In [1]:
```python
# importing some of the main packages we'll use in this class
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```python
import warnings
# Suppress unwanted warning messages.
warnings.filterwarnings("ignore")
```

In [3]:
```python
df = pd.read_csv('Mroz.csv') # print the data set
df
```

Out[3]:

	Unnamed: 0	work	hoursw	child6	child618	agew	educw	hearnw	wagew	hoursh	ageh	educh	wageh	income	educwm	educwf	unemprate	city	experience
0	1	yes	1610	1	0	32	12	3.3540	2.65	2708	34	12	4.0288	16310	12	7	5.0	no	14
1	2	yes	1656	0	2	30	12	1.3889	2.65	2310	30	9	8.4416	21800	7	7	11.0	yes	5
2	3	yes	1980	1	3	35	12	4.5455	4.04	3072	40	12	3.5807	21040	12	7	5.0	no	15
3	4	yes	456	0	3	34	12	1.0965	3.25	1920	53	10	3.5417	7300	7	7	5.0	no	6
4	5	yes	1568	1	2	31	14	4.5918	3.60	2000	32	12	10.0000	27300	12	14	9.5	yes	7
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
748	749	no	0	0	2	40	13	0.0000	0.00	3020	43	16	9.2715	28200	10	10	9.5	yes	5
749	750	no	0	2	3	31	12	0.0000	0.00	2056	33	12	4.8638	10000	12	12	7.5	no	14
750	751	no	0	0	0	43	12	0.0000	0.00	2383	43	12	1.0898	9952	10	3	7.5	no	4
751	752	no	0	0	0	60	12	0.0000	0.00	1705	55	8	12.4400	24984	12	12	14.0	yes	15
752	753	no	0	0	3	39	9	0.0000	0.00	3120	48	12	6.0897	28363	7	7	11.0	yes	12

753 rows × 19 columns

Part 1: Choose a dataset

1. The dataset we chose is: US Married Couples in 1976.
2. The dataset contains information about families units and economic factors, including whether the wife is working in 1975. The hours worked by the wife, husband, number of children in different age ranges, age of the parnets and their educations, earnings, family income, and more.
3. Features in the dataset include: The dataset consists of 17 features. Among these, there are 10 numeric variables and 7 categorical variables.
4. Number of records in the dataset: 753.

In [4]:
```python
# List of the features and their types
features_and_types = df.dtypes

# List of the number of records in the dataset
num_records = len(df)

print("Features and their types:")
print(features_and_types)

print("\nNumber of records in the dataset:", num_records)
```

```
Features and their types:
Unnamed: 0      int64
work           object
hoursw          int64
child6          int64
child618        int64
agew            int64
educw           int64
hearnw        float64
wagew         float64
hoursh          int64
ageh            int64
educh           int64
wageh         float64
income          int64
educwm          int64
educwf          int64
unemprate     float64
city           object
experience      int64
dtype: object

Number of records in the dataset: 753
```

Part 2: Exploratory data analysis:

1.

- In this section, our exploration delves into various key factors within the family, with a particular emphasis on the wife's educational. We examine aspects such as the wife's work hours, the number of children in the household, her age, educational attainment, average hourly earnings, wage, family income, the educational backgrounds of her parents, and her previous work experience.

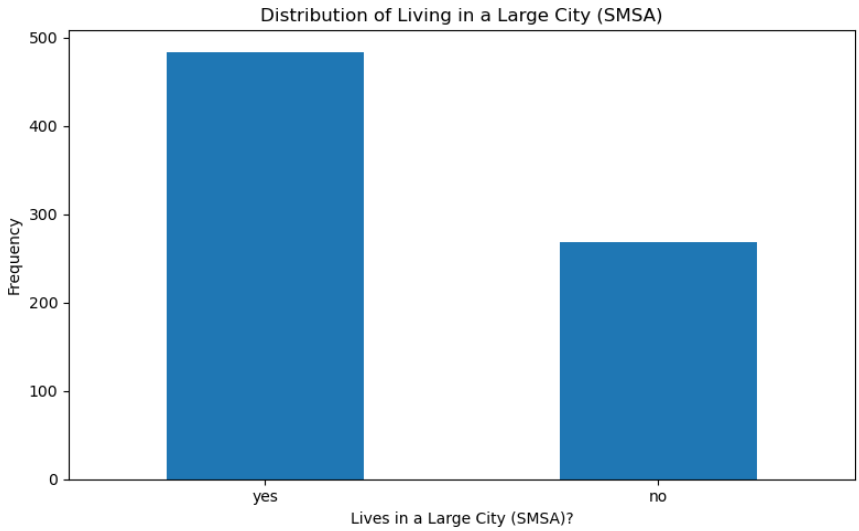
In [5]:
```python
# Count the frequency of each category
work_at_home_counts = df['work'].value_counts()

# Creating a bar plot for the distribution of "Work at home in 1975?"
plt.figure(figsize=(8, 5))
work_at_home_counts.plot(kind='bar')
plt.xlabel("is the wife working in 1975?")
plt.ylabel("Frequency")
plt.title("Distribution of Working women in 1975")
plt.xticks(rotation=0)
plt.tight_layout()
plt.show()
```

- The phrase "Work at home in 1975? (Same as labor force participation)" refers to the variable in the dataset that indicates whether a person engaged in any work-related activities from their home in the year 1975. It's directly tied to the concept of labor force participation, indicating whether an individual was actively involved in work while being at home during that particular year.

In [6]:
```
# Count the number of working women and non-working women
working_women_count = df[df['work'] == 'yes']['work'].count()
non_working_women_count = df[df['work'] == 'no']['work'].count()

# Calculate the rate of working women
total_women = len(df)
working_women_rate = (working_women_count / total_women) * 100

# Display the results
print(f"Number of Working Women: {working_women_count}")
print(f"Number of Non-Working Women: {non_working_women_count}")
print(f"Rate of Working Women (%): {working_women_rate:.2f}%")
```

```
Number of Working Women: 428
Number of Non-Working Women: 325
Rate of Working Women (%): 56.84%
```

In [7]:
```
# Count the frequency of each category
city_counts = df['city'].value_counts()

# Creating a bar plot for the distribution of "Lives in a large city (SMSA)?"
plt.figure(figsize=(8, 5))
city_counts.plot(kind='bar')
plt.xlabel("Lives in a Large City (SMSA)?")
plt.ylabel("Frequency")
plt.title("Distribution of Living in a Large City (SMSA)")
plt.xticks(rotation=0)
plt.tight_layout()
plt.show()
```

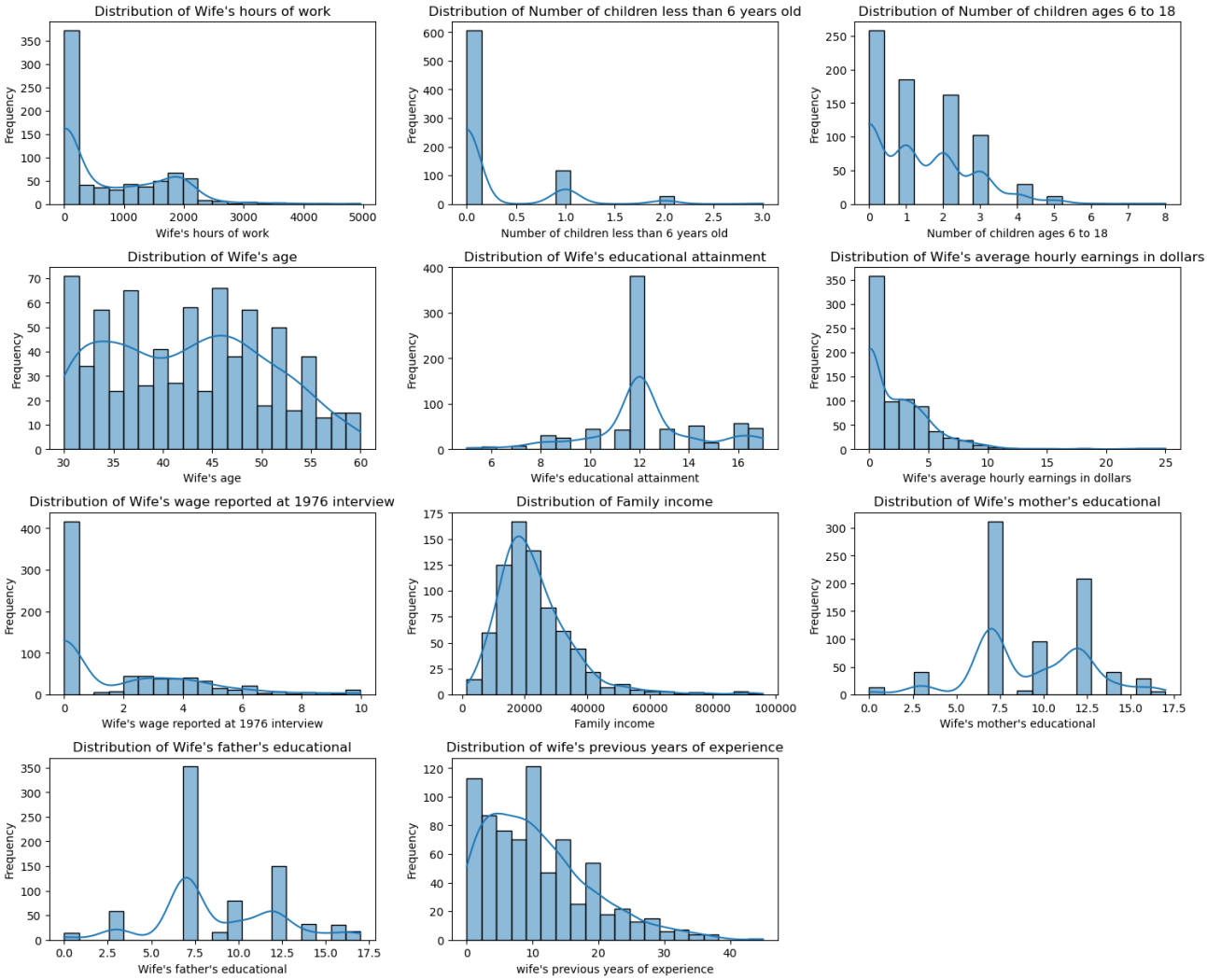


In [8]:
```
# Count the number of people living in a large city and not living in a large city
large_city_count = df[df['city'] == 'yes']['city'].count()
non_large_city_count = df[df['city'] == 'no']['city'].count()

# Calculate the rate of living in a large city
total_people = len(df)
large_city_rate = (large_city_count / total_people) * 100

# Display the results
print(f"Number of People Living in a Large City: {large_city_count}")
print(f"Number of People Not Living in a Large City: {non_large_city_count}")
print(f"Rate of Living in a Large City (%): {large_city_rate:.2f}%")
```

```
Number of People Living in a Large City: 484
Number of People Not Living in a Large City: 269
Rate of Living in a Large City (%): 64.28%
```

In [9]:
```
# Selected variables and their corresponding meanings
selected_vars = ["hoursw", "child6", "child618", "agew", "educw", "hearnw", "wagew",
```

```
                 "income", "educwm", "educwf", "experience"]
meaning_vars = ["Wife's hours of work", "Number of children less than 6 years old",
                "Number of children ages 6 to 18", "Wife's age", "Wife's educational attainment",
                "Wife's average hourly earnings in dollars", "Wife's wage reported at 1976 interview",
                "Family income", "Wife's mother's educational",
                "Wife's father's educational",
                "wife's previous years of experience"]

# Creating distribution plots for selected variables
plt.figure(figsize=(15, 18))

for i, var in enumerate(selected_vars):
    plt.subplot(6, 3, i + 1)
    sns.histplot(data=df, x=var, bins=20, kde=True)
    plt.xlabel(meaning_vars[i])
    plt.ylabel("Frequency")
    plt.title(f"Distribution of {meaning_vars[i]}")

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



Some interesting points we found:

- More than half of the women seems not to be working.
- There are more families in the data living in large cities rather than in small cities
- Wife's educational attainment seems to be mainly centered around 12 years of education, When all other options are relatively low.
- Wife's perantes educational attainment seems to be mainly centered around 7.5 and 12.5 years of education, When all other options are relatively low.
- The graph for family income is normally distributed around 20000.
- Most families do not have children under 6 years old.

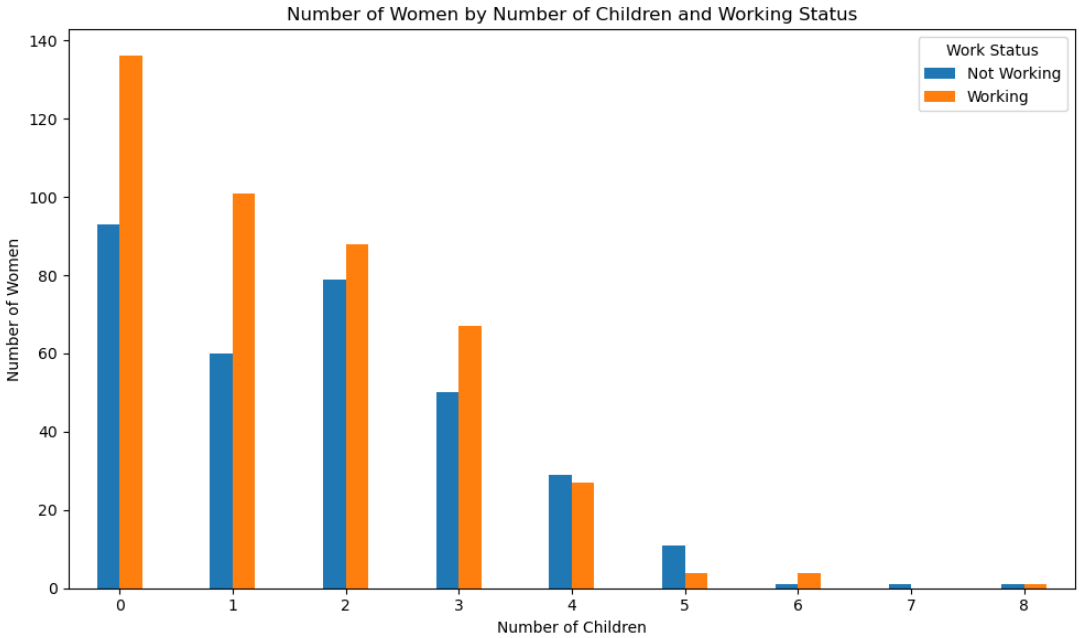
2.

- In this section we will Show plots illustrating bivariate relationships for at least 2 pairs of variables.

Number of Women by Number of Children and Working Status

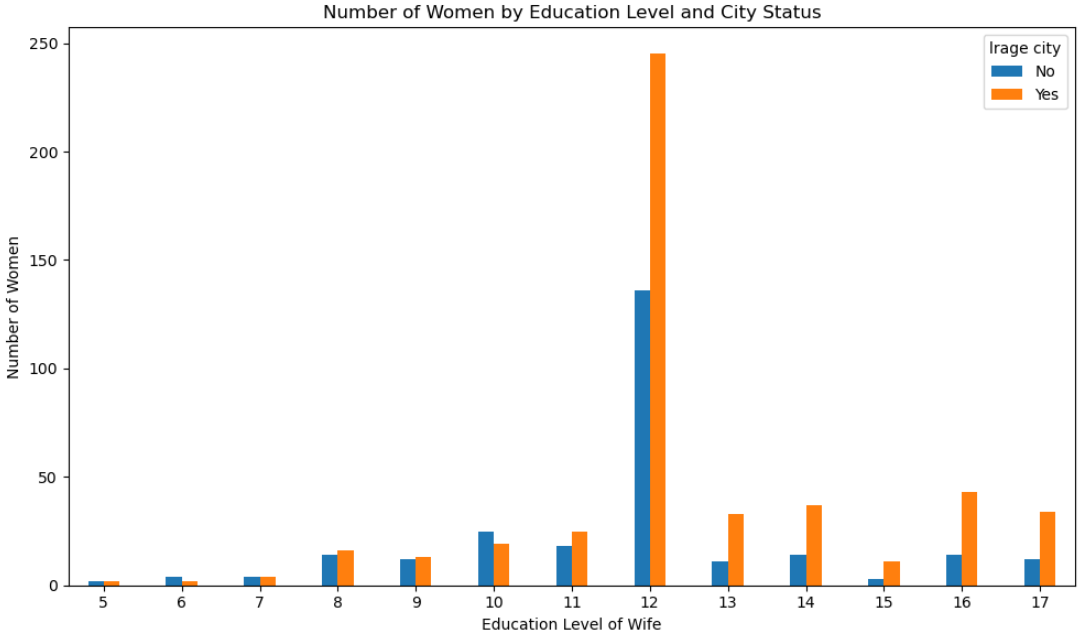
In [10]:
```
# making a copy dataframe
df_copy = df.copy()

# Combine the two separate child categories
df_copy['total_children'] = df_copy['child6'] + df_copy['child618']
df_copy.drop(['child6', 'child618'], axis='columns', inplace=True)

# Remove the 'Unnamed: 0' column because it's not relevant
df_copy.drop('Unnamed: 0', axis='columns', inplace=True)

# Mapping dictionary to convert 'yes' to 1(large city, and working wife)
#and 'no' to 0 (living not in a large city, and the woman is not working)
mapping = {'yes': 1, 'no': 0}
df_copy['city'] = df_copy['city'].map(mapping)
df_copy['work'] = df_copy['work'].map(mapping)
```

In [11]:
```
# Grouping the data and calculating the count of working women
graph = df_copy.groupby(['total_children', 'work']).size().unstack()

# Creating the grouped bar plot for the count of working women
ax = graph.plot(kind='bar', figsize=(10, 6), width=0.4)

# Adjusting plot labels and layout
plt.xlabel("Number of Children")
plt.ylabel("Number of Women")
plt.title("Number of Women by Number of Children and Working Status")
plt.xticks(rotation=0)
plt.legend(["Not Working", "Working"], title="Work Status")
plt.tight_layout()

# Displaying the plot
plt.show()
```



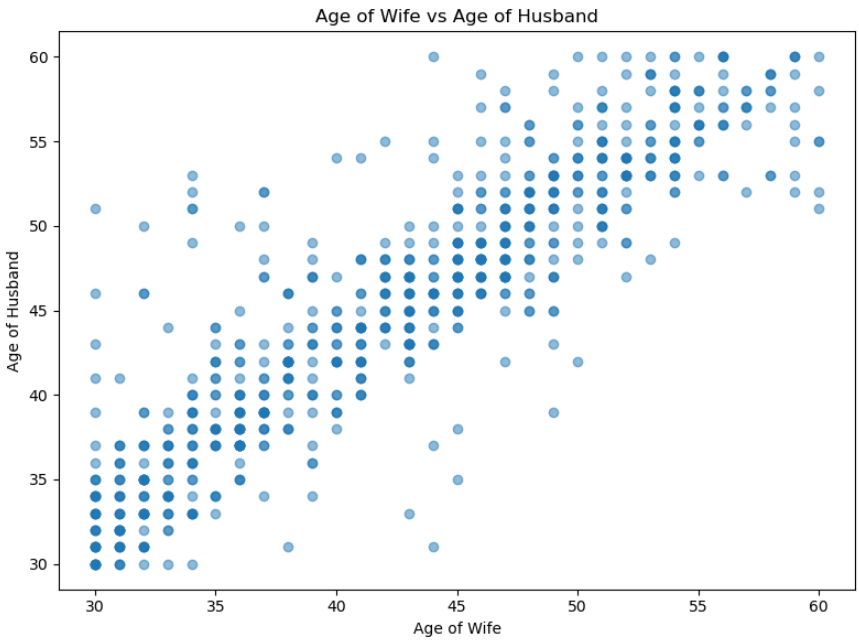
- We can observe from this plot that there is strong correlation between the numbers of childrens to the numbers of working woman.

In [12]:
```
# Grouping the data and calculating the count of working women
graph = df_copy.groupby(['educw', 'city']).size().unstack()

# Creating the grouped bar plot for the count of working women
ax = graph.plot(kind='bar', figsize=(10, 6), width=0.4)

# Adjusting plot labels and layout
plt.xlabel("Education Level of Wife")
plt.ylabel("Number of Women")
plt.title("Number of Women by Education Level and City Status")
plt.xticks(rotation=0)
plt.legend(["No", "Yes"], title="lrage city")

# Displaying the plot
plt.tight_layout()
plt.show()
```



We can observe from the graph that there is a significant correlation between living in a large city and the number of women with more than 12 years of education

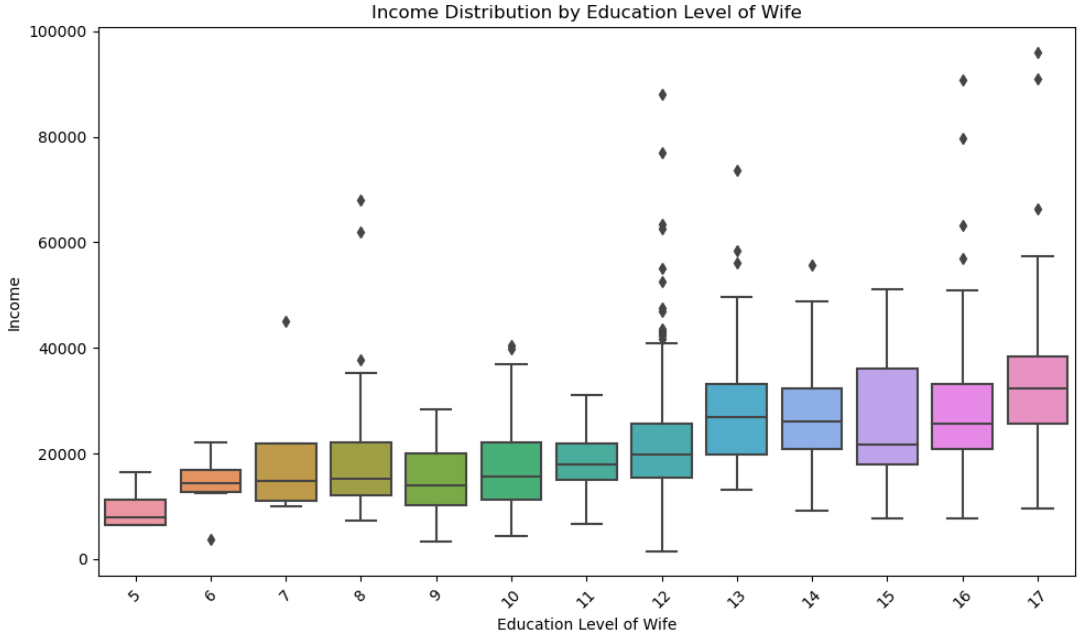
In [13]:
```
# Creating a scatter plot for the age of wife vs age of husband
plt.figure(figsize=(8, 6))
plt.scatter(df_copy['agew'], df_copy['ageh'], alpha=0.5)
plt.xlabel("Age of Wife")
plt.ylabel("Age of Husband")
plt.title("Age of Wife vs Age of Husband")
plt.tight_layout()
plt.show()
correlation_coefficient = df_copy['agew'].corr(df_copy['ageh'])
print("Correlation Coefficient:", correlation_coefficient)
```




```
Correlation Coefficient: 0.8881379748786074
```

Based on the calculated correlation coefficient of 0.888, we can observe a substantial positive correlation between the ages of the wives and the ages of their husbands. This correlation suggests that, in general, as the age of the wife increases, the age of the husband tends to increase as well.

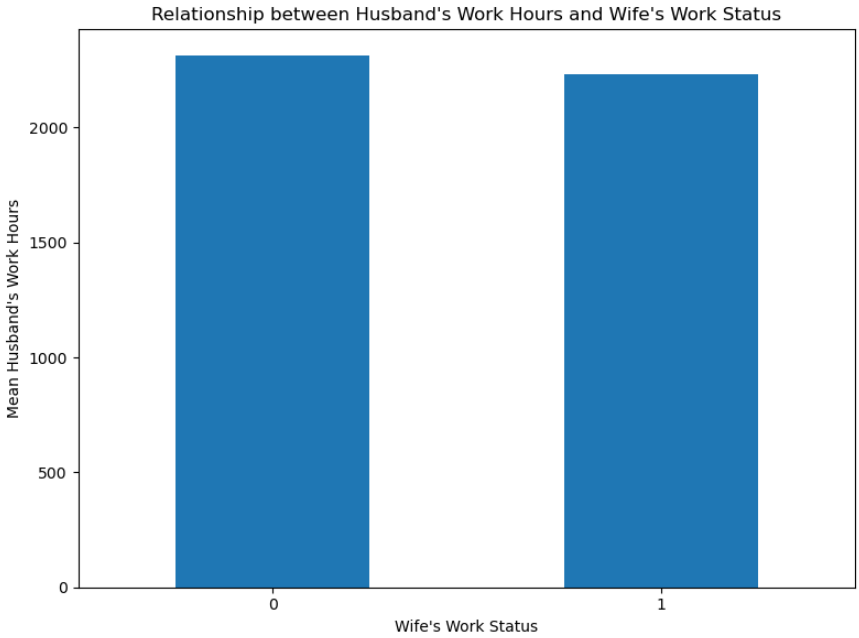
In [14]:
```
plt.figure(figsize=(10, 6))
sns.boxplot(data=df_copy, x='educw', y='income')
plt.xlabel("Education Level of Wife")
plt.ylabel("Income")
plt.title("Income Distribution by Education Level of Wife")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



From the graph, we can discern that there exists a correlation between the mean income of wives who possess more than 12 years of education and their income levels.

Relationship between Husband's Work Hours and Wife's Work Status

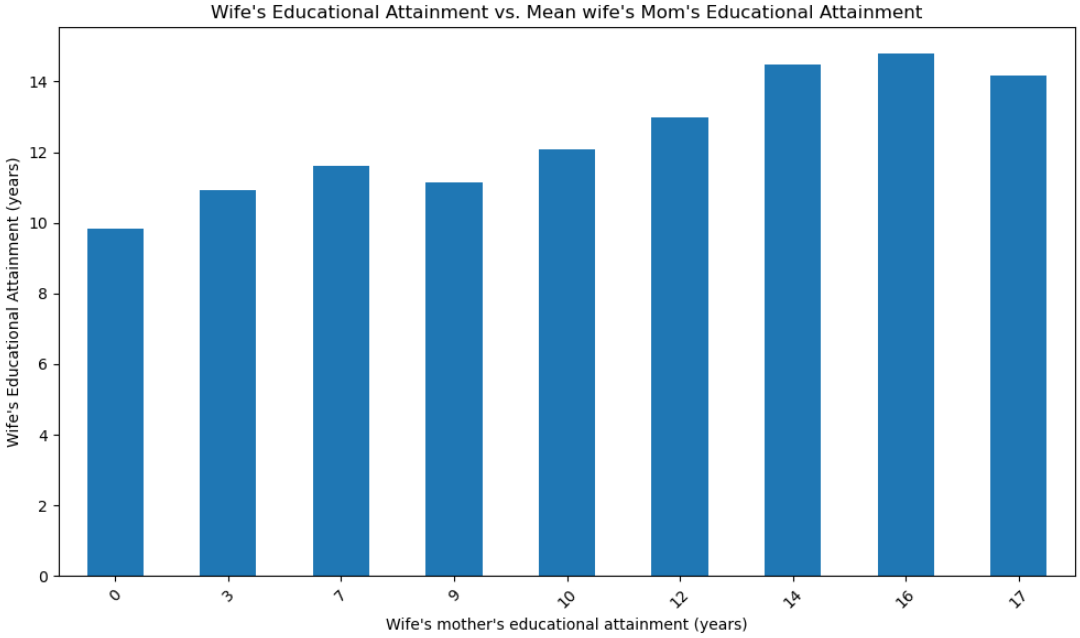
In [15]:
```
plt.figure(figsize=(8, 6))
df_copy.groupby("work")['hoursh'].mean().plot(kind='bar')  # Grouping and plotting mean work hours
plt.xlabel("Wife's Work Status")
plt.ylabel("Mean Husband's Work Hours")
plt.title("Relationship between Husband's Work Hours and Wife's Work Status")
plt.xticks(rotation=0)
plt.tight_layout()

# Displaying the plot
plt.show()
```



- We can observe from this plot that there is some week negative correlation between Mean wife's wage and wife's work experience.

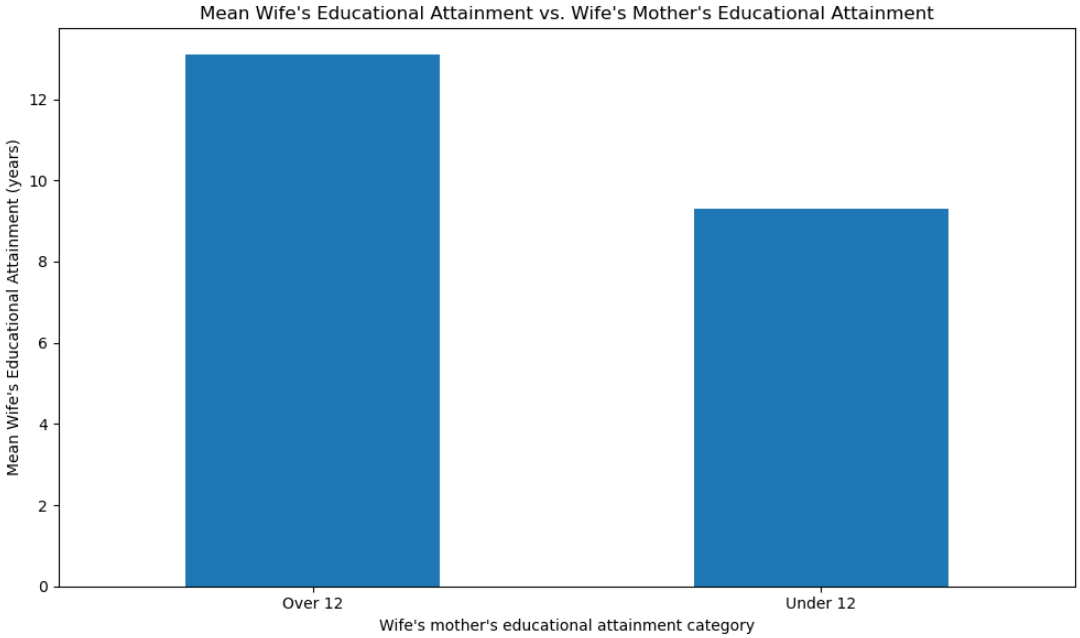
Mean wife's wage vs. Wife's mother's educational attainment (years).

In [16]:
```
plt.figure(figsize=(10, 6))
df.groupby("educwm")['educw'].mean().plot(kind='bar')  # Grouping and plotting mean educational attainment
plt.xlabel("Wife's mother's educational attainment (years)")
plt.ylabel("Wife's Educational Attainment (years)")
plt.title("Wife's Educational Attainment vs. Mean wife's Mom's Educational Attainment")
plt.xticks(rotation=45)
plt.tight_layout()

# Displaying the third plot
plt.show()
```



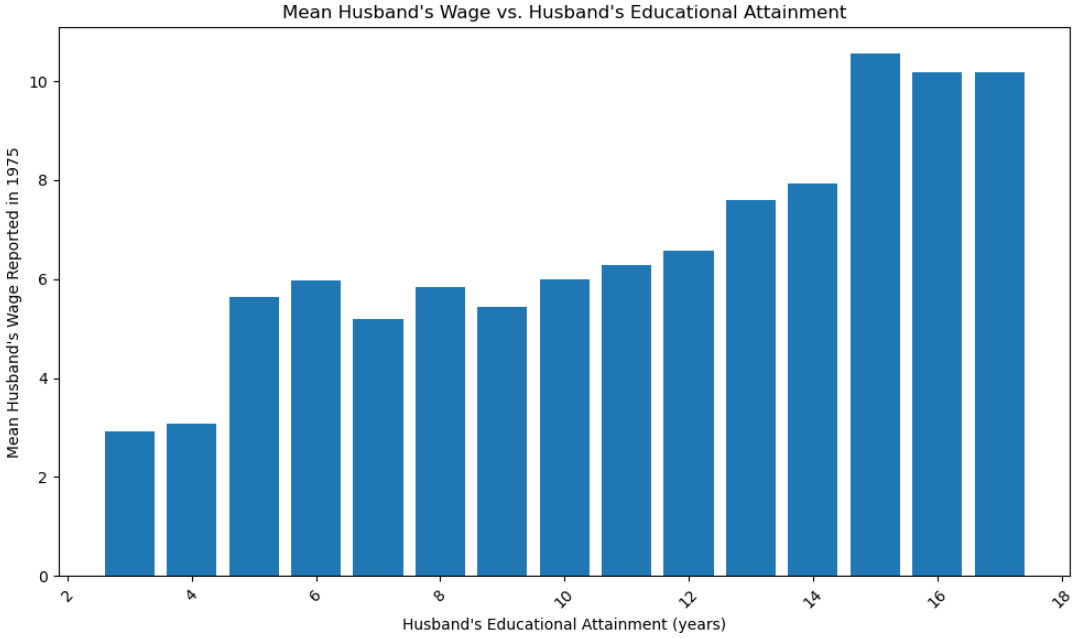
- We can observe from this plot that there is some relatively strong positive correlation between wife's wage and Wife's mother's educational attainment.

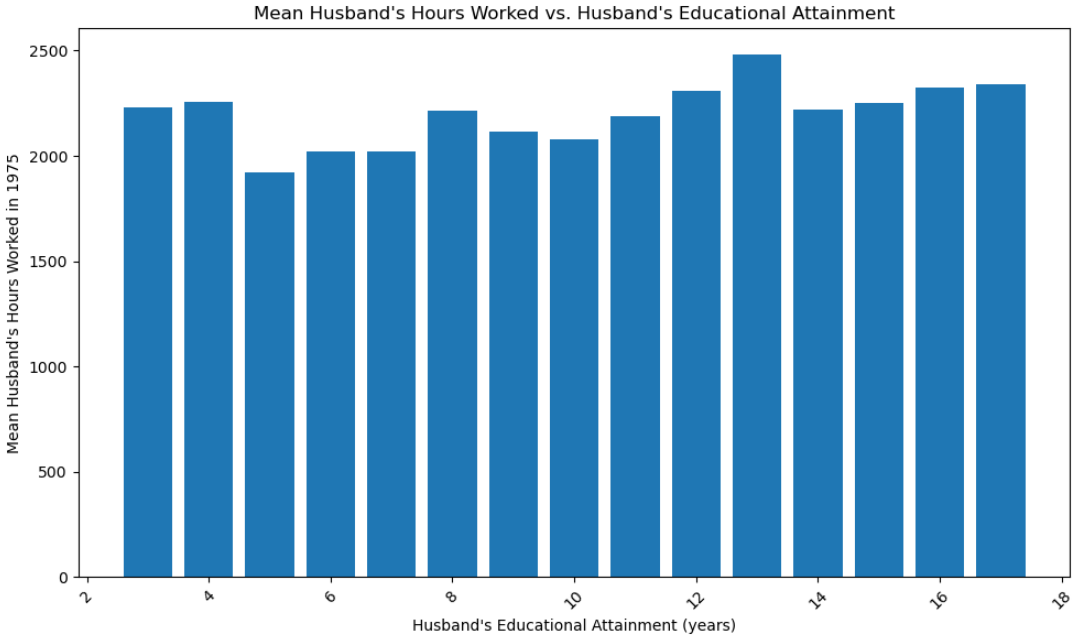
In [17]:
```
# Creating a new categorical column for Wife's mother's educational attainment
df_copy['educw_category'] = df_copy['educw'].apply(lambda x: 'Under 12' if x < 12 else 'Over 12')

# Grouping and calculating mean educational attainment
mean_educ_by_category = df_copy.groupby("educw_category")['educw'].mean()

# Mapping dictionary to convert 'yes' to 1 and 'no' to 0
mapping = {'Over 12': 1, 'Under 12': 0}
df_copy['educw_category'] = df_copy['educw_category'].map(mapping)



# Creating the bar plot
plt.figure(figsize=(10, 6))
mean_educ_by_category.plot(kind='bar')
plt.xlabel("Wife's mother's educational attainment category")
plt.ylabel("Mean Wife's Educational Attainment (years)")
plt.title("Mean Wife's Educational Attainment vs. Wife's Mother's Educational Attainment")
plt.xticks(rotation=0)
plt.tight_layout()

# Displaying the plot
plt.show()
```

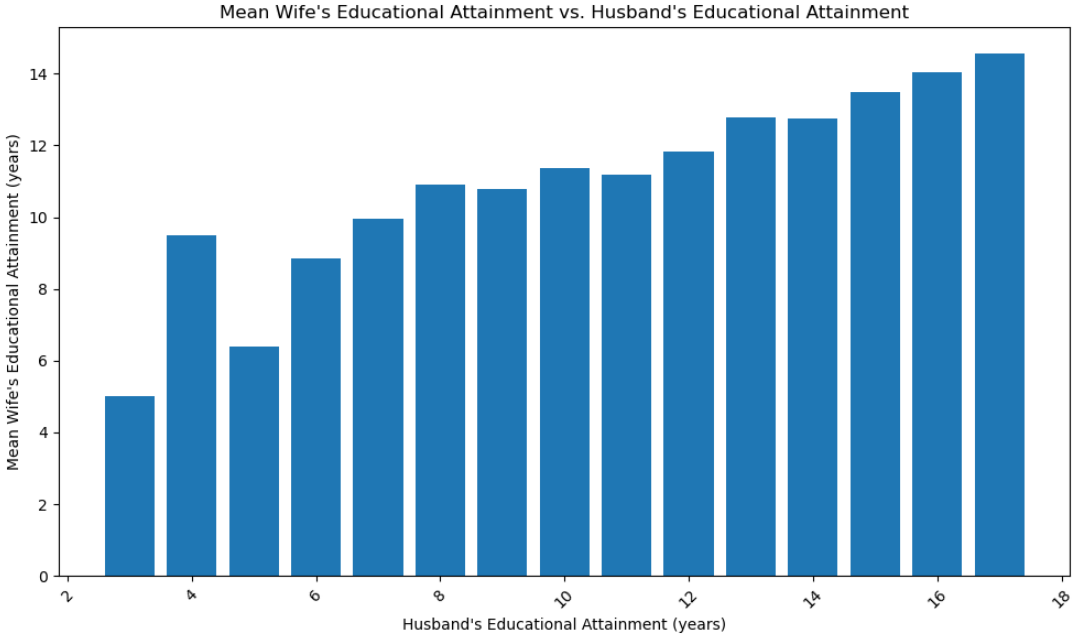


In [18]:
```
def create_bar_plot(x_values, y_values, x_label, y_label, title):
    plt.figure(figsize=(10, 6))
    plt.bar(x_values, y_values)
    plt.xlabel(x_label)
    plt.ylabel(y_label)
    plt.title(title)
    plt.xticks(rotation=45)
    plt.tight_layout()
    plt.show()

# Grouping the data by Husband's educational attainment and calculating the mean of wages, hours worked, and wife's mother's educational attainment
husband_graph1 = df.groupby('educh')['wageh'].mean()
husband_graph2 = df.groupby('educh')['hoursh'].mean()
husband_graph3 = df.groupby('educh')['educw'].mean()
```

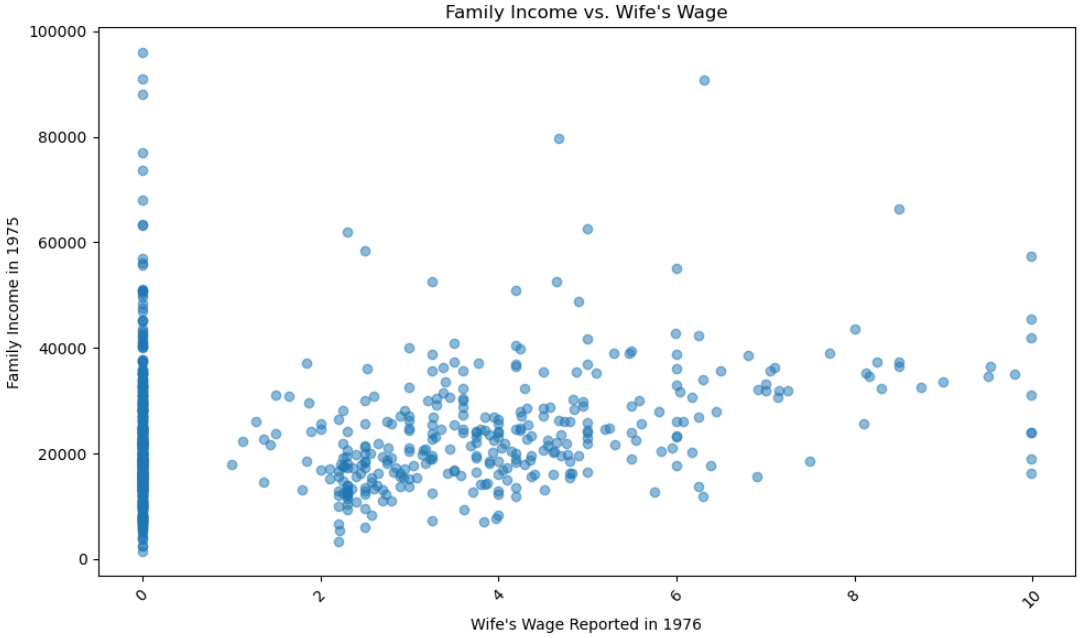
```python
# Creating bar plots
create_bar_plot(husband_graph1.index, husband_graph1, "Husband's Educational Attainment (years)",
                "Mean Husband's Wage Reported in 1975", "Mean Husband's Wage vs. Husband's Educational Attainment")
create_bar_plot(husband_graph2.index, husband_graph2, "Husband's Educational Attainment (years)",
                "Mean Husband's Hours Worked in 1975", "Mean Husband's Hours Worked vs. Husband's Educational Attainment")
create_bar_plot(husband_graph3.index, husband_graph3, "Husband's Educational Attainment (years)", "Mean Wife's Educational
                Attainment (years)", "Mean Wife's Educational Attainment vs. Husband's Educational Attainment")
```

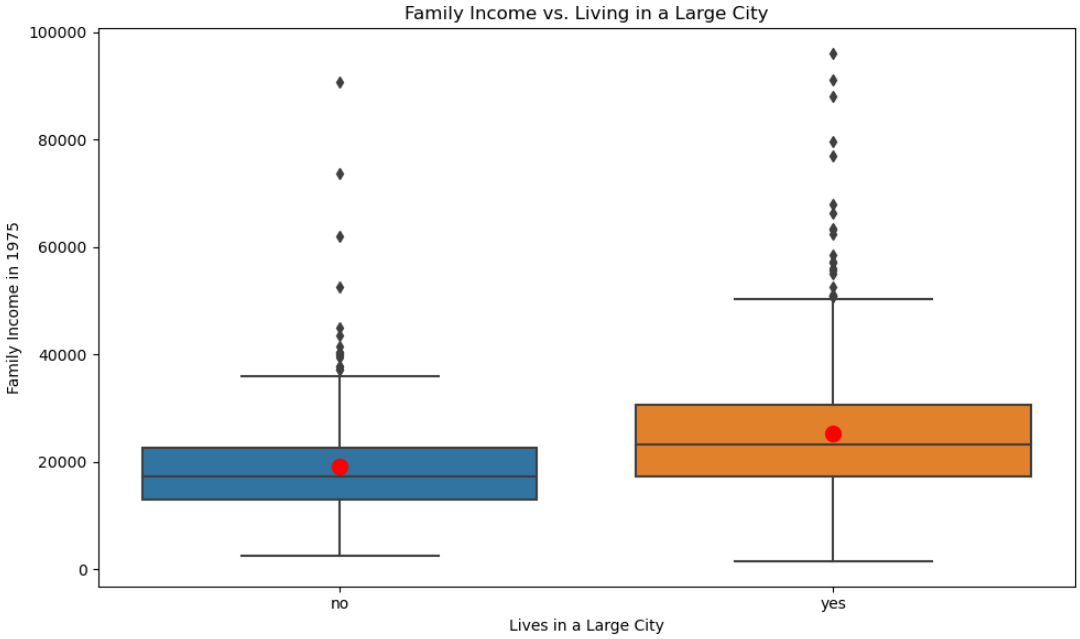




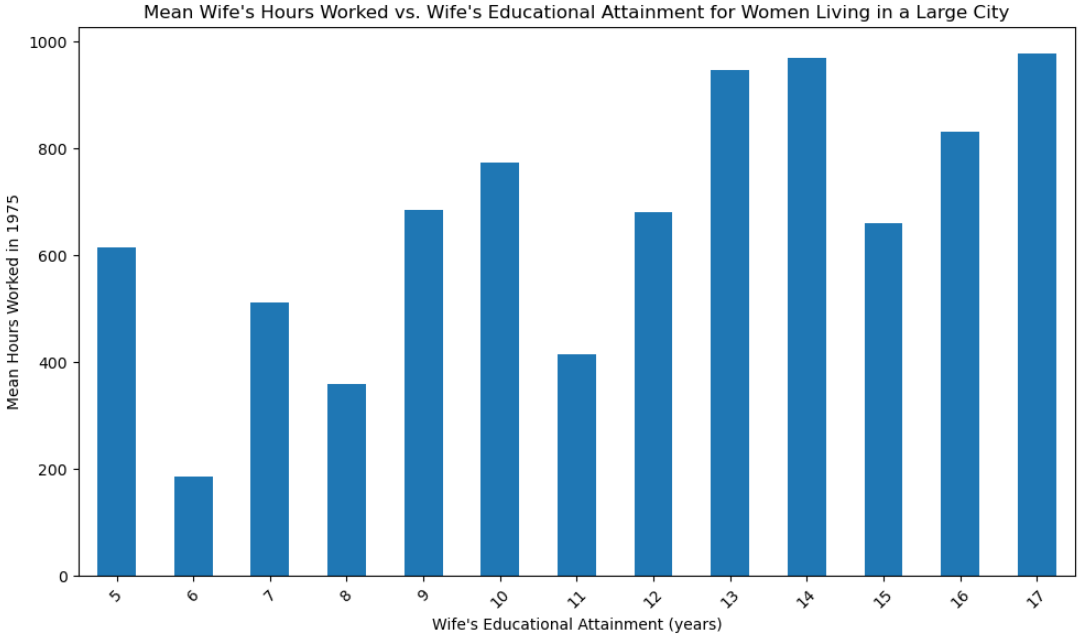


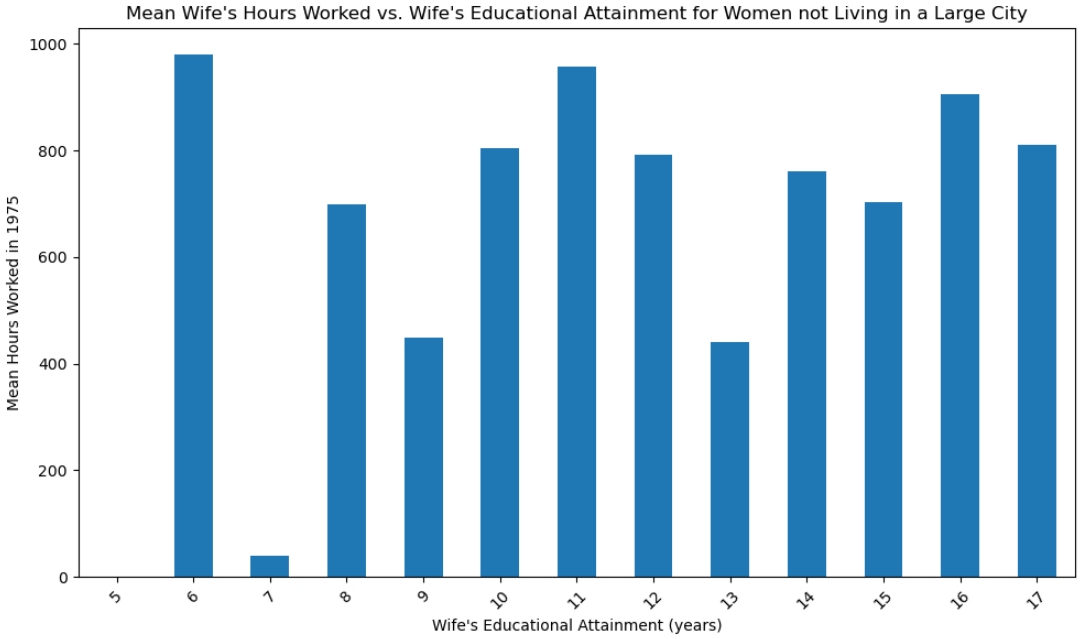
In [19]:
```
def create_scatter_plot(x_data, y_data, x_label, y_label, title):
    plt.figure(figsize=(10, 6))
    plt.scatter(x_data, y_data, alpha=0.5)
    plt.xlabel(x_label)
    plt.ylabel(y_label)
    plt.title(title)
    plt.xticks(rotation=45)
    plt.tight_layout()
    plt.show()

# Creating scatter plot for Husband's wage vs. Family income
create_scatter_plot(df['wageh'], df['income'], "Husband's Wage Reported in 1975", "Family Income in 1975",
                    "Family Income vs. Husband's Wage")

# Creating scatter plot for Wife's wage vs. Family income
create_scatter_plot(df['wagew'], df['income'], "Wife's Wage Reported in 1976", "Family Income in 1975",
                    "Family Income vs. Wife's Wage")

def create_box_plot_with_mean(data, x_label, y_label, title):
    plt.figure(figsize=(10, 6))
    sns.boxplot(x=data['city'], y=data['income'])

    # Calculate the mean income for each category
    mean_incomes = data.groupby('city')['income'].mean()

    # Add red points for mean incomes
    plt.scatter(x=mean_incomes.index, y=mean_incomes.values, color='red', marker='o', s=100, zorder=3)

    plt.xlabel(x_label)
    plt.ylabel(y_label)
    plt.title(title)
    plt.tight_layout()
    plt.show()

# Creating box plot with mean points for Income vs. Living in a Large City
create_box_plot_with_mean(df, "Lives in a Large City", "Family Income in 1975", "Family Income vs. Living in a Large City")
```





In [20]:
```
def create_bar_plot(data, x_label, y_label, title):
    plt.figure(figsize=(10, 6))
    data.plot(kind='bar')
    plt.xlabel(x_label)
    plt.ylabel(y_label)
    plt.title(title)
    plt.xticks(rotation=45)
    plt.tight_layout()
    plt.show()

# Grouping the data by Wife educational attainment and calculating the mean of hours worked for women living in a large city
graph_large_city = df[df['city'] == 'yes'].groupby('educw')['hoursw'].mean()

create_bar_plot(graph_large_city, "Wife's Educational Attainment (years)", "Mean Hours Worked in 1975",
                "Mean Wife's Hours Worked vs. Wife's Educational Attainment for Women Living in a Large City")


# Grouping the data by Wife educational and calculating the mean of hours worked for women not living in a large city
graph_large_city = df[df['city'] == 'no'].groupby('educw')['hoursw'].mean()

create_bar_plot(graph_large_city, "Wife's Educational Attainment (years)", "Mean Hours Worked in 1975",
                "Mean Wife's Hours Worked vs. Wife's Educational Attainment for Women not Living in a Large City")
```





Part 3: Estimation and hypothesis testing:

- In this part we will formally test a hypothesis using our data.
- The question we want to explore is whether the percentage of working women in big cities is greater than the percentage of working women in small cities.
- This question is interesting because it examines whether the percentage of employed women varies between big cities and small cities can shed light on urbanization's impact on women's workforce participation. It holds significance in assessing whether larger urban centers offer more employment opportunities for women compared to smaller urban areas.

- Null Hypothesis (H0): the percentage of working women in big cities is equal to the percentage of working women in small cities.(the diffrence is 0).
- Alternative Hypothesis (H1): the percentage of working women in big cities is greater to the percentage of working women in small cities.(the diffrence is not 0).

First, we need to find the Test Statistic - calaculate the mean of interest in our data.

For that we need to check if there are any missing values.

In [21]:
```
nan_counts = df.isna().sum()

print(nan_counts)
```

```
Unnamed: 0    0
work          0
hoursw        0
child6        0
child618      0
agew          0
educw         0
hearnw        0
wagew         0
hoursh        0
ageh          0
educh         0
wageh         0
income        0
educwm        0
educwf        0
unemprate     0
city          0
experience    0
dtype: int64
```

There is not none value in the data

In [22]:
```
# Calculate the mean percentage of working women in big cities
mean_big_city = df_copy[df_copy['city'] == 1]['work'].mean()

# Calculate the mean percentage of working women in small cities
mean_small_city = df_copy[df_copy['city'] == 0]['work'].mean()

# Calculate the test statistic (difference in means)
test_statistic = mean_big_city - mean_small_city
print(test_statistic)
```

```
-0.006375003840363713
```

- The Test Statistic: the diffrence between the means, is -0.006375003840363713.

Calculating the confidence interval

In [ ]:

In [23]:
```
import numpy as np

def bootstrap_mean(original_sample, num_replications, grouping_var):
    original_sample_size = original_sample.shape[0]
    bstrap_mean_diffs = np.empty(num_replications)

    for i in range(num_replications):
        bootstrap_sample = original_sample.sample(original_sample_size, replace=True)
        working_big_city = bootstrap_sample[(bootstrap_sample[grouping_var] == 1) &
                            (bootstrap_sample['city'] == 1)]['work']
        working_small_city = bootstrap_sample[(bootstrap_sample[grouping_var] == 0) &
                            (bootstrap_sample['city'] == 0)]['work']

        resampled_mean_diff = working_big_city.mean() - working_small_city.mean()
        bstrap_mean_diffs[i] = resampled_mean_diff

    return bstrap_mean_diffs

# Assuming your data is stored in the variable df_copy
means_bootstrapped = bootstrap_mean(df_copy, 5000, 'city')
print(means_bootstrapped)
```

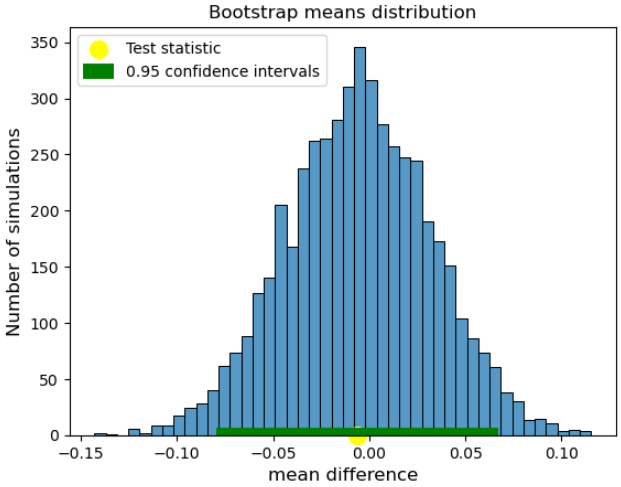
```
[-0.004647    0.01303419 -0.02660559 ... -0.00402724 -0.00796913
 -0.07558633]
```

In [24]:
```
# 95% confidence interval for the mean
left_end_data = np.percentile(means_bootstrapped, 2.5)
right_end_data = np.percentile(means_bootstrapped, 97.5)
print('lower value of 0.95 confidence interval: ', left_end_data)
print('upper value of 0.95 confidence interval: ', right_end_data)
```

```
lower value of 0.95 confidence interval:  -0.07961341707070532
upper value of 0.95 confidence interval:  0.06726610055432354
```

In [25]:
```
fig, ax = plt.subplots()

sns.histplot(means_bootstrapped, ax=ax)
ax.set(title='Bootstrap means distribution', xlabel='salary mean', ylabel='Number of simulations')
ax.set_xlabel('mean difference', fontsize=12)
ax.set_ylabel('Number of simulations', fontsize=12)


plt.scatter(test_statistic, 0, color='yellow', s=130, clip_on=False, label='Test statistic')
ax.hlines(y=0, xmin=left_end_data, xmax = right_end_data, colors='green', linestyles='solid', lw=10, label='0.95 confidence intervals')
ax.legend()
```

Out[25]:
```
<matplotlib.legend.Legend at 0x17242f2ef70>
```



- As we can observe, the Test statistic is inside the confidence interval. # Based on that, In conclusion - we cannot reject the null hypothesis.

Part 4: Prediction/clustering:

In this section of our study, we are venturing into a classification task. Our aim is to discern whether a wife's educational attainment surpasses the threshold of 12 years. The rationale behind this task stems from the intriguing connections we have observed between this particular parameter and other variables within our dataset. By predicting a wife's educational attainment, we aspire to shed light on potentially influential factors.

The target variable we are seeking to predict is whether a wife's educational attainment exceeds 12 years. To accomplish this, we will leverage various features within our dataset, including her work status, age, family income, and more. These features were chosen due to their plausible influence on educational attainment we observed.

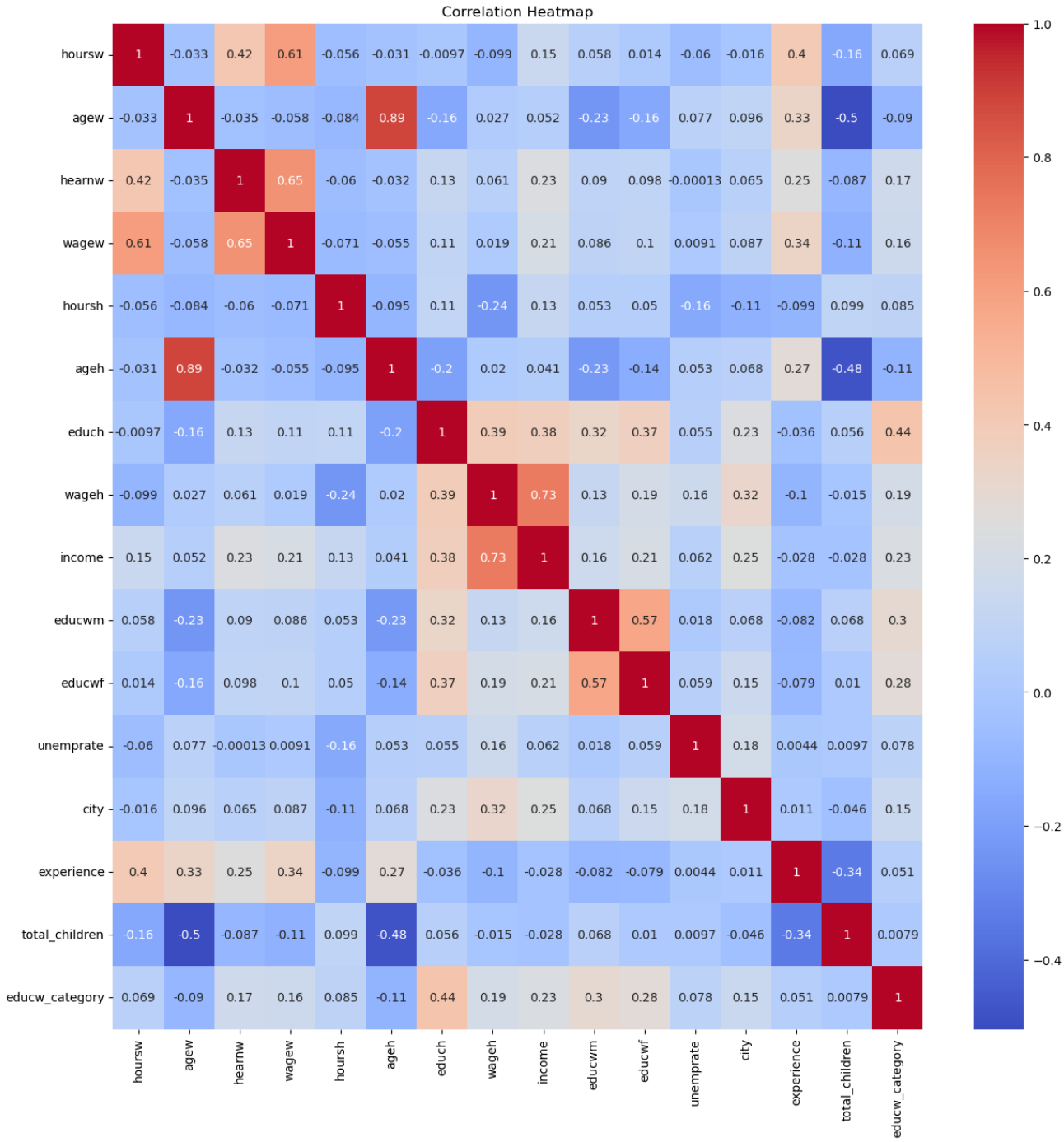
To perform the classification task, we will use the k-Nearest Neighbors (kNN) algorithm. By utilizing kNN, we intend to capture patterns and relationships within the data that may aid in the accurate classification of whether a wife's educational attainment surpasses 12 years.

- We have converted our categorical variables into numerical representations, enabling us to effectively assess their correlations. Additionally, we implemented several data transformations that we deemed valuable for implementing the k-Nearest Neighbors (kNN) algorithm. For instance, we removed the unnamed column and transformed the 'city' and 'work' columns into binary values. Furthermore, we consolidated the 'children' columns into a single representation, as the age range information was not necessary for our analysis. We created a new categorical column named 'educw_category' to represent the wife's educational attainment. This new column categorizes individuals into two groups(binary): '0' if the educational attainment of the wife's mother is less than 12 years and '1' if it's 12 years or more. This categorization helps in simplifying and categorizing the educational attainment for analysis.

In [26]:
```
# Remove the 'educw' column because have educw_category thats hold it already in numeric
df_copy.drop('educw', axis='columns', inplace=True)

# Remove the 'work' column because "hoursw" already holds the same data(if the wife is working hoursw !=0)
df_copy.drop('work', axis='columns', inplace=True)
```

This step is important to ensure that the model does not have access to information that it shouldn't during training or evaluation. Data leakage can lead to overly optimistic performance estimates and a model that does not generalize well to new data.

In [27]:
```
# Compute correlation between each pair of variables in the data frame
correlations = df_copy.corr()

# Plot heat map with increased width, focusing on variables most correlated with "educw_category"
plt.figure(figsize=(15, 15))  # Adjust the figsize to make the heatmap wider
g = sns.heatmap(correlations, annot=True, cmap='coolwarm')
plt.title("Correlation Heatmap")
plt.show()
```



we can observe that we have a strong connection between

In [28]:
```
# Calculate correlations with the "educw_category" column
correlations = df_copy.corr()["educw_category"].sort_values(ascending=False)

# Print the correlations with the "educwm_category" column
print("Correlations with the 'educw_category' column:")
for column, correlation in correlations.items():
    if column != "educw_category":  # Exclude the correlation with itself (educw_category column)
        print(f"{column}: {correlation}")
```

```
Correlations with the 'educw_category' column:
educh: 0.4362111832820417
educwm: 0.2963203453663941
educwf: 0.27591389424396884
income: 0.2261129212468726
wageh: 0.19239060812624675
hearnw: 0.16818742848765927
wagew: 0.15695883044077658
city: 0.1479791517738185
hoursh: 0.0845988309452169
unemprate: 0.0779699559693585
hoursw: 0.06911851681025757
experience: 0.051101951213513026
total_children: 0.007907857707850383
agew: -0.09012163414156003
ageh: -0.11031863775065064
```

- We will select the five most correlated variables. However, we encountered an issue - the education level of the wife's mother and the education level of the wife's father exhibit a significant correlation with each other, rendering them inefficient when considered together. As a result, we opted not to include both the husband's income and wage due to a similar reason - given that the husband is the primary breadwinner, the household income demonstrates a strong correlation with the husband's wage.
- Consequently, we concluded that it is preferable to include the income and city variables, which display a high correlation with the variable of educational, while maintaining a relatively lower correlation with the other variables.

In [29]:
```
#we will keep only the variables 'educh', 'educwm', 'ageh', 'income', 'city' because
# Select the 5 best relevant columns for KNN
selected_features = ['educh', 'educwm', 'ageh', 'income', 'city']
knn_df = df_copy[selected_features]
display(knn_df)
```

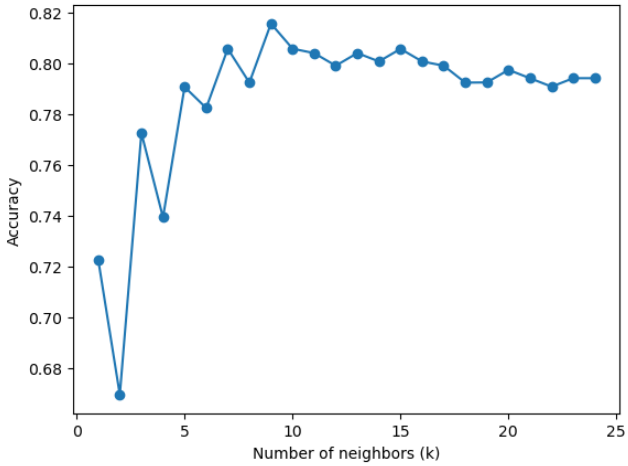
	educh	educwm	ageh	income	city
0	12	12	34	16310	0
1	9	7	30	21800	1
2	12	12	40	21040	0
3	10	7	53	7300	0
4	12	12	32	27300	1
...	...	...	...	...	...
748	16	10	43	28200	1
749	12	12	33	10000	0
750	12	10	43	9952	0
751	8	12	55	24984	1
752	12	7	48	28363	1

753 rows × 5 columns

In [ ]:

In [33]:
```
#performing the KNN.

from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt

# Randomly shuffle the dataframe
knn_df = df_copy.sample(frac=1)

# Handle missing values (for example, filling with mean)
knn_df.fillna(knn_df.mean(), inplace=True)


# Split into features (X) and target (y)
X = knn_df.drop('educw_category', axis=1)  # Features
y = knn_df['educw_category'].values  # Target

# Standardize the features (mean=0 and variance=1)
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Split into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.20)

# Create and train the KNN classifier
knn_classifier = KNeighborsClassifier(n_neighbors=5)  # You can adjust the number of neighbors
knn_classifier.fit(X_train, y_train)

# Run 10-fold CV to find optimal k - using only the training data
mean_cv_scores = []
k_list = range(1, 25)
for nn in k_list:
    knn_cv = KNeighborsClassifier(n_neighbors=nn)
    cv_scores = cross_val_score(knn_cv, X_train, y_train, cv=10)
    mean_cv_scores.append(cv_scores.mean())

# Output results
best_k = mean_cv_scores.index(max(mean_cv_scores))+1  # gets index of best performing k and adds 1
print('Highest accuracy is obtained for k =', best_k, 'and equals', max(mean_cv_scores))
plt.plot(k_list, mean_cv_scores, '-o')
plt.xlabel('Number of neighbors (k)')
plt.ylabel('Accuracy')
plt.show()
```

```
Highest accuracy is obtained for k = 9 and equals 0.8156284153005464
```



In [31]:
```
from sklearn.metrics import confusion_matrix
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
```

In [32]:
```
# Create a StandardScaler and fit-transform the training features
scaler = StandardScaler()
scaled_X_train = scaler.fit_transform(X_train)
```

```python
# Retrain the chosen kNN classifier on the entire training data
knn_classifier = KNeighborsClassifier(n_neighbors=best_k)
knn_classifier.fit(scaled_X_train, y_train)

# Transform the test features using the same scaler
scaled_X_test = scaler.transform(X_test)

# Evaluate the accuracy of the classifier on the test data
accuracy = knn_classifier.score(scaled_X_test, y_test)
print("Accuracy of the classifier:", accuracy)

# Compute a confusion matrix
predictions = knn_classifier.predict(X=scaled_X_test)  # Get the classifier's predictions
conf_matrix = confusion_matrix(y_true=y_test, y_pred=predictions, labels=[0, 1])
print("Confusion matrix:\n", conf_matrix)

# Calculate precision and recall
precision = precision_score(y_true=y_test, y_pred=predictions, average=None, labels=[0, 1])
recall = recall_score(y_true=y_test, y_pred=predictions, average=None, labels=[0, 1])

# Print precision and recall for each class
print("Precision:", precision[0])
print("Recall:", recall[1])

print("The classifier has an accuracy percentage of {:.2f}%".format(accuracy * 100))
```

```
Accuracy of the classifier: 0.8013245033112583
Confusion matrix:
 [[  9  26]
 [  4 112]]
Precision: 0.6923076923076923
Recall: 0.9655172413793104
The classifier has an accuracy percentage of 80.13%
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

ברק צבי שרון 207283888 ומטר רביץ 207036211

חלק 5:

בעבודה זו, נחקרו נתוני השכלה ותעסוקה במשפחות נשואות בארה"ב, על מנת לחשוף את הקשרים בין השכלת האישה ונתונים אחרים כמו השכלת הוריה ובעלה, הכנסה, מקום מגורים ועוד. חשיפת הקשרים הפוטנציאליים החזקים יכולה לספק הבנה מעמיקה על גורמים שמשפיעים על תעסוקת האישה במשפחה והשכלתה.

שאלות המחקר

האם קיים קשר בין ההשכלה של האישה למשתנים אחרים במשפחה, כמו השכלת ההורים והבעל, הכנסה ועוד?

האם אחוז הנשים שעובדות בערים גדולות גבוה יותר מאחוז הנשים שעובדות בערים אחרות?

האם ניתן לסווג באופן יעיל האם נשים במשפחה עובדות, בהתבסס על הנתונים הזמינים?

שאלות אלו מעניינות אותנו מאחר ואנו מעוניינים להבין את הגורמים המשפיעים על תעסוקת הנשים וההשכלה שלהן במשפחה. על פי התשובות לשאלות אלו, אנו מקווים שנוכל לקבוע את ההשפעה של השכלה של האישה על התעסוקה שלה ולהבין אילו גורמים משפיעים על ההחלטות שלהן. בנוסף, אפשר למצוא אילו משתנים קשורים לעבודה בערים גדולות ולבחון האם ניתן לסווג אם הנשים במשפחות עובדות על פי הנתונים הזמינים. התשובות לשאלות אלו יכולות לספק תובנות משמעותיות לגבי הקשרים בין פרמטרים שונים בחיי הנשים והמשפחה, ולתרום להבנה עמוקה יותר של התהליכים והגורמים שמשפיעים עליהם.

במאגר נתונים זה קיימים בכל שורה 17 פרמטרים על התא המשפחתי. ביניהם ניתן למצוא את מספר הילדים במשפחה בטווח גילים מסוים, האם האישה עובדת, השכלת הבעל, האישה, הוריהם ועוד.

מאגר הנתונים הנ"ל רלוונטי עבור חיפוש תשובות לשאלותינו משום שקיימים בו נתונים רבים על המשפחה, אשר עשויים לספק תובנות לגבי דינמיקת המשפחה ואחריות הטיפוח האפשרית. מידע זה יכול לעזור להעריך האם מספרם וגילם של הילדים משפיע על ההשכלה וסטטוס התעסוקה של האישה. השכלתו של הבעל וההורים של האישה רלוונטיים משום שהם מאפשרת לנו לחקור יחסים אפשריים בין רמות ההשכלה של חברי המשפחה וסטטוס התעסוקה של האישה. מידע זה יכול לסייע בזיהוי אם קיים דפוס של רמות השכלה גבוהות יותר במשפחות בהן נשים יותר סביר שיעבדו. גם ההכנסה היא משתנה חשוב מאוד מכיוון שהיא משקפת את המצב הכלכלי של המשפחה. היא יכולה לספק תובנות לגבי האם שיקולים פיננסיים משפיעים על ההחלטה של האישה לעבוד, במיוחד בהתחשבות בהוצאות טיפוח לילדים. גם המידע על מקום המגורים של המשפחה (בעיר גדולה או קטנה) יכול לספק תובנות משמעותיות לגבי ההשפעה של גורמים חברתיים וכלכליים על ההשתתפות התעסוקתית של הנשים.

חקרנו את הקורלציות בין השכלת האישה ומשתנים נוספים כמו השכלת הבעל, הכנסה, מצב העבודה ועוד. דבר נוסף שבדקנו הוא הקשר בין מספר הילדים למעמד התעסוקתי של נשים. ניסינו לבדוק קורלציות שקשורות גם להשכלת הבעל עם נתונים נוספים. ולבסוף בדקנו את אחוז הנשים שעובדות בערים גדולות לעומת אחוז הנשים שעובדות בערים אחרות באמצעות בדיקת השערות(רווח סמך).

גילינו קורלציה בין השכלת האישה והשכלת הבעל, המציינת מגמה של שיתוף ההשכלה בתוך המשפחה, כנ"ל עם הוריהם. מצאנו קורלציה נוספת בין מספר הילדים במשפחה לסטטוס עבודת האישה. יצרנו גם תרשים עמודות להצגת הממוצע שכר והשכלת האישה, שמציג את הקשר החיובי ביניהם.

מספר הילדים ומצב עבודה:

בחנו את היחס בין מספר הילדים, מצב העבודה של הנשים והקורלציה שביניהם.

ברק צבי שרון 207283888 ומטר רביץ 207036211

יצרנו גרף עמודות מוערך להצגת מספר הנשים העובדות לפי מספר הילדים.

גילינו קורלציה שלילית חלשה בין מספר הילדים ומספר הנשים העובדות.

שעות העבודה של הבעל ומצב עבודה של האישה:

חקרנו את היחס בין שעות העבודה של הבעל ומצב העבודה של האישה.

יצרנו גרף עמודות להצגת ממוצע שעות העבודה של הבעל לפי מצב העבודה של האישה.

זיהינו קורלציה שלילית חלשה בין שעות העבודה של הבעל ומצב העבודה של האישה.

השכלת האישה והשכלת האמא:

ניתחנו את הקורלציה בין השכלת האישה להשכלת האמא שלה.

יצרנו גרף עמודות להצגת היחס בין שני המשתנים.

גילינו קורלציה חיובית יחסית חזקה בין השכלת האישה להשכלת אמא שלה.

הניתוחים מספקים תובנות ביחס ליחסים בין משתנים כגון השכלה, מצב עבודה, מאפייני המשפחה ומקום מגורים. הממצאים מראים קורלציות בין הגורמים הללו, היכולות לסייע בהבנת השתתפות בכוח העבודה ובזיהוי גורמים המעשירים את ההבנה בנוגע להשתתפות בכוח העבודה והשפעת ההשכלה של האישה.

בדיקת השערות - אחוז הנשים העובדות בערים גדולות וקטנות:

השערות פוטנציאליות אודות אחוז הנשים העובדות בערים גדולות וקטנות.

ביצענו את השערות על בסיס דגימה חוזרת (בוטסטראפ) כדי לחשב את הרווח סמך.

הסקנו שסטטיסטי המבחן נפל בתוך הרווח סמך, לכן אין לנו אפשרות לדחות את השערת האפס.

בבחירת המשתנים למודל KNN לא לקחנו מספר משתנים שיש בינהם דמיון גבוה כמו, השכלת אמא ואבא של האישה. בחרנו לקחת רק אחד מהם משום שמצאנו כי יש בינהם קשר חזק שעלול להשפיע משמעותית על התוצאות. הורדנו גם את העמודה של work אשר אומרת האם האישה עובדת(כן או לא), משום שקיימים נתונים שמצביעים על אותו הקשר- כלומר הנתון של מספר שעות העבודה שלה מכיל כבר בתוכו את התשובה לאם האישה עובדת או לא(כאשר מספר השעות שווה ל 0 אזי האישה וודאי עובדת).

דבר נוסף הינו שבחרנו להוריד את ה income משום שמצאנו קורלציה גבוהה בינו לבין משכורת האב ולכן החלטנו להשאיר רק אחד מהם מאותן סיבות.

שינינו את עמודת ההשכלה של האישה לבינארי בצורה כזו: מעל 12 שנות לימוד: ' 1' ומתחת ל12 שנות לימוד '0', על מנת שנוכל למצוא את הקשרים בצורה המיטבית.

בחרנו לבסוף את המאפיינים הרלוונטיים לחיזוי האם לאישה יש מעל ל12 שנות לימוד.

נרמלנו את הנתונים וחילקנו את הנתונים ללימוד וסט מבחן.

השתמשנו במודל סמיכות שכנים (k-NN) k-Nearest Neighbors .

השגנו דיוק של כ-81.5% בחיזוי מצב ההשכלה של האישה.

הגבלות-

אחת מההגבלות במחקרנו הינה כמות מידע מוגבל, במאגר נתונים זה ישנם 753 שורות אודות משפחות, ולכן קשה להסיק מכך על אוכלוסיית ארצות הברית.

יכול להיות שהאוכלוסייה שנדגמה הינה אוכלוסייה שאינה מייצגת נאמנה את המציאות. הטיית בחירה היא בהחלט גורם שמשפיע על הנתונים. אם המדגם נבחר בצורה לא מאוזנת או מאזור ספציפי, יתכן שהתוצאות יוטו לאותו הכיוון. לדוגמה, בחירת אוכלוסייה מאזור בארה"ב שבו אחוז הנשים שעובדות הוא הגבוה ביותר עלול להטות את התוצאות ליצור תמונת מצב שונה מאשר במציאות.

במאגר הנתונים חסר מידע על ילדים שמעל לגיל 18, והשכלת הוריי האב.

כיוונים עתידיים

לאחר החקירה של הנתונים, עלו שאלות חדשות שעניינו אותנו, שאלה אחת שלא ניתן היה לענות עליה באופן מלא באמצעות המאגר הנתונים הנוכחי היא השפעת ההבדלים התרבותיים והאזוריים על התבניות שראינו.

לדוגמה, נרצה לחקור כיצד התרבויות וההבדלים האזוריים משפיעים על ההחלטות של נשים ללמוד ולעבוד במשפחות במבנים משפחתיים שונים. הדבר יכול לכלול הבנות כיצד היחסים כלפי השכלת הנשים משתנות בין מדינות ואזורים שונים בארצות הברית. כדי לענות לשאלה זו, נצטרך לאסוף נתונים נוספים בנוגע להשפעות התרבותיות והאזוריות.

אספקטים נוספים שנדרש לאסוף עלולים לכלול נתונים על הגישה המקומית והאזורית להשכלת הנשים, ערכים ועמדות מסובכות בנושאים מגדריים, ותנאי הכלכלה המקומיים בשונות חלקי המדינה.

על ידי הכללת מידע נוסף זה, נוכל להבין יותר את המורכבות ביחסי השכלה, דינמיקת המשפחה וההכנסות באופן יותר מעמיק וממוקד. זה יאפשר לנו לשלב מסקנות יציבות יותר ולהציע ניתוח עמוק ומורכב יותר של הקשרים שחקרנו במחקר הנוכחי.