Untitled

October 21, 2025

[473]:
```
import pandas as pd
import altair as alt
pd.set_option('display.max_rows', None)

# Display all columns
pd.set_option('display.max_columns', None)
```

[632]:
```
data =pd.read_csv('poverty_Neighborhoods_4633402121131220268.csv',index_col=0)
data.columns = data.columns.str.replace('\xa0', ' ')
data.head()
```

[632]:

OBJECTID	NEIGH_NO	Neighborhood Name	Neighborhood Type	Neighborhood Subtype	ACS Vinatage	Population 16 years and over	Population 16 years and over not in labor force	Population 16 years and over in civilian labor force
1	8.2	Olympic Hills/Victory Heights	CRA	North	5Y23	14516	3751	10745
2	8.3	Cedar Park/Meadowbrook	CRA	North	5Y23	12513	3677	
3	9.1	Broadview/Bitter Lake	CRA	Northwest	5Y23	13260	4753	
4	9.2	Licton Springs	CRA	Northwest	5Y23	8798	1895	
5	9.3	Greenwood/Phinney Ridge	CRA	Northwest	5Y23	23396	4985	

2	8836
3	8507
4	6903
5	18381

OBJECTID	Population 16 years and over in civilian labor force unemployed
1	574
2	259
3	356
4	267
5	565

OBJECTID	Population for whom poverty status is determined
1	17039
2	14445
3	15645
4	9597
5	26912

OBJECTID	People with Income to Poverty Ratio Under .50
1	670
2	498
3	676
4	631
5	854

OBJECTID	People with Income to Poverty .50 to .99
1	717
2	824
3	1105
4	429
5	371

OBJECTID	People with Income to Poverty 1.00 to 1.24
1	376
2	497
3	173
4	145
5	241

OBJECTID	People with Income to Poverty 1.25 to 1.49

1	426
2	287
3	291
4	310
5	395

OBJECTID	People with Income to Poverty 1.50 to 1.84 \
1	594
2	582
3	898
4	411
5	417

OBJECTID	People with Income to Poverty 1.85 to 1.99 \
1	168
2	119
3	118
4	155
5	335

OBJECTID	People with Income to Poverty 2.00 and over \
1	14088
2	11638
3	12384
4	7516
5	24299

OBJECTID	Families for whom poverty status is determined \
1	3765
2	3442
3	3478
4	1580
5	6376

OBJECTID	Families with income in the past 12 months below poverty level \
1	156
2	169
3	242
4	92
5	117

Married-couple family with related children of the householder under

```
18 years below poverty  \
OBJECTID
1                                                    0
2                                                   54
3                                                    9
4                                                    0
5                                                   35

          Married-couple family without related children of the householder
under 18 years below poverty  \
OBJECTID
1                                                   29
2                                                   28
3                                                   49
4                                                   13
5                                                   12

          Single parents with related children of the householder under 18 years
below poverty  \
OBJECTID
1                                                   67
2                                                   49
3                                                  161
4                                                   13
5                                                   65

          Other family below poverty  \
OBJECTID
1                                  60
2                                  38
3                                  23
4                                  66
5                                   5

          Population 20 to 64 years for whom poverty status is determined  \
OBJECTID
1                                                11633
2                                                 9389
3                                                 9295
4                                                 7609
5                                                19323

          Population 20 to 64 years below poverty  \
OBJECTID
1                                           887
2                                           843
3                                           861
```

```
4                                          712
5                                          886

          Population 20 to 64 years employed  \
OBJECTID
1                                     9637
2                                     7774
3                                     7454
4                                     6339
5                                    16629

          Population 20 to 64 years below poverty employed  \
OBJECTID
1                                                      392
2                                                      250
3                                                      285
4                                                      201
5                                                      324

          People with Income to Poverty below 2.00  \
OBJECTID
1                                          2951
2                                          2807
3                                          3261
4                                          2081
5                                          2613

          Population 20 to 64 years in civilian labor force  \
OBJECTID
1                                                     10165
2                                                      7985
3                                                      7776
4                                                      6601
5                                                     17105

          Population 20 to 64 years in civilian labor force below poverty  \
OBJECTID
1                                                       594
2                                                       321
3                                                       371
4                                                       301
5                                                       456

          Population 20 to 64 years not in labor force  \
OBJECTID
1                                               1448
2                                               1404
```

```
3                                                    1519
4                                                    1008
5                                                    2188

          Population 20 to 64 years not in labor force below poverty  \
OBJECTID
1                                                                 293
2                                                                 522
3                                                                 490
4                                                                 411
5                                                                 430

          Population 20 to 64 years unemployed  \
OBJECTID
1                                          528
2                                          211
3                                          322
4                                          262
5                                          476

          Population 20 to 64 years unemployed below poverty  \
OBJECTID
1                                                         202
2                                                          71
3                                                          86
4                                                         100
5                                                         132

         Neighborhood Type (outside comp plan areas id)
OBJECTID
1                                                   CRA
2                                                   CRA
3                                                   CRA
4                                                   CRA
5                                                   CRA
```

[605]:
```
poverty_data = data[['Neighborhood Name','Neighborhood Subtype','Population 16␣
↪years and over','Population 16 years and over not in labor␣
↪force','Population 16 years and over in civilian labor force','Population 16␣
↪years and over in civilian labor force unemployed',
```

```
'People with Income to Poverty Ratio Under .50','People with Income to Poverty .
 ↪50 to .99','People with Income to Poverty 1.00 to 1.24','People with Income␣
 ↪to Poverty 1.25 to 1.49','People with Income to Poverty 1.50 to 1.
 ↪84','People with Income to Poverty 1.85 to 1.99','People with Income to␣
 ↪Poverty 2.00 and over','Married-couple family with related children of the␣
 ↪householder under 18 years below poverty','Single parents with related␣
 ↪children of the householder under 18 years below poverty','Population 20 to␣
 ↪64 years below poverty','Population 20 to 64 years employed',
'Population 20 to 64 years below poverty employed','Population 20 to 64 years␣
 ↪in civilian labor force','Population 20 to 64 years in civilian labor force␣
 ↪below poverty','Population 20 to 64 years not in labor force below poverty',
'Population 20 to 64 years unemployed','Population 20 to 64 years unemployed␣
 ↪below poverty']]
poverty_data.head()
```

[605]:

OBJECTID	Neighborhood Name	Neighborhood Subtype	Population 16 years and over	Population 16 years and over not in labor force	Population 16 years and over in civilian labor force
1	Olympic Hills/Victory Heights	North	14516	3751	10745
2	Cedar Park/Meadowbrook	North	12513	3677	8836
3	Broadview/Bitter Lake	Northwest	13260	4753	8507
4	Licton Springs	Northwest	8798	1895	6903
5	Greenwood/Phinney Ridge	Northwest	23396	4985	18381

OBJECTID	Population 16 years and over in civilian labor force unemployed	People with Income to Poverty Ratio Under .50	People with Income to Poverty .50 to .99	People with Income to Poverty 1.00 to 1.24	People with Income to Poverty 1.25 to 1.49	People with Income to Poverty 1.50 to 1.84
1	574	670	717	376	426	594
2	259	498	824	497	287	582
3	356	676	1105	173	291	898
4	267	631	429	145	310	411
5	565	854	371	241	395	417

OBJECTID	People with Income to Poverty 1.85 to 1.99
1	168
2	119
3	118
4	155
5	335

OBJECTID	People with Income to Poverty 2.00 and over
1	14088
2	11638
3	12384
4	7516
5	24299

OBJECTID	Married-couple family with related children of the householder under 18 years below poverty
1	0
2	54
3	9
4	0
5	35

OBJECTID	Single parents with related children of the householder under 18 years below poverty
1	67
2	49
3	161
4	13
5	65

OBJECTID	Population 20 to 64 years below poverty
1	887
2	843
3	861
4	712
5	886

OBJECTID	Population 20 to 64 years employed
1	9637
2	7774

```
3                                    7454
4                                    6339
5                                   16629

          Population 20 to 64 years below poverty employed  \
OBJECTID
1                                                        392
2                                                        250
3                                                        285
4                                                        201
5                                                        324

          Population 20 to 64 years in civilian labor force  \
OBJECTID
1                                                      10165
2                                                       7985
3                                                       7776
4                                                       6601
5                                                      17105

          Population 20 to 64 years in civilian labor force below poverty  \
OBJECTID
1                                                         594
2                                                         321
3                                                         371
4                                                         301
5                                                         456

          Population 20 to 64 years not in labor force below poverty  \
OBJECTID
1                                                         293
2                                                         522
3                                                         490
4                                                         411
5                                                         430

          Population 20 to 64 years unemployed  \
OBJECTID
1                                       528
2                                       211
3                                       322
4                                       262
5                                       476

          Population 20 to 64 years unemployed below poverty
OBJECTID
1                                                         202
```

2	71
3	86
4	100
5	132

[697]:
```
rename_dict = {
    'Population 16 years and over': 'Population (16+)',
    'Population 16 years and over not in labor force': 'Not in Labor Force␣
↪(16+)',
    'Population 16 years and over in civilian labor force': 'Civilian Labor␣
↪Force (16+)',
    'Population 16 years and over in civilian labor force unemployed':␣
↪'Civilian labor force unemployed (16+)',
    'People with Income to Poverty Ratio Under .50': 'Income to Poverty Ratio␣
↪Under 0.50',
    'People with Income to Poverty .50 to .99':'Income to Poverty Ratio 0.50 to␣
↪0.99',
    'People with Income to Poverty 1.00 to 1.24': 'Income to Poverty 1.00 to 1.
↪24',
    'People with Income to Poverty 1.25 to 1.49':'Income to Poverty 1.25 to 1.
↪49',
    'People with Income to Poverty 1.50 to 1.84':'Income to Poverty 1.50 to 1.
↪84',
    'People with Income to Poverty 1.85 to 1.99':'Income to Poverty 1.85 to 1.
↪99',
    'People with Income to Poverty 2.00 and over': 'Income to Poverty 2.00 and␣
↪over',
    'Families with income in the past 12 months below poverty level': 'Families␣
↪Below Poverty',
    'Married-couple family with related children of the householder under 18␣
↪years below poverty': 'Married Families Below Poverty',
    'Single parents with related children of the householder under 18 years␣
↪below poverty': 'Single-Parent Families Below Poverty',
    'Population 20 to 64 years below poverty': 'Pop 20-64 Below Poverty',
    'Population 20 to 64 years employed': 'Pop 20-64 Employed',
    'Population 20 to 64 years below poverty employed': 'Pop 20-64 Employed␣
↪Below Poverty',
    'Population 20 to 64 years in civilian labor force': 'Pop 20-64 in Civilian␣
↪Labor Force',
    'Population 20 to 64 years in civilian labor force below poverty': 'Pop 20-
64 Civilian Labor Force Below Poverty',
    'Population 20 to 64 years not in labor force below poverty': 'Pop 20-64␣
↪Not in Labor Force Below Poverty',
    'Population 20 to 64 years unemployed': 'Pop 20-64 Unemployed',
    'Population 20 to 64 years unemployed below poverty': 'Pop 20-64 Unemployed␣
↪below poverty'
```

```
}

poverty_data = poverty_data.rename(columns=rename_dict)
poverty_data.head()
```

[697]:

OBJECTID	Neighborhood Name	Neighborhood Subtype	Population (16+)	Not in Labor Force (16+)	Civilian Labor Force (16+)	Civilian labor force unemployed (16+)	Income to Poverty Ratio Under 0.50	Income to Poverty Ratio 0.50 to 0.99	Income to Poverty 1.00 to 1.24	Income to Poverty 1.25 to 1.49
1	Olympic Hills/Victory Heights	North	14516	3751	10745	574	670	717		
2	Cedar Park/Meadowbrook	North	12513	3677	8836	259	498	824		
3	Broadview/Bitter Lake	Northwest	13260	4753	8507	356	676	1105		
4	Licton Springs	Northwest	8798	1895	6903	267	631	429		
5	Greenwood/Phinney Ridge	Northwest	23396	4985	18381	565	854	371		

1	376	426
2	497	287
3	173	291
4	145	310
5	241	395

OBJECTID	Income to Poverty 1.50 to 1.84	Income to Poverty 1.85 to 1.99 \
1	594	168
2	582	119
3	898	118
4	411	155
5	417	335

OBJECTID	Income to Poverty 2.00 and over	Married Families Below Poverty \
1	14088	0
2	11638	54
3	12384	9
4	7516	0
5	24299	35

OBJECTID	Single-Parent Families Below Poverty	Pop 20-64 Below Poverty \
1	67	887
2	49	843
3	161	861
4	13	712
5	65	886

OBJECTID	Pop 20-64 Employed	Pop 20-64 Employed Below Poverty \
1	9637	392
2	7774	250
3	7454	285
4	6339	201
5	16629	324

OBJECTID	Pop 20-64 in Civilian Labor Force \
1	10165
2	7985
3	7776
4	6601
5	17105

OBJECTID	Pop 20-64 Civilian Labor Force Below Poverty \

```
OBJECTID
1                                                    594
2                                                    321
3                                                    371
4                                                    301
5                                                    456

          Pop 20-64 Not in Labor Force Below Poverty  Pop 20-64 Unemployed  \
OBJECTID
1                                                293                   528
2                                                522                   211
3                                                490                   322
4                                                411                   262
5                                                430                   476

          Pop 20-64 Unemployed below poverty
OBJECTID
1                                        202
2                                         71
3                                         86
4                                        100
5                                        132
```

The dataset was obtained from the City of Seattle Open Data Portal(https://data.seattle.gov/dataset/Poverty-and-Employment-Status-Seattle-Neighborhood/9f8r-eu9y/about_data). It provides demographic and economic statistics for Seattle neighborhoods based on the American Community Survey (ACS) five-year estimates. The data captures population distributions and poverty measures for residents aged 16 years and older, with a focus on the 20–64 age group. Key variables include population counts for employment status categories such as employed, unemployed, and not in labor force and their corresponding poverty subgroups. It also includes measures like income-to-poverty ratios, families below poverty level, and population in the civilian labor force below poverty. Together, these variables enable analysis of how employment patterns, labor participation, and family structure relate to economic inequality and poverty across different Seattle neighborhoods.

0.1 EDA

[610]:
```
import altair as alt
alt.data_transformers.disable_max_rows()
chart1 = (
    alt.Chart(poverty_data)
    .mark_bar()
    .encode(
        x=alt.X("Neighborhood Subtype:N", sort="-y", title="Neighborhood␣
 ↪Subtype"),
        y=alt.Y("Population (16+):Q", title="Population (16+)"),
        tooltip=[
            alt.Tooltip("Neighborhood Name:N", title="Neighborhood"),
```

```
            alt.Tooltip("Neighborhood Subtype:N", title="Subtype"),
            alt.Tooltip("Population (16+):Q", title="Population (16+)")
        ]
    )
    .properties(
        title="Population (16+) by Neighborhood Subtype",
        width=500,
        height=300
    )
)
chart1
```

```
/opt/anaconda3/lib/python3.12/site-packages/altair/utils/core.py:395:
FutureWarning: the convert_dtype parameter is deprecated and will be removed in
a future version.  Do ``ser.astype(object).apply()`` instead if you want
``convert_dtype=False``.
  col = df[col_name].apply(to_list_if_array, convert_dtype=False)
/opt/anaconda3/lib/python3.12/site-packages/altair/utils/core.py:395:
FutureWarning: the convert_dtype parameter is deprecated and will be removed in
a future version.  Do ``ser.astype(object).apply()`` instead if you want
``convert_dtype=False``.
  col = df[col_name].apply(to_list_if_array, convert_dtype=False)
```

[610]:
```
alt.Chart(…)
```

The above visualization shows the total population aged 16years or older in each neighbouring subtype.

[612]:
```
chart2 = (
    alt.Chart(poverty_data)
    .mark_circle(size=100, opacity=0.8)
    .encode(
        x=alt.X("Civilian Labor Force (16+):Q", title="In Civilian Labor␣
 ↪Force"),
        y=alt.Y("Not in Labor Force (16+):Q", title="Not in Labor Force"),
        tooltip=["Neighborhood Name", "Civilian Labor Force (16+)", "Not in␣
 ↪Labor Force (16+)"]
    )
    .properties(title="Labor Force vs. Not in Labor Force (16+)", width=500)
)
chart2
```

```
/opt/anaconda3/lib/python3.12/site-packages/altair/utils/core.py:395:
FutureWarning: the convert_dtype parameter is deprecated and will be removed in
a future version.  Do ``ser.astype(object).apply()`` instead if you want
``convert_dtype=False``.
  col = df[col_name].apply(to_list_if_array, convert_dtype=False)
/opt/anaconda3/lib/python3.12/site-packages/altair/utils/core.py:395:
FutureWarning: the convert_dtype parameter is deprecated and will be removed in
```

```
a future version.  Do ``ser.astype(object).apply()`` instead if you want
``convert_dtype=False``.
  col = df[col_name].apply(to_list_if_array, convert_dtype=False)
```

[612]: alt.Chart(…)

This scatter plot compares the population of the civilian labor force and the population not in the labor force across Seattle neighborhoods.

[614]:
```
family_long = poverty_data.melt(
    id_vars=["Neighborhood Subtype"],
    value_vars=[
        "Married Families Below Poverty",
        "Single-Parent Families Below Poverty"
    ],
    var_name="Family Type",
    value_name="Count"
)

chart5 = (
    alt.Chart(family_long)
    .mark_bar()
    .encode(
        x=alt.X("Neighborhood Subtype:N", sort="-y", title=None),
        y=alt.Y("Count:Q", title="Families Below Poverty"),
        color=alt.Color("Family Type:N", title="Family Type"),
        tooltip=["Neighborhood Subtype", "Family Type", "Count"]
    )
    .properties(title="Families Below Poverty by Type", width=500)
)
chart5
```

```
/opt/anaconda3/lib/python3.12/site-packages/altair/utils/core.py:395:
FutureWarning: the convert_dtype parameter is deprecated and will be removed in
a future version.  Do ``ser.astype(object).apply()`` instead if you want
``convert_dtype=False``.
  col = df[col_name].apply(to_list_if_array, convert_dtype=False)
/opt/anaconda3/lib/python3.12/site-packages/altair/utils/core.py:395:
FutureWarning: the convert_dtype parameter is deprecated and will be removed in
a future version.  Do ``ser.astype(object).apply()`` instead if you want
``convert_dtype=False``.
  col = df[col_name].apply(to_list_if_array, convert_dtype=False)
```

[614]: alt.Chart(…)

This grouped bar chart compares the number of married and single-parent families living below the poverty line across different Seattle neighborhood subtypes.

[616]:
```
chart7 = (
    alt.Chart(poverty_data)
    .mark_circle()
    .encode(
        x=alt.X("Neighborhood Subtype:N", title="Neighborhood Subtype"),
        y=alt.Y("sum(Pop 20-64 Below Poverty):Q", title="Total People Below␣
 ↪Poverty (20-64)"),
        size=alt.Size("sum(Population (16+)):Q", title="Population (16+)",␣
 ↪scale=alt.Scale(range=[100, 1000])),
        tooltip=["Neighborhood Subtype", alt.Tooltip("sum(Pop 20-64 Below␣
 ↪Poverty):Q", title="People Below Poverty")]
    )
    .properties(title="People Below Poverty (20-64) by Neighborhood Subtype",␣
 ↪width=500)
)
chart7
```

```
/opt/anaconda3/lib/python3.12/site-packages/altair/utils/core.py:395:
FutureWarning: the convert_dtype parameter is deprecated and will be removed in
a future version.  Do ``ser.astype(object).apply()`` instead if you want
``convert_dtype=False``.
  col = df[col_name].apply(to_list_if_array, convert_dtype=False)
/opt/anaconda3/lib/python3.12/site-packages/altair/utils/core.py:395:
FutureWarning: the convert_dtype parameter is deprecated and will be removed in
a future version.  Do ``ser.astype(object).apply()`` instead if you want
``convert_dtype=False``.
  col = df[col_name].apply(to_list_if_array, convert_dtype=False)
```

[616]:
```
alt.Chart(…)
```

This bubble chart shows the total number of people aged 20–64 living below the poverty line across different Seattle neighborhood subtypes. The size of each bubble represents the total neighborhood population

[618]:
```
df_long = poverty_data.melt(
    id_vars=["Neighborhood Name"],
    value_vars=["Pop 20-64 Civilian Labor Force Below Poverty", "Pop 20-64 Not␣
 ↪in Labor Force Below Poverty"],
    var_name="Group",
    value_name="Count"
)

chart8 = (
    alt.Chart(df_long)
    .mark_bar()
    .encode(
        x="Neighborhood Name:N",
        y="Count:Q",
```

```
        color="Group:N",
        tooltip=["Neighborhood Name", "Group", "Count"]
    )
    .properties(title="Poverty Distribution by Labor Force Status", width=500)
)
chart8
```

```
/opt/anaconda3/lib/python3.12/site-packages/altair/utils/core.py:395:
FutureWarning: the convert_dtype parameter is deprecated and will be removed in
a future version.  Do ``ser.astype(object).apply()`` instead if you want
``convert_dtype=False``.
  col = df[col_name].apply(to_list_if_array, convert_dtype=False)
/opt/anaconda3/lib/python3.12/site-packages/altair/utils/core.py:395:
FutureWarning: the convert_dtype parameter is deprecated and will be removed in
a future version.  Do ``ser.astype(object).apply()`` instead if you want
``convert_dtype=False``.
  col = df[col_name].apply(to_list_if_array, convert_dtype=False)
```

[618]: `alt.Chart(…)`

This bar chart compares the number of people aged 20–64 living below the poverty line who are either in the civilian labor force or not in the labor force, across Seattle neighborhoods. It highlights how poverty is distributed between working and non-working populations

0.2 Q1 - What proportion of Seattle's population falls within each income-to-poverty category?

[679]:
```
import altair as alt
alt.data_transformers.disable_max_rows()

color_scale = alt.Scale(
    domain=['<50% (deep poverty)', '50-99%', '100-199% (near poverty)',␣
 ↪'200%+'],
    range=['#b27415', '#e0b43a', '#fff176', '#6ab26a']
)

bar_chart = (
    alt.Chart(poverty_data)
    .transform_calculate(
        under50 = "(datum['Income to Poverty Ratio Under 0.50'])",
        fifty_99 = "(datum['Income to Poverty Ratio 0.50 to 0.99'])",
        hundred_199 = "(datum['Income to Poverty 1.00 to 1.24'])"
                      " + (datum['Income to Poverty 1.25 to 1.49'])"
                      " + (datum['Income to Poverty 1.50 to 1.84'])"
                      " + (datum['Income to Poverty 1.85 to 1.99'])",
        twohundred_plus = "(datum['Income to Poverty 2.00 and over'])"
    )
    .transform_fold(
```

```
        ['under50', 'fifty_99', 'hundred_199', 'twohundred_plus'],
        as_=['group','value']
    )
    .transform_aggregate(total='sum(value)', groupby=['group'])
    .transform_joinaggregate(grand_total='sum(total)')
    .transform_calculate(
        pct="datum.total / datum.grand_total * 100",
        label="{'under50':'<50% (deep poverty)', 'fifty_99':'50-99%', "
              "'hundred_199':'100-199% (near poverty)', 'twohundred_plus':
↪'200%+'}[datum.group]"
    )
    .mark_bar(height=30)
    .encode(
        y=alt.Y('label:N',␣
↪sort=['under50','fifty_99','hundred_199','twohundred_plus'],
                title='Income-to-Poverty Category'),
        x=alt.X('pct:Q', title='Percentage of Population'),
        color=alt.Color('label:N', scale=color_scale, legend=alt.
↪Legend(title='Category')),
        tooltip=[
            alt.Tooltip('label:N', title='Category'),
            alt.Tooltip('pct:Q', title='Percent', format='.1f')
        ]
    )
    .properties(
        title='Population Distribution by Income-to-Poverty Ratio',
        width=550,
        height=220
    )
)
bar_chart
```

```
/opt/anaconda3/lib/python3.12/site-packages/altair/utils/core.py:395:
FutureWarning: the convert_dtype parameter is deprecated and will be removed in
a future version.  Do ``ser.astype(object).apply()`` instead if you want
``convert_dtype=False``.
  col = df[col_name].apply(to_list_if_array, convert_dtype=False)
/opt/anaconda3/lib/python3.12/site-packages/altair/utils/core.py:395:
FutureWarning: the convert_dtype parameter is deprecated and will be removed in
a future version.  Do ``ser.astype(object).apply()`` instead if you want
``convert_dtype=False``.
  col = df[col_name].apply(to_list_if_array, convert_dtype=False)
```

[679]: `alt.Chart(…)`

This horizontal bar chart displays the percentage of Seattle's population in four income-to-poverty categories ranging from deep poverty ($< 50\%$) to above 200% of the poverty line. The bar lengths encode the proportion of residents in each group, while the sequential color scale from brown to

green visually represents economic improvement. The chart effectively shows that more than 80 % of Seattle's population lives above 200 % of the poverty threshold, whereas smaller portions experience deep or near poverty. Its simple quantitative encoding makes the differences between categories immediately clear, although the dominance of the "200 %+" category leaves little visual room to highlight smaller groups. Overall, the graphic effectively summarizes city-wide economic inequality but is more descriptive than exploratory in revealing neighborhood-level variation.

0.3 Q2 - How does the share of Seattle's population living below 200% of the poverty line vary across different neighborhood subtypes?

[745]:
```
import altair as alt
alt.data_transformers.disable_max_rows()

u050 = 'Income to Poverty Ratio Under 0.50'
p5099 = 'Income to Poverty Ratio 0.50 to 0.99'
p100  = 'Income to Poverty 1.00 to 1.24'
p125  = 'Income to Poverty 1.25 to 1.49'
p150  = 'Income to Poverty 1.50 to 1.84'
p185  = 'Income to Poverty 1.85 to 1.99'
p200p = 'Income to Poverty 2.00 and over'

bar_chart = (
    alt.Chart(poverty_data)
    .transform_calculate(
        below200=f"datum['{u050}'] + datum['{p5099}'] + datum['{p100}'] +␣
 ↪datum['{p125}'] + datum['{p150}'] + datum['{p185}']",
        total   =f"datum['{u050}'] + datum['{p5099}'] + datum['{p100}'] +␣
 ↪datum['{p125}'] + datum['{p150}'] + datum['{p185}'] + datum['{p200p}']"
    )
    # 2) Aggregate to neighborhood (group-by)
    .transform_aggregate(
        sum_below200='sum(below200)',
        sum_total='sum(total)',
        groupby=['Neighborhood Subtype']
    )
    .transform_calculate(pct_below200='datum.sum_below200 / datum.sum_total')
    .mark_bar()
    .encode(
        y=alt.Y('Neighborhood Subtype:N', title='Neighborhood'),
        x=alt.X('pct_below200:Q', title='% with income < 200% of poverty line'),
        color=alt.Color('Neighborhood Subtype:N', title='Subtype'),
        tooltip=[
            alt.Tooltip('Neighborhood Name:N', title='Neighborhood'),
            alt.Tooltip('Neighborhood Subtype:N', title='Subtype'),
            alt.Tooltip('pct_below200:Q', title='% < 200%'),
            alt.Tooltip('sum_below200:Q', title='Count < 200%'),
            alt.Tooltip('sum_total:Q', title='Total (categorized)')
```

```
        ]
    )
    .properties(
        title='Share of Population with Income < 200% of Poverty Line (by␣
 ↪Neighborhood)'
    )
)
bar_chart
```

```
/opt/anaconda3/lib/python3.12/site-packages/altair/utils/core.py:395:
FutureWarning: the convert_dtype parameter is deprecated and will be removed in
a future version.  Do ``ser.astype(object).apply()`` instead if you want
``convert_dtype=False``.
  col = df[col_name].apply(to_list_if_array, convert_dtype=False)
/opt/anaconda3/lib/python3.12/site-packages/altair/utils/core.py:395:
FutureWarning: the convert_dtype parameter is deprecated and will be removed in
a future version.  Do ``ser.astype(object).apply()`` instead if you want
``convert_dtype=False``.
  col = df[col_name].apply(to_list_if_array, convert_dtype=False)
```

[745]: `alt.Chart(…)`

In the plot, I looked at the percentage of population in the Seattle area for four income-to-poverty categories ranging from deep poverty (< 50 %) to above 200 % of the poverty line. Since there is a considerable proportion of residents with income-to-poverty ratios below 200 %, I wanted to visualize the percentage of the population with income-to-poverty < 200 % at each neighborhood level in Seattle.

This horizontal bar chart shows the share of residents earning less than 200 % of the poverty threshold across Seattle's neighborhood subtypes. Each bar represents one neighborhood subtype, with color distinguishing categories for easier comparison. The length of the bars clearly communicates the relative proportion of low-income residents, making it easy to identify that neighborhoods such as Downtown, Greater Duwamish, and Urban Centers have substantially higher concentrations of economically vulnerable populations than others like Ballard or Outside Villages.

0.4 Q3 - How does poverty vary across different employment status groups—employed, unemployed, not in labor force, and civilian labor force—within Seattle neighborhoods?

[751]:
```
import altair as alt
alt.data_transformers.disable_max_rows()

chart1 = (
    alt.Chart(poverty_data)
    # Step 1: compute each group count directly
    .transform_calculate(
        employed_pov="datum['Pop 20-64 Employed Below Poverty']",
        unemployed_pov="datum['Pop 20-64 Unemployed below poverty']",
```

```
        notlf_pov="datum['Pop 20-64 Not in Labor Force Below Poverty']",
        civillbr_pov="datum['Pop 20-64 Civilian Labor Force Below Poverty']",
        total_pov="datum['Pop 20-64 Below Poverty']"
    )
    # Step 2: fold calculated fields into one tidy structure
    .transform_fold(
        ['employed_pov', 'unemployed_pov', 'notlf_pov','civillbr_pov'],
        as_=['Category', 'Value']
    )
    # Step 3: compute proportion of total poverty within each neighborhood
    .transform_calculate(
        pct="datum.Value / datum.total_pov"
    )
    .mark_bar()
    .encode(
        x=alt.X('Neighborhood Subtype:N', title='Neighborhood'),
        y=alt.Y('sum(pct):Q', stack='normalize', title='Share of Total Poverty␣
↪(20-64)'),
        color=alt.Color('Category:N',
                        title='Employment Group',
                        scale=alt.
↪Scale(domain=['employed_pov','unemployed_pov','notlf_pov','civillbr_pov']),
                        legend=alt.Legend(labelExpr="{'employed_pov':
↪'Employed','unemployed_pov':'Unemployed','notlf_pov':'Not in Labor␣
↪Force','civillbr_pov':'Civilian Labor Force'}[datum.label]")),
        tooltip=[
            alt.Tooltip('Neighborhood Name:N'),
            alt.Tooltip('Category:N', title='Employment Status'),
            alt.Tooltip('sum(pct):Q', format='.1%', title='Share of Poverty')
        ]
    )
    .properties(
        title='Composition of Poverty by Employment Status (20-64)'
    )
)
chart1
```

```
/opt/anaconda3/lib/python3.12/site-packages/altair/utils/core.py:395:
FutureWarning: the convert_dtype parameter is deprecated and will be removed in
a future version.  Do ``ser.astype(object).apply()`` instead if you want
``convert_dtype=False``.
  col = df[col_name].apply(to_list_if_array, convert_dtype=False)
/opt/anaconda3/lib/python3.12/site-packages/altair/utils/core.py:395:
FutureWarning: the convert_dtype parameter is deprecated and will be removed in
a future version.  Do ``ser.astype(object).apply()`` instead if you want
``convert_dtype=False``.
  col = df[col_name].apply(to_list_if_array, convert_dtype=False)
```

[751]: `alt.Chart(…)`

This stacked bar chart illustrates how poverty among Seattle's 20–64-year-old population varies by employment status across different neighborhoods. Each bar represents a neighborhood, and the color segments indicate the proportion of people in poverty belonging to each employment group—employed, unemployed, not in the labor force, and civilian labor force. The visualization effectively highlights how the composition of poverty differs between neighborhoods, showing that in many areas, individuals who are not in the labor force or part of the civilian labor force make up the largest proportions of people below the poverty line, while the unemployed and employed groups represent smaller shares. The color contrast and normalized scale make these comparisons intuitive and visually engaging. However, the exact proportions are difficult to read directly from the chart without the aid of tooltips or annotations. Overall, the plot effectively conveys relative differences across neighborhoods but is less precise for interpreting exact percentage values.

0.5 Refined Graphic

I have choosen Q2 (How does the share of Seattle's population living below 200% of the poverty line vary across different neighborhood subtypes?) to further refine the visualization.

[914]:
```
import altair as alt
alt.data_transformers.disable_max_rows()

u050 = 'Income to Poverty Ratio Under 0.50'
p5099 = 'Income to Poverty Ratio 0.50 to 0.99'
p100  = 'Income to Poverty 1.00 to 1.24'
p125  = 'Income to Poverty 1.25 to 1.49'
p150  = 'Income to Poverty 1.50 to 1.84'
p185  = 'Income to Poverty 1.85 to 1.99'
p200p = 'Income to Poverty 2.00 and over'

bar_chart = (
    alt.Chart(poverty_data)
    .transform_calculate(
        below200=f"datum['{u050}'] + datum['{p5099}'] + datum['{p100}'] +␣
 ↪datum['{p125}'] + datum['{p150}'] + datum['{p185}']",
        total   =f"datum['{u050}'] + datum['{p5099}'] + datum['{p100}'] +␣
 ↪datum['{p125}'] + datum['{p150}'] + datum['{p185}'] + datum['{p200p}']"
    )
    # 2) Aggregate to neighborhood (group-by)
    .transform_aggregate(
        sum_below200='sum(below200)',
        sum_total='sum(total)',
        groupby=['Neighborhood Subtype']
    )
    .transform_calculate(pct_below200='datum.sum_below200 / datum.sum_total')
    .mark_bar()
    .encode(
        y=alt.Y('Neighborhood Subtype:N', title='Neighborhood'),
```

```
        x=alt.X('pct_below200:Q', title='% with income < 200% of poverty line'),
        color=alt.Color('Neighborhood Subtype:N', title='Subtype'),
        tooltip=[
            alt.Tooltip('Neighborhood Name:N', title='Neighborhood'),
            alt.Tooltip('Neighborhood Subtype:N', title='Subtype'),
            alt.Tooltip('pct_below200:Q', title='% < 200%'),
            alt.Tooltip('sum_below200:Q', title='Count < 200%'),
            alt.Tooltip('sum_total:Q', title='Total (categorized)')
        ]
    )
    .properties(
        title='% of Population with Income < 200% of Poverty Line (by␣
 ↪Neighborhood)'
    )
)
bar_chart
```

```
/opt/anaconda3/lib/python3.12/site-packages/altair/utils/core.py:395:
FutureWarning: the convert_dtype parameter is deprecated and will be removed in
a future version.  Do ``ser.astype(object).apply()`` instead if you want
``convert_dtype=False``.
  col = df[col_name].apply(to_list_if_array, convert_dtype=False)
/opt/anaconda3/lib/python3.12/site-packages/altair/utils/core.py:395:
FutureWarning: the convert_dtype parameter is deprecated and will be removed in
a future version.  Do ``ser.astype(object).apply()`` instead if you want
``convert_dtype=False``.
  col = df[col_name].apply(to_list_if_array, convert_dtype=False)
```

[914]: `alt.Chart(…)`

The above barplot shows the information on % of Population with Income $< 200\%$ of Poverty Line but it need to be refined to make it more effective.

0.5.1 First Iteration

[917]:
```
base = (
    alt.Chart(poverty_data)
    .transform_calculate(
        below200="datum['Income to Poverty Ratio Under 0.50'] + datum['Income␣
 ↪to Poverty Ratio 0.50 to 0.99'] + datum['Income to Poverty 1.00 to 1.24'] +␣
 ↪datum['Income to Poverty 1.25 to 1.49'] + datum['Income to Poverty 1.50 to 1.
 ↪84'] + datum['Income to Poverty 1.85 to 1.99']",
        total   ="datum['Income to Poverty Ratio Under 0.50'] + datum['Income␣
 ↪to Poverty Ratio 0.50 to 0.99'] + datum['Income to Poverty 1.00 to 1.24'] +␣
 ↪datum['Income to Poverty 1.25 to 1.49'] + datum['Income to Poverty 1.50 to 1.
 ↪84'] + datum['Income to Poverty 1.85 to 1.99'] + datum['Income to Poverty 2.
 ↪00 and over']"
    )
```

```
    .transform_aggregate(
        sum_below200='sum(below200)',
        sum_total='sum(total)',
        groupby=['Neighborhood Subtype']
    )
    .transform_calculate(pct_below200='datum.sum_below200 / datum.sum_total')
)

it1 = (
    base.mark_bar()
    .encode(
        x=alt.X('Neighborhood Subtype:N', sort='-y', title='Neighborhood␣
 ↪Subtype'),
        y=alt.Y('pct_below200:Q', title='% with income < 200% of poverty'),
        color=alt.Color('Neighborhood Subtype:N', title='Subtype'),
        tooltip=[
            alt.Tooltip('Neighborhood Subtype:N', title='Subtype'),
            alt.Tooltip('pct_below200:Q', title='% < 200%', format='.1%'),
            alt.Tooltip('sum_below200:Q', title='Count < 200%', format=','),
            alt.Tooltip('sum_total:Q', title='Total (categorized)', format=',')
        ]
    )
    .properties(title='% of Population with Income to poverty ratio < 200%')
)
it1
```

```
/opt/anaconda3/lib/python3.12/site-packages/altair/utils/core.py:395:
FutureWarning: the convert_dtype parameter is deprecated and will be removed in
a future version.  Do ``ser.astype(object).apply()`` instead if you want
``convert_dtype=False``.
  col = df[col_name].apply(to_list_if_array, convert_dtype=False)
/opt/anaconda3/lib/python3.12/site-packages/altair/utils/core.py:395:
FutureWarning: the convert_dtype parameter is deprecated and will be removed in
a future version.  Do ``ser.astype(object).apply()`` instead if you want
``convert_dtype=False``.
  col = df[col_name].apply(to_list_if_array, convert_dtype=False)
```

[917]: alt.Chart(…)

In this first iteration, I added sorting to arrange the bar chart, which shows the percentage of the population with income below 200% of the poverty line across different neighborhood subtypes, in descending order. This makes it easier for viewers to quickly identify which neighborhoods have the highest and lowest proportions of residents with lower income levels. However, the visualization needs refinement the y-axis labels should be formatted as percentages to clearly represent the metric being measured. The current aspect ratio of the plot is slightly narrow, which compresses the bars and reduces readability; increasing the chart's width would improve visual balance and make the differences between neighborhoods easier to interpret.

[922]:
```
it2 = (
    base.mark_bar()
    .encode(
        x=alt.X('Neighborhood Subtype:N', sort='-y', title='Neighborhood␣
↪Subtype'),
        y=alt.Y('pct_below200:Q', title='% with income < 200% of poverty',
                axis=alt.Axis(format='.0%')),
        color=alt.Color('Neighborhood Subtype:N', title='Subtype'),
        tooltip=[
            alt.Tooltip('Neighborhood Subtype:N', title='Subtype'),
            alt.Tooltip('pct_below200:Q', title='% < 200%', format='.1%'),
            alt.Tooltip('sum_below200:Q', title='Count < 200%', format=','),
            alt.Tooltip('sum_total:Q', title='Total (categorized)', format=',')
        ]
    )
    .properties(title='% of Population with Income < 200% of Poverty Line',␣
↪width=700, height=420)
)
it2
```

```
/opt/anaconda3/lib/python3.12/site-packages/altair/utils/core.py:395:
FutureWarning: the convert_dtype parameter is deprecated and will be removed in
a future version.  Do ``ser.astype(object).apply()`` instead if you want
``convert_dtype=False``.
  col = df[col_name].apply(to_list_if_array, convert_dtype=False)
/opt/anaconda3/lib/python3.12/site-packages/altair/utils/core.py:395:
FutureWarning: the convert_dtype parameter is deprecated and will be removed in
a future version.  Do ``ser.astype(object).apply()`` instead if you want
``convert_dtype=False``.
  col = df[col_name].apply(to_list_if_array, convert_dtype=False)
```

[922]:
```
alt.Chart(…)
```

In this iteration, I added an axis format to display the y-axis values as percentages and adjusted the chart's width and height properties to improve readability and visual balance. These changes make the scale clearer and ensure the bars are easier to compare across neighborhoods. However, the visualization still needs refinement—using a consistent color for all bars would reduce visual distraction, and adding value labels directly on each bar would make it easier to interpret exact percentage values without relying on tooltips. These adjustments would enhance both the clarity and precision of the visualization.

[925]:
```
it2_bars = (
    base.mark_bar(color='#5B8FF9')
    .encode(
        x=alt.X('Neighborhood Subtype:N', sort='-y', title='Neighborhood␣
↪Subtype'),
        y=alt.Y('pct_below200:Q', axis=alt.Axis(format='.0%'), title='% with␣
↪income < 200% of poverty')
```

```
    )
    .properties(width=700, height=420)
)

it2_labels = (
    base.mark_text(align='center', dy=-5)
    .encode(
        x=alt.X('Neighborhood Subtype:N', sort='-y'),
        y='pct_below200:Q',
        text=alt.Text('pct_below200:Q', format='.1%')
    )
)

it2 = it2_bars + it2_labels
it2.properties(title='% of Population with Income < 200% of Poverty Line - with␣
 ↪Value Labels')
```

```
/opt/anaconda3/lib/python3.12/site-packages/altair/utils/core.py:395:
FutureWarning: the convert_dtype parameter is deprecated and will be removed in
a future version.  Do ``ser.astype(object).apply()`` instead if you want
``convert_dtype=False``.
  col = df[col_name].apply(to_list_if_array, convert_dtype=False)
/opt/anaconda3/lib/python3.12/site-packages/altair/utils/core.py:395:
FutureWarning: the convert_dtype parameter is deprecated and will be removed in
a future version.  Do ``ser.astype(object).apply()`` instead if you want
``convert_dtype=False``.
  col = df[col_name].apply(to_list_if_array, convert_dtype=False)
```

[925]:
```
alt.LayerChart(…)
```

In this iteration, I refined the chart by changing all bars to the same color and adding percentage value labels above each bar for better readability. Using a single color removes unnecessary categorical emphasis and helps viewers focus on comparing bar lengths rather than color differences. The addition of direct value labels makes it easier to interpret exact percentages without relying on tooltips, improving the chart's precision and accessibility. The consistent formatting and balanced aspect ratio make the visualization clearer and more professional. However, the chart still lacks contextual reference such as how these neighborhood-level percentages compare to the overall city average which could be added in the next iteration through a reference line to provide broader interpretive context.

[928]:
```
it3 = (
    base.mark_bar(color='#5B8FF9').encode(
        x=alt.X('Neighborhood Subtype:N', sort='-y'),
        y=alt.Y('pct_below200:Q', axis=alt.Axis(format='.0%'), scale=alt.
 ↪Scale(domain=[0, 0.4]),title='% with income < 200% of poverty'),
    ).properties(width=700, height=420, title='% population with income to␣
 ↪poverty ratio <200% - with City Average') +
```

```
    base.transform_aggregate(city_avg='mean(pct_below200)').
  ↪mark_rule(color='black', strokeDash=[6,3]).encode(y='city_avg:Q') +
    base.mark_text(align='center', dy=-5).encode(x=alt.X('Neighborhood Subtype:
  ↪N', sort='-y'), y='pct_below200:Q', text=alt.Text('pct_below200:Q', format='.
  ↪1%'))
)
it3
```

```
/opt/anaconda3/lib/python3.12/site-packages/altair/utils/core.py:395:
FutureWarning: the convert_dtype parameter is deprecated and will be removed in
a future version.  Do ``ser.astype(object).apply()`` instead if you want
``convert_dtype=False``.
  col = df[col_name].apply(to_list_if_array, convert_dtype=False)
/opt/anaconda3/lib/python3.12/site-packages/altair/utils/core.py:395:
FutureWarning: the convert_dtype parameter is deprecated and will be removed in
a future version.  Do ``ser.astype(object).apply()`` instead if you want
``convert_dtype=False``.
  col = df[col_name].apply(to_list_if_array, convert_dtype=False)
```

[928]: `alt.LayerChart(…)`

In this iteration, I added a citywide reference line to provide contextual comparison and help viewers interpret how each neighborhood subtype's percentage of residents with income below 200% of the poverty line compares to the overall city average. The dashed black line indicating the average serves as a clear visual benchmark, allowing for quick identification of neighborhoods performing above or below the citywide level. I also constrained the y-axis scale to a fixed range (0–40%) to maintain consistency and prevent exaggerated differences between neighborhoods. These changes make the chart more informative and balanced by grounding neighborhood-level data within a broader context. The next refinement will involve adding a diverging color scale.

[931]:
```
sort_by_val = alt.SortField(field='pct_below200', order='descending')

bars = (
    base
    .mark_bar()
    .encode(
        x=alt.X('Neighborhood Subtype:N', sort=sort_by_val, title='Neighborhood␣
  ↪Subtype'),
        y=alt.Y('pct_below200:Q', axis=alt.Axis(format='.0%'), scale=alt.
  ↪Scale(domain=[0, 0.4]),title='% with income < 200% of poverty'),
        color=alt.Color('pct_below200:Q',title='Deviation from City Avg',␣
  ↪legend=alt.Legend(format='.1%'))
    )
    .properties(width=700, height=420, title='% population with income to␣
  ↪poverty ratio <200% - with City Average')
)

rule = (
```

```
    base.transform_aggregate(city_avg='mean(pct_below200)')
    .mark_rule(color='black', strokeDash=[6,3])
    .encode(y='city_avg:Q')
)

labels = (
    base.mark_text(align='center', dy=-5)
    .encode(
        x=alt.X('Neighborhood Subtype:N', sort=sort_by_val),
        y='pct_below200:Q',
        text=alt.Text('pct_below200:Q', format='.1%')
    )
)

it3_with_color = bars + rule + labels
it3_with_color
```

```
/opt/anaconda3/lib/python3.12/site-packages/altair/utils/core.py:395:
FutureWarning: the convert_dtype parameter is deprecated and will be removed in
a future version.  Do ``ser.astype(object).apply()`` instead if you want
``convert_dtype=False``.
  col = df[col_name].apply(to_list_if_array, convert_dtype=False)
/opt/anaconda3/lib/python3.12/site-packages/altair/utils/core.py:395:
FutureWarning: the convert_dtype parameter is deprecated and will be removed in
a future version.  Do ``ser.astype(object).apply()`` instead if you want
``convert_dtype=False``.
  col = df[col_name].apply(to_list_if_array, convert_dtype=False)
```

[931]:
```
alt.LayerChart(…)
```

In this final refinement, I added a quantitative color fill to the bars based on each subtype's percentage and included a legend. This change provides an extra visual channel besides bar length and labels to convey magnitude, so the neighborhoods with higher percentages stand out immediately in darker hues while lower values appear lighter.

[ ]:
```

```