UCI Adult Income Dataset - Exploratory and Descriptive Analysis

In this notebook, we carry out an in-depth exploratory and descriptive analysis of the UCI Adult Income Dataset, a widely used dataset for income prediction tasks based on individual demographic and employment attributes.

This phase of analysis is essential for uncovering patterns, detecting potential biases, and gaining intuition about the dataset's structure before applying any modelling procedures. We examine the distribution of key numerical and categorical variables, investigate relationships between demographic features and income levels, and use visualizations to summarize insights. Particular focus is placed on income disparities across **age groups**, **geographical regions**, **races**, and **education-occupation combinations**, helping lay a solid foundation for downstream modeling and policy-relevant interpretation.

We begin our analysis by importing the core Python libraries required for **data handling**, **numerical computation**, **visualization**, and **directory management**:

- `pandas`: Enables efficient manipulation, filtering, and aggregation of structured tabular data, forming the backbone of our analysis pipeline.
- `numpy`: Provides support for fast numerical operations, array-based computation, and statistical routines.
- `os`: Facilitates interaction with the file system, allowing us to construct flexible and portable directory paths for data and output management.
- `plotly.express`: A high-level graphing library that enables the creation of interactive, publication-quality visualizations, which we use extensively to uncover patterns and present insights throughouutputs for reproducibility.

```
# import libraries
import pandas as pd
import numpy as np
import os
import plotly.express as px
```

Define and Create Directory Paths

To ensure reproducibility and organized storage, we programmatically create directories if they don't already exist for:

- **raw data**
- **processed data**
- **results**
- **documentation**

These directories will store intermediate and final outputs for reproducibility.

```python
# get working directory
Current_dir = os.getcwd()
# on one directory up to the root directory
project_root_dir = os.path.dirname(Current_dir)
# define paths to the data folders
data_dir = os.path.join(project_root_dir,'data')
raw_dir = os.path.join(data_dir,'raw')
processed_dir = os.path.join(data_dir,'processed')
# define paths to result folder
results_dir = os.path.join(project_root_dir,'results')
# define paths to docs folder
docs_dir = os.path.join(project_root_dir,'docs')

# create directories if they do not eFileExists
os.makedirs(raw_dir, exist_ok = True)
os.makedirs(processed_dir, exist_ok = True)
os.makedirs(results_dir, exist_ok = True)
os.makedirs(docs_dir, exist_ok = True)
```

Loading the Cleaned Dataset

We load the cleaned version of the UCI Adult Income Dataset from the processed data directory into a Pandas DataFrame. The `head(10)` function shows the first ten records, giving a glimpse into the data columns such as `age`, `workclass`, `education_num`, etc.

```python
adult_data_filename = os.path.join(processed_dir,'adult_cleaned.csv')
adult_df = pd.read_csv(adult_data_filename)
adult_df.head(10)
```

	age	workclass	fnwgt	education_num	marital_status	relationship	race	sex	c
0	39	government	77516	13	single	single	white	male	2
1	50	self-employed	83311	13	married	male spouse	white	male	0
2	38	private	215646	9	divorced or separated	single	white	male	0
3	53	private	234721	7	married	male spouse	black	male	0
4	28	private	338409	13	married	female spouse	black	female	0
5	37	private	284582	14	married	female spouse	white	female	0
6	49	private	160187	5	divorced or separated	single	black	female	0
7	52	self-employed	209642	9	married	male spouse	white	male	0
8	31	private	45781	14	single	single	white	female	1
9	42	private	159449	13	married	male spouse	white	male	5

Dataset Dimensions and Data Types

Here, we examine the structure of the dataset:

- There are *32,513* entries and *16* variables.
- The dataset includes both **numerical** (e.g., `age`, `hours_per_week`) and **categorical** variables (e.g., `sex`, `education_level`).

Understanding data types and null entries is essential before proceeding with analysis.

```
adult_df.shape
```

```
(32513, 16)
```

```
adult_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32513 entries, 0 to 32512
Data columns (total 16 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   age                 32513 non-null  int64
 1   workclass           32513 non-null  object
 2   fnwgt               32513 non-null  int64
 3   education_num       32513 non-null  int64
 4   marital_status      32513 non-null  object
 5   relationship        32513 non-null  object
 6   race                32513 non-null  object
```

```
 7   sex                 32513 non-null  object
 8   capital_gain        32513 non-null  int64
 9   capital_loss        32513 non-null  int64
 10  hours_per_week      32513 non-null  int64
 11  income              32513 non-null  object
 12  education_level     32513 non-null  object
 13  occupation_grouped  32513 non-null  object
 14  native_region       32513 non-null  object
 15  age_group           32513 non-null  object
dtypes: int64(6), object(10)
memory usage: 4.0+ MB
```

Summary Statistics: Numerical Variables

This summary provides a snapshot of key distribution characteristics. We see that:

- Age ranges from 17 to 90, with a mean of 38.6 years. It is slightly right-skewed (positively skewed). While the average age is approximately 38.6 years, an examination of the percentiles reveals that the majority of individuals are clustered in the younger to middle-age range, with fewer observations in the older age brackets. This skewed age distribution might suggest labor force participation is concentrated in specific age groups, which could reflect broader demographic or economic realities.
- Capital gains/losses are highly skewed, with most values at 0 (the 75th percentile is 0). This indicates that a small number of individuals report very large gains or losses, especially evident in the capital gain variable which reaches up to $99,999. These variables act as proxies for wealth-related income that goes beyond regular wages or salaries. Individuals with non-zero values for capital gains or losses often represent a distinct socioeconomic subset of the population — typically more financially literate, or with access to investment assets. The stark inequality in their distributions mirrors real-world disparities in asset ownership and investment returns.
- The dataset has individuals working anywhere from 1 to 99 hours per week, with a median of 40. This aligns with the standard full-time work week in many countries (8 hours per day for 5 working days). The mean is slightly above that at 40.4 hours, suggesting a mild right skew, with a small subset of individuals working significantly longer hours. The mode is also 40, further reinforcing the prevalence of full-time work. A non-trivial number of individuals report working very few hours, possibly due to part-time work, unemployment, or semi-retirement. On the other extreme, some report working more than 45 hours per week, which may indicate multiple jobs, weekend-work, self-employment, or informal labor, and could reflect socioeconomic necessity.

```
adult_df.describe()
```

	age	fnwgt	education_num	capital_gain	capital_loss	hours_per_week
count	32513.000000	3.251300e+04	32513.000000	32513.000000	32513.000000	32513.000000
mean	38.590256	1.897942e+05	10.081629	1079.239812	87.432719	40.440962
std	13.638932	1.055788e+05	2.572015	7390.625650	403.243596	12.350184
min	17.000000	1.228500e+04	1.000000	0.000000	0.000000	1.000000
25%	28.000000	1.178330e+05	9.000000	0.000000	0.000000	40.000000
50%	37.000000	1.783560e+05	10.000000	0.000000	0.000000	40.000000
75%	48.000000	2.370510e+05	12.000000	0.000000	0.000000	45.000000
max	90.000000	1.484705e+06	16.000000	99999.000000	4356.000000	99.000000

Summary Statistics: Categorical Variables

`workclass`

The private sector dominates, employing ~69.7% of the population. The government sector (13.4%) and self-employment (11.2%) also make up substantial portions of the workforce. A small fraction is labeled as "unknown" (5.6%), which may correspond to missing or ambiguous data entries. Tiny proportions are voluntary (0.04%) or unemployed (0.02%), possibly underreported or underrepresented in the sample.

`marital_status`

Married individuals make up the largest group (46.1%), followed by those who are single (32.8%) and divorced or separated (18.1%). Widowed individuals represent a small minority (~3.1%).

`relationship`

The majority are labeled as "male spouse" (40.5%) or "single" (36.1%). Smaller categories include children (15.6%), female spouses (4.8%), and extended relatives (3.0%). The dominance of `male spouse` reflects the dataset's gendered structure and may point to traditional family roles. The relative scarcity of "female spouse" roles suggests potential gender imbalances in how income-earning is reported within households.

`race`

The dataset is overwhelmingly composed of White individuals (~85.4%). Other racial groups include Black (9.6%), Asian or Pacific Islander (3.2%), American Indian or Eskimo (1.0%), and Other (0.8%). The racial imbalance limits the generalizability of models trained on this data. Smaller racial groups may suffer from limited statistical power, affecting fairness and performance in predictive modeling.

`sex`

Males constitute 66.9% of the dataset, with females making up the remaining 33.1%. This male-skewed distribution could be due to sampling (e.g., primary earners in households), workforce participation patterns, or reporting biases.

`education_level`

Secondary-school graduates form the largest educational group (~32%), highlighting the central role of high school completion in the labor force. Tertiary education holders — those with university or equivalent degrees — account for nearly 25% of the population, representing a substantial segment with advanced qualifications. A notable 22.4% have attended some college without necessarily earning a degree, suggesting that partial post-secondary education is common, yet may not always translate into formal certification. The remaining 20% are distributed among those with only secondary education (9.4%), associate degrees (7.5%), primary school (3.5%), and a very small group with only preschool education (0.15%). It is ecident that the education distribution is skewed toward mid- to high-level education, with relatively few individuals having only basic schooling. This reflects a dataset that largely captures working-age adults in formal labor, which may underrepresent the least-educated populations.

`occupation_grouped`

White-collar occupations are the most prevalent (~51%), followed by blue-collar, service, and unknown. Smaller categories include military, which is marginal. Essentially, slightly over half of individuals in the dataset work in professional, managerial, sales, clerical, or tech-support roles. This suggests the dataset is heavily weighted toward professional and administrative occupations. Nearly a third of the population works in manual labor or skilled trade positions (craft, transport, machine operation, farming, etc.). This indicates a significant segment engaged in physically intensive or technical labor.

`native_region`

The vast majority of individuals are from North America (~92.3%). Smaller proportions are from Central America, Asia, Europe, South America, and a generic Other category. The heavy concentration of North American individuals reflects the U.S. focus of the dataset.

`age_group`

The largest groups are 26–35 and 36–45, followed by 46–60. These three age groups represent about 73% of the dataset. Very few individuals are under 18 or above 75, consistent with the dataset's focus on the working-age population.

```
adult_df.describe(include='object')
```

	workclass	marital_status	relationship	race	sex	income	education_level	occ
count	32513	32513	32513	32513	32513	32513	32513	325
unique	6	4	5	5	2	2	7	5
top	private	married	male spouse	white	male	<=50k	secondary-school graduate	wh
freq	22650	14984	13178	27771	21758	24677	10484	165

```
adult_df['workclass'].value_counts()
```

```
workclass
private          22650
government        4350
self-employed     3656
unknown           1836
voluntary           14
unemployed           7
Name: count, dtype: int64
```

```
adult_df['workclass'].value_counts(normalize=True)
```

```
workclass
private          0.696644
government       0.133793
self-employed    0.112447
unknown          0.056470
voluntary        0.000431
unemployed       0.000215
Name: proportion, dtype: float64
```

```
adult_df['marital_status'].value_counts(normalize=True)
```

```
marital_status
married                  0.460862
single                   0.327684
divorced or separated    0.180912
widowed                  0.030542
Name: proportion, dtype: float64
```

```
adult_df['relationship'].value_counts(normalize=True)
```

```
relationship
male spouse          0.405315
single               0.360686
child                0.155599
female spouse        0.048227
extended relative    0.030173
Name: proportion, dtype: float64
```

```
adult_df['race'].value_counts(normalize=True)
```

```
race
white                          0.854151
black                          0.096023
asian or pacific islander      0.031926
american indian or eskimo      0.009565
other                          0.008335
Name: proportion, dtype: float64
```

Income Distribution

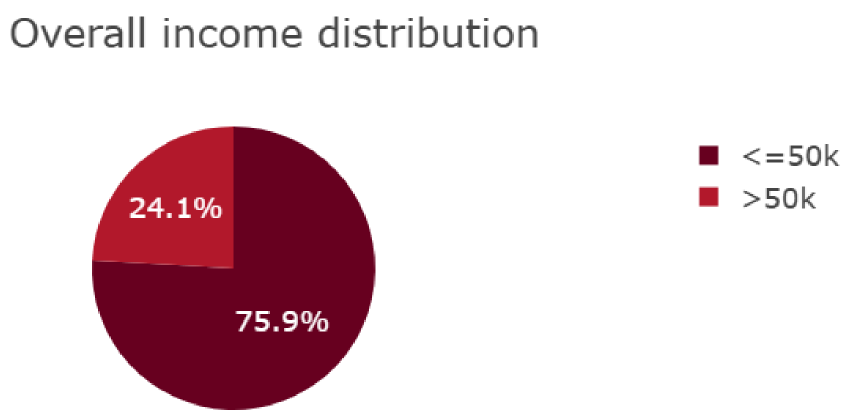
Given that `income` is the target variable, most of the analysis hereafter will be based on it. We first of all examine the income distribution in the dataset.

```
adult_df_income = adult_df.groupby('income').size().reset_index(name='total')
adult_df_income
```

	income	total
0	<=50k	24677
1	>50k	7836

```
fig = px.pie(
    adult_df_income,
    names='income',
    values='total',
    title='Overall income distribution',
    color_discrete_sequence=px.colors.sequential.RdBu
```

```
)

fig.update_layout(
    template="presentation",
    paper_bgcolor="rgba(0,0,0,0)",
    plot_bgcolor="rgba(0,0,0,0)"
)

fig.show()

fig.write_image(os.path.join(results_dir, 'income_distribution_pie_chart.jpg'))
fig.write_image(os.path.join(results_dir, 'income_distribution_pie_chart.png'))
fig.write_html(os.path.join(results_dir, 'income_distribution_pie_chart.html'))
```




```
WARNING Thread(Thread-25 (run)) Task(Task-378) choreographer.browser_async:browser_async.py:
WARNING Thread(Thread-27 (run)) Task(Task-414) choreographer.browser_async:browser_async.py:

RuntimeError: Couldn't close or kill browser subprocess
---------------------------------------------------------------------------
RuntimeError                              Traceback (most recent call last)
Cell In[50], line 18
     15 fig.show()
     17 fig.write_image(os.path.join(results_dir, 'income_distribution_pie_chart.jpg'))
---> 18 fig.write_image(os.path.join(results_dir, 'income_distribution_pie_chart.png'))
     19 fig.write_html(os.path.join(results_dir, 'income_distribution_pie_chart.html'))
```

```
File ~\PYTHON\Lib\site-packages\plotly\basedatatypes.py:3911, in BaseFigure.write_image(self
   3907     if kwargs.get("engine", None):
   3908         warnings.warn(
   3909             ENGINE_PARAM_DEPRECATION_MSG, DeprecationWarning, stacklevel=2
   3910         )
-> 3911 return pio.write_image(self, *args, **kwargs)
File ~\PYTHON\Lib\site-packages\plotly\io\_kaleido.py:509, in write_image(fig, file, format,
    505 format = infer_format(path, format)
    507 # Request image
    508 # Do this first so we don't create a file if image conversion fails
--> 509 img_data = to_image(
    510     fig,
    511     format=format,
    512     scale=scale,
    513     width=width,
    514     height=height,
    515     validate=validate,
    516     engine=engine,
    517 )
    519 # Open file
    520 if path is None:
    521     # We previously failed to make sense of `file` as a pathlib object.
    522     # Attempt to write to `file` as an open file descriptor.
File ~\PYTHON\Lib\site-packages\plotly\io\_kaleido.py:373, in to_image(fig, format, width, he
    369 from kaleido.errors import ChromeNotFoundError
    371 try:
    372     # TODO: Refactor to make it possible to use a shared Kaleido instance here
--> 373     img_bytes = kaleido.calc_fig_sync(
    374         fig_dict,
    375         opts=dict(
    376             format=format or defaults.default_format,
    377             width=width or defaults.default_width,
    378             height=height or defaults.default_height,
    379             scale=scale or defaults.default_scale,
    380         ),
    381         topojson=defaults.topojson,
    382         kopts=(
    383             dict(
    384                 mathjax=defaults.mathjax,
    385             )
    386             if defaults.mathjax
    387             else None
    388         ),
```

```
    389     )
    390 except ChromeNotFoundError:
    391     raise RuntimeError(PLOTLY_GET_CHROME_ERROR_MSG)
File ~\PYTHON\Lib\site-packages\kaleido\__init__.py:145, in calc_fig_sync(*args, **kwargs)
    143 def calc_fig_sync(*args, **kwargs):
    144     """Call `calc_fig` but blocking."""
--> 145     return _async_thread_run(calc_fig, args=args, kwargs=kwargs)
File ~\PYTHON\Lib\site-packages\kaleido\__init__.py:138, in _async_thread_run(func, args, kwa
    136 res = q.get()
    137 if isinstance(res, BaseException):
--> 138     raise res
    139 else:
    140     return res
File ~\PYTHON\Lib\site-packages\kaleido\__init__.py:129, in _async_thread_run.<locals>.run(*a
    126 def run(*args, **kwargs):
    127     # func is a closure
    128     try:
--> 129         q.put(asyncio.run(func(*args, **kwargs)))
    130     except BaseException as e:  # noqa: BLE001
    131         q.put(e)
File ~\PYTHON\Lib\asyncio\runners.py:194, in run(main, debug, loop_factory)
    190     raise RuntimeError(
    191         "asyncio.run() cannot be called from a running event loop")
    193 with Runner(debug=debug, loop_factory=loop_factory) as runner:
--> 194     return runner.run(main)
File ~\PYTHON\Lib\asyncio\runners.py:118, in Runner.run(self, coro, context)
    116 self._interrupt_count = 0
    117 try:
--> 118     return self._loop.run_until_complete(task)
    119 except exceptions.CancelledError:
    120     if self._interrupt_count > 0:
File ~\PYTHON\Lib\asyncio\base_events.py:687, in BaseEventLoop.run_until_complete(self, futu
    684 if not future.done():
    685     raise RuntimeError('Event loop stopped before Future completed.')
--> 687 return future.result()
File ~\PYTHON\Lib\site-packages\kaleido\__init__.py:54, in calc_fig(fig, path, opts, topojson
     52 kopts = kopts or {}
     53 kopts["n"] = 1
---> 54 async with Kaleido(**kopts) as k:
     55     return await k.calc_fig(
     56         fig,
     57         path=path,
     58         opts=opts,
```

```
     59         topojson=topojson,
     60     )
File ~\PYTHON\Lib\site-packages\kaleido\kaleido.py:76, in Kaleido.__aexit__(self, exc_type, e
     74 await asyncio.gather(*self._background_render_tasks, return_exceptions=True)
     75 _logger.info("Exiting Kaleido")
---> 76 return await super().__aexit__(exc_type, exc_value, exc_tb)
File ~\PYTHON\Lib\site-packages\choreographer\browser_async.py:249, in Browser.__aexit__(sel
    242 async def __aexit__(
    243     self,
    244     type_: type[BaseException] | None,
    245     value: BaseException | None,
    246     traceback: TracebackType | None,
    247 ) -> None:  # None instead of False is fine, eases type checking
    248     """Close the browser."""
--> 249     await self.close()
File ~\PYTHON\Lib\site-packages\kaleido\kaleido.py:69, in Kaleido.close(self)
     67         task.cancel()
     68 _logger.info("Exiting Kaleido/Choreo")
---> 69 return await super().close()
File ~\PYTHON\Lib\site-packages\choreographer\browser_async.py:228, in Browser.close(self)
    226 try:
    227     _logger.debug("Starting browser close methods.")
--> 228     await self._close()
    229     _logger.debug("Browser close methods finished.")
    230 except ProcessLookupError:
File ~\PYTHON\Lib\site-packages\choreographer\browser_async.py:216, in Browser._close(self)
    214     return
    215 else:
--> 216     raise RuntimeError("Couldn't close or kill browser subprocess")
RuntimeError: Couldn't close or kill browser subprocess
```

This pie chart visualizes the overall income split: 76% of individuals earn 50K, while 24% earn >50K. This means that nearly 3 out of 4 individuals fall into the lower income bracket (<=50K). This shows that there is a significant imbalance.

Income by Age Group

```
adult_df_income_age= adult_df.groupby(['age_group','income']).size().reset_index(name='total_
adult_df_income_age
```

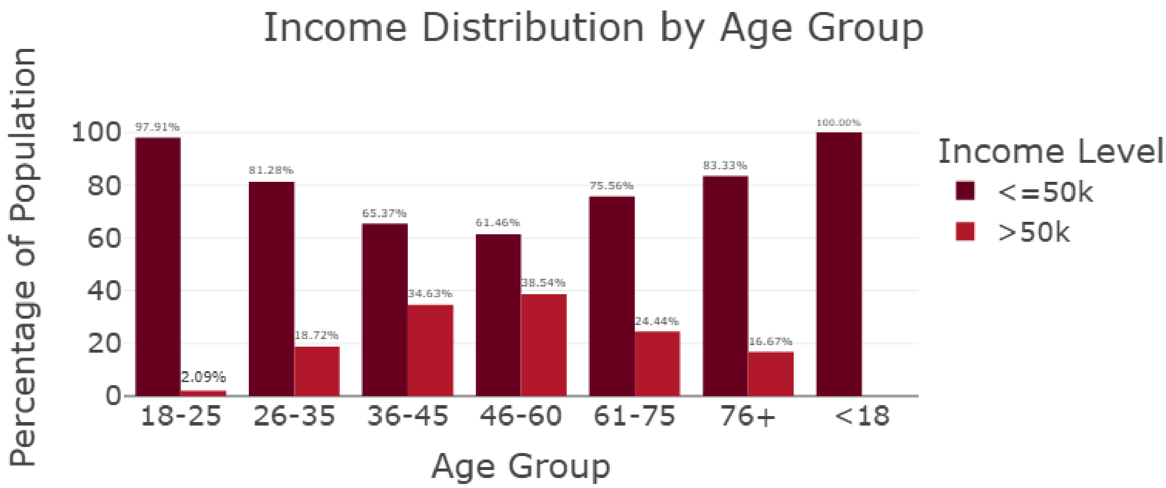
	age_group	income	total_by_age
0	18-25	<=50k	5333
1	18-25	>50k	114
2	26-35	<=50k	6910
3	26-35	>50k	1591
4	36-45	<=50k	5230
5	36-45	>50k	2771
6	46-60	<=50k	4479
7	46-60	>50k	2809
8	61-75	<=50k	1580
9	61-75	>50k	511
10	76+	<=50k	200
11	76+	>50k	40
12	<18	<=50k	945

```
total_per_group = adult_df_income_age.groupby('age_group')['total_by_age'].transform('sum')
adult_df_income_age['percentage'] = (adult_df_income_age['total_by_age'] / total_per_group *
adult_df_income_age['percentage']
```

```
0      97.91
1       2.09
2      81.28
3      18.72
4      65.37
5      34.63
6      61.46
7      38.54
8      75.56
9      24.44
10     83.33
11     16.67
12    100.00
Name: percentage, dtype: float64
```

```
fig = px.bar(adult_df_income_age,
             x='age_group',
             y='percentage',
             color='income',
             title='Income Distribution by Age Group',
             barmode='group',
```

```
                height=500,
                color_discrete_sequence=px.colors.sequential.RdBu,
                text='percentage'
)

fig.update_traces(texttemplate='%{text:.2f}%', textposition='outside')

fig.update_layout(
    template="presentation",
    xaxis_title='Age Group',
    yaxis_title='Percentage of Population',
    legend_title_text='Income Level',
    paper_bgcolor="rgba(0,0,0,0)",
    plot_bgcolor="rgba(0,0,0,0)"
)

fig.show()

fig.write_image(os.path.join(results_dir, 'income_distribution_by_agegroup_bar_plot.jpg'))
fig.write_image(os.path.join(results_dir, 'income_distribution_by_agegroup_bar_plot.png'))
fig.write_html(os.path.join(results_dir, 'income_distribution_by_agegroup_bar_plot.html'))
```

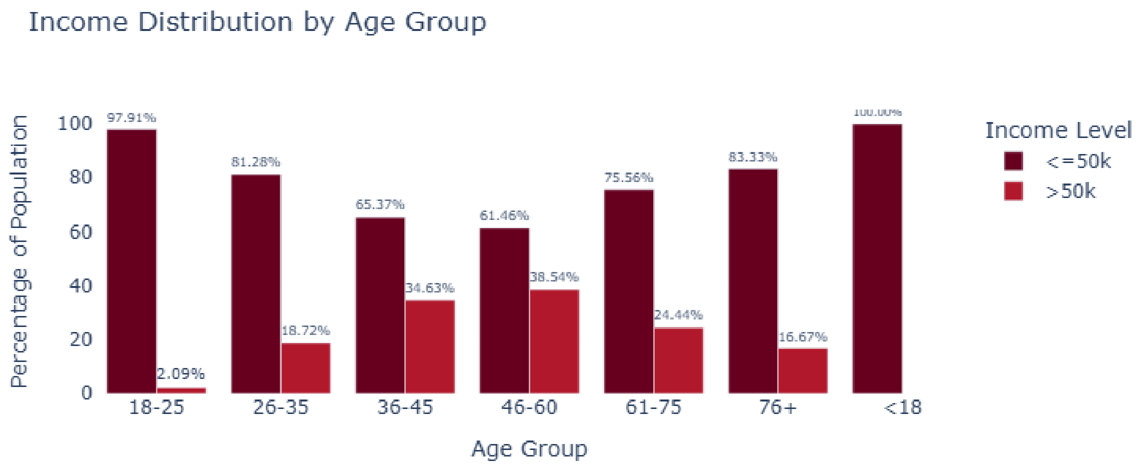


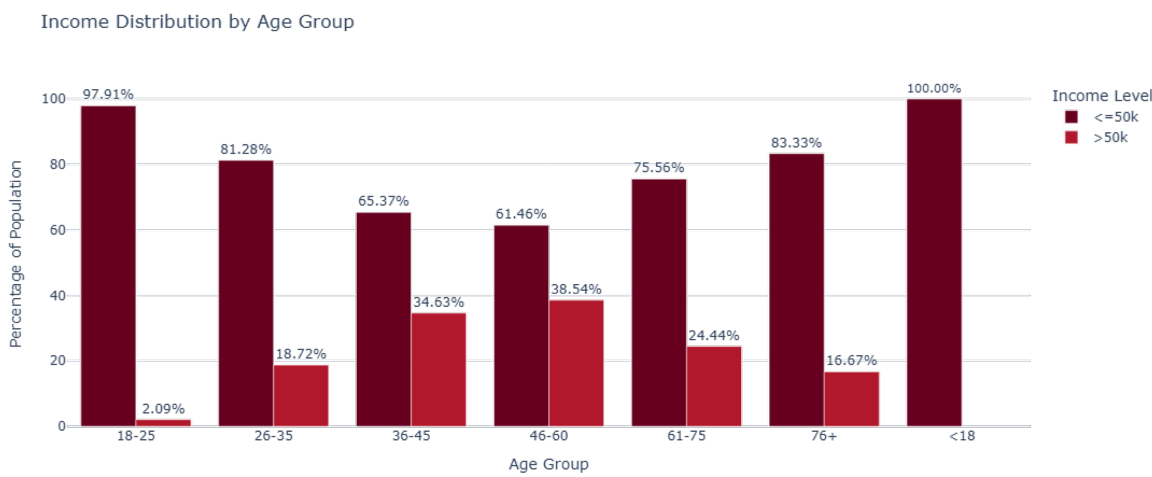

```
WARNING Thread(Thread-29 (run)) Task(Task-452) choreographer.browser_async:browser_async.py:
WARNING Thread(Thread-31 (run)) Task(Task-490) choreographer.browser_async:browser_async.py:
```

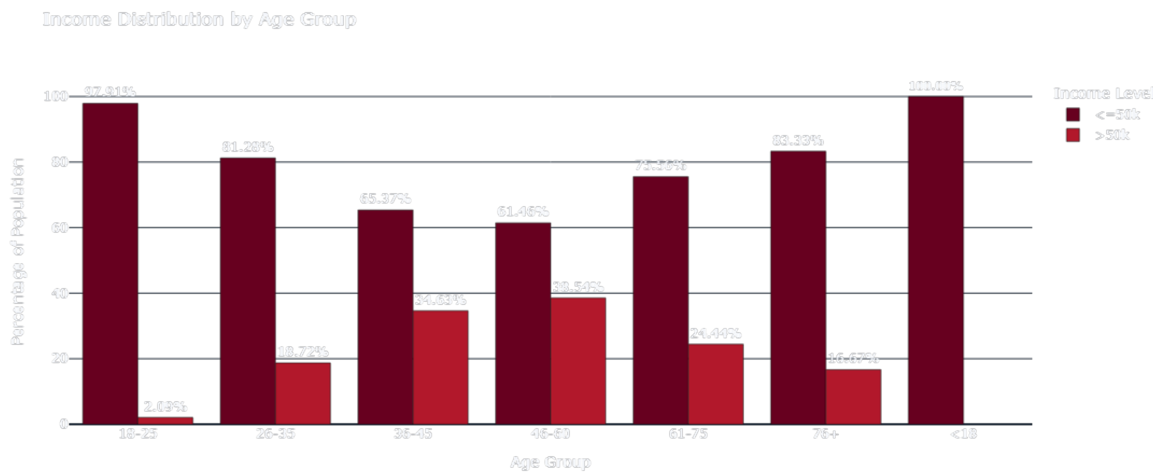
The bar chart visualizes the income distribution across age groups, using percentages within each group. There is an evident pattern in terms of income progression over the years with a gradual increase in terms of the number of people earning >50K starting from 0 amongst those aged 18 and below, peaking between 36 and 60 years, then declining after 60 years but not to zero.

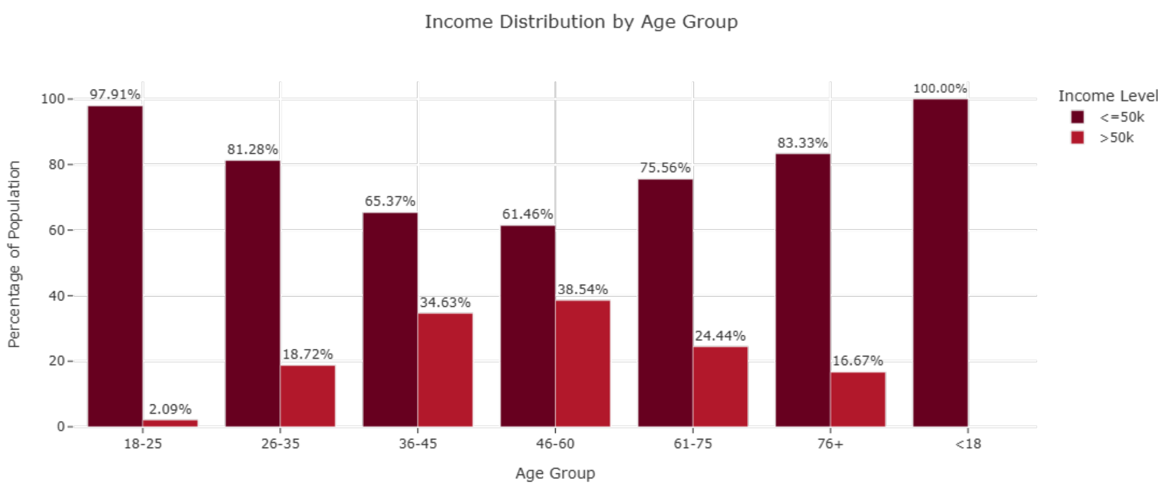
All individuals under 18 earn <=50K, likely due to being students, minors, or ineligible for full-time employment. Extremely few young adults (2.1%) exceed 50K, as most are early in their careers, pursuing education, or in entry-level jobs. For the 26-35 age group, there's a noticeable improvement — roughly 1 in 5 individuals in this group earn >50K, reflecting early career progression and accumulation of qualifications/experience. A substantial income increase is seen in the 36-45 age group: over a third now earn >50K. This is typically considered prime earning age where individuals settle into stable, higher-paying positions. Highest proportion of >50K earners is seen amongst individuals aged between 46 and 60— nearly 4 in 10. This reflects career maturity, peak seniority levels, and accumulated experience. There's a drop-off in high incomes as many transition to retirement, part-time, or less demanding roles in the age group 61-75. Yet about 1 in 4 still earn >50K. Most in 76+ age group earn <=50K, likely due to retirement, pensions, or fixed incomes — but a small minority still earn higher incomes, possibly through continued work or investments.

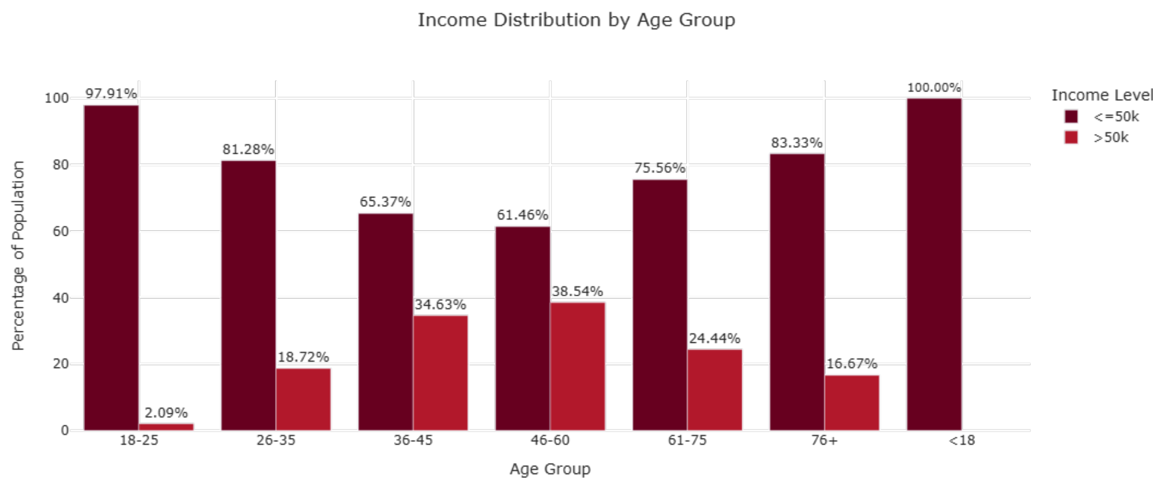
```python
themes = ["plotly", "plotly_white", "plotly_dark", "ggplot2", "seaborn",
          "simple_white", "presentation", "gridon", "gridon", "none"]


for theme in themes:
    fig.update_layout(template=theme)
    fig.show()
```

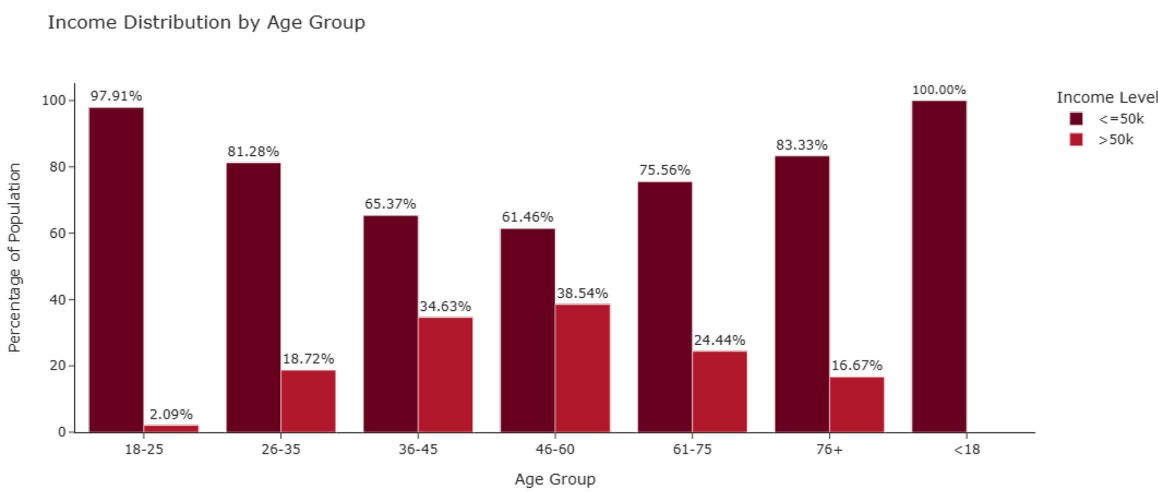

Income Distribution by Age Group
Income Level
<=50k
>50k
Percentage of Population
Age Group
18-25
26-35
36-45
46-60
61-75
76+
<18
97.91%
2.09%
81.28%
18.72%
65.37%
34.63%
61.46%
38.54%
75.56%
24.44%
83.33%
16.67%
100.00%

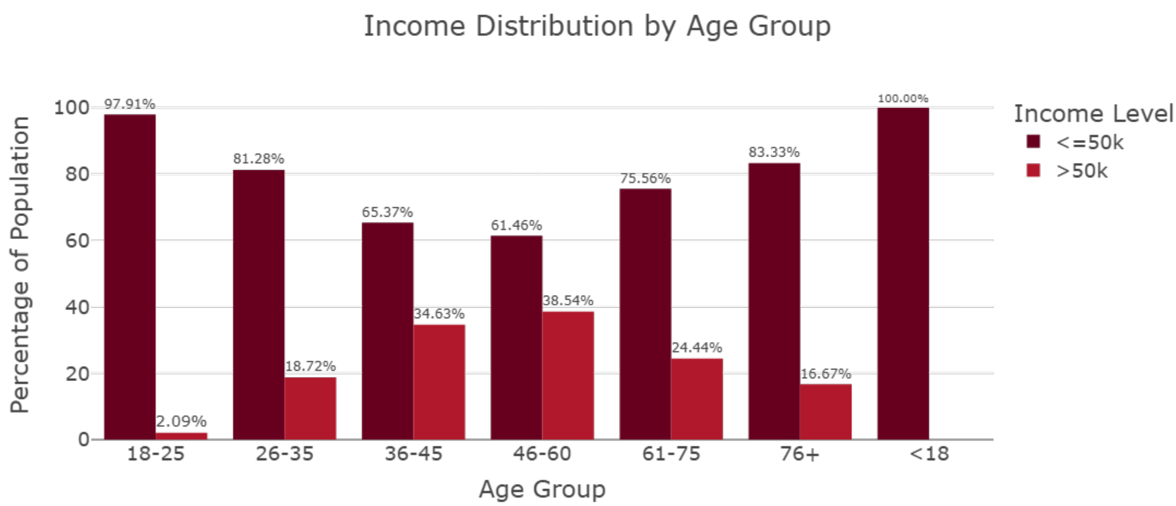

Income Distribution by Age Group
Income Level
<=50k
>50k
Percentage of Population
Age Group
18-25
26-35
36-45
46-60
61-75
76+
<18
97.91%
2.09%
81.28%
18.72%
65.37%
34.63%
61.46%
38.54%
75.56%
24.44%
83.33%
16.67%
100.00%

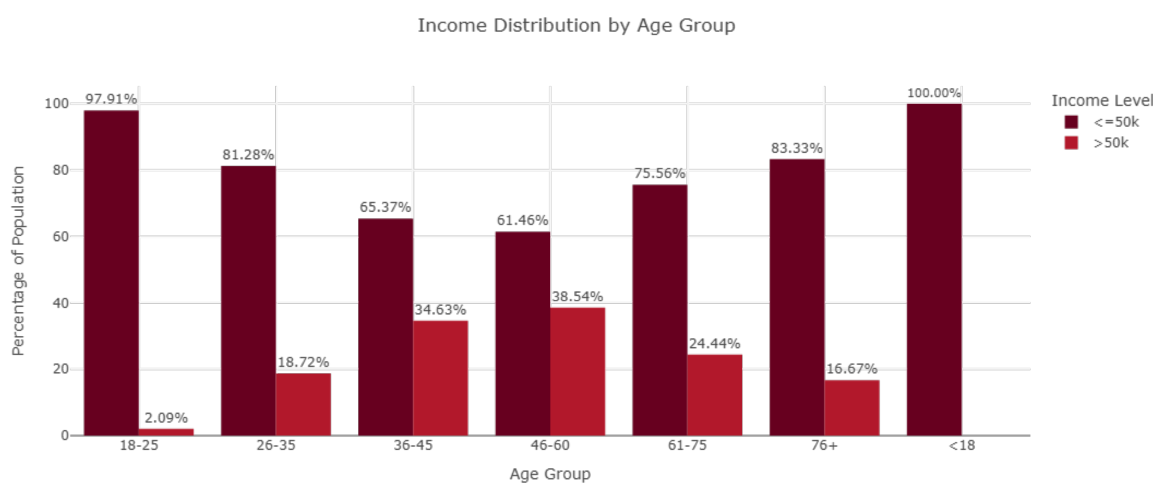

Income Distribution by Age Group
Income Level
<=50k
>50k
Percentage of Population
Age Group
18-25
26-35
36-45
46-60
61-75
76+
<18

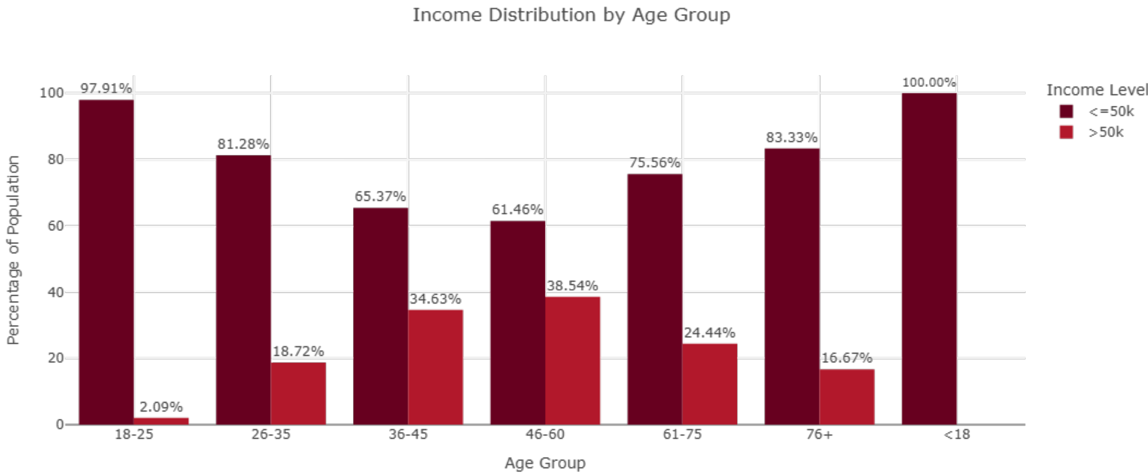

Income Distribution by Age Group
Income Level
<=50k
>50k
Percentage of Population
0
20
40
60
80
100
97.91%
2.09%
81.28%
18.72%
65.37%
34.63%
61.46%
38.54%
75.56%
24.44%
83.33%
16.67%
100.00%
18-25
26-35
36-45
46-60
61-75
76+
<18
Age Group

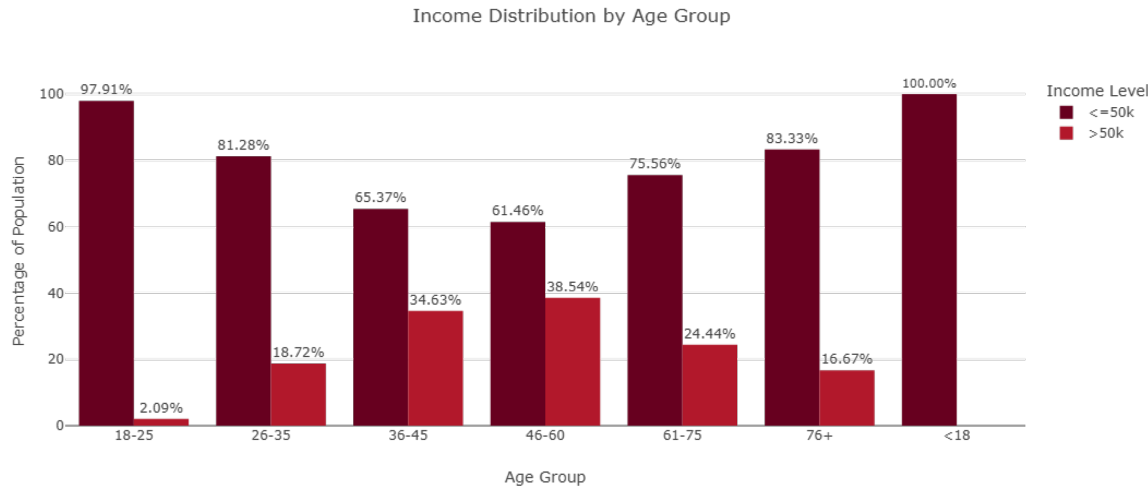

Income Distribution by Age Group
Income Level
<=50k
>50k
Percentage of Population
Age Group
18-25
26-35
36-45
46-60
61-75
76+
<18
97.91%
2.09%
81.28%
18.72%
65.37%
34.63%
61.46%
38.54%
75.56%
24.44%
83.33%
16.67%
100.00%


Income Distribution by Age Group
Income Level
<=50k
>50k
Percentage of Population
Age Group
18-25
26-35
36-45
46-60
61-75
76+
<18
97.91%
2.09%
81.28%
18.72%
65.37%
34.63%
61.46%
38.54%
75.56%
24.44%
83.33%
16.67%
100.00%


Income Distribution by Age Group
Percentage of Population
Age Group
Income Level
<=50k
>50k
18-25
26-35
36-45
46-60
61-75
76+
<18
97.91%
2.09%
81.28%
18.72%
65.37%
34.63%
61.46%
38.54%
75.56%
24.44%
83.33%
16.67%
100.00%


Income Distribution by Age Group
Percentage of Population
Age Group
Income Level
<=50k
>50k
18-25
26-35
36-45
46-60
61-75
76+
<18
97.91%
2.09%
81.28%
18.72%
65.37%
34.63%
61.46%
38.54%
75.56%
24.44%
83.33%
16.67%
100.00%





Income by Native region

```
adult_df_income_native_region = adult_df.groupby(['native_region', 'income']).size().reset_i
adult_df_income_native_region
```

	native_region	income	total_income_distr
6	north_america	<=50k	22768
7	north_america	>50k	7250

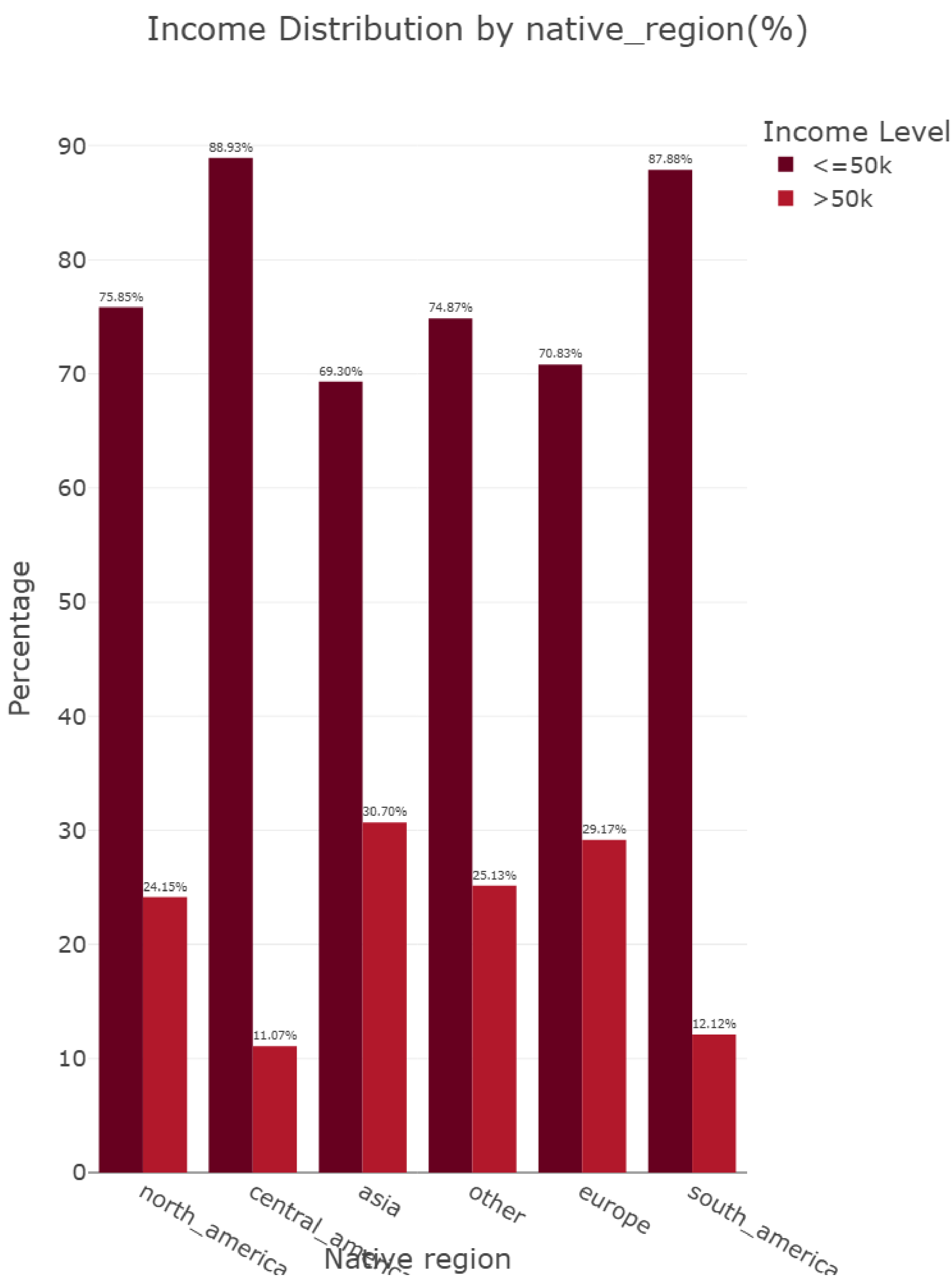
	native_region	income	total_income_distr
2	central_america	<=50k	466
0	asia	<=50k	465
8	other	<=50k	435
4	europe	<=50k	369
1	asia	>50k	206
10	south_america	<=50k	174
5	europe	>50k	152
9	other	>50k	146
3	central_america	>50k	58
11	south_america	>50k	24

```
total_per_region = adult_df_income_native_region.groupby('native_region')['total_income_dist
adult_df_income_native_region['percentage'] = (adult_df_income_native_region['total_income_di
adult_df_income_native_region
```

	native_region	income	total_income_distr	percentage
6	north_america	<=50k	22768	75.847825
7	north_america	>50k	7250	24.152175
2	central_america	<=50k	466	88.931298
0	asia	<=50k	465	69.299553
8	other	<=50k	435	74.870912
4	europe	<=50k	369	70.825336
1	asia	>50k	206	30.700447
10	south_america	<=50k	174	87.878788
5	europe	>50k	152	29.174664
9	other	>50k	146	25.129088
3	central_america	>50k	58	11.068702
11	south_america	>50k	24	12.121212

```
fig = px.bar(
    adult_df_income_native_region,
    x='native_region',
    y='percentage',
    color='income',
    title='Income Distribution by native_region(%)',
    barmode='group',
    color_discrete_sequence=px.colors.sequential.RdBu,
    text='percentage',
```

```
    width=900,
    height=1000
)

fig.update_traces(texttemplate='%{text:.2f}%', textposition='outside')

fig.update_layout(
    template="presentation",
    xaxis_title='Native region',
    yaxis_title='Percentage',
    legend_title_text='Income Level',
    paper_bgcolor="rgba(0,0,0,0)",
    plot_bgcolor="rgba(0,0,0,0)"
)

fig.show()

fig.write_image(os.path.join(results_dir, 'income_distribution_by_native_region_bar_plot.jpg
fig.write_image(os.path.join(results_dir, 'income_distribution_by_native_region_bar_plot.png
fig.write_html(os.path.join(results_dir, 'income_distribution_by_native_region_bar_plot.html
```


Income Distribution by native_region(%)
Income Level
<=50k
>50k
Percentage
Native region
north_america
central_america
asia
other
europe
south_america
75.85%
24.15%
88.93%
11.07%
69.30%
30.70%
74.87%
25.13%
70.83%
29.17%
87.88%
12.12%
0
10
20
30
40
50
60
70
80
90

Asia (30.7%) and Europe (29.2%) have the highest proportions of high-income earners. This suggests these immigrant groups might be better integrated into high-paying professional roles, or may represent a more skilled migrant profile in the dataset. Central America (11.1%) and South America (12.1%) have the lowest proportions of >50K earners. With 24.2% of North Americans earning >50K, this serves as a middle-ground baseline. Interestingly, both Asian and European groups outperform the native-born population proportionally in high-income brackets. The 'Other' group sits around 25.1%, close to North America's rate. This likely reflects a diverse mix of regions not explicitly listed.

Asian or Pacific Islander (26.6%) and White (25.6%) populations have the highest proportions of >50K earners. Asians/Pacific Islanders marginally outperform Whites, a pattern often attributed to occupational concentration in high-paying sectors like technology and medicine. On the other hand, American Indian or Eskimo (11.6%), Black (12.4%), and Other (9.2%) groups show significantly lower rates of high-income earners. These figures reflect long-standing economic disparities rooted in historical exclusion, occupational segregation, and systemic inequality.

The stark differences in high-income proportions:

- **Between Whites and Blacks**: 25.6% vs 12.4% — slightly over double the proportion.
- **Between Asians and Others**: 26.6% vs 9.2% — nearly triple.

These disparities are consistent with well-documented wage gaps and underrepresentation of marginalized groups in higher-paying roles.

Income by Race

```
adult_df_income_race = adult_df.groupby(['race', 'income']).size().reset_index(name='total_i
adult_df_income_race
```

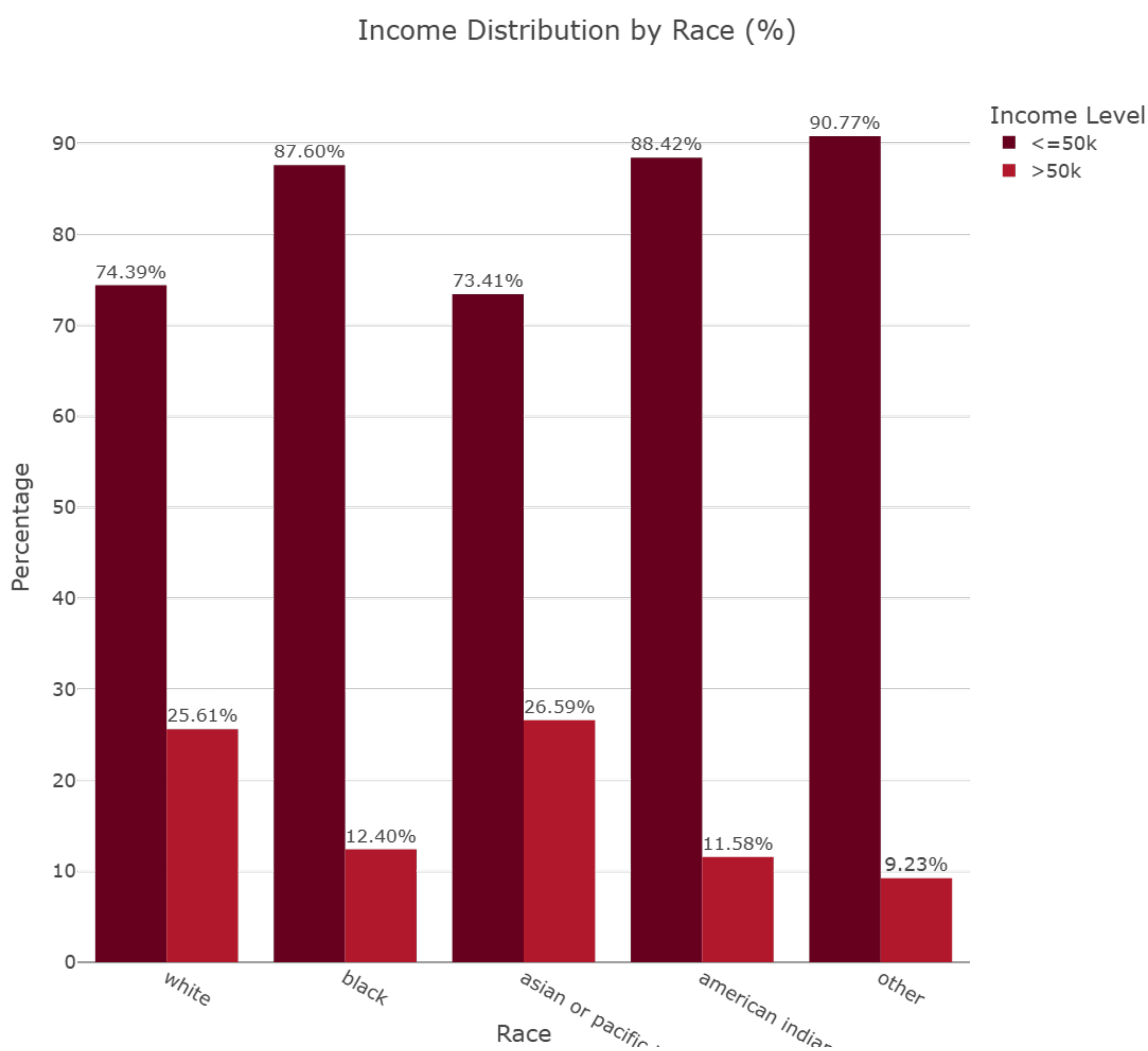
	race	income	total_income_distr
8	white	<=50k	20659
9	white	>50k	7112
4	black	<=50k	2735
2	asian or pacific islander	<=50k	762
5	black	>50k	387
3	asian or pacific islander	>50k	276
0	american indian or eskimo	<=50k	275
6	other	<=50k	246
1	american indian or eskimo	>50k	36
7	other	>50k	25

```
total_per_race = adult_df_income_race.groupby('race')['total_income_distr'].transform('sum')
adult_df_income_race['percentage'] = (adult_df_income_race['total_income_distr']/total_per_ra
adult_df_income_race
```

	race	income	total_income_distr	percentage
8	white	<=50k	20659	74.390551
9	white	>50k	7112	25.609449
4	black	<=50k	2735	87.604100
2	asian or pacific islander	<=50k	762	73.410405
5	black	>50k	387	12.395900
3	asian or pacific islander	>50k	276	26.589595
0	american indian or eskimo	<=50k	275	88.424437
6	other	<=50k	246	90.774908
1	american indian or eskimo	>50k	36	11.575563
7	other	>50k	25	9.225092

```
fig = px.bar(
    adult_df_income_race,
    x='race',
    y='percentage',
    color='income',
    title='Income Distribution by Race (%)',
    barmode='group',
    color_discrete_sequence=px.colors.sequential.RdBu,
    text='percentage',
    width=900,
    height=1000
)

fig.update_traces(texttemplate='%{text:.2f}%', textposition='outside')

fig.update_layout(
    template="presentation",
    xaxis_title='Race',
    yaxis_title='Percentage',
    legend_title_text='Income Level',
    paper_bgcolor="rgba(0,0,0,0)",
    plot_bgcolor="rgba(0,0,0,0)"
)
```

```
fig.show()

fig.write_image(os.path.join(results_dir, 'income_distribution_by_race_bar_plot.jpg'))
fig.write_image(os.path.join(results_dir, 'income_distribution_by_race_bar_plot.png'))
fig.write_html(os.path.join(results_dir, 'income_distribution_by_race_bar_plot.html'))
```




```
WARNING Thread(Thread-45 (run)) Task(Task-750) choreographer.browser_async:browser_async.py:
WARNING Thread(Thread-47 (run)) Task(Task-787) choreographer.browser_async:browser_async.py:
```

```
adult_df_income_edu_occ = adult_df.groupby(['education_level', 'occupation_grouped','income']
adult_df_income_edu_occ
```

	education_level	occupation_grouped	income	total
29	secondary-school graduate	blue collar	<=50k	3976
56	tertiary	white collar	>50k	3545
55	tertiary	white collar	<=50k	3369
45	some-college	white collar	<=50k	3003
36	secondary-school graduate	white collar	<=50k	2900
38	some-college	blue collar	<=50k	1503
32	secondary-school graduate	service	<=50k	1444
20	secondary	blue collar	<=50k	1349
6	associate	white collar	<=50k	1015
41	some-college	service	<=50k	902
46	some-college	white collar	>50k	858
30	secondary-school graduate	blue collar	>50k	796
37	secondary-school graduate	white collar	>50k	731
23	secondary	service	<=50k	663
12	primary	blue collar	<=50k	634
27	secondary	white collar	<=50k	552
34	secondary-school graduate	unknown	<=50k	487
0	associate	blue collar	<=50k	482
43	some-college	unknown	<=50k	481
39	some-college	blue collar	>50k	397
7	associate	white collar	>50k	397
47	tertiary	blue collar	<=50k	375
25	secondary	unknown	<=50k	307
14	primary	service	<=50k	243
2	associate	service	<=50k	237
51	tertiary	service	<=50k	232
48	tertiary	blue collar	>50k	183
53	tertiary	unknown	<=50k	172
1	associate	blue collar	>50k	166
21	secondary	blue collar	>50k	116
16	primary	unknown	<=50k	111
33	secondary-school graduate	service	>50k	100
52	tertiary	service	>50k	97
42	some-college	service	>50k	95
18	primary	white collar	<=50k	93
4	associate	unknown	<=50k	89
54	tertiary	unknown	>50k	82

	education_level	occupation_grouped	income	total
28	secondary	white collar	>50k	49
35	secondary-school graduate	unknown	>50k	46
3	associate	service	>50k	44
13	primary	blue collar	>50k	40
44	some-college	unknown	>50k	35
8	preschool	blue collar	<=50k	25
5	associate	unknown	>50k	19
9	preschool	service	<=50k	17
19	primary	white collar	>50k	17
24	secondary	service	>50k	12
10	preschool	unknown	<=50k	5
26	secondary	unknown	>50k	5
17	primary	unknown	>50k	4
31	secondary-school graduate	military	<=50k	4
11	preschool	white collar	<=50k	3
40	some-college	military	<=50k	2
49	tertiary	military	<=50k	1
50	tertiary	military	>50k	1
15	primary	service	>50k	1
22	secondary	military	<=50k	1

```
adult_df_income_edu_occ['edu_occ']=(adult_df_income_edu_occ['education_level']+"|")+adult_df_
adult_df_income_edu_occ
```

	education_level	occupation_grouped	income	total	edu_occ
29	secondary-school graduate	blue collar	<=50k	3976	secondary-school graduate|blue collar
56	tertiary	white collar	>50k	3545	tertiary|white collar
55	tertiary	white collar	<=50k	3369	tertiary|white collar
45	some-college	white collar	<=50k	3003	some-college|white collar
36	secondary-school graduate	white collar	<=50k	2900	secondary-school graduate|white colla
38	some-college	blue collar	<=50k	1503	some-college|blue collar
32	secondary-school graduate	service	<=50k	1444	secondary-school graduate|service
20	secondary	blue collar	<=50k	1349	secondary|blue collar
6	associate	white collar	<=50k	1015	associate|white collar
41	some-college	service	<=50k	902	some-college|service
46	some-college	white collar	>50k	858	some-college|white collar
30	secondary-school graduate	blue collar	>50k	796	secondary-school graduate|blue collar
37	secondary-school graduate	white collar	>50k	731	secondary-school graduate|white colla
23	secondary	service	<=50k	663	secondary|service

	education_level	occupation_grouped	income	total	edu_occ
12	primary	blue collar	<=50k	634	primary|blue collar
27	secondary	white collar	<=50k	552	secondary|white collar
34	secondary-school graduate	unknown	<=50k	487	secondary-school graduate|unknown
0	associate	blue collar	<=50k	482	associate|blue collar
43	some-college	unknown	<=50k	481	some-college|unknown
39	some-college	blue collar	>50k	397	some-college|blue collar
7	associate	white collar	>50k	397	associate|white collar
47	tertiary	blue collar	<=50k	375	tertiary|blue collar
25	secondary	unknown	<=50k	307	secondary|unknown
14	primary	service	<=50k	243	primary|service
2	associate	service	<=50k	237	associate|service
51	tertiary	service	<=50k	232	tertiary|service
48	tertiary	blue collar	>50k	183	tertiary|blue collar
53	tertiary	unknown	<=50k	172	tertiary|unknown
1	associate	blue collar	>50k	166	associate|blue collar
21	secondary	blue collar	>50k	116	secondary|blue collar
16	primary	unknown	<=50k	111	primary|unknown
33	secondary-school graduate	service	>50k	100	secondary-school graduate|service
52	tertiary	service	>50k	97	tertiary|service
42	some-college	service	>50k	95	some-college|service
18	primary	white collar	<=50k	93	primary|white collar
4	associate	unknown	<=50k	89	associate|unknown
54	tertiary	unknown	>50k	82	tertiary|unknown
28	secondary	white collar	>50k	49	secondary|white collar
35	secondary-school graduate	unknown	>50k	46	secondary-school graduate|unknown
3	associate	service	>50k	44	associate|service
13	primary	blue collar	>50k	40	primary|blue collar
44	some-college	unknown	>50k	35	some-college|unknown
8	preschool	blue collar	<=50k	25	preschool|blue collar
5	associate	unknown	>50k	19	associate|unknown
9	preschool	service	<=50k	17	preschool|service
19	primary	white collar	>50k	17	primary|white collar
24	secondary	service	>50k	12	secondary|service
10	preschool	unknown	<=50k	5	preschool|unknown
26	secondary	unknown	>50k	5	secondary|unknown
17	primary	unknown	>50k	4	primary|unknown
31	secondary-school graduate	military	<=50k	4	secondary-school graduate|military
11	preschool	white collar	<=50k	3	preschool|white collar
40	some-college	military	<=50k	2	some-college|military
49	tertiary	military	<=50k	1	tertiary|military
50	tertiary	military	>50k	1	tertiary|military

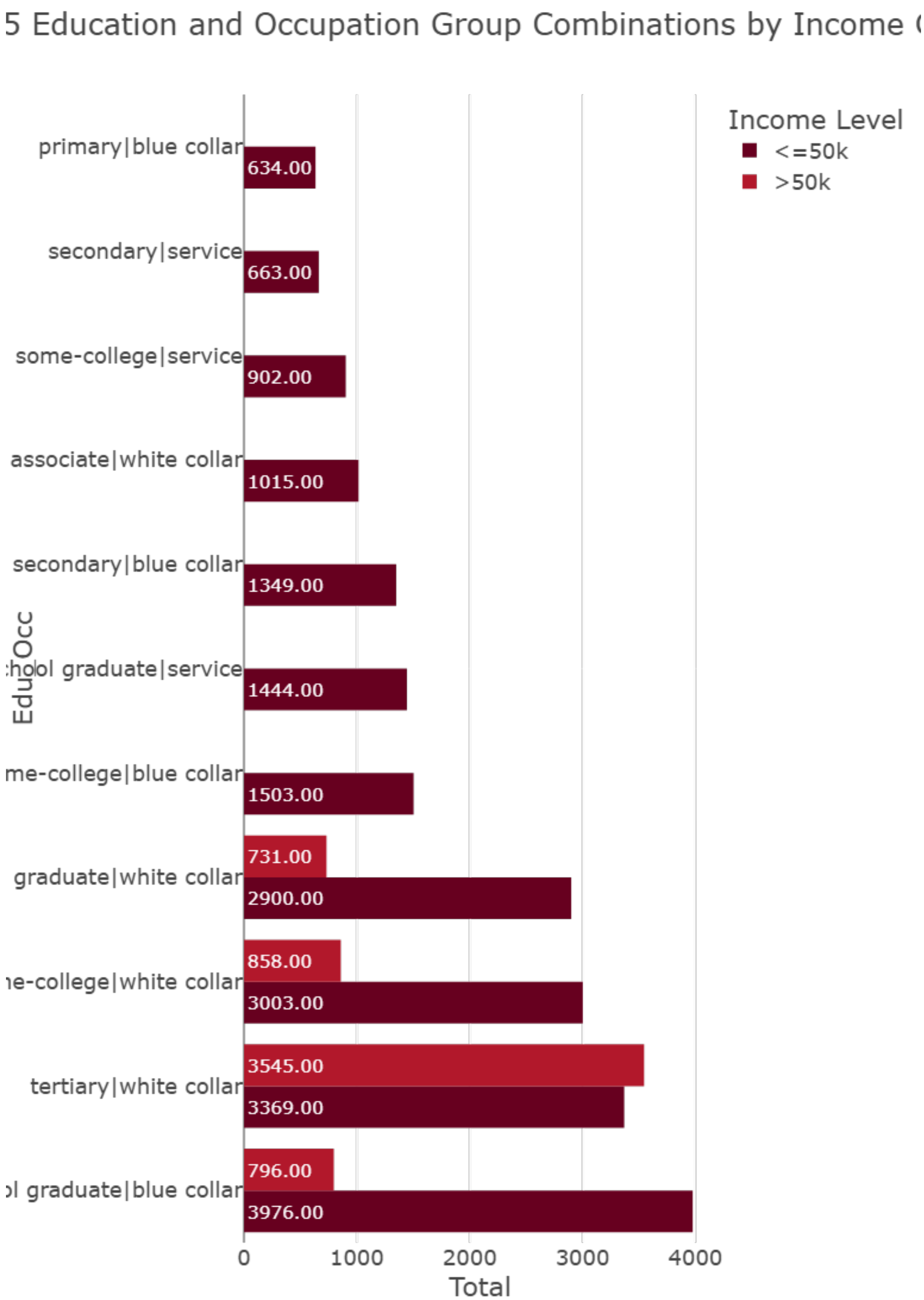
<table>
<tr><th></th><th>education_level</th><th>occupation_grouped</th><th>income</th><th>total</th><th>edu_occ</th></tr>
<tr><td>15</td><td>primary</td><td>service</td><td>>50k</td><td>1</td><td>primary|service</td></tr>
<tr><td>22</td><td>secondary</td><td>military</td><td><=50k</td><td>1</td><td>secondary|military</td></tr>
</table>

```
adult_df_income_edu_occ.head(15)
```

<table>
<tr><th></th><th>education_level</th><th>occupation_grouped</th><th>income</th><th>total</th><th>edu_occ</th></tr>
<tr><td>29</td><td>secondary-school graduate</td><td>blue collar</td><td><=50k</td><td>3976</td><td>secondary-school graduate|blue collar</td></tr>
<tr><td>56</td><td>tertiary</td><td>white collar</td><td>>50k</td><td>3545</td><td>tertiary|white collar</td></tr>
<tr><td>55</td><td>tertiary</td><td>white collar</td><td><=50k</td><td>3369</td><td>tertiary|white collar</td></tr>
<tr><td>45</td><td>some-college</td><td>white collar</td><td><=50k</td><td>3003</td><td>some-college|white collar</td></tr>
<tr><td>36</td><td>secondary-school graduate</td><td>white collar</td><td><=50k</td><td>2900</td><td>secondary-school graduate|white colla</td></tr>
<tr><td>38</td><td>some-college</td><td>blue collar</td><td><=50k</td><td>1503</td><td>some-college|blue collar</td></tr>
<tr><td>32</td><td>secondary-school graduate</td><td>service</td><td><=50k</td><td>1444</td><td>secondary-school graduate|service</td></tr>
<tr><td>20</td><td>secondary</td><td>blue collar</td><td><=50k</td><td>1349</td><td>secondary|blue collar</td></tr>
<tr><td>6</td><td>associate</td><td>white collar</td><td><=50k</td><td>1015</td><td>associate|white collar</td></tr>
<tr><td>41</td><td>some-college</td><td>service</td><td><=50k</td><td>902</td><td>some-college|service</td></tr>
<tr><td>46</td><td>some-college</td><td>white collar</td><td>>50k</td><td>858</td><td>some-college|white collar</td></tr>
<tr><td>30</td><td>secondary-school graduate</td><td>blue collar</td><td>>50k</td><td>796</td><td>secondary-school graduate|blue collar</td></tr>
<tr><td>37</td><td>secondary-school graduate</td><td>white collar</td><td>>50k</td><td>731</td><td>secondary-school graduate|white colla</td></tr>
<tr><td>23</td><td>secondary</td><td>service</td><td><=50k</td><td>663</td><td>secondary|service</td></tr>
<tr><td>12</td><td>primary</td><td>blue collar</td><td><=50k</td><td>634</td><td>primary|blue collar</td></tr>
</table>

```
num = 15
adult_df_combos = adult_df_income_edu_occ.head(num)

fig = px.bar(
    data_frame=adult_df_combos,
    x='total',
    y='edu_occ',
    color='income',
    orientation='h',
    title='Top 15 Education and Occupation Group Combinations by Income Group',
    barmode='group',
    color_discrete_sequence=px.colors.sequential.RdBu,
    text='total',
    width=1000,
    height=1100
)
```

```
fig.update_traces(
    texttemplate='%{text:.2f}',
    textposition='inside',
    insidetextanchor='start'
)

fig.update_layout(
    template="presentation",
    xaxis_title='Total',
    yaxis_title='Edu_Occ',
    legend_title_text='Income Level',
    uniformtext_minsize=10,
    uniformtext_mode='hide',
    margin=dict(l=200, r=50, t=100, b=50),
    paper_bgcolor="rgba(0,0,0,0)",
    plot_bgcolor="rgba(0,0,0,0)"
)

fig.show()

fig.write_image(os.path.join(results_dir, 'Top 15 Education and Occupation Group Combinations
fig.write_image(os.path.join(results_dir, 'Top 15 Education and Occupation Group Combinations
fig.write_html(os.path.join(results_dir, 'Top 15 Education and Occupation Group Combinations
```


5 Education and Occupation Group Combinations by Income (
Income Level
<=50k
>50k
primary|blue collar
634.00
secondary|service
663.00
some-college|service
902.00
associate|white collar
1015.00
secondary|blue collar
1349.00
hool graduate|service
1444.00
me-college|blue collar
1503.00
graduate|white collar
731.00
2900.00
ne-college|white collar
858.00
3003.00
tertiary|white collar
3545.00
3369.00
ol graduate|blue collar
796.00
3976.00
0
1000
2000
3000
4000
Total
EduOcc

```
WARNING Thread(Thread-113 (run)) Task(Task-2025) choreographer.browser_async:browser_async.py
WARNING Thread(Thread-115 (run)) Task(Task-2061) choreographer.browser_async:browser_async.py
```

From the bar chart, we can pick out the largest groups per income-level. We see that secondary-school graduates working a blue collar job occupy the largest group in the dataset (3976). This reflects a common socio-economic profile: individuals with basic schooling in manual or technical trades predominantly earning lower incomes. The largest high-income group are tertiary-educated individuals in white collar roles. This highlights the strong earning advantage conferred by higher education and skilled jobs.

Some of the key patterns we can get from the dataset are:

- **Education matters, but isn't deterministic**

Tertiary education combined with white-collar work offers the highest income prospects. Yet a substantial number of tertiary-educated white-collar workers earn <=50K, likely early career, part-time, or structural pay gaps.

- **Blue-collar and service work predominantly pay <=50K, regardless of education.**

Even some college education doesn't guarantee high incomes in these sectors. Manual and service sector income is highly occupation-dependent (some skilled trades can break the 50K mark).

- **Some non-tertiary education groups do reach >50K**

Secondary-school graduates in blue-collar and white-collar work have decent representation among >50K earners. This reflects upward mobility possible through skilled trades, tenure, or niche role