.0
9	168.0	57.0	134.0
10	280.0	780.0	1328.0
11	40.0	163.0	175.0
12	330.0	330.0	436.0
13	342.0	384.0	699.0
14	15460.0	36924.0	53568.0
15	2276.0	6252.0	11127.0
16	288.0	290.0	551.0
17	69.0	591.0	158.0
18	356.0	648.0	1017.0
19	96.0	132.0	105.0
20	3078.0	5972.0	8810.0
21	691.0	1151.0	1007.0
22	0.0	30.0	0.0
23	292.0	839.0	865.0
24	408.0	1067.0	1720.0
25	1056.0	2439.0	4385.0
26	1141.0	2270.0	3796.0
27	851.0	1890.0	3940.0
28	447.0	1304.0	1665.0
29	329.0	752.0	1148.0
30	70.0	64.0	101.0
31	1654.0	4864.0	7546.0
32	64.0	316.0	565.0
33	147.0	78.0	201.0
34	507.0	759.0	496.0
35	77.0	54.0	87.0
36	1149.0	3266.0	4132.0

37	937.0	2530.0	2841.0
38	3077.0	7639.0	14063.0

	20.0 to 24.9 percent	25.0 to 29.9 percent	30.0 to 34.9 percent \
0	1324.0	1098.0	1079.0
1	16.0	10.0	30.0
2	1350.0	918.0	1135.0
3	823.0	592.0	430.0
4	1007.0	922.0	533.0
5	11.0	55.0	37.0
6	213.0	158.0	140.0
7	234.0	387.0	273.0
8	951.0	862.0	835.0
9	102.0	68.0	101.0
10	897.0	847.0	670.0
11	242.0	199.0	92.0
12	362.0	324.0	313.0
13	553.0	452.0	292.0
14	54386.0	48377.0	35643.0
15	12912.0	12365.0	10858.0
16	730.0	527.0	251.0
17	130.0	226.0	148.0
18	1051.0	876.0	875.0
19	148.0	94.0	44.0
20	9362.0	9072.0	7494.0
21	1634.0	1498.0	722.0
22	2.0	45.0	8.0
23	795.0	612.0	386.0
24	1781.0	1906.0	1396.0
25	4806.0	4120.0	4239.0
26	3782.0	3925.0	3489.0
27	3971.0	3493.0	2595.0
28	1483.0	1709.0	1765.0
29	1073.0	830.0	560.0
30	85.0	101.0	57.0
31	7752.0	8050.0	5697.0
32	258.0	318.0	259.0
33	153.0	245.0	404.0
34	494.0	397.0	430.0
35	41.0	33.0	44.0
36	3835.0	3894.0	2604.0
37	3165.0	2663.0	2156.0
38	16708.0	13715.0	11967.0

	35.0 to 39.9 percent	40.0 to 49.9 percent	50.0 percent or more \
0	662.0	639.0	1855.0
1	0.0	10.0	39.0

2	647.0	825.0	1885.0
3	428.0	475.0	2284.0
4	715.0	810.0	1744.0
5	11.0	15.0	70.0
6	102.0	127.0	422.0
7	195.0	179.0	517.0
8	584.0	730.0	1533.0
9	89.0	124.0	198.0
10	482.0	1185.0	1866.0
11	169.0	195.0	385.0
12	356.0	171.0	540.0
13	278.0	95.0	854.0
14	26496.0	32446.0	83511.0
15	6655.0	8340.0	22115.0
16	220.0	235.0	1360.0
17	102.0	161.0	420.0
18	390.0	781.0	1745.0
19	16.0	42.0	80.0
20	5164.0	7388.0	17766.0
21	442.0	1014.0	1891.0
22	22.0	39.0	116.0
23	586.0	832.0	3785.0
24	1139.0	1293.0	3091.0
25	2255.0	2891.0	10028.0
26	2454.0	2653.0	6947.0
27	2808.0	3335.0	9355.0
28	824.0	1404.0	3320.0
29	543.0	900.0	1521.0
30	33.0	72.0	186.0
31	4568.0	6091.0	13274.0
32	203.0	194.0	431.0
33	42.0	265.0	385.0
34	180.0	368.0	698.0
35	47.0	66.0	101.0
36	2004.0	1976.0	6644.0
37	1449.0	2502.0	4517.0
38	9113.0	12071.0	26999.0

Not computed

0	1324.0
1	62.0
2	637.0
3	627.0
4	698.0
5	68.0
6	342.0
7	121.0



```

8          705.0
9          333.0
10         899.0
11         250.0
12         583.0
13         705.0
14        14502.0
15        3812.0
16         853.0
17         207.0
18         786.0
19         247.0
20        3293.0
21        1066.0
22          26.0
23        848.0
24        702.0
25        1646.0
26        1524.0
27        1491.0
28        1157.0
29         841.0
30        134.0
31       3343.0
32         504.0
33         485.0
34         809.0
35         171.0
36       2730.0
37       1365.0
38      5787.0

```

```
[36]: county_pop.head(1)
```

```
[36]:
```

	name	state	county	city	population
0	Adams County	Washington	NaN	NaN	20557

```
[37]: pop_col_drop = ['county', 'city']
      county_pop = county_pop.drop(columns=pop_col_drop)
```

```
[38]: county_pop.head()
```

```
[38]:
```

	name	state	population
0	Adams County	Washington	20557
1	Asotin County	Washington	22370
2	Benton County	Washington	207560
3	Chelan County	Washington	79076

```
[39]: merged_rent_data = merged_rent_data.merge(county_pop, left_on='County',
        ↪right_on='name', how='left')
```

```
[40]: merged_rent_data['% of 50 percent'] = (merged_rent_data['50.0 percent or more']
        ↪ / merged_rent_data['Total']) * 100
merged_rent_data['% of 30 percent'] = (merged_rent_data['30.0 to 34.9 percent']
        ↪ / merged_rent_data['Total']) * 100
merged_rent_data['% of 40 percent'] = (merged_rent_data['40.0 to 49.9 percent']
        ↪ / merged_rent_data['Total']) * 100
```

```
[41]: merged_rent_data
```

```
[41]:
```

	County	FIPS	\
0	Grant County	53025	
1	Garfield County	53023	
2	Island County	53029	
3	Kittitas County	53037	
4	Walla Walla County	53071	
5	Columbia County	53013	
6	San Juan County	53055	
7	Asotin County	53003	
8	Franklin County	53021	
9	Pend Oreille County	53051	
10	Clallam County	53009	
11	Pacific County	53049	
12	Stevens County	53065	
13	Douglas County	53017	
14	King County	53033	
15	Snohomish County	53061	
16	Mason County	53045	
17	Adams County	53001	
18	Lewis County	53041	
19	Lincoln County	53043	
20	Spokane County	53063	
21	Chelan County	53007	
22	Wahkiakum County	53069	
23	Whitman County	53075	
24	Cowlitz County	53015	
25	Thurston County	53067	
26	Kitsap County	53035	
27	Whatcom County	53073	
28	Skagit County	53057	
29	Grays Harbor County	53027	
30	Skamania County	53059	
31	Clark County	53011	

32	Jefferson County	53031
33	Klickitat County	53039
34	Okanogan County	53047
35	Ferry County	53019
36	Yakima County	53077
37	Benton County	53005
38	Pierce County	53053

		geometry	Total \
0	POLYGON	((-13245041.204 6100462.041, -13245049...	11861.0
1	POLYGON	((-13077215.155 5893282.479, -13076922...	227.0
2	POLYGON	((-13645903.473 6175425.382, -13645833...	9291.0
3	POLYGON	((-13482428.890 6040101.397, -13482417...	7402.0
4	POLYGON	((-13170470.944 5878093.595, -13170220...	7778.0
5	POLYGON	((-13119277.932 5875313.561, -13119349...	450.0
6	POLYGON	((-13678738.782 6239384.542, -13678710...	1945.0
7	POLYGON	((-13046628.720 5847940.517, -13046278...	2640.0
8	POLYGON	((-13159766.598 5899385.299, -13159880...	8526.0
9	POLYGON	((-13080281.031 6138101.060, -13080277...	1374.0
10	POLYGON	((-13762933.712 6146044.720, -13752545...	9234.0
11	POLYGON	((-13794176.032 5908293.996, -13792142...	1910.0
12	POLYGON	((-13080281.031 6138101.060, -13080284...	3745.0
13	POLYGON	((-13257560.068 6129730.973, -13257263...	4654.0
14	POLYGON	((-13593254.964 6069657.588, -13592053...	401313.0
15	POLYGON	((-13601415.957 6156455.175, -13601292...	96712.0
16	POLYGON	((-13736648.080 6041327.062, -13734265...	5305.0
17	POLYGON	((-13131350.696 5984709.557, -13131339...	2212.0
18	POLYGON	((-13726667.316 5908398.973, -13724294...	8525.0
19	POLYGON	((-13228357.287 6097273.339, -13228274...	1004.0
20	POLYGON	((-13029073.626 6071491.429, -13029068...	77399.0
21	POLYGON	((-13451695.289 6198988.419, -13451524...	11116.0
22	POLYGON	((-13736747.915 5842104.061, -13735718...	288.0
23	POLYGON	((-13028819.631 5984496.998, -13028817...	9840.0
24	POLYGON	((-13702664.169 5842750.924, -13700419...	14503.0
25	POLYGON	((-13668140.191 5968532.266, -13667828...	37865.0
26	POLYGON	((-13632070.339 6053935.446, -13632151...	31981.0
27	POLYGON	((-13661948.246 6275217.332, -13660730...	33729.0
28	POLYGON	((-13646976.232 6211587.326, -13646829...	15078.0
29	POLYGON	((-13818358.748 6029600.620, -13817157...	8497.0
30	POLYGON	((-13593638.701 5842796.230, -13591537...	903.0
31	POLYGON	((-13609304.472 5789347.279, -13609297...	62839.0
32	POLYGON	((-13653405.352 6126390.566, -13653235...	3112.0
33	POLYGON	((-13527802.842 5787404.070, -13526218...	2405.0
34	POLYGON	((-13228831.248 6274874.740, -13228837...	5138.0
35	POLYGON	((-13157667.072 6274864.610, -13157440...	721.0
36	POLYGON	((-13500270.861 5956159.912, -13500209...	32234.0
37	POLYGON	((-13292341.455 5883579.005, -13292340...	24125.0

38 POLYGON ((-13651653.941 6008143.837, -13651640... 121139.0

	Less than 10.0 percent	10.0 to 14.9 percent	15.0 to 19.9 percent	\
0	931.0	1199.0	1750.0	
1	9.0	41.0	10.0	
2	238.0	842.0	814.0	
3	272.0	420.0	1051.0	
4	248.0	440.0	661.0	
5	54.0	21.0	108.0	
6	75.0	174.0	192.0	
7	172.0	230.0	332.0	
8	521.0	902.0	903.0	
9	168.0	57.0	134.0	
10	280.0	780.0	1328.0	
11	40.0	163.0	175.0	
12	330.0	330.0	436.0	
13	342.0	384.0	699.0	
14	15460.0	36924.0	53568.0	
15	2276.0	6252.0	11127.0	
16	288.0	290.0	551.0	
17	69.0	591.0	158.0	
18	356.0	648.0	1017.0	
19	96.0	132.0	105.0	
20	3078.0	5972.0	8810.0	
21	691.0	1151.0	1007.0	
22	0.0	30.0	0.0	
23	292.0	839.0	865.0	
24	408.0	1067.0	1720.0	
25	1056.0	2439.0	4385.0	
26	1141.0	2270.0	3796.0	
27	851.0	1890.0	3940.0	
28	447.0	1304.0	1665.0	
29	329.0	752.0	1148.0	
30	70.0	64.0	101.0	
31	1654.0	4864.0	7546.0	
32	64.0	316.0	565.0	
33	147.0	78.0	201.0	
34	507.0	759.0	496.0	
35	77.0	54.0	87.0	
36	1149.0	3266.0	4132.0	
37	937.0	2530.0	2841.0	
38	3077.0	7639.0	14063.0	
	20.0 to 24.9 percent	25.0 to 29.9 percent	30.0 to 34.9 percent	\
0	1324.0	1098.0	1079.0	
1	16.0	10.0	30.0	
2	1350.0	918.0	1135.0	

3	823.0	592.0	430.0
4	1007.0	922.0	533.0
5	11.0	55.0	37.0
6	213.0	158.0	140.0
7	234.0	387.0	273.0
8	951.0	862.0	835.0
9	102.0	68.0	101.0
10	897.0	847.0	670.0
11	242.0	199.0	92.0
12	362.0	324.0	313.0
13	553.0	452.0	292.0
14	54386.0	48377.0	35643.0
15	12912.0	12365.0	10858.0
16	730.0	527.0	251.0
17	130.0	226.0	148.0
18	1051.0	876.0	875.0
19	148.0	94.0	44.0
20	9362.0	9072.0	7494.0
21	1634.0	1498.0	722.0
22	2.0	45.0	8.0
23	795.0	612.0	386.0
24	1781.0	1906.0	1396.0
25	4806.0	4120.0	4239.0
26	3782.0	3925.0	3489.0
27	3971.0	3493.0	2595.0
28	1483.0	1709.0	1765.0
29	1073.0	830.0	560.0
30	85.0	101.0	57.0
31	7752.0	8050.0	5697.0
32	258.0	318.0	259.0
33	153.0	245.0	404.0
34	494.0	397.0	430.0
35	41.0	33.0	44.0
36	3835.0	3894.0	2604.0
37	3165.0	2663.0	2156.0
38	16708.0	13715.0	11967.0

	35.0 to 39.9 percent	40.0 to 49.9 percent	50.0 percent or more	\
0	662.0	639.0	1855.0	
1	0.0	10.0	39.0	
2	647.0	825.0	1885.0	
3	428.0	475.0	2284.0	
4	715.0	810.0	1744.0	
5	11.0	15.0	70.0	
6	102.0	127.0	422.0	
7	195.0	179.0	517.0	
8	584.0	730.0	1533.0	

9	89.0	124.0	198.0
10	482.0	1185.0	1866.0
11	169.0	195.0	385.0
12	356.0	171.0	540.0
13	278.0	95.0	854.0
14	26496.0	32446.0	83511.0
15	6655.0	8340.0	22115.0
16	220.0	235.0	1360.0
17	102.0	161.0	420.0
18	390.0	781.0	1745.0
19	16.0	42.0	80.0
20	5164.0	7388.0	17766.0
21	442.0	1014.0	1891.0
22	22.0	39.0	116.0
23	586.0	832.0	3785.0
24	1139.0	1293.0	3091.0
25	2255.0	2891.0	10028.0
26	2454.0	2653.0	6947.0
27	2808.0	3335.0	9355.0
28	824.0	1404.0	3320.0
29	543.0	900.0	1521.0
30	33.0	72.0	186.0
31	4568.0	6091.0	13274.0
32	203.0	194.0	431.0
33	42.0	265.0	385.0
34	180.0	368.0	698.0
35	47.0	66.0	101.0
36	2004.0	1976.0	6644.0
37	1449.0	2502.0	4517.0
38	9113.0	12071.0	26999.0

	Not computed	name	state	population \
0	1324.0	Grant County	Washington	99145
1	62.0	Garfield County	Washington	2310
2	637.0	Island County	Washington	86510
3	627.0	Kittitas County	Washington	44424
4	698.0	Walla Walla County	Washington	62150
5	68.0	Columbia County	Washington	3980
6	342.0	San Juan County	Washington	18001
7	121.0	Asotin County	Washington	22370
8	705.0	Franklin County	Washington	96692
9	333.0	Pend Oreille County	Washington	13570
10	899.0	Clallam County	Washington	77333
11	250.0	Pacific County	Washington	23396
12	583.0	Stevens County	Washington	46774
13	705.0	Douglas County	Washington	43189
14	14502.0	King County	Washington	2254371

15	3812.0	Snohomish County	Washington	828337
16	853.0	Mason County	Washington	66053
17	207.0	Adams County	Washington	20557
18	786.0	Lewis County	Washington	82663
19	247.0	Lincoln County	Washington	11036
20	3293.0	Spokane County	Washington	538711
21	1066.0	Chelan County	Washington	79076
22	26.0	Wahkiakum County	Washington	4476
23	848.0	Whitman County	Washington	47141
24	702.0	Cowlitz County	Washington	110621
25	1646.0	Thurston County	Washington	294272
26	1524.0	Kitsap County	Washington	275411
27	1491.0	Whatcom County	Washington	226523
28	1157.0	Skagit County	Washington	129480
29	841.0	Grays Harbor County	Washington	75672
30	134.0	Skamania County	Washington	12118
31	3343.0	Clark County	Washington	504091
32	504.0	Jefferson County	Washington	33006
33	485.0	Klickitat County	Washington	22798
34	809.0	Okanogan County	Washington	42336
35	171.0	Ferry County	Washington	7260
36	2730.0	Yakima County	Washington	256143
37	1365.0	Benton County	Washington	207560
38	5787.0	Pierce County	Washington	918993

	% of 50 percent	% of 30 percent	% of 40 percent
0	15.639491	9.097041	5.387404
1	17.180617	13.215859	4.405286
2	20.288451	12.216123	8.879561
3	30.856525	5.809241	6.417185
4	22.422217	6.852661	10.413988
5	15.555556	8.222222	3.333333
6	21.696658	7.197943	6.529563
7	19.583333	10.340909	6.780303
8	17.980296	9.793573	8.562046
9	14.410480	7.350801	9.024745
10	20.207927	7.255794	12.833008
11	20.157068	4.816754	10.209424
12	14.419226	8.357810	4.566088
13	18.349807	6.274173	2.041255
14	20.809443	8.881596	8.084961
15	22.866862	11.227149	8.623542
16	25.636192	4.731385	4.429783
17	18.987342	6.690778	7.278481
18	20.469208	10.263930	9.161290
19	7.968127	4.382470	4.183267
20	22.953785	9.682296	9.545343

21	17.011515	6.495142	9.121986
22	40.277778	2.777778	13.541667
23	38.465447	3.922764	8.455285
24	21.312832	9.625595	8.915397
25	26.483560	11.195035	7.635019
26	21.722273	10.909603	8.295550
27	27.735776	7.693676	9.887634
28	22.018835	11.705797	9.311580
29	17.900435	6.590561	10.591974
30	20.598007	6.312292	7.973422
31	21.123824	9.066026	9.693025
32	13.849614	8.322622	6.233933
33	16.008316	16.798337	11.018711
34	13.585053	8.369015	7.162320
35	14.008322	6.102635	9.153953
36	20.611776	8.078427	6.130173
37	18.723316	8.936788	10.370984
38	22.287620	9.878734	9.964586

### 6.0.1 Map displaying population density of households experience high burden of rent cost (Households who spend more than 50 percent of their income on rent)

dataset used: rent\_income.csv & population dataset

```
[42]: fig, ax = plt.subplots(figsize=(10, 8))

merged_rent_data.plot(ax=ax, column='% of 50 percent', cmap='Reds',
    ↪edgecolor='black', legend=True)

title_text = "Percentage of Households Spending More Than\n50 Percent of Income
    ↪on Rent By County"
ax.set_title(title_text, fontsize=16, fontweight='bold', loc='center')

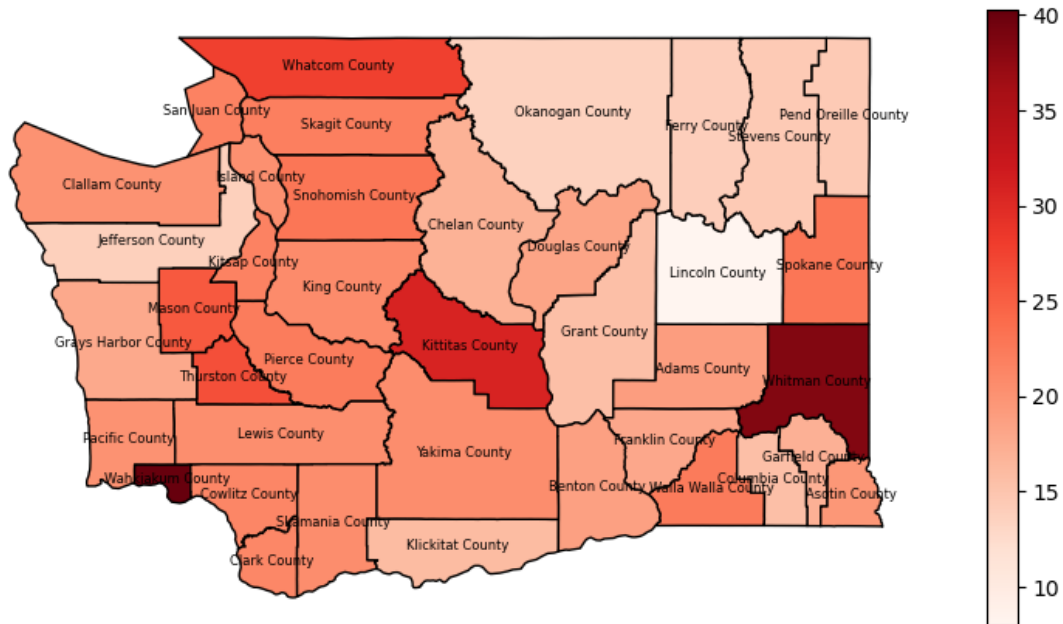
for idx, row in merged_rent_data.iterrows():
    centroid = row.geometry.centroid
    ax.text(centroid.x, centroid.y, row['County'], fontsize=6, ha='center',
    ↪va='center')

cbar = fig.axes[-1]
ax_pos = ax.get_position()
cbar.set_position([cbar.get_position().x0, ax_pos.y0, cbar.get_position().
    ↪width, ax_pos.height])
ax.set_axis_off()

plt.show()
```



## Percentage of Households Spending More Than 50 Percent of Income on Rent By County



Merging homelessness 2020 data with shapefile to map out those rates.

```
[43]: merged_homeless2020_data = counties_layer.merge(homeless2020, on='County',
→how='left')
```

```
[44]: merged_homeless2020_data
```

```
[44]:
```

	County	FIPS	\
0	Grant County	53025	
1	Garfield County	53023	
2	Island County	53029	
3	Kittitas County	53037	
4	Walla Walla County	53071	
5	Columbia County	53013	
6	San Juan County	53055	
7	Asotin County	53003	
8	Franklin County	53021	
9	Pend Oreille County	53051	
10	Clallam County	53009	
11	Pacific County	53049	
12	Stevens County	53065	
13	Douglas County	53017	
14	King County	53033	

15	Snohomish County	53061
16	Mason County	53045
17	Adams County	53001
18	Lewis County	53041
19	Lincoln County	53043
20	Spokane County	53063
21	Chelan County	53007
22	Wahkiakum County	53069
23	Whitman County	53075
24	Cowlitz County	53015
25	Thurston County	53067
26	Kitsap County	53035
27	Whatcom County	53073
28	Skagit County	53057
29	Grays Harbor County	53027
30	Skamania County	53059
31	Clark County	53011
32	Jefferson County	53031
33	Klickitat County	53039
34	Okanogan County	53047
35	Ferry County	53019
36	Yakima County	53077
37	Benton County	53005
38	Pierce County	53053

```

                                geometry \
0  POLYGON ((-13245041.204 6100462.041, -13245049...
1  POLYGON ((-13077215.155 5893282.479, -13076922...
2  POLYGON ((-13645903.473 6175425.382, -13645833...
3  POLYGON ((-13482428.890 6040101.397, -13482417...
4  POLYGON ((-13170470.944 5878093.595, -13170220...
5  POLYGON ((-13119277.932 5875313.561, -13119349...
6  POLYGON ((-13678738.782 6239384.542, -13678710...
7  POLYGON ((-13046628.720 5847940.517, -13046278...
8  POLYGON ((-13159766.598 5899385.299, -13159880...
9  POLYGON ((-13080281.031 6138101.060, -13080277...
10 POLYGON ((-13762933.712 6146044.720, -13752545...
11 POLYGON ((-13794176.032 5908293.996, -13792142...
12 POLYGON ((-13080281.031 6138101.060, -13080284...
13 POLYGON ((-13257560.068 6129730.973, -13257263...
14 POLYGON ((-13593254.964 6069657.588, -13592053...
15 POLYGON ((-13601415.957 6156455.175, -13601292...
16 POLYGON ((-13736648.080 6041327.062, -13734265...
17 POLYGON ((-13131350.696 5984709.557, -13131339...
18 POLYGON ((-13726667.316 5908398.973, -13724294...
19 POLYGON ((-13228357.287 6097273.339, -13228274...
20 POLYGON ((-13029073.626 6071491.429, -13029068...

```

21 POLYGON ((-13451695.289 6198988.419, -13451524...  
 22 POLYGON ((-13736747.915 5842104.061, -13735718...  
 23 POLYGON ((-13028819.631 5984496.998, -13028817...  
 24 POLYGON ((-13702664.169 5842750.924, -13700419...  
 25 POLYGON ((-13668140.191 5968532.266, -13667828...  
 26 POLYGON ((-13632070.339 6053935.446, -13632151...  
 27 POLYGON ((-13661948.246 6275217.332, -13660730...  
 28 POLYGON ((-13646976.232 6211587.326, -13646829...  
 29 POLYGON ((-13818358.748 6029600.620, -13817157...  
 30 POLYGON ((-13593638.701 5842796.230, -13591537...  
 31 POLYGON ((-13609304.472 5789347.279, -13609297...  
 32 POLYGON ((-13653405.352 6126390.566, -13653235...  
 33 POLYGON ((-13527802.842 5787404.070, -13526218...  
 34 POLYGON ((-13228831.248 6274874.740, -13228837...  
 35 POLYGON ((-13157667.072 6274864.610, -13157440...  
 36 POLYGON ((-13500270.861 5956159.912, -13500209...  
 37 POLYGON ((-13292341.455 5883579.005, -13292340...  
 38 POLYGON ((-13651653.941 6008143.837, -13651640...

	Persons of Household with no Minor	Households with no Minors \
0	104.0	97.0
1	11.0	11.0
2	105.0	94.0
3	11.0	11.0
4	123.0	122.0
5	11.0	11.0
6	55.0	55.0
7	13.0	13.0
8	44.0	44.0
9	11.0	11.0
10	151.0	147.0
11	48.0	44.0
12	35.0	33.0
13	12.0	12.0
14	7707.0	7222.0
15	818.0	776.0
16	90.0	86.0
17	0.0	0.0
18	97.0	89.0
19	0.0	0.0
20	1171.0	1118.0
21	229.0	215.0
22	11.0	11.0
23	11.0	11.0
24	244.0	223.0
25	672.0	645.0
26	390.0	366.0

27	521.0	496.0
28	181.0	162.0
29	92.0	91.0
30	36.0	35.0
31	536.0	491.0
32	119.0	112.0
33	28.0	27.0
34	55.0	49.0
35	11.0	11.0
36	457.0	442.0
37	50.0	50.0
38	1527.0	1445.0

	Persons in Households with Minors	Households with Minors \
0	75.0	19.0
1	11.0	11.0
2	24.0	11.0
3	11.0	11.0
4	11.0	11.0
5	11.0	11.0
6	11.0	11.0
7	11.0	11.0
8	11.0	11.0
9	29.0	11.0
10	46.0	16.0
11	11.0	11.0
12	11.0	11.0
13	11.0	11.0
14	3743.0	1190.0
15	284.0	92.0
16	83.0	25.0
17	0.0	0.0
18	45.0	16.0
19	0.0	0.0
20	363.0	104.0
21	92.0	30.0
22	0.0	0.0
23	14.0	11.0
24	81.0	28.0
25	310.0	95.0
26	133.0	42.0
27	165.0	55.0
28	130.0	36.0
29	15.0	11.0
30	11.0	11.0
31	372.0	120.0
32	20.0	11.0

33	11.0	11.0
34	11.0	11.0
35	0.0	0.0
36	176.0	49.0
37	81.0	23.0
38	358.0	113.0

	Persons in Households with only Minors	Households with only Minors \
0	11.0	11.0
1	0.0	0.0
2	0.0	0.0
3	11.0	11.0
4	11.0	11.0
5	0.0	0.0
6	0.0	0.0
7	0.0	0.0
8	11.0	11.0
9	11.0	11.0
10	11.0	11.0
11	11.0	11.0
12	0.0	0.0
13	0.0	0.0
14	301.0	210.0
15	30.0	29.0
16	11.0	11.0
17	0.0	0.0
18	0.0	0.0
19	0.0	0.0
20	25.0	22.0
21	16.0	11.0
22	0.0	0.0
23	11.0	0.0
24	11.0	11.0
25	13.0	0.0
26	11.0	11.0
27	11.0	11.0
28	11.0	0.0
29	11.0	0.0
30	0.0	0.0
31	11.0	11.0
32	0.0	0.0
33	11.0	11.0
34	11.0	11.0
35	0.0	0.0
36	0.0	0.0
37	11.0	11.0
38	12.0	12.0

	Total Persons	Total Households
0	190.0	127.0
1	22.0	22.0
2	129.0	105.0
3	33.0	33.0
4	145.0	144.0
5	22.0	22.0
6	66.0	66.0
7	24.0	24.0
8	66.0	66.0
9	51.0	33.0
10	208.0	174.0
11	70.0	66.0
12	46.0	44.0
13	23.0	23.0
14	11751.0	8622.0
15	1132.0	897.0
16	184.0	122.0
17	0.0	0.0
18	142.0	105.0
19	0.0	0.0
20	1559.0	1244.0
21	337.0	256.0
22	11.0	11.0
23	36.0	22.0
24	336.0	262.0
25	995.0	740.0
26	534.0	419.0
27	697.0	562.0
28	322.0	198.0
29	118.0	102.0
30	47.0	46.0
31	919.0	622.0
32	139.0	123.0
33	50.0	49.0
34	77.0	71.0
35	11.0	11.0
36	633.0	491.0
37	142.0	84.0
38	1897.0	1570.0

```
[45]: merged_homeless2020_data['Proportion of Persons Homeless'] =_
      ↪(merged_homeless2020_data['Total Persons'] / merged_rent_data['Total'])
```

```
[46]: # check current crs, as we may need to change it to convert the geometries to_
      ↪area per square mile
```

```
current_crs = merged_homeless2020_data.crs
print(current_crs)
```

EPSG:3857

```
[47]: merged_homeless2020_data
```

```
[47]:
```

	County	FIPS	\
0	Grant County	53025	
1	Garfield County	53023	
2	Island County	53029	
3	Kittitas County	53037	
4	Walla Walla County	53071	
5	Columbia County	53013	
6	San Juan County	53055	
7	Asotin County	53003	
8	Franklin County	53021	
9	Pend Oreille County	53051	
10	Clallam County	53009	
11	Pacific County	53049	
12	Stevens County	53065	
13	Douglas County	53017	
14	King County	53033	
15	Snohomish County	53061	
16	Mason County	53045	
17	Adams County	53001	
18	Lewis County	53041	
19	Lincoln County	53043	
20	Spokane County	53063	
21	Chelan County	53007	
22	Wahkiakum County	53069	
23	Whitman County	53075	
24	Cowlitz County	53015	
25	Thurston County	53067	
26	Kitsap County	53035	
27	Whatcom County	53073	
28	Skagit County	53057	
29	Grays Harbor County	53027	
30	Skamania County	53059	
31	Clark County	53011	
32	Jefferson County	53031	
33	Klickitat County	53039	
34	Okanogan County	53047	
35	Ferry County	53019	
36	Yakima County	53077	
37	Benton County	53005	
38	Pierce County	53053	

```

                                geometry \
0  POLYGON ((-13245041.204 6100462.041, -13245049...
1  POLYGON ((-13077215.155 5893282.479, -13076922...
2  POLYGON ((-13645903.473 6175425.382, -13645833...
3  POLYGON ((-13482428.890 6040101.397, -13482417...
4  POLYGON ((-13170470.944 5878093.595, -13170220...
5  POLYGON ((-13119277.932 5875313.561, -13119349...
6  POLYGON ((-13678738.782 6239384.542, -13678710...
7  POLYGON ((-13046628.720 5847940.517, -13046278...
8  POLYGON ((-13159766.598 5899385.299, -13159880...
9  POLYGON ((-13080281.031 6138101.060, -13080277...
10 POLYGON ((-13762933.712 6146044.720, -13752545...
11 POLYGON ((-13794176.032 5908293.996, -13792142...
12 POLYGON ((-13080281.031 6138101.060, -13080284...
13 POLYGON ((-13257560.068 6129730.973, -13257263...
14 POLYGON ((-13593254.964 6069657.588, -13592053...
15 POLYGON ((-13601415.957 6156455.175, -13601292...
16 POLYGON ((-13736648.080 6041327.062, -13734265...
17 POLYGON ((-13131350.696 5984709.557, -13131339...
18 POLYGON ((-13726667.316 5908398.973, -13724294...
19 POLYGON ((-13228357.287 6097273.339, -13228274...
20 POLYGON ((-13029073.626 6071491.429, -13029068...
21 POLYGON ((-13451695.289 6198988.419, -13451524...
22 POLYGON ((-13736747.915 5842104.061, -13735718...
23 POLYGON ((-13028819.631 5984496.998, -13028817...
24 POLYGON ((-13702664.169 5842750.924, -13700419...
25 POLYGON ((-13668140.191 5968532.266, -13667828...
26 POLYGON ((-13632070.339 6053935.446, -13632151...
27 POLYGON ((-13661948.246 6275217.332, -13660730...
28 POLYGON ((-13646976.232 6211587.326, -13646829...
29 POLYGON ((-13818358.748 6029600.620, -13817157...
30 POLYGON ((-13593638.701 5842796.230, -13591537...
31 POLYGON ((-13609304.472 5789347.279, -13609297...
32 POLYGON ((-13653405.352 6126390.566, -13653235...
33 POLYGON ((-13527802.842 5787404.070, -13526218...
34 POLYGON ((-13228831.248 6274874.740, -13228837...
35 POLYGON ((-13157667.072 6274864.610, -13157440...
36 POLYGON ((-13500270.861 5956159.912, -13500209...
37 POLYGON ((-13292341.455 5883579.005, -13292340...
38 POLYGON ((-13651653.941 6008143.837, -13651640...

```

```

Persons of Household with no Minor  Households with no Minors \
0                                     104.0                        97.0
1                                     11.0                        11.0
2                                     105.0                       94.0
3                                     11.0                        11.0

```



4	123.0	122.0
5	11.0	11.0
6	55.0	55.0
7	13.0	13.0
8	44.0	44.0
9	11.0	11.0
10	151.0	147.0
11	48.0	44.0
12	35.0	33.0
13	12.0	12.0
14	7707.0	7222.0
15	818.0	776.0
16	90.0	86.0
17	0.0	0.0
18	97.0	89.0
19	0.0	0.0
20	1171.0	1118.0
21	229.0	215.0
22	11.0	11.0
23	11.0	11.0
24	244.0	223.0
25	672.0	645.0
26	390.0	366.0
27	521.0	496.0
28	181.0	162.0
29	92.0	91.0
30	36.0	35.0
31	536.0	491.0
32	119.0	112.0
33	28.0	27.0
34	55.0	49.0
35	11.0	11.0
36	457.0	442.0
37	50.0	50.0
38	1527.0	1445.0

	Persons in Households with Minors	Households with Minors \
0	75.0	19.0
1	11.0	11.0
2	24.0	11.0
3	11.0	11.0
4	11.0	11.0
5	11.0	11.0
6	11.0	11.0
7	11.0	11.0
8	11.0	11.0
9	29.0	11.0

10	46.0	16.0
11	11.0	11.0
12	11.0	11.0
13	11.0	11.0
14	3743.0	1190.0
15	284.0	92.0
16	83.0	25.0
17	0.0	0.0
18	45.0	16.0
19	0.0	0.0
20	363.0	104.0
21	92.0	30.0
22	0.0	0.0
23	14.0	11.0
24	81.0	28.0
25	310.0	95.0
26	133.0	42.0
27	165.0	55.0
28	130.0	36.0
29	15.0	11.0
30	11.0	11.0
31	372.0	120.0
32	20.0	11.0
33	11.0	11.0
34	11.0	11.0
35	0.0	0.0
36	176.0	49.0
37	81.0	23.0
38	358.0	113.0

	Persons in Households with only Minors	Households with only Minors \
0	11.0	11.0
1	0.0	0.0
2	0.0	0.0
3	11.0	11.0
4	11.0	11.0
5	0.0	0.0
6	0.0	0.0
7	0.0	0.0
8	11.0	11.0
9	11.0	11.0
10	11.0	11.0
11	11.0	11.0
12	0.0	0.0
13	0.0	0.0
14	301.0	210.0
15	30.0	29.0

16	11.0	11.0
17	0.0	0.0
18	0.0	0.0
19	0.0	0.0
20	25.0	22.0
21	16.0	11.0
22	0.0	0.0
23	11.0	0.0
24	11.0	11.0
25	13.0	0.0
26	11.0	11.0
27	11.0	11.0
28	11.0	0.0
29	11.0	0.0
30	0.0	0.0
31	11.0	11.0
32	0.0	0.0
33	11.0	11.0
34	11.0	11.0
35	0.0	0.0
36	0.0	0.0
37	11.0	11.0
38	12.0	12.0

	Total Persons	Total Households	Proportion of Persons Homeless
0	190.0	127.0	0.016019
1	22.0	22.0	0.096916
2	129.0	105.0	0.013884
3	33.0	33.0	0.004458
4	145.0	144.0	0.018642
5	22.0	22.0	0.048889
6	66.0	66.0	0.033933
7	24.0	24.0	0.009091
8	66.0	66.0	0.007741
9	51.0	33.0	0.037118
10	208.0	174.0	0.022525
11	70.0	66.0	0.036649
12	46.0	44.0	0.012283
13	23.0	23.0	0.004942
14	11751.0	8622.0	0.029281
15	1132.0	897.0	0.011705
16	184.0	122.0	0.034684
17	0.0	0.0	0.000000
18	142.0	105.0	0.016657
19	0.0	0.0	0.000000
20	1559.0	1244.0	0.020142
21	337.0	256.0	0.030317

22	11.0	11.0	0.038194
23	36.0	22.0	0.003659
24	336.0	262.0	0.023168
25	995.0	740.0	0.026278
26	534.0	419.0	0.016697
27	697.0	562.0	0.020665
28	322.0	198.0	0.021356
29	118.0	102.0	0.013887
30	47.0	46.0	0.052049
31	919.0	622.0	0.014625
32	139.0	123.0	0.044666
33	50.0	49.0	0.020790
34	77.0	71.0	0.014986
35	11.0	11.0	0.015257
36	633.0	491.0	0.019638
37	142.0	84.0	0.005886
38	1897.0	1570.0	0.015660

```
[48]: merged_homeless2020_data = merged_homeless2020_data.to_crs(epsg=5070)
# the value was found online "2.58998811e+6", this gives us the square mile of
# each county.
merged_homeless2020_data['Area Square Mile'] =
    merged_homeless2020_data['geometry'].area / 2.58998811e+6

# create a new column calculating homeless density per county
merged_homeless2020_data['Homeless Density'] = merged_homeless2020_data['Total
    Persons'] / merged_homeless2020_data['Area Square Mile']
```

```
[49]: # change back to correct projection
merged_homeless2020_data = merged_homeless2020_data.to_crs(epsg=3857)
```

```
[50]: # assert statements to check if area square mile was correctly calculated,
# information on square mile obtained from Google search

correct_sq_mi_king_county = 2307
correct_grant_county = 2791

sq_mi_king_county = int(merged_homeless2020_data.
    loc[merged_homeless2020_data['County'] == 'King County', 'Area Square Mile'].
    iloc[0])
sq_mi_grant_county = int(merged_homeless2020_data.
    loc[merged_homeless2020_data['County'] == 'Grant County', 'Area Square
    Mile'].iloc[0])

assert sq_mi_king_county == correct_sq_mi_king_county, "No Match"
assert sq_mi_grant_county == correct_grant_county, "No Match"
```

```
[51]: merged_homeless2020_data
```

```
[51]:
```

	County	FIPS	\
0	Grant County	53025	
1	Garfield County	53023	
2	Island County	53029	
3	Kittitas County	53037	
4	Walla Walla County	53071	
5	Columbia County	53013	
6	San Juan County	53055	
7	Asotin County	53003	
8	Franklin County	53021	
9	Pend Oreille County	53051	
10	Clallam County	53009	
11	Pacific County	53049	
12	Stevens County	53065	
13	Douglas County	53017	
14	King County	53033	
15	Snohomish County	53061	
16	Mason County	53045	
17	Adams County	53001	
18	Lewis County	53041	
19	Lincoln County	53043	
20	Spokane County	53063	
21	Chelan County	53007	
22	Wahkiakum County	53069	
23	Whitman County	53075	
24	Cowlitz County	53015	
25	Thurston County	53067	
26	Kitsap County	53035	
27	Whatcom County	53073	
28	Skagit County	53057	
29	Grays Harbor County	53027	
30	Skamania County	53059	
31	Clark County	53011	
32	Jefferson County	53031	
33	Klickitat County	53039	
34	Okanogan County	53047	
35	Ferry County	53019	
36	Yakima County	53077	
37	Benton County	53005	
38	Pierce County	53053	

```
                                geometry \
```

0	POLYGON ((-13245041.204 6100462.041, -13245049...
1	POLYGON ((-13077215.155 5893282.479, -13076922...
2	POLYGON ((-13645903.473 6175425.382, -13645833...

```

3  POLYGON ((-13482428.890 6040101.397, -13482417...
4  POLYGON ((-13170470.944 5878093.595, -13170220...
5  POLYGON ((-13119277.932 5875313.561, -13119349...
6  POLYGON ((-13678738.782 6239384.542, -13678710...
7  POLYGON ((-13046628.720 5847940.517, -13046278...
8  POLYGON ((-13159766.598 5899385.299, -13159880...
9  POLYGON ((-13080281.031 6138101.060, -13080277...
10 POLYGON ((-13762933.712 6146044.720, -13752545...
11 POLYGON ((-13794176.032 5908293.996, -13792142...
12 POLYGON ((-13080281.031 6138101.060, -13080284...
13 POLYGON ((-13257560.068 6129730.973, -13257263...
14 POLYGON ((-13593254.964 6069657.588, -13592053...
15 POLYGON ((-13601415.957 6156455.175, -13601292...
16 POLYGON ((-13736648.080 6041327.062, -13734265...
17 POLYGON ((-13131350.696 5984709.557, -13131339...
18 POLYGON ((-13726667.316 5908398.973, -13724294...
19 POLYGON ((-13228357.287 6097273.339, -13228274...
20 POLYGON ((-13029073.626 6071491.429, -13029068...
21 POLYGON ((-13451695.289 6198988.419, -13451524...
22 POLYGON ((-13736747.915 5842104.061, -13735718...
23 POLYGON ((-13028819.631 5984496.998, -13028817...
24 POLYGON ((-13702664.169 5842750.924, -13700419...
25 POLYGON ((-13668140.191 5968532.266, -13667828...
26 POLYGON ((-13632070.339 6053935.446, -13632151...
27 POLYGON ((-13661948.246 6275217.332, -13660730...
28 POLYGON ((-13646976.232 6211587.326, -13646829...
29 POLYGON ((-13818358.748 6029600.620, -13817157...
30 POLYGON ((-13593638.701 5842796.230, -13591537...
31 POLYGON ((-13609304.472 5789347.279, -13609297...
32 POLYGON ((-13653405.352 6126390.566, -13653235...
33 POLYGON ((-13527802.842 5787404.070, -13526218...
34 POLYGON ((-13228831.248 6274874.740, -13228837...
35 POLYGON ((-13157667.072 6274864.610, -13157440...
36 POLYGON ((-13500270.861 5956159.912, -13500209...
37 POLYGON ((-13292341.455 5883579.005, -13292340...
38 POLYGON ((-13651653.941 6008143.837, -13651640...

```

	Persons of Household with no Minor	Households with no Minors \
0	104.0	97.0
1	11.0	11.0
2	105.0	94.0
3	11.0	11.0
4	123.0	122.0
5	11.0	11.0
6	55.0	55.0
7	13.0	13.0
8	44.0	44.0

9	11.0	11.0
10	151.0	147.0
11	48.0	44.0
12	35.0	33.0
13	12.0	12.0
14	7707.0	7222.0
15	818.0	776.0
16	90.0	86.0
17	0.0	0.0
18	97.0	89.0
19	0.0	0.0
20	1171.0	1118.0
21	229.0	215.0
22	11.0	11.0
23	11.0	11.0
24	244.0	223.0
25	672.0	645.0
26	390.0	366.0
27	521.0	496.0
28	181.0	162.0
29	92.0	91.0
30	36.0	35.0
31	536.0	491.0
32	119.0	112.0
33	28.0	27.0
34	55.0	49.0
35	11.0	11.0
36	457.0	442.0
37	50.0	50.0
38	1527.0	1445.0

	Persons in Households with Minors	Households with Minors \
0	75.0	19.0
1	11.0	11.0
2	24.0	11.0
3	11.0	11.0
4	11.0	11.0
5	11.0	11.0
6	11.0	11.0
7	11.0	11.0
8	11.0	11.0
9	29.0	11.0
10	46.0	16.0
11	11.0	11.0
12	11.0	11.0
13	11.0	11.0
14	3743.0	1190.0

15	284.0	92.0
16	83.0	25.0
17	0.0	0.0
18	45.0	16.0
19	0.0	0.0
20	363.0	104.0
21	92.0	30.0
22	0.0	0.0
23	14.0	11.0
24	81.0	28.0
25	310.0	95.0
26	133.0	42.0
27	165.0	55.0
28	130.0	36.0
29	15.0	11.0
30	11.0	11.0
31	372.0	120.0
32	20.0	11.0
33	11.0	11.0
34	11.0	11.0
35	0.0	0.0
36	176.0	49.0
37	81.0	23.0
38	358.0	113.0

	Persons in Households with only Minors	Households with only Minors \
0	11.0	11.0
1	0.0	0.0
2	0.0	0.0
3	11.0	11.0
4	11.0	11.0
5	0.0	0.0
6	0.0	0.0
7	0.0	0.0
8	11.0	11.0
9	11.0	11.0
10	11.0	11.0
11	11.0	11.0
12	0.0	0.0
13	0.0	0.0
14	301.0	210.0
15	30.0	29.0
16	11.0	11.0
17	0.0	0.0
18	0.0	0.0
19	0.0	0.0
20	25.0	22.0



21	16.0	11.0
22	0.0	0.0
23	11.0	0.0
24	11.0	11.0
25	13.0	0.0
26	11.0	11.0
27	11.0	11.0
28	11.0	0.0
29	11.0	0.0
30	0.0	0.0
31	11.0	11.0
32	0.0	0.0
33	11.0	11.0
34	11.0	11.0
35	0.0	0.0
36	0.0	0.0
37	11.0	11.0
38	12.0	12.0

	Total Persons	Total Households	Proportion of Persons Homeless \
0	190.0	127.0	0.016019
1	22.0	22.0	0.096916
2	129.0	105.0	0.013884
3	33.0	33.0	0.004458
4	145.0	144.0	0.018642
5	22.0	22.0	0.048889
6	66.0	66.0	0.033933
7	24.0	24.0	0.009091
8	66.0	66.0	0.007741
9	51.0	33.0	0.037118
10	208.0	174.0	0.022525
11	70.0	66.0	0.036649
12	46.0	44.0	0.012283
13	23.0	23.0	0.004942
14	11751.0	8622.0	0.029281
15	1132.0	897.0	0.011705
16	184.0	122.0	0.034684
17	0.0	0.0	0.000000
18	142.0	105.0	0.016657
19	0.0	0.0	0.000000
20	1559.0	1244.0	0.020142
21	337.0	256.0	0.030317
22	11.0	11.0	0.038194
23	36.0	22.0	0.003659
24	336.0	262.0	0.023168
25	995.0	740.0	0.026278
26	534.0	419.0	0.016697

27	697.0	562.0	0.020665
28	322.0	198.0	0.021356
29	118.0	102.0	0.013887
30	47.0	46.0	0.052049
31	919.0	622.0	0.014625
32	139.0	123.0	0.044666
33	50.0	49.0	0.020790
34	77.0	71.0	0.014986
35	11.0	11.0	0.015257
36	633.0	491.0	0.019638
37	142.0	84.0	0.005886
38	1897.0	1570.0	0.015660

	Area Square Mile	Homeless Density
0	2791.509308	0.068064
1	718.303625	0.030628
2	519.285584	0.248418
3	2332.997125	0.014145
4	1298.849463	0.111637
5	873.656769	0.025182
6	621.780599	0.106147
7	640.378302	0.037478
8	1265.038501	0.052172
9	1425.674397	0.035773
10	2675.816485	0.077733
11	1230.748350	0.056876
12	2540.095539	0.018110
13	1848.765120	0.012441
14	2307.374698	5.092801
15	2195.782909	0.515534
16	1051.071585	0.175059
17	1929.866014	0.000000
18	2435.560659	0.058303
19	2339.558258	0.000000
20	1781.052426	0.875325
21	2994.368562	0.112545
22	286.202018	0.038434
23	2177.588950	0.016532
24	1166.432160	0.288058
25	774.091854	1.285377
26	565.788502	0.943816
27	2504.781911	0.278268
28	1916.298450	0.168032
29	2240.984262	0.052655
30	1686.562169	0.027867
31	655.710125	1.401534
32	2186.831385	0.063562

33	1904.416480	0.026255
34	5312.680914	0.014494
35	2257.456975	0.004873
36	4311.135371	0.146829
37	1759.728417	0.080694
38	1805.524704	1.050664

## 7 Four subplot maps displaying proportion measures, density, and counts

dataset used: rent\_income.csv

```
[52]: fig, axs = plt.subplots(2, 2, figsize=(14, 7))

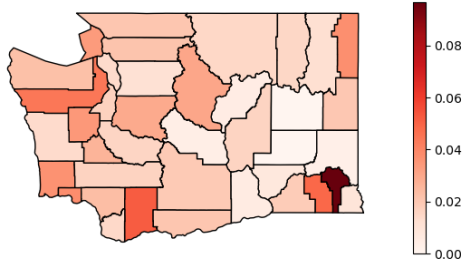
titles = [
    "Proportion of Persons Homeless (%)",
    "Total Persons Homeless (Count)",
    "Homeless Density (square mile)" ,
    "Total Households Homeless (Count)"
]

data_columns = ['Proportion of Persons Homeless', 'Total Persons', 'Homeless_
↳Density', 'Total Households'] # Adjust these column names as per your
↳DataFrame

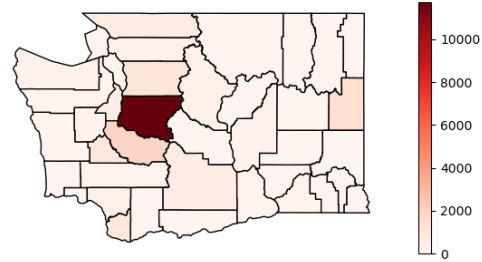
for ax, title, column in zip(axs.flatten(), titles, data_columns):
    merged_homeless2020_data.plot(column=column, cmap='Reds',
↳edgecolor='black', ax=ax, legend=True)
    ax.set_title(title, fontsize=16, fontweight='bold')
    ax.set_axis_off()

plt.tight_layout()
plt.show()
```

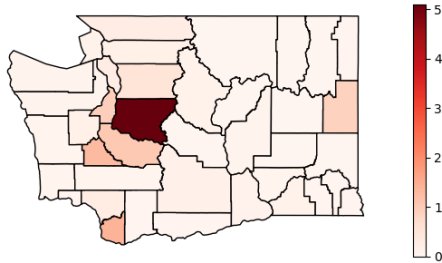
**Proportion of Persons Homeless (%)**



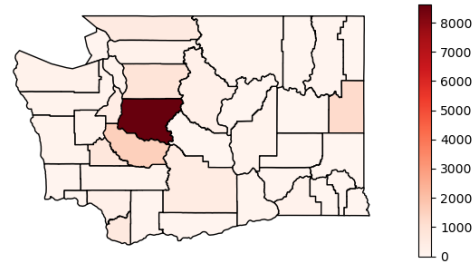
**Total Persons Homeless (Count)**



**Homeless Density (square mile)**



**Total Households Homeless (Count)**



## 8 Homeless Interactive Bar Chart Displaying Values Of Different Shelter Categories

This section of code uses the dataset `homeless_shelter_data` and melts it into one category of measure with values added up, then we visualized it using `plotly` library showing a bar graph of values comparison between different factors.

```
[53]: homeless_shelter2020.head()
```

```
[53]:
```

	Type	Category	Emergency_Shelter	\
0	Households_one adult_one child	Households	781.0	
1	Households_one adult_one child	Persons	2505.0	
2	Households_ only_children	Households	68.0	
3	Households_ only_children	Persons	95.0	
4	Households_without_children	Households	5813.0	

	Transitional_Housing	Safe_Haven	Total_Sheltered	Unsheltered	TOTAL
0	767	-	1548.0	570.0	2118.0
1	2336	-	4841.0	1891.0	6732.0
2	26	-	94.0	224.0	318.0
3	47	-	142.0	308.0	450.0
4	1048	81	6942.0	7888.0	14830.0

```
[54]: # Removing underscores and leading/trailing spaces from column names
homeless_shelter2020.columns = homeless_shelter2020.columns.str.replace('_', ' ')
      ↪).str.strip()
```

```
homeless_shelter2020['Type'] = homeless_shelter2020['Type'].str.replace('_', '␣')
                                ↪).str.strip()
homeless_shelter2022.columns = homeless_shelter2022.columns.str.replace('_', '␣')
                                ↪).str.strip()
homeless_shelter2022['Type'] = homeless_shelter2022['Type'].str.replace('_', '␣')
                                ↪).str.strip()
```

```
[55]: def convert_to_numeric(value):
        """
        Converts a given value to a numeric type. If the value is a hyphen ('-')
        or contains a less-than sign ('<'), it is interpreted as '0'.

        Parameters:
        - value: The value to convert, which can be a number, a string representing
                  a number, a hyphen ('-'), or a string with a less-than sign ('<').

        Returns:
        - The numeric representation of the input value if it can be converted
        """
        if value == '-':
            return 0
        elif '<' in str(value):
            return 0
        else:
            return pd.to_numeric(value)

        ##### 2020 #####
        for column in ['Emergency Shelter', 'Transitional Housing', 'Safe Haven']:
            homeless_shelter2020[column] = homeless_shelter2020[column].
            ↪apply(convert_to_numeric)
        homeless_shelter2020['Safe Haven'] = pd.to_numeric(homeless_shelter2020['Safe_
            ↪Haven']).replace('-', 0))

        homeless_shelter2020['Group'] = homeless_shelter2020['Type'].apply(lambda x:
            ↪'With Children' if 'child' in x else 'Without Children')

        grouped_data = homeless_shelter2020.groupby(['Group', 'Category']).sum()

        households_data = grouped_data.loc[(slice(None), 'Households'), :]
        persons_data = grouped_data.loc[(slice(None), 'Persons'), :]
```

```
[56]: ##### 2022 #####
        for column in ['Emergency Shelter', 'Transitional Housing', 'Safe Haven']:
```

```

homeless_shelter2022[column] = homeless_shelter2022[column].
    ↪apply(convert_to_numeric)
homeless_shelter2022['Safe Haven'] = pd.to_numeric(homeless_shelter2022['Safe_
    ↪Haven']).replace('-', 0)
homeless_shelter2022['Group'] = homeless_shelter2022['Type'].apply(lambda x:
    ↪'With Children' if 'child' in x else 'Without Children')
grouped_data2 = homeless_shelter2022.groupby(['Group', 'Category']).sum()

households_data2 = grouped_data2.loc[(slice(None), 'Households'), :]
persons_data2 = grouped_data2.loc[(slice(None), 'Persons'), :]

```

```
[57]: print(homeless_shelter2022.columns)
```

```

Index(['Type', 'Category', 'Emergency Shelter', 'Transitional Housing',
      'Safe Haven', 'Total Sheltered', 'Unsheltered', 'Total', 'Group'],
      dtype='object')

```

### 8.0.1 Shelter Statistics by Category Plot

```

[59]: ##### 2020 Plot #####

melted_shelter_data = pd.melt(homeless_shelter2020, id_vars=['Group',
    ↪'Category'],
                                value_vars=['Emergency Shelter', 'Transitional_
    ↪Housing',
                                'Safe Haven', 'Unsheltered'], var_name='Shelter_
    ↪Type',
                                value_name='Count')

# Grouped data for plotting
grouped_data = melted_shelter_data.groupby(['Shelter Type',
    ↪'Category'])['Count'].sum().unstack().reset_index()

fig = go.Figure()

for category in grouped_data.columns[1:]:
    fig.add_trace(go.Bar(x=grouped_data['Shelter Type'],
    ↪y=grouped_data[category], name=category))

fig.update_layout(
    title='Shelter Statistics by Category(2020)',
    xaxis_title='Shelter Type',
    yaxis_title='Count',
    barmode='group',
    width=1000,
    height=600
)

```

```

fig.show()

##### 2022 Plot #####
# Color bar is different colors, we couldn't figure out how to make it be
↳consistent.

melted_shelter_data2 = pd.melt(homeless_shelter2022, id_vars=['Group',
↳'Category'],
                                value_vars=['Emergency Shelter', 'Transitional_
↳Housing',
                                'Safe Haven', 'Unsheltered'], var_name='Shelter_
↳Type',
                                value_name='Count')

# Replacing the string representation of the field to a plottable one
melted_shelter_data2['Count'] = melted_shelter_data2['Count'].replace('< 11',
↳11)
melted_shelter_data2['Count'] = pd.to_numeric(melted_shelter_data2['Count'])

grouped_data2 = melted_shelter_data2.groupby(['Shelter Type',
↳'Category'])['Count'].sum().unstack().reset_index()

fig = go.Figure()

for category in grouped_data2.columns[1:]:
    fig.add_trace(go.Bar(x=grouped_data2['Shelter Type'],
↳y=grouped_data2[category], name=category))

fig.update_layout(
    title='Shelter Statistics by Category(2022)',
    xaxis_title='Shelter Type',
    yaxis_title='Count',
    barmode='group',
    width=1000,
    height=600
)

fig.show()

```

```
[60]: melted_shelter_data2
```

```
[60]:
```

	Group	Category \
0	With Children	Households
1	With Children	Persons
2	With Children	Households

```

3      With Children      Persons
4      With Children      Households
..      ...
83 Without Children      Unaccompanied Youth households
84 Without Children      Parenting Youth Households
85 Without Children      Total number of persons
86 Without Children      Persons in parenting youth household
87 Without Children      Persons in unaccompanied youth household

```

```

      Shelter Type      Count
0  Emergency Shelter    958.0
1  Emergency Shelter    3120.0
2  Emergency Shelter     71.0
3  Emergency Shelter     71.0
4  Emergency Shelter    6341.0
..      ...
83      Unsheltered    2793.0
84      Unsheltered     11.0
85      Unsheltered   1056.0
86      Unsheltered     11.0
87      Unsheltered   1050.0

```

[88 rows x 4 columns]

```
[61]: homeless2020
```

```

[61]:      County  Persons of Household with no Minor \
0      Adams County      0.0
1      Asotin County     13.0
2      Benton County     50.0
3      Chelan County    229.0
4      Clallam County    151.0
5      Clark County     536.0
6      Columbia County     11.0
7      Cowlitz County    244.0
8      Douglas County     12.0
9      Ferry County      11.0
10     Franklin County     44.0
11     Garfield County     11.0
12     Grant County     104.0
13     Grays Harbor County     92.0
14     Island County     105.0
15     Jefferson County    119.0
16     King County     7707.0
17     Kitsap County     390.0
18     Kittitas County     11.0
19     Klickitat County     28.0

```



20	Lewis County	97.0
21	Lincoln County	0.0
22	Mason County	90.0
23	Okanogan County	55.0
24	Pacific County	48.0
25	Pend Oreille County	11.0
26	Pierce County	1527.0
27	San Juan County	55.0
28	Skagit County	181.0
29	Skamania County	36.0
30	Snohomish County	818.0
31	Spokane County	1171.0
32	Stevens County	35.0
33	Thurston County	672.0
34	Wahkiakum County	11.0
35	Walla Walla County	123.0
36	Whatcom County	521.0
37	Whitman County	11.0
38	Yakima County	457.0

	Households with no Minors	Persons in Households with Minors \
0	0.0	0.0
1	13.0	11.0
2	50.0	81.0
3	215.0	92.0
4	147.0	46.0
5	491.0	372.0
6	11.0	11.0
7	223.0	81.0
8	12.0	11.0
9	11.0	0.0
10	44.0	11.0
11	11.0	11.0
12	97.0	75.0
13	91.0	15.0
14	94.0	24.0
15	112.0	20.0
16	7222.0	3743.0
17	366.0	133.0
18	11.0	11.0
19	27.0	11.0
20	89.0	45.0
21	0.0	0.0
22	86.0	83.0
23	49.0	11.0
24	44.0	11.0
25	11.0	29.0

26	1445.0	358.0
27	55.0	11.0
28	162.0	130.0
29	35.0	11.0
30	776.0	284.0
31	1118.0	363.0
32	33.0	11.0
33	645.0	310.0
34	11.0	0.0
35	122.0	11.0
36	496.0	165.0
37	11.0	14.0
38	442.0	176.0

	Households with Minors	Persons in Households with only Minors \
0	0.0	0.0
1	11.0	0.0
2	23.0	11.0
3	30.0	16.0
4	16.0	11.0
5	120.0	11.0
6	11.0	0.0
7	28.0	11.0
8	11.0	0.0
9	0.0	0.0
10	11.0	11.0
11	11.0	0.0
12	19.0	11.0
13	11.0	11.0
14	11.0	0.0
15	11.0	0.0
16	1190.0	301.0
17	42.0	11.0
18	11.0	11.0
19	11.0	11.0
20	16.0	0.0
21	0.0	0.0
22	25.0	11.0
23	11.0	11.0
24	11.0	11.0
25	11.0	11.0
26	113.0	12.0
27	11.0	0.0
28	36.0	11.0
29	11.0	0.0
30	92.0	30.0
31	104.0	25.0

32	11.0	0.0
33	95.0	13.0
34	0.0	0.0
35	11.0	11.0
36	55.0	11.0
37	11.0	11.0
38	49.0	0.0

	Households with only Minors	Total Persons	Total Households
0	0.0	0.0	0.0
1	0.0	24.0	24.0
2	11.0	142.0	84.0
3	11.0	337.0	256.0
4	11.0	208.0	174.0
5	11.0	919.0	622.0
6	0.0	22.0	22.0
7	11.0	336.0	262.0
8	0.0	23.0	23.0
9	0.0	11.0	11.0
10	11.0	66.0	66.0
11	0.0	22.0	22.0
12	11.0	190.0	127.0
13	0.0	118.0	102.0
14	0.0	129.0	105.0
15	0.0	139.0	123.0
16	210.0	11751.0	8622.0
17	11.0	534.0	419.0
18	11.0	33.0	33.0
19	11.0	50.0	49.0
20	0.0	142.0	105.0
21	0.0	0.0	0.0
22	11.0	184.0	122.0
23	11.0	77.0	71.0
24	11.0	70.0	66.0
25	11.0	51.0	33.0
26	12.0	1897.0	1570.0
27	0.0	66.0	66.0
28	0.0	322.0	198.0
29	0.0	47.0	46.0
30	29.0	1132.0	897.0
31	22.0	1559.0	1244.0
32	0.0	46.0	44.0
33	0.0	995.0	740.0
34	0.0	11.0	11.0
35	11.0	145.0	144.0
36	11.0	697.0	562.0
37	0.0	36.0	22.0

38

0.0

633.0

491.0

### 8.0.2 2020 HOMELESS PEOPLE FOR EACH COUNTY

Plan: create a graph that visually displays the comparison between households w/out minors, households with minors, and the households with ONLY minors

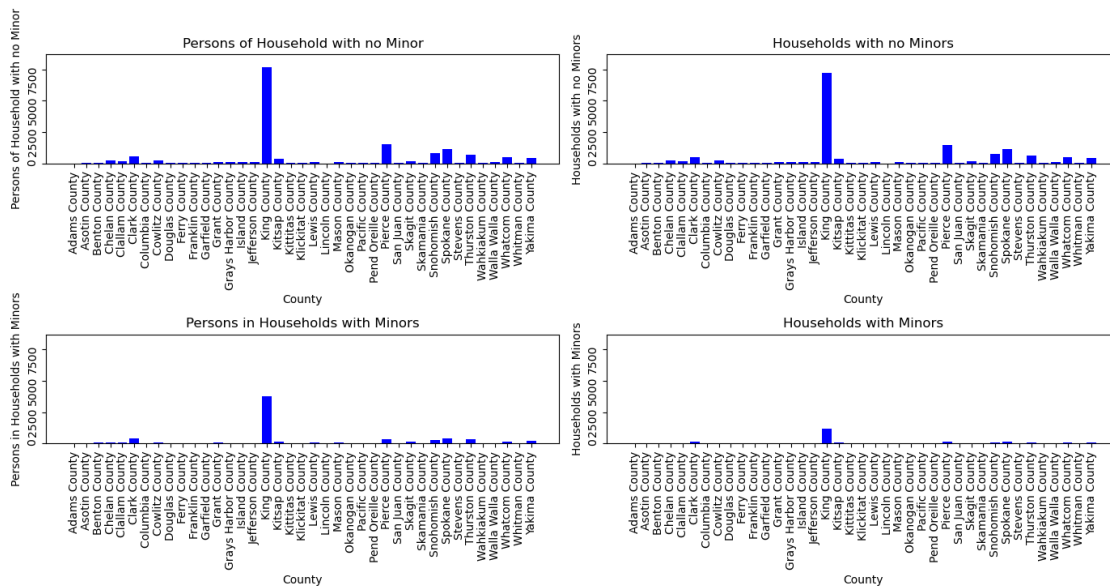
```
[62]: # Define the measures you want to plot, excluding total columns
measures = [
    'Persons of Household with no Minor',
    'Households with no Minors',
    'Persons in Households with Minors',
    'Households with Minors'
]

max_value = homeless2020[measures].max().max()
fig, axs = plt.subplots(2, 2, figsize=(14, 8))
fig.suptitle('Homeless Count by Measure by County in Washington')
axs = axs.flatten()

for i, measure in enumerate(measures):
    axs[i].bar(homeless2020['County'], homeless2020[measure], color='blue')
    axs[i].set_title(measure)
    axs[i].set_xlabel('County')
    axs[i].set_ylabel(measure)
    axs[i].tick_params(labelrotation=90)
    axs[i].set_ylim(0, max_value + 1000) # to evenly space out the y axis
    ↪ labels and also set a consistent scale across all four subplots

plt.tight_layout(rect=[0, 0.03, 1, 0.95]) # Adjust the layout to make room for
    ↪ the overall title
plt.show()
```

Homeless Count by Measure by County in Washington



```
[63]: columns_graph = ['Persons of Household with no Minor',
                        'Households with no Minors', 'Persons in Households with Minors',
                        'Households with Minors', 'Persons in Households with only Minors',
                        'Households with only Minors', 'Total Persons', 'Total Households']

# Calculate the percentage change between 2020 and 2022 datasets for each
category
percentage_changes = {}
for column in columns_graph:
    percentage_changes[column] = ((homeless2022[column] - homeless2020[column]) /
    homeless2020[column]) * 100

# Create a horizontal bar graph
plt.figure(figsize=(10, 8))
plt.barh(homeless2022['County'], percentage_changes['Total Persons'],
color='skyblue', label='Total Persons')
plt.xlabel('Percentage Change', fontweight='bold', fontsize=12, color='black')
plt.ylabel('County', fontweight='bold', fontsize=12, color='black')
plt.title('Percentage Change of Total Homeless Persons from 2020 to 2022',
fontweight='bold', fontsize=14, color='black') # Brighter font
plt.gca().invert_yaxis()
plt.grid(axis='x')

# Add labels to the bars
for county, change in zip(homeless2022['County'], percentage_changes['Total
Persons']):
```

```
plt.text(change, county, f'{change:.2f}%', va='center', ha='left',
↪fontsize=10, fontweight='bold', color='black')
```

```
plt.tight_layout()
plt.show()
```

posx and posy should be finite values  
posx and posy should be finite values  
posx and posy should be finite values  
posx and posy should be finite values

