UNIVERSITY OF TURKU

Statistical Data Analysis

Project Work Report

Masa Cirkovic *1.2.246.562.24.54054064815*
Mete Harun Akcay *1.2.246.562.24.61590076240*

Turku, November 2024

Contents

1 Introduction **2**

2 Functions & Imports **3**

3 Data Preparation **6**

4 Descriptive Statistics **11**

5 K means **21**

6 Statistical Testing **35**
6.1 Normality Testing 35
6.2 Correlation Analysis 36
6.3 Statistical Comparison 38
6.3.1 Mann-Whitney U Test 38
6.3.2 Kruskal-Wallis Test 41
6.3.3 Wilcoxon Signed-rank Test 44
6.3.4 Chi-squared test 45

7 Conclusions **52**

1 Introduction

Processing large amounts of data has become a norm nowadays, given that a great number of devices generate data. Collecting the data, cleaning and standardizing it, and then visualizing it are all part of the standard process when working with data.

Visualization is arguably the most important part, because humans respond to and process visual data better than any other type of data. In fact, the human brain processes images 60,000 times faster than text, and 90 percent of the information transmitted to the brain is visual, as said in this article. Different visualizations types work better for different target groups. Whether is it to show distribution, correlation, or some statistics, we comprehend information the best when we can see and read it.

For this project, we were given data which was obtained from a survey on *living habits of individual persons and households*. Our task was to analyze which activities people spend their time on and whether there are differences between groups. The files habits.data and habits.txt contain the data and its documentation, respectively. Given that the complete data set is huge, we were tasked with focusing only on specific parts of the data. More about that is explained in the section 3.

In the current phase of the research process, we were tasked with addressing the following tasks and questions:

1. Characterise the individuals that are present in the data. Are there groups of similar persons?
2. Estimate how much time on average households spend daily on each activity.
3. With respect to which activities do living environments differ?
4. With respect to which activities do working days and weekends differ?
5. Which activities are associated with each other?

Out task was to design and execute a **statistical analysis** that addresses the above mentioned tasks and questions. This comprehensive report documents our analysis and its results. The *objective* of each step of the analysis will be described along with the *methodology*, starting with necessary imports and our custom functions, both given in 2, followed by **data preparation** in 3, then with **descriptive statistics** in section 4, **K means clustering** in section 5. We explore **statistical tests** and their results in section 6, and lastly give concrete answers to research questions in the section 7.

2 Functions & Imports

At the beginning of the project, we imported the necessary libraries to handle data manipulation, visualization, statistical tests, clustering, and dimensionality reduction. The following imports were used:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import wilcoxon, shapiro, spearmanr, chi2_contingency,
    mannwhitneyu, kruskal

from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from sklearn.manifold import TSNE
from sklearn.metrics import silhouette_score, davies_bouldin_score
from sklearn.decomposition import PCA
from statsmodels.stats.multitest import multipletests
```

Listing 1: Importing Required Libraries

The following helper functions were written to streamline the analysis and avoid code repetitiveness:

```
# Check the unique values for each column
def check_unique_values(df):
  for column in df.columns:
    unique_values = df[column].unique()
    print(f"Column '{column}':")
    print(unique_values)
    print("\n" + "-"*50 + "\n")
```

Listing 2: Checking Unique Values for Each Column of the Provided Dataframe

```
# Function to convert 0 to 1.0 (no) and any other integer to 2.0 (yes)
def convert_number_to_1_or_2(value):
    if value in [1.0, 2.0]:
        return value  # Keep original values as they are
    elif isinstance(value, (int, float)) and value == 0:
        return 1.0  # Convert 0 to 1.0 which is no
    elif isinstance(value, (int, float)) and value != 0:
        return 2.0  # Convert any other number to 2.0 which is yes
    return None
```

Listing 3: Converting Numbers to Binary Categories (Yes/No)

```
# Function to transform timestamps to minutes
def time_to_minutes(value):
  if pd.isnull(value):  # Check for NaN or None values using pandas
      function
    return None
  elif isinstance(value, str) and ':' in value:  # Case 2: "HH:MM" format
      and ensure it's a string
```

```
    hours, minutes = map(int, value.split(':'))
    return hours * 60 + minutes  # Convert hours to minutes and add the minutes
  else:  # Case 1: "minutes" format or other numeric types
    return int(value)
```

Listing 4: Transforming Timestamps to Minutes

```
def countplots(x_columns, y_columns, titles, x_labels):
  length = len(x_columns)
  plt.figure(figsize=(6 * length, 6))
  for i in range(length):
    plt.subplot(1, length, i + 1)
    sns.countplot(data=df, x=x_columns[i], hue=y_columns[i])
    plt.title(titles[i])
    plt.xlabel(x_labels[i])
  plt.tight_layout()
```

Listing 5: Creating Count Plots Using Seaborn and Matplotlib

```
def histoplots(x_columns, y_columns, titles, x_labels):
  length = len(x_columns)
  plt.figure(figsize=(6 * length, 6))
  for i in range(length):
    plt.subplot(1, length, i + 1)
    sns.histplot(data=df, x=x_columns[i], hue=y_columns[i], kde=True, edgecolor="black")
    plt.title(titles[i])
    plt.xlabel(x_labels[i])
  plt.tight_layout()
```

Listing 6: Creating Histograms Using Seaborn and Matplotli

```
def kdeplots(x_columns, y_columns, titles, x_labels):
  length = len(x_columns)
  plt.figure(figsize=(6 * length, 6))
  for i in range(length):
    plt.subplot(1, length, i + 1)
    sns.kdeplot(data=df, x=x_columns[i], hue=y_columns[i], fill=False, warn_singular=False, clip=(0, None))
    plt.title(titles[i])
    plt.xlabel(x_labels[i])
  plt.tight_layout()
```

Listing 7: Creating KDE Plots Using Seaborn and Matplotli

```
def print_test_results(stat, p, H0, p_adj = None, alpha=0.05):
  if p_adj == None:
    print('Statistics=%.3f, p=%.3f' % (stat, p))
  else:
    print('Statistics=%.3f, p=%.3f, p_adj=%.3f' % (stat, p, p_adj))
    p = p_adj

  print(f"H0: {H0}")
  if p > alpha:
```

```
10        print('Fail to reject H0. There is no statistically significant
              difference between the groups.')
11    else:
12        print('Reject H0. There is a statistically significant difference
              between the groups.')
```

Listing 8: Printing Statistical Test Results

```
1  def print_normality_test_results(data, H0, alpha=0.05):
2      stat, p = shapiro(data)
3      print('Statistics=%.3f, p=%.3f' % (stat, p))
4      print(f"H0: {H0}")
5      if p > alpha:
6          print('Sample looks Gaussian (fail to reject H0)')
7      else:
8          print('Sample does not look Gaussian (reject H0)')
```

Listing 9: Printing Normality Test Results

3 Data Preparation

To begin the analysis, the dataset was loaded using the pandas library from the provided file, habits.data. The file was read with the following specifications:
Separator: The file uses ; as the delimiter, which was specified using the sep parameter.
Missing Values: The placeholder ? in the dataset was automatically recognized and converted to NaN values using the na_values parameter for easier handling.
Once the dataset was loaded, only the columns specified in the project instructions were retained to focus on the analysis of the relevant data. These selected columns include:

- **Demographic Variables:**
  - `kohde` (household ID)
  - `jasen` (member ID)
  - `pvknro` (day)
  - `sp` (sex)
  - `IKAL1` (age)
  - `ASALUE` (place)
- **Activity Variables:**
  - `V1` (working)
  - `V40` (studying)
  - `V7` (washing dishes)
  - `V70` (watching TV)
  - `H1a_A` (cinema attendance)
  - `H1i_A` (sports attendance)

```
data = pd.read_csv('habits.data', sep=';', na_values='?')
dataframe = data[['kohde', 'jasen', 'pvknro', 'sp', 'IKAL1', 'ASALUE', 'V1'
    , 'V40', 'V7', 'V70', 'H1a_A', 'H1i_A']]
```

Listing 10: Loading and Selecting Relevant Data

Using the provided habits.txt file for reference, we translated the column names to English for clarity and convenience. Furthermore, we mapped categorical values to more descriptive labels to improve interpretability. For example: values of column *day* (1 and 2) were converted to "working day" and "weekend," respectively; values of column *sex* (1 and 2) were replaced with "male" and "female." *Age groups* (e.g., 1, 2, 3) were transformed into descriptive ranges such as "10-14", "15-19" etc. The complete mapping of these transformations can be seen in the code snippet below.

```
# Change columns to English & descriptive labels for clarity
df = dataframe.copy()

df.rename(columns={
    'kohde': 'household_ID',
    'jasen': 'member_ID',
    'pvknro': 'day',
    'sp': 'sex',
```

```
    'IKAL1': 'age',
    'ASALUE': 'place',
    'V1': 'working',
    'V40': 'studying',
    'V7': 'washing_dishes',
    'V70': 'watching_TV',
    'H1a_A': 'cinema',
    'H1i_A': 'sport'
}, inplace=True)

df['day'] = df['day'].map({1: 'working day', 2: 'weekend'}).astype('
    category')

df['sex'] = df['sex'].map({1: 'male', 2: 'female'}).astype('category')

df['age'] = df['age'].map({
    1: '10-14',
    2: '15-19',
    3: '20-24',
    4: '25-34',
    5: '35-44',
    6: '45-54',
    7: '55-64',
    8: '65-74',
    9: '75+'
}).astype('category')

df['place'] = df['place'].map({1: 'city', 2: 'municipality', 3: 'rural'}).
    astype('category')

df['member_ID'] = df['member_ID'].astype('category')
df['household_ID'] = df['household_ID'].astype('category')
```

Listing 11: Renaming Columns and Mapping Categorical Variables

During the data preparation phase, we observed that the columns `cinema` and `sport` contained numeric values that were not limited to the expected categories `1` (yes) and `2` (no). These additional numeric values were interpreted as activity durations in minutes. To address this:

- Values of `0` were treated as "no" for the activity.
- Non-zero values were treated as "yes."
- Subsequently, the categorical mapping was applied to convert these values into "yes" and "no" categories for easier interpretation and consistency.

The following code snippet illustrates this process:

```
# Some values for cinema and sport are numbers other than 1.0 and 2.0

df['cinema'] = df['cinema'].apply(convert_number_to_1_or_2)
df['sport'] = df['sport'].apply(convert_number_to_1_or_2)

df['cinema'] = df['cinema'].map({1.0: 'yes', 2.0: 'no'}).astype('category')
df['sport'] = df['sport'].map({1.0: 'yes', 2.0: 'no'}).astype('category')
```

Listing 12: Handling Cinema and Sport Variables

Based on the renaming and mapping operations performed to make the dataset more interpretable, the updated DataFrame is shown below.

	household_ID	member_ID	day	sex	age	place	working	studying	washing_dishes	watching_TV	cinema	sport
0	50002	1	weekend	male	45-54	city	380	0	0	130	no	yes
1	50002	2	weekend	female	45-54	city	470	0	10	0	yes	yes
2	50009	1	weekend	female	55-64	city	350	0	10	120	yes	yes
3	50011	1	weekend	female	25-34	rural	360	0	0	20	yes	yes
4	50012	1	working day	female	65-74	city	0	0	70	350	no	no
...	...	...	...	...	...	...	...	...	...	...	...	...
775	51980	1	weekend	female	45-54	municipality	460	0	0	80	yes	yes
776	51981	1	weekend	female	35-44	city	0	0	0	160	yes	yes
777	51981	2	weekend	male	35-44	city	0	0	0	190	no	no
778	51982	1	weekend	female	45-54	city	0	0	0	210	no	no
779	51982	2	weekend	male	35-44	city	420	0	NaN	190	no	no

780 rows × 12 columns

Figure 1: Modified DataFrame After Renaming and Mapping

To handle missing values effectively, we first examined the unique values in the dataset. It was observed that some activity values were provided in minutes as strings, while others were recorded in the `HH:MM` format. To ensure uniformity, we converted all activity values to integer minutes using the following code:

```
# Some activity values are given in minutes, some are given in HH:MM format
for col in ['working', 'studying', 'washing_dishes', 'watching_TV']:
    df[col] = df[col].apply(time_to_minutes).astype('Int64')  # Convert to
        integer type, allowing NaNs.
```

Listing 13: Converting Activity Values to Minutes

To handle missing values in the dataset, we first checked for `NaN` values across all columns using the following code:

```
# Check for NaN values
df.isnull().sum()
```

Listing 14: Checking for Missing Values

The result of this operation revealed missing values in the activity-related columns:

- `working`: 5 missing values
- `studying`: 7 missing values
- `washing_dishes`: 12 missing values
- `watching_TV`: 8 missing values

To impute these missing values, we used a group-based approach. Specifically, the mean values for each activity were calculated based on combinations of the demographic columns (`sex`, `day`, `age`, `place`). If no group mean was available, the overall column mean was used as a fallback. The following code snippet illustrates this process:

```
# Impute missing values based on the mean of each group defined by 'sex', 'day', 'age', and 'place';
# if no group mean is available, use the overall column mean.

df_cleaned = df.copy()
columns_to_impute = ['working', 'studying', 'washing_dishes', 'watching_TV']
print("Missing values before imputation:")
print(df_cleaned[columns_to_impute].isnull().sum())

grouped_means = df_cleaned.groupby(['sex', 'day', 'age', 'place'], observed=False)[columns_to_impute].transform('mean')
grouped_means = grouped_means.round().astype('Int64')
overall_means = df_cleaned[columns_to_impute].mean().round().astype('Int64')

for column in columns_to_impute:
    df_cleaned[column] = df_cleaned[column].fillna(grouped_means[column]).fillna(overall_means[column])

print("\nMissing values after imputation:")
print(df_cleaned[columns_to_impute].isnull().sum())
```

Listing 15: Imputing Missing Values Based on Group Means

This approach ensured that missing values were imputed *logically* based on group-level patterns while maintaining data consistency. The final result showed no remaining missing values in the activity columns, as confirmed by the output of the imputation process, which is given below.

Column	Before Imputation	After Imputation
working	5	0
studying	7	0
washing_dishes	12	0
watching_TV	8	0

Table 1: Missing Values Before and After Imputation

To ensure that all columns were correctly converted to their appropriate data types and that missing values were properly handled, we used the `info()` function on the cleaned DataFrame. The following code snippet was executed:

```
df_cleaned.info()
```

Listing 16: Checking DataFrame Information

The output, shown below, confirms that:

- The dataframe has 780 rows and 12 columns.
- All columns have been converted to the correct data types (e.g., `category` for categorical variables and `Int64` for activity-related columns).
- All missing values have been successfully filled.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 780 entries, 0 to 779
Data columns (total 12 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   household_ID    780 non-null    category
 1   member_ID       780 non-null    category
 2   day             780 non-null    category
 3   sex             780 non-null    category
 4   age             780 non-null    category
 5   place           780 non-null    category
 6   working         780 non-null    Int64
 7   studying        780 non-null    Int64
 8   washing_dishes  780 non-null    Int64
 9   watching_TV     780 non-null    Int64
 10  cinema          780 non-null    category
 11  sport           780 non-null    category
dtypes: Int64(4), category(8)
memory usage: 55.5 KB
```

Figure 2: Summary of the Cleaned DataFrame

As a result, the DataFrame was fully prepared for further analysis.

4 Descriptive Statistics

To streamline the analysis process, we grouped the columns into two categories for convenience:

- **Demographic Columns**: These include categorical variables: `sex`, `age`, `day`, and `place`.
- **Numerical Columns**: These include quantitative variables: `working`, `studying`, `washing_dishes`, and `watching_TV`.

The following code snippet was used to define these categories:

```
df = df_cleaned.copy()
demographic_columns = ['sex', 'age', 'day', 'place']
numerical_columns = ['working', 'studying', 'washing_dishes', 'watching_TV'
    ]
```

Listing 17: Categorizing Demographic and Numerical Columns

To summarize the demographic variables, we calculated descriptive statistics, including the count of observations, number of unique categories, most frequent category (`top`), and its frequency (`freq`). The following code was used to generate these statistics:

```
# Summary statistics for demographic variables
demographic_summary = df[demographic_columns].describe()
print(demographic_summary)
```

Listing 18: Summary Statistics for Demographic Variables

The output is shown below:

Statistic	sex	age	day	place
Count	780	780	780	780
Unique	2	7	2	3
Top	female	45-54	working day	city
Frequency	406	170	418	519

Table 2: Summary Statistics for Demographic Variables

These statistics provide a high-level overview of the dataset:

- The most frequent category for `sex` is `female`, representing 406 observations.
- The most common age group is `45-54`, with 170 participants.
- The majority of observations are for `working day` and `city`, with frequencies of 418 and 519, respectively.

Similarly, we computed descriptive statistics for the numerical variables to summarize their central tendency, dispersion, and range. The code and the output are given below:

```
# Summary statistics for numerical variables
numerical_summary = df[numerical_columns].describe()
print(numerical_summary)
```

Listing 19: Summary Statistics for Numerical Variables

Statistic	working	studying	washing_dishes	watching_TV
Count	780.0	780.0	780.0	780.0
Mean	105.13	2.29	11.02	147.12
Standard Deviation	203.06	26.55	16.92	122.95
Minimum	0.0	0.0	0.0	0.0
25th Percentile	0.0	0.0	0.0	60.0
Median (50th %)	0.0	0.0	0.0	123.5
75th Percentile	32.5	0.0	20.0	210.0
Maximum	1150.0	430.0	90.0	950.0

Table 3: Summary Statistics for Numerical Variables

These statistics reveal the following insights:

- The average time spent `working` is 105 minutes, with highest variability amongst the other activities.
- Most participants spend little to no time `studying` as the 25th, 50th, and 75th percentiles for this variable are all 0.
- Time spent `watching_TV` is the greatest amongst the others, with an average of 147 minutes and a maximum of 950 minutes.

To visualize the distribution of demographic variables, we created pie charts for each variable. The charts display the proportion of each category within the variables `sex`, `age`, `day`, and `place`. The following code was used to generate the visualizations:

```
plt.figure(figsize=(22, 6))
i = 0
for column in demographic_columns:
    plt.subplot(1, len(demographic_columns), i + 1)
    df[column].value_counts().plot.pie(autopct='%1.1f%%', startangle=90,
        cmap='Set3')
    plt.title(f'Distribution of {column.capitalize()}')
    plt.ylabel('')
    i += 1
plt.tight_layout()
```

Listing 20: Pie Charts for Demographic Variables

The resulting visualizations are shown below:

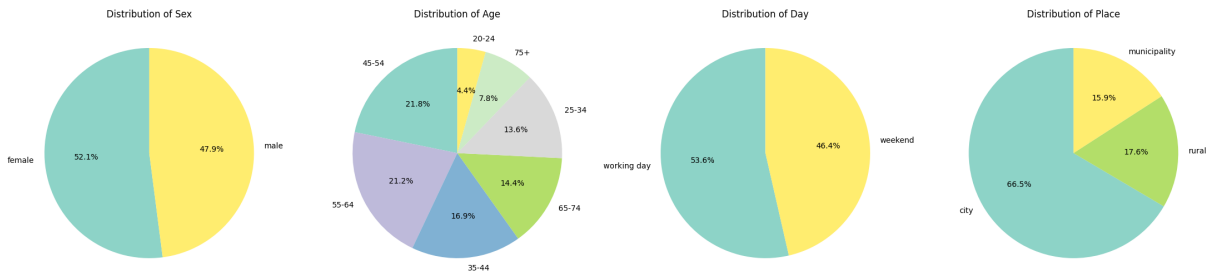


Figure 3: Pie Charts Showing the Distribution of Demographic Variables

To estimate how much time on average households spend daily on each activity, we first calculated the total time spent on activities for each household. Then, the average time was

derived across all households for each activity to estimate daily averages. The Python code used for this calculation is as follows:

```
# Calculate the average time spent on each activity per household
household_average_activity_time = df.groupby('household_ID', observed=False
    )[numerical_columns].sum()

# Calculate the cumulative average time spent on activities
# Answer to the question: "Estimate how much time on average households
    spend daily on each activity"
average_activity_time = household_average_activity_time.mean(axis=0)
print(average_activity_time)
```

Listing 21: Calculating Average Time Spent Daily on Activities by Households

The results are presented in the table below:

Activity	Average Time (minutes)
working	165.99
studying	3.62
washing_dishes	17.41
watching_TV	232.30

Table 4: Average Time Spent Daily on Activities by Households

We then explored how activities vary across different demographic groups. To achieve this, we calculated descriptive statistics (`count`, `mean`, `std`, `min`, `max`) for each numerical variable grouped by the demographic categories. The Python code snippet used for this analysis is as follows:

```
# Grouped summary statistics for numerical variables based on each
    demographic variable
summary_stats = ['count', 'mean', 'std', 'min', 'max']
for col in demographic_columns:
    grouped_summary = df.groupby([col], observed=False)[numerical_columns].
        describe().loc[:, (slice(None), summary_stats)]

    styled_summary = grouped_summary.style.format("{:.2f}") \
                                          .set_caption(f"Summary Statistics
                                              for Activities by {col.
                                              capitalize()}") \
                                          .background_gradient(cmap="YlGnBu
                                              ", subset=pd.IndexSlice[:, pd.
                                              IndexSlice[:, :]])

    display(styled_summary)
```

Listing 22: Grouped Summary Statistics by Demographic Variables

This code groups the data by each demographic variable separately and computes summary statistics for the activity columns. The resulting grouped summary statistics are visualized in the image below:

Summary Statistics for Activities by Sex

	working					studying					washing_dishes					watching_TV				
	count	mean	std	min	max	count	mean	std	min	max	count	mean	std	min	max	count	mean	std	min	max
sex																				
female	406.00	85.48	175.40	0.00	830.00	406.00	1.92	23.17	0.00	350.00	406.00	17.45	19.56	0.00	90.00	406.00	132.21	117.78	0.00	950.00
male	374.00	126.45	227.67	0.00	1150.00	374.00	2.70	29.81	0.00	430.00	374.00	4.05	9.42	0.00	70.00	374.00	163.31	126.51	0.00	740.00

Summary Statistics for Activities by Age

	working					studying					washing_dishes					watching_TV				
	count	mean	std	min	max	count	mean	std	min	max	count	mean	std	min	max	count	mean	std	min	max
age																				
20-24	34.00	94.53	178.42	0.00	520.00	34.00	34.41	103.23	0.00	430.00	34.00	5.62	11.87	0.00	60.00	34.00	97.26	94.27	0.00	310.00
25-34	106.00	170.39	253.19	0.00	940.00	106.00	5.85	38.76	0.00	350.00	106.00	8.87	15.20	0.00	70.00	106.00	95.85	110.44	0.00	540.00
35-44	132.00	165.61	235.32	0.00	1150.00	132.00	0.00	0.00	0.00	0.00	132.00	9.07	15.06	0.00	70.00	132.00	119.38	98.20	0.00	480.00
45-54	170.00	166.20	229.47	0.00	830.00	170.00	0.00	0.00	0.00	0.00	170.00	9.28	15.10	0.00	80.00	170.00	127.55	105.48	0.00	740.00
55-64	165.00	60.61	163.14	0.00	910.00	165.00	0.00	0.00	0.00	0.00	165.00	12.52	18.00	0.00	90.00	165.00	176.91	127.87	0.00	510.00
65-74	112.00	4.46	33.26	0.00	250.00	112.00	0.00	0.00	0.00	0.00	112.00	14.09	18.36	0.00	70.00	112.00	186.96	136.50	0.00	950.00
75+	61.00	1.80	14.08	0.00	110.00	61.00	0.00	0.00	0.00	0.00	61.00	17.21	21.99	0.00	80.00	61.00	224.89	134.53	0.00	640.00

Summary Statistics for Activities by Day

	working					studying					washing_dishes					watching_TV				
	count	mean	std	min	max	count	mean	std	min	max	count	mean	std	min	max	count	mean	std	min	max
day																				
weekend	362.00	113.49	212.62	0.00	1150.00	362.00	2.07	22.13	0.00	350.00	362.00	9.56	15.81	0.00	90.00	362.00	145.56	126.38	0.00	740.00
working day	418.00	97.88	194.36	0.00	910.00	418.00	2.49	29.88	0.00	430.00	418.00	12.30	17.75	0.00	90.00	418.00	148.48	120.04	0.00	950.00

Summary Statistics for Activities by Place

	working					studying					washing_dishes					watching_TV				
	count	mean	std	min	max	count	mean	std	min	max	count	mean	std	min	max	count	mean	std	min	max
place																				
city	519.00	107.12	203.66	0.00	1150.00	519.00	2.27	26.38	0.00	430.00	519.00	10.71	17.12	0.00	90.00	519.00	144.26	121.55	0.00	950.00
municipality	124.00	99.23	200.62	0.00	940.00	124.00	0.00	0.00	0.00	0.00	124.00	10.33	14.15	0.00	60.00	124.00	171.81	139.19	0.00	710.00
rural	137.00	102.92	204.31	0.00	910.00	137.00	4.45	37.12	0.00	350.00	137.00	12.86	18.38	0.00	90.00	137.00	135.64	109.87	0.00	540.00

Figure 4: Grouped Summary Statistics for Activities by Demographic Variables

Following this, we calculated grouped summary statistics for numerical variables based on all demographic variables. This approach provides a comprehensive view of how activities vary when considering all demographic factors simultaneously. The Python code used for this analysis is as follows:

```
# Grouped summary statistics for numerical variables based on all
    demographic variables
summary_stats = ['count', 'mean', 'std', 'min', 'max']
grouped_summary = df.groupby(demographic_columns, observed=False)[
    numerical_columns].describe().loc[:, (slice(None), summary_stats)]

# Modify the styling to correctly access the 'mean' level in the MultiIndex
styled_summary = grouped_summary.style.format("{:.2f}") \
                                      .set_caption(f"Summary Statistics for
                                          Activities") \
                                      .background_gradient(cmap="YlGnBu",
                                         subset=pd.IndexSlice[:, pd.
                                         IndexSlice[:, :]]) \
                                      .highlight_max(axis=0, color="
                                         lightcoral", subset=pd.IndexSlice
                                         [:, pd.IndexSlice[:, :]]) # Apply
                                         to all columns within each level

display(styled_summary)
```

Listing 23: Grouped Summary Statistics by All Demographic Variables

This code computes detailed statistics for all activities by taking into account all possible demographic combinations. The results are styled to highlight the maximum values within each group. Since the original visualization for the grouped summary statistics was too large to fit into a single image, it was manually cropped and divided into two parts. The first image represents statistics for `females`, while the second image represents statistics for `males`. The visualized results are shown below:

Summary Statistics for Activities

				working						studying					washing_dishes					watching_TV				
				count	mean	std	min	max	count	mean	std	min	max	count	mean	std	min	max	count	mean	std	min	max
sex	age	day	place																					
female	20-24	weekend	city	5.00	74.00	165.47	0.00	370.00	5.00	0.00	0.00	0.00	0.00	5.00	18.00	24.90	0.00	60.00	5.00	56.00	99.65	0.00	230.00
			municipality	2.00	230.00	325.27	0.00	460.00	2.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	2.00	100.00	28.28	80.00	120.00
			rural	1.00	410.00	[redacted]	410.00	410.00	1.00	0.00	[redacted]	0.00	0.00	1.00	11.00	[redacted]	11.00	11.00	1.00	50.00	[redacted]	50.00	50.00
		working day	city	7.00	188.57	227.04	0.00	460.00	7.00	0.00	0.00	0.00	0.00	7.00	8.57	10.69	0.00	20.00	7.00	126.71	87.88	0.00	270.00
			municipality	1.00	0.00	[redacted]	0.00	0.00	1.00	0.00	[redacted]	0.00	0.00	1.00	0.00	[redacted]	0.00	0.00	1.00	60.00	[redacted]	60.00	60.00
			rural	3.00	20.00	34.64	0.00	60.00	3.00	203.33	181.75	0.00	350.00	3.00	3.33	5.77	0.00	10.00	3.00	90.00	147.31	0.00	260.00
		weekend	city	16.00	110.69	187.42	0.00	500.00	16.00	10.62	42.50	0.00	170.00	16.00	10.62	20.48	0.00	70.00	16.00	80.00	98.59	0.00	310.00
			municipality	3.00	0.00	0.00	0.00	0.00	3.00	0.00	0.00	0.00	0.00	3.00	3.33	5.77	0.00	10.00	3.00	93.33	15.28	80.00	110.00
			rural	5.00	80.00	157.48	0.00	360.00	5.00	0.00	0.00	0.00	0.00	5.00	12.00	10.95	0.00	20.00	5.00	90.00	91.65	0.00	230.00
	25-34		city	16.00	157.50	187.07	0.00	460.00	16.00	0.00	0.00	0.00	0.00	16.00	11.88	19.40	0.00	70.00	16.00	86.88	70.40	0.00	240.00
		working day	municipality	7.00	135.71	234.79	0.00	540.00	7.00	0.00	0.00	0.00	0.00	7.00	25.71	21.49	0.00	60.00	7.00	67.14	102.75	0.00	290.00
			rural	6.00	141.67	248.71	0.00	610.00	6.00	0.00	0.00	0.00	0.00	6.00	18.33	9.83	10.00	30.00	6.00	85.00	88.49	0.00	220.00
			city	26.00	209.62	228.60	0.00	510.00	26.00	0.00	0.00	0.00	0.00	26.00	12.38	12.74	0.00	40.00	26.00	111.15	109.30	0.00	480.00
		weekend	municipality	4.00	0.00	0.00	0.00	0.00	4.00	0.00	0.00	0.00	0.00	4.00	12.50	5.00	10.00	20.00	4.00	102.50	123.39	0.00	270.00
			rural	4.00	0.00	0.00	0.00	0.00	4.00	0.00	0.00	0.00	0.00	4.00	20.00	23.09	0.00	40.00	4.00	62.50	45.00	0.00	100.00
	35-44		city	25.00	80.00	186.97	0.00	600.00	25.00	0.00	0.00	0.00	0.00	25.00	12.00	20.41	0.00	70.00	25.00	100.84	91.92	0.00	350.00
		working day	municipality	10.00	190.00	229.88	0.00	590.00	10.00	0.00	0.00	0.00	0.00	10.00	15.00	18.41	0.00	50.00	10.00	138.00	89.17	50.00	320.00
			rural	5.00	62.00	138.64	0.00	310.00	5.00	0.00	0.00	0.00	0.00	5.00	6.00	5.48	0.00	10.00	5.00	92.40	85.26	10.00	220.00
			city	22.00	103.18	198.53	0.00	620.00	22.00	0.00	0.00	0.00	0.00	22.00	14.09	20.62	0.00	70.00	22.00	123.64	96.39	0.00	300.00
		weekend	municipality	9.00	113.78	197.04	0.00	460.00	9.00	0.00	0.00	0.00	0.00	9.00	13.33	16.58	0.00	50.00	9.00	97.78	70.49	0.00	180.00
			rural	7.00	185.71	242.13	0.00	540.00	7.00	0.00	0.00	0.00	0.00	7.00	20.00	18.26	0.00	50.00	7.00	68.57	53.36	20.00	170.00
	45-54		city	40.00	172.25	228.49	0.00	830.00	40.00	0.00	0.00	0.00	0.00	40.00	17.00	17.86	0.00	80.00	40.00	88.25	86.91	0.00	320.00
		working day	municipality	6.00	66.67	163.30	0.00	400.00	6.00	0.00	0.00	0.00	0.00	6.00	13.33	15.06	0.00	40.00	6.00	121.67	71.67	0.00	200.00
			rural	5.00	156.00	214.08	0.00	410.00	5.00	0.00	0.00	0.00	0.00	5.00	7.60	12.99	0.00	30.00	5.00	106.00	114.37	10.00	250.00
			city	24.00	59.58	159.74	0.00	580.00	24.00	0.00	0.00	0.00	0.00	24.00	22.50	24.54	0.00	90.00	24.00	125.00	96.59	0.00	310.00
		weekend	municipality	10.00	45.00	142.30	0.00	450.00	10.00	0.00	0.00	0.00	0.00	10.00	13.00	10.59	0.00	30.00	10.00	178.00	156.97	0.00	460.00
			rural	3.00	0.00	0.00	0.00	0.00	3.00	0.00	0.00	0.00	0.00	3.00	20.00	0.00	20.00	20.00	3.00	260.00	227.16	0.00	420.00
	55-64		city	35.00	39.71	133.05	0.00	510.00	35.00	0.00	0.00	0.00	0.00	35.00	18.83	18.11	0.00	70.00	35.00	155.43	109.88	0.00	400.00
		working day	municipality	5.00	0.00	0.00	0.00	0.00	5.00	0.00	0.00	0.00	0.00	5.00	18.00	21.68	0.00	50.00	5.00	182.00	136.09	0.00	340.00
			rural	8.00	0.00	0.00	0.00	0.00	8.00	0.00	0.00	0.00	0.00	8.00	37.50	27.12	10.00	90.00	8.00	140.00	59.04	40.00	210.00
			city	16.00	0.00	0.00	0.00	0.00	16.00	0.00	0.00	0.00	0.00	16.00	16.88	17.02	0.00	50.00	16.00	146.88	89.27	0.00	370.00
		weekend	municipality	5.00	0.00	0.00	0.00	0.00	5.00	0.00	0.00	0.00	0.00	5.00	15.00	20.62	0.00	50.00	5.00	228.00	154.18	0.00	380.00
			rural	6.00	0.00	0.00	0.00	0.00	6.00	0.00	0.00	0.00	0.00	6.00	36.67	18.62	10.00	60.00	6.00	171.67	97.86	0.00	300.00
	65-74		city	15.00	0.00	0.00	0.00	0.00	15.00	0.00	0.00	0.00	0.00	15.00	26.67	23.80	0.00	70.00	15.00	238.67	223.92	40.00	950.00
		working day	municipality	1.00	0.00	[redacted]	0.00	0.00	1.00	0.00	[redacted]	0.00	0.00	1.00	40.00	[redacted]	40.00	40.00	1.00	220.00	[redacted]	220.00	220.00
			rural	12.00	0.00	0.00	0.00	0.00	12.00	0.00	0.00	0.00	0.00	12.00	27.50	17.65	0.00	50.00	12.00	155.83	87.54	0.00	330.00
			city	12.00	0.00	0.00	0.00	0.00	12.00	0.00	0.00	0.00	0.00	12.00	17.50	17.12	0.00	50.00	12.00	247.25	196.89	0.00	640.00
		weekend	municipality	2.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	2.00	10.00	14.14	0.00	20.00	2.00	255.00	219.20	100.00	410.00
			rural	1.00	0.00	[redacted]	0.00	0.00	1.00	0.00	[redacted]	0.00	0.00	1.00	80.00	[redacted]	80.00	80.00	1.00	120.00	[redacted]	120.00	120.00
	75+		city	13.00	0.00	0.00	0.00	0.00	13.00	0.00	0.00	0.00	0.00	13.00	32.31	28.62	0.00	80.00	13.00	220.85	88.55	0.00	350.00
		working day	municipality	2.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	2.00	15.00	7.07	10.00	20.00	2.00	405.00	106.07	330.00	480.00
			rural	1.00	0.00	[redacted]	0.00	0.00	1.00	0.00	[redacted]	0.00	0.00	1.00	20.00	[redacted]	20.00	20.00	1.00	270.00	[redacted]	270.00	270.00
			city	6.00	0.00	0.00	0.00	0.00	6.00	21.67	53.07	0.00	130.00	6.00	0.00	0.00	0.00	0.00	6.00	91.67	106.85	0.00	250.00

Figure 5: Grouped Summary Statistics for Activities by All Demographic Variables (Females)

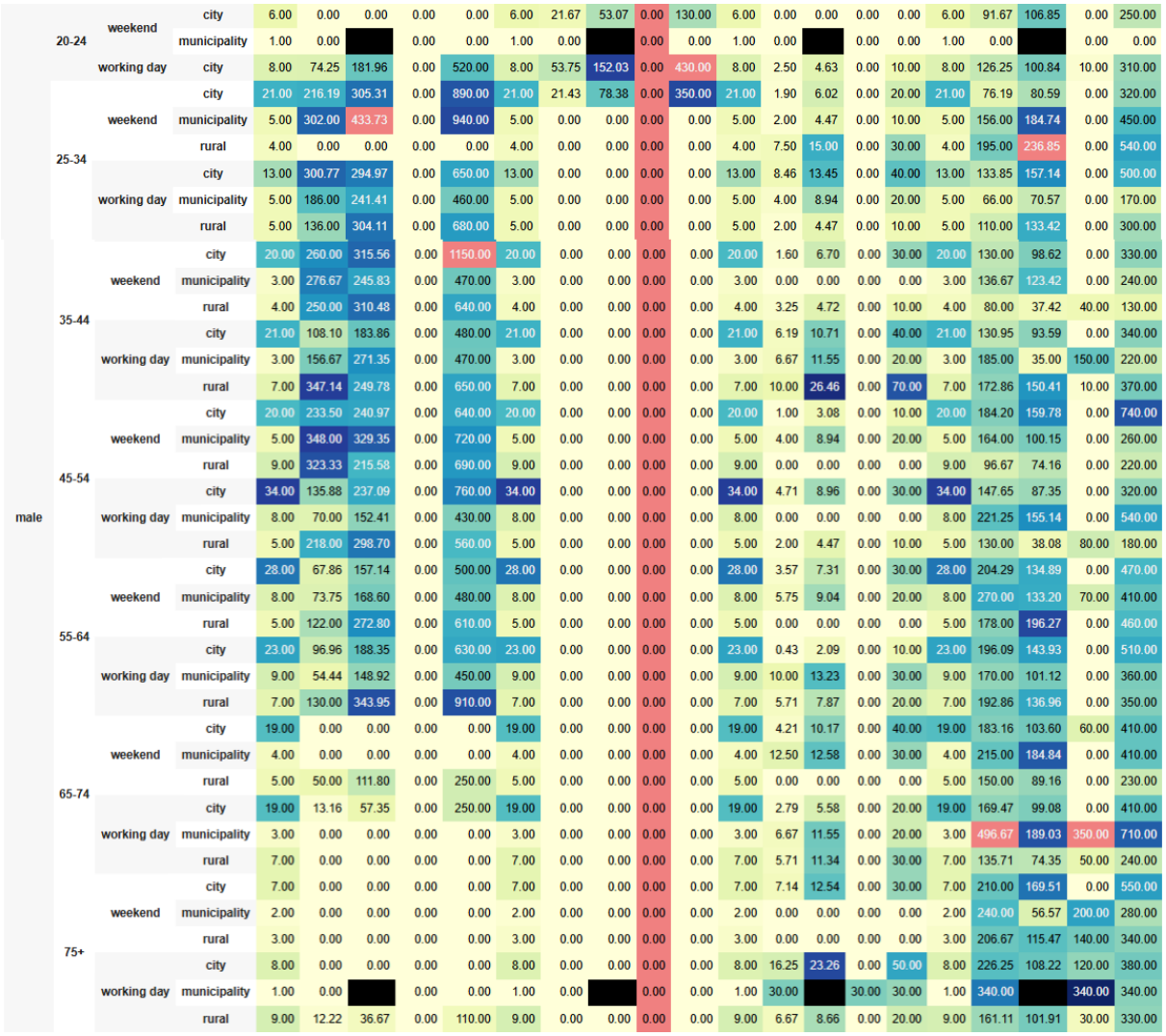


Figure 6: Grouped Summary Statistics for Activities by All Demographic Variables (Males)

To visualize the general distribution of age groups and explore how age and gender interact with other demographic variables, we created bar plots for the following relationships:

- Distribution of age groups by gender.
- Distribution of age groups by place.
- Distribution of age groups by day type.
- Distribution of gender by place.

The following code snippet was used to generate these visualizations:

```
# General distribution of age groups, age groups based on gender, age
    groups based on place, and gender based on place
x_columns = ['age', 'age', 'age', 'sex']
y_columns = ['sex', 'place', 'day', 'place']
titles = ['Distribution of Age Groups by Sex', 'Distribution of Age Groups
    by Place',
          'Distribution of Age Groups by Day', 'Distribution of Sex by
             Place']
x_labels = ['Age Group', 'Age Group', 'Age Group', 'Sex']
```

```
countplots(x_columns, y_columns, titles, x_labels)
```

Listing 24: Generating Bar Plots for Demographic Relationships

The resulting visualizations are shown below:

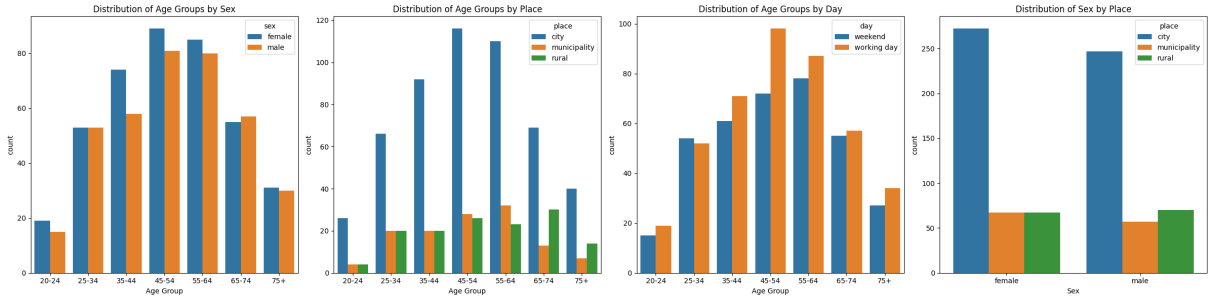


Figure 7: Demographic Relationships Between Age and Gender, Age and Place, Age and Day, Gender and Place

To further explore the distribution of time spent on different activities, we visualized histograms and box plots for each variable. Histograms help in understanding the frequency distribution, while box plots provide insights into the spread, outliers, and central tendency of the data. The Python code used for these visualizations is as follows:

```
# Histograms of working, studying, washing dishes, and watching tv
plt.figure(figsize=(22, 6))
i = 0
for column in numerical_columns:
    plt.subplot(1, len(numerical_columns), i + 1)
    sns.histplot(data=df, x=column, kde=True)
    plt.title(f'Distribution of {column.capitalize()} Time')
    plt.xlabel(f'{column.capitalize()} Time')
    i += 1
plt.tight_layout()

# Box plots of working, studying, washing dishes, and watching tv
plt.figure(figsize=(22, 6))
i = 0
for column in numerical_columns:
    plt.subplot(1, len(numerical_columns), i + 1)
    sns.boxplot(data=df, x=column)
    plt.title(f'Box plot of {column.capitalize()} Time')
    plt.xlabel(f'{column.capitalize()} Time')
    i += 1

plt.tight_layout()
```

Listing 25: Histograms and Box Plots for Numerical Variables

The histograms for the numerical variables are shown below:

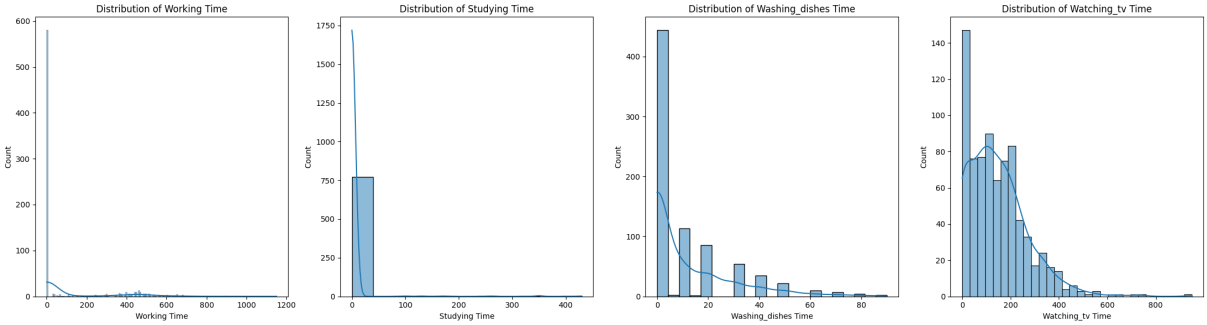


Figure 8: Histograms Showing the Distribution of Activity-Related Numerical Variables

Additionally, the box plots for these variables are presented here:

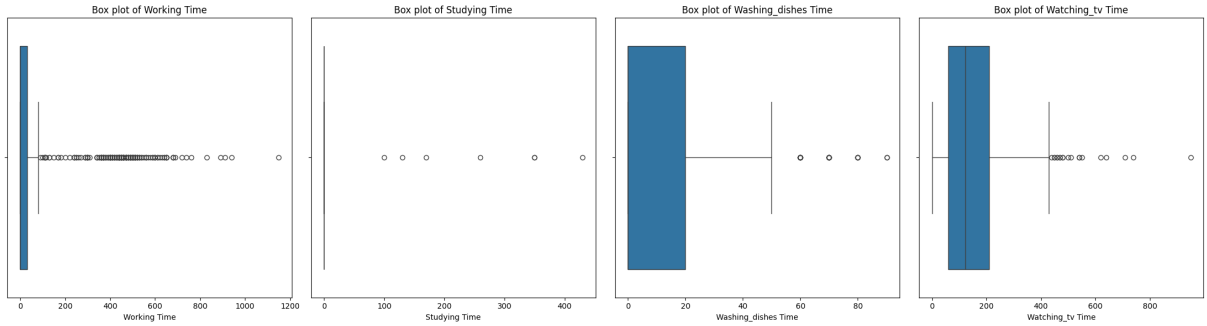


Figure 9: Box Plots Showing the Variability and Outliers of Activity-Related Numerical Variables

From the visualizations, the following key findings were observed:

- `Working`:
  - The histogram reveals a highly skewed distribution, with most individuals reporting minimal working times.
  - The box plot shows numerous outliers at high values, with some individuals reporting working times exceeding 1000 minutes.
- `Studying`:
  - The majority of individuals report no or little studying time.
  - Based on the box plot, we can say that there is low variability, with very few individuals dedicating significant time to study.
- `Washing Dishes`:
  - Most individuals spend minimal time on this activity, with a steep drop in frequency as time increases.
  - The box plot shows some outliers for washing dishes, but the majority of values are concentrated below 20 minutes.
- `Watching TV`:
  - The histogram shows a broader distribution compared to other activities.

– The box plot also highlights significant variability in watching TV times.

Moving forward, we analyzed the distributions of the activity variables based on demographic categories. The Python code used for these visualizations is shown below:

```
# Distributions of working, studying, washing dishes and watching TV based
    on demographics
x_columns = numerical_columns

for col in demographic_columns:
    y_columns = len(x_columns) * [col]
    name = col.capitalize()
    titles = [f'Distribution of Working Time by {name}',
              f'Distribution of Studying Time by {name}',
              f'Distribution of Washing Dishes Time by {name}',
              f'Distribution of Watching TV Time by {name}']
    x_labels = ['Working Time', 'Studying Time', 'Washing Dishes Time', '
        Watching TV Time']

    kdeplots(x_columns, y_columns, titles, x_labels)
```

Listing 26: Distributions of Activities by Demographics

The resulting visualizations are split into four figures based on the demographic categories:

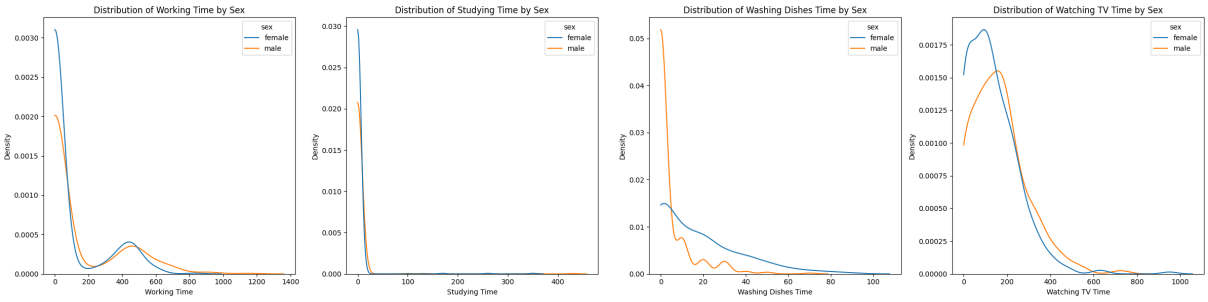


Figure 10: Distributions of Activities Based on Gender

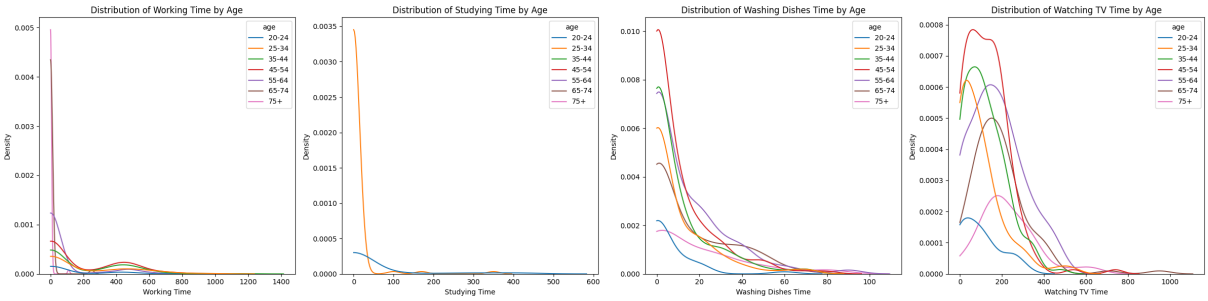


Figure 11: Distributions of Activities Based on Age Groups

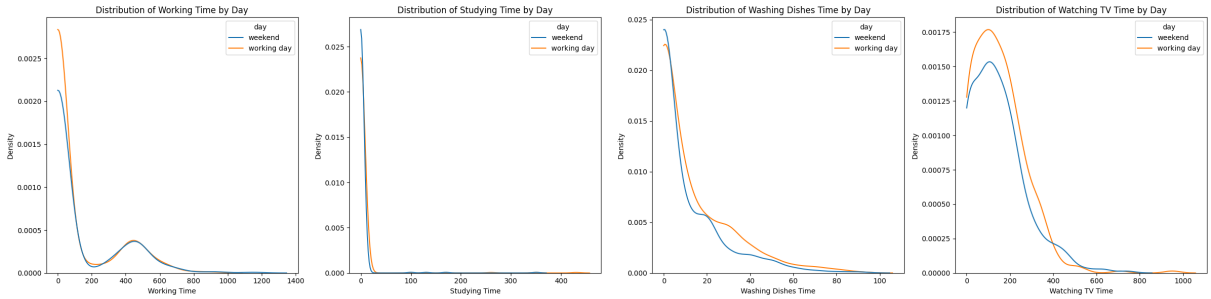


Figure 12: Distributions of Activities Based on Day Type

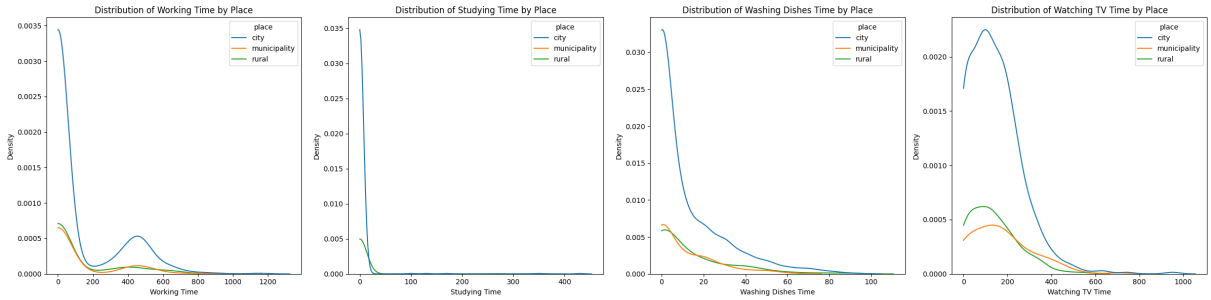


Figure 13: Distributions of Activities Based on Place

From these visualizations, the following key findings were observed:

- `Sex`:
  - Males report slightly higher `working` times compared to females.
  - On the other hand, females spend more time on `washing dishes`.
  - `Watching TV` and `studying` distributions are similar for both genders.
- `Age`:
  - The distributions reveal differences between age groups, but no clear or consistent pattern can be identified.
- `Day`:
  - `Washing dishes` and `watching TV` times are higher on `weekends`.
  - `Studying` and `working` remain relatively consistent across day types.
- `Place`:
  - There is a lot more individuals living in the city, so the histogram is imbalanced, which may lead to the belief there is a difference. Statistical tests are needed to confirm or reject this.

5 K means

In order to find groups of similar individuals, we decided to employ the K Means algorithm. K Means requires data to be scaled, i.e the data mean to be 0 and the standard deviation to be 1. To do this, we used the StandardScaler from skicit. Code for this is given below. It should be noted that clustering was done **only based on the numerical columns** of the data.

```
scaler = StandardScaler()
k_means_scaled_data = scaler.fit_transform(k_means_data[numerical_columns])
```

Listing 27: Normalizing the data using StandardScaler

Since K Means requires the number of clusters to be given as a parameter, we employed 3 different techniques in order to try and find the optimal value for the number of clusters (k). The methods we used are:

- The Elbow Method. The Elbow Method involves plotting the sum of squared distances (also called inertia or within-cluster sum of squares) between points and their respective cluster centroids as a function of the number of clusters, k. It is a graphical way of determining the optimal k. The person examines the graph and at the *elbow point* reads the optimal value for k. More can be read at this link. The optimal k we got from this method was $k = 5$. Code for this is given below, and the corresponding plot in 14.

```
    # Range of k values to try
    inertia_values = []
    K_range = range(1, 12)
    for k in K_range:
        kmeans = KMeans(n_clusters=k, random_state=42)
        kmeans.fit(k_means_scaled_data)
        inertia_values.append(kmeans.inertia_)

    # Plotting the results
    plt.plot(K_range, inertia_values, 'bo-')
    plt.xlabel('Number of Clusters (k)')
    plt.ylabel('Inertia')
    plt.title('Elbow Method for Optimal k')
    plt.show()
```

Listing 28: Using the Elbow Method to Find Optimal k

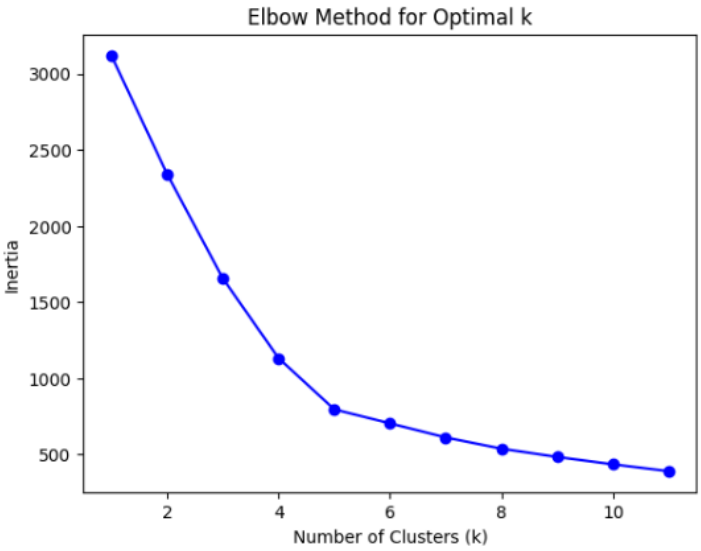


Figure 14: The Elbow Method for Finding Optimal Number of Clusters

- Silhouette Score. The Silhouette Score measures how similar each point is to its cluster compared to other clusters. The score ranges from -1 to 1, with higher values indicating better-defined clusters. More can be read at this link. The optimal k we got from this method was $k = 4$. The code for this method is given below, and the corresponding plot in 15.

```
    silhouette_scores = []
    silhouette_range = range(2, 10)
    for k in silhouette_range:  # Starting from 2 as silhouette score
        is not defined for k=1
        kmeans = KMeans(n_clusters=k, random_state=42)
        cluster_labels = kmeans.fit_predict(k_means_scaled_data)
        silhouette_avg = silhouette_score(k_means_scaled_data,
            cluster_labels)
        silhouette_scores.append(silhouette_avg)

    plt.plot(silhouette_range, silhouette_scores, 'bo-')
    plt.xlabel('Number of Clusters (k)')
    plt.ylabel('Silhouette Score')
    plt.title('Silhouette Score for Optimal k')
    plt.show()
```

Listing 29: Using the Silhouette Score to Find Optimal k

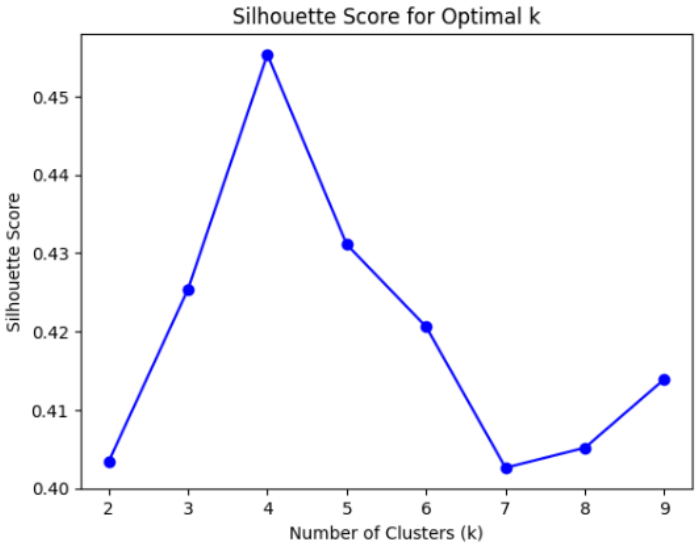


Figure 15: The Silhouette Score for Finding Optimal Number of Clusters

- Davies-Bouldin Index. The Davies-Bouldin Index measures the average similarity ratio of each cluster to its most similar cluster. Lower values indicate better clustering. More can be read at this link. The optimal k we got from this method was $k = 3$ and $k = 5$. The code for this method is given below, and the corresponding plot in 16.

```
    davies_bouldin_scores = []
    db_range = range(2, 11)
    for k in db_range:
        kmeans = KMeans(n_clusters=k, random_state=42)
        cluster_labels = kmeans.fit_predict(k_means_scaled_data)
        db_score = davies_bouldin_score(k_means_scaled_data,
            cluster_labels)
        davies_bouldin_scores.append(db_score)

    plt.plot(db_range, davies_bouldin_scores, 'bo-')
    plt.xlabel('Number of Clusters (k)')
    plt.ylabel('Davies-Bouldin Index')
    plt.title('Davies-Bouldin Index for Optimal k')
    plt.show()
```

Listing 30: Using the Davis-Boulding Index to Find Optimal k

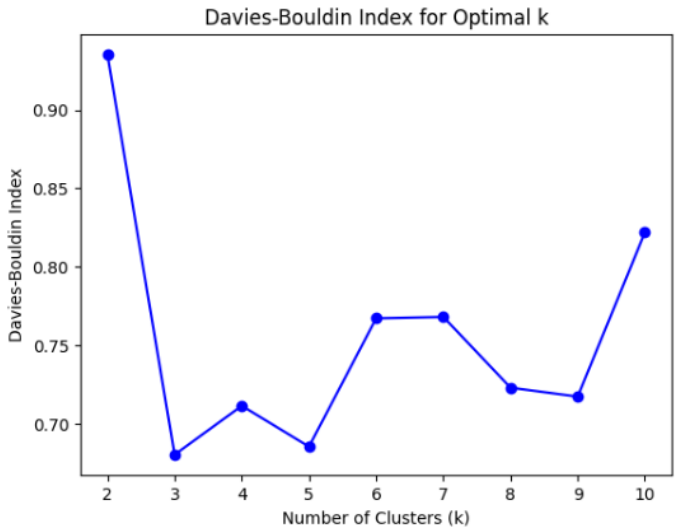


Figure 16: The Davies-Boulding Index for Finding Optimal Number of Clusters

Given that all three methods produced the optimal k in the range $[3, 4, 5]$, with two of them outputting 5, we made the conclusion to settle for $k = 5$ for further analysis.
K Means algorithm is given in the code below for $k = 5$.

```
kmeans = KMeans(n_clusters=5, random_state=42)
k_means_data['cluster'] = kmeans.fit_predict(k_means_scaled_data)
```

Listing 31: K Means for k equals 5

Firstly the size od each cluster is calculated and listed down below:

```
# Access the cluster sizes
cluster_sizes = k_means_data['cluster'].value_counts()

# Print the size of each cluster
print(cluster_sizes)
```

Listing 32: Cluster Sizes

Cluster	Size
1	321
4	171
0	161
3	123
2	4

Table 5: Size of Each Cluster

Next, we analyzed the properties of different clusters obtained with the K Means algorithm. Firstly we checked the characteristics of categorical and numerical columns within each cluster. Code for this is given below:

```
# Analyze clusters by checking the characteristics of columns within each
    cluster
# Get a summary of categorical columns within each cluster
categorical_columns = ['sex', 'age', 'day', 'place', 'cinema', 'sport']
```

```
extended_demographic_columns = ['cluster'] + demographic_columns

cluster_summary = k_means_data.groupby('cluster', observed=False)[
    categorical_columns].agg(lambda x: x.value_counts().index[0])

# Display cluster summaries
print("Cluster summary based on most common categorical values in each
    group:")
print(cluster_summary)

# Show average values of numerical columns per cluster
numerical_summary = k_means_data.groupby('cluster', observed=False)[
    numerical_columns].mean()
print("\nNumerical summary (mean values) per cluster:")
print(numerical_summary)
```

Listing 33: Cluster summary based on categorical and numerical columns

The results are given below:

Cluster	Sex	Age	Day	Place	Cinema	Sport
0	Male	45-54	Weekend	City	Yes	Yes
1	Male	55-64	Working Day	City	No	No
2	Female	20-24	Working Day	City	Yes	No
3	Female	65-74	Working Day	City	No	No
4	Male	55-64	Working Day	City	No	No

Table 6: Cluster Summary Based on Most Common Categorical Values in Each Group

Cluster	Working	Studying	Washing Dishes	Watching TV
0	478.70	0.00	4.18	86.19
1	5.48	1.25	4.69	98.23
2	15.00	347.50	2.50	30.00
3	9.11	0.00	43.66	127.46
4	11.64	0.00	6.09	313.16

Table 7: Numerical Summary (Mean Values) Per Cluster Given in Minutes

We then explored how activities vary across different demographic groups and clusters. To achieve this, we calculated descriptive statistics (`count`, `mean`, `std`, `min`, `max`) for each numerical variable grouped by the demographic categories, extended with clusters. The Python code snippet used for this analysis is as follows:

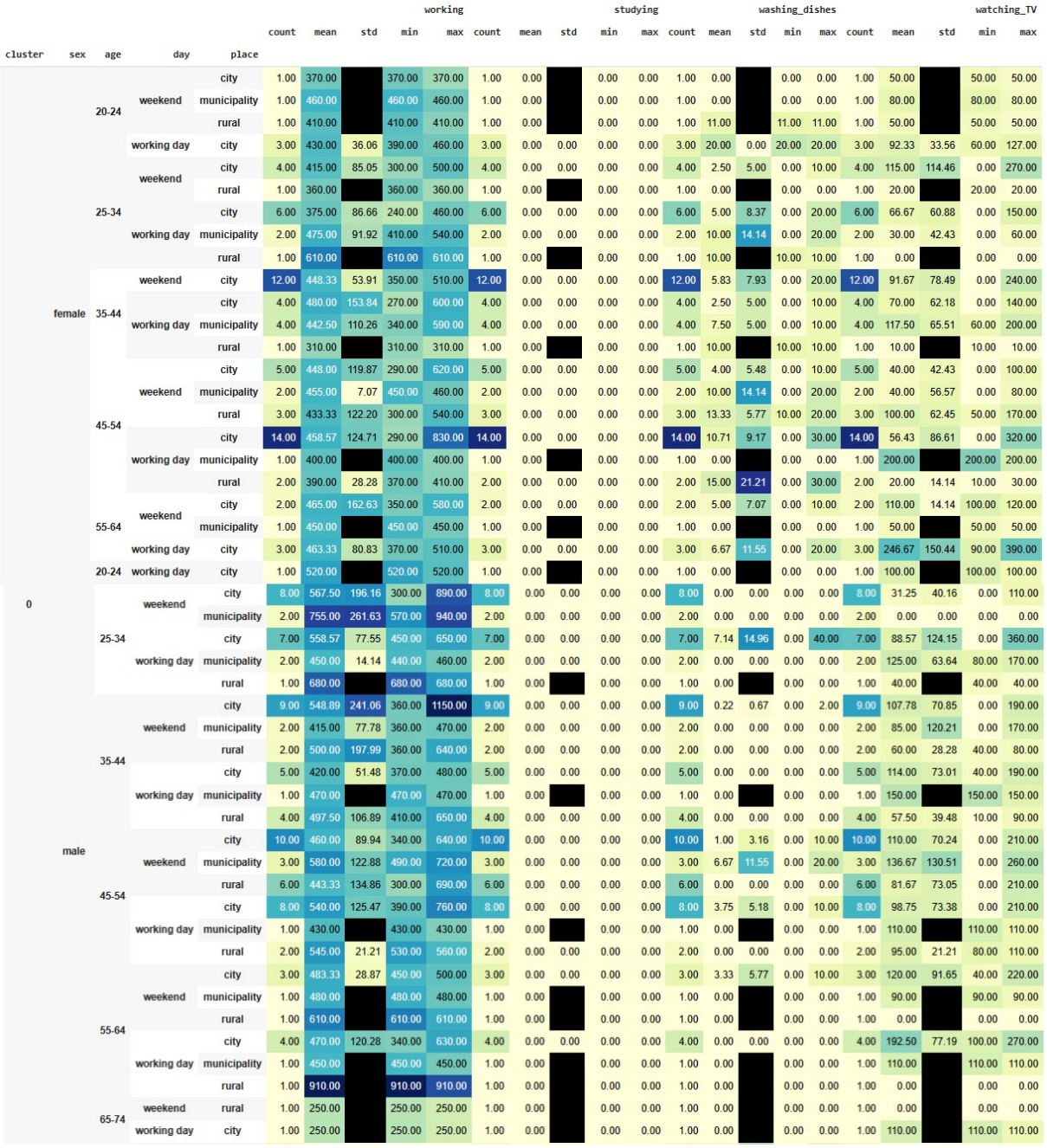
```
# Grouped cluster summary
grouped_summary = k_means_data.groupby(extended_demographic_columns,
    observed=False)[numerical_columns].describe().loc[:, (slice(None),
    summary_stats)]
styled_summary = grouped_summary.style.format("{:.2f}") \
                                      .set_caption(f"Summary Statistics for
                                          Activities by cluster") \
                                      .background_gradient(cmap="YlGnBu",
                                         subset=pd.IndexSlice[:, pd.
                                         IndexSlice[:, :]])

```

```
display(styled_summary)
```

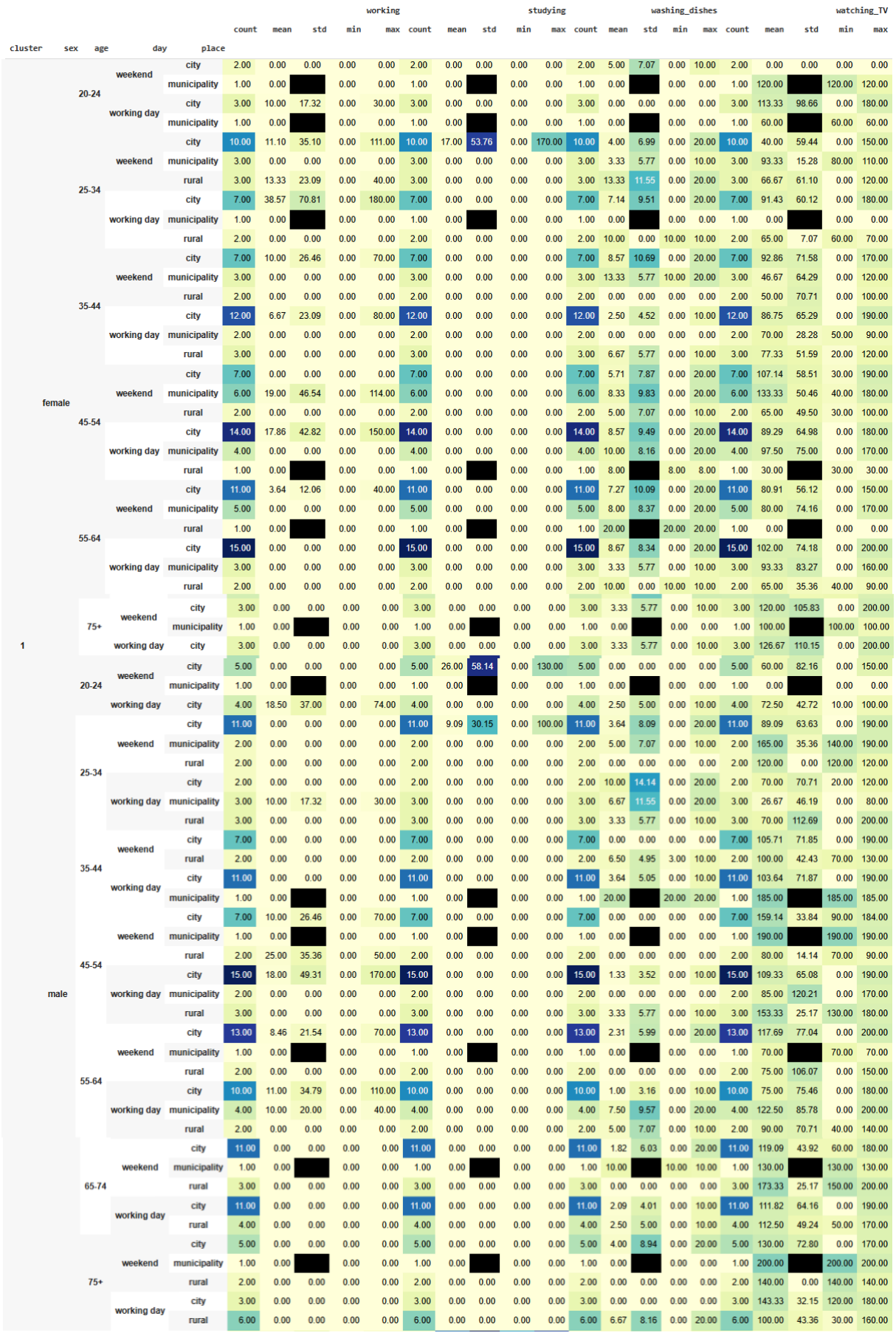
Listing 34: Grouped Cluster Summary

The output of the clustering analysis code is presented below. Each figure corresponds to one of the five clusters identified during the K-means clustering process.



					working					studying					washing_dishes					watching_TV			
					count	mean	std	min	max	count	mean	std	min	max	count	mean	std	min	max	count	mean	std	min	max
cluster	sex	age	day	place																			
0	female	20-24	weekend	city	1.00	370.00		370.00	370.00	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	50.00		50.00	50.00
				municipality	1.00	460.00		460.00	460.00	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	80.00		80.00	80.00
				rural	1.00	410.00		410.00	410.00	1.00	0.00		0.00	0.00	1.00	11.00		11.00	11.00	1.00	50.00		50.00	50.00
			working day	city	3.00	430.00	36.06	390.00	460.00	3.00	0.00	0.00	0.00	0.00	3.00	20.00	0.00	20.00	20.00	3.00	92.33	33.56	60.00	127.00
			weekend	city	4.00	415.00	85.05	300.00	500.00	4.00	0.00	0.00	0.00	0.00	4.00	2.50	5.00	0.00	10.00	4.00	115.00	114.46	0.00	270.00
				rural	1.00	360.00		360.00	360.00	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	20.00		20.00	20.00
		25-34		city	6.00	375.00	86.66	240.00	460.00	6.00	0.00	0.00	0.00	0.00	6.00	5.00	8.37	0.00	20.00	6.00	66.67	60.88	0.00	150.00
			working day	municipality	2.00	475.00	91.92	410.00	540.00	2.00	0.00	0.00	0.00	0.00	2.00	10.00	14.14	0.00	20.00	2.00	30.00	42.43	0.00	60.00
				rural	1.00	610.00		610.00	610.00	1.00	0.00		0.00	0.00	1.00	10.00		10.00	10.00	1.00	0.00		0.00	0.00
			weekend	city	12.00	448.33	53.91	350.00	510.00	12.00	0.00	0.00	0.00	0.00	12.00	5.83	7.93	0.00	20.00	12.00	91.67	78.49	0.00	240.00
			working day	city	4.00	480.00	153.84	270.00	600.00	4.00	0.00	0.00	0.00	0.00	4.00	2.50	5.00	0.00	10.00	4.00	70.00	62.18	0.00	140.00
				municipality	4.00	442.50	110.26	340.00	590.00	4.00	0.00	0.00	0.00	0.00	4.00	7.50	5.00	0.00	10.00	4.00	117.50	65.51	60.00	200.00
				rural	1.00	310.00		310.00	310.00	1.00	0.00		0.00	0.00	1.00	10.00		10.00	10.00	1.00	10.00		10.00	10.00
			weekend	city	5.00	448.00	119.87	290.00	620.00	5.00	0.00	0.00	0.00	0.00	5.00	4.00	5.48	0.00	10.00	5.00	40.00	42.43	0.00	100.00
				municipality	2.00	455.00	7.07	450.00	460.00	2.00	0.00	0.00	0.00	0.00	2.00	10.00	14.14	0.00	20.00	2.00	40.00	56.57	0.00	80.00
				rural	3.00	433.33	122.20	300.00	540.00	3.00	0.00	0.00	0.00	0.00	3.00	13.33	5.77	10.00	20.00	3.00	100.00	62.45	50.00	170.00
		45-54		city	14.00	458.57	124.71	290.00	830.00	14.00	0.00	0.00	0.00	0.00	14.00	10.71	9.17	0.00	30.00	14.00	56.43	86.61	0.00	320.00
			working day	municipality	1.00	400.00		400.00	400.00	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	200.00		200.00	200.00
				rural	2.00	390.00	28.28	370.00	410.00	2.00	0.00	0.00	0.00	0.00	2.00	15.00	21.21	0.00	30.00	2.00	20.00	14.14	10.00	30.00
			weekend	city	2.00	465.00	162.63	350.00	580.00	2.00	0.00	0.00	0.00	0.00	2.00	5.00	7.07	0.00	10.00	2.00	110.00	14.14	100.00	120.00
		55-64		municipality	1.00	450.00		450.00	450.00	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	50.00		50.00	50.00
			working day	city	3.00	463.33	80.83	370.00	510.00	3.00	0.00	0.00	0.00	0.00	3.00	6.67	11.55	0.00	20.00	3.00	246.67	150.44	90.00	390.00
		20-24	working day	city	1.00	520.00		520.00	520.00	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	100.00		100.00	100.00
	male	25-34	weekend	city	8.00	567.50	196.16	300.00	890.00	8.00	0.00	0.00	0.00	0.00	8.00	0.00	0.00	0.00	0.00	8.00	31.25	40.16	0.00	110.00
				municipality	2.00	755.00	261.63	570.00	940.00	2.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00
			working day	city	7.00	558.57	77.55	450.00	650.00	7.00	0.00	0.00	0.00	0.00	7.00	7.14	14.96	0.00	40.00	7.00	88.57	124.15	0.00	360.00
				municipality	2.00	450.00	14.14	440.00	460.00	2.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	2.00	125.00	63.64	80.00	170.00
				rural	1.00	680.00		680.00	680.00	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	40.00		40.00	40.00
		35-44	weekend	city	9.00	548.89	241.06	360.00	1150.00	9.00	0.00	0.00	0.00	0.00	9.00	0.22	0.67	0.00	2.00	9.00	107.78	70.85	0.00	190.00
				municipality	2.00	415.00	77.78	360.00	470.00	2.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	2.00	85.00	120.21	0.00	170.00
				rural	2.00	500.00	197.99	360.00	640.00	2.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	2.00	60.00	28.28	40.00	80.00
			working day	city	5.00	420.00	51.48	370.00	480.00	5.00	0.00	0.00	0.00	0.00	5.00	0.00	0.00	0.00	0.00	5.00	114.00	73.01	40.00	190.00
				municipality	1.00	470.00		470.00	470.00	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	150.00		150.00	150.00
				rural	4.00	497.50	106.89	410.00	650.00	4.00	0.00	0.00	0.00	0.00	4.00	0.00	0.00	0.00	0.00	4.00	57.50	39.48	10.00	90.00
			weekend	city	10.00	460.00	89.94	340.00	640.00	10.00	0.00	0.00	0.00	0.00	10.00	1.00	3.16	0.00	10.00	10.00	110.00	70.24	0.00	210.00
				municipality	3.00	580.00	122.88	490.00	720.00	3.00	0.00	0.00	0.00	0.00	3.00	6.67	11.55	0.00	20.00	3.00	136.67	130.51	0.00	260.00
				rural	6.00	443.33	134.86	300.00	690.00	6.00	0.00	0.00	0.00	0.00	6.00	0.00	0.00	0.00	0.00	6.00	81.67	73.05	0.00	210.00
		45-54		city	8.00	540.00	125.47	390.00	760.00	8.00	0.00	0.00	0.00	0.00	8.00	3.75	5.18	0.00	10.00	8.00	98.75	73.38	0.00	210.00
			working day	municipality	1.00	430.00		430.00	430.00	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	110.00		110.00	110.00
				rural	2.00	545.00	21.21	530.00	560.00	2.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	2.00	95.00	21.21	80.00	110.00
				city	3.00	483.33	28.87	450.00	500.00	3.00	0.00	0.00	0.00	0.00	3.00	3.33	5.77	0.00	10.00	3.00	120.00	91.65	40.00	220.00
			weekend	municipality	1.00	480.00		480.00	480.00	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	90.00		90.00	90.00
				rural	1.00	610.00		610.00	610.00	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00
		55-64		city	4.00	470.00	120.28	340.00	630.00	4.00	0.00	0.00	0.00	0.00	4.00	0.00	0.00	0.00	0.00	4.00	192.50	77.19	100.00	270.00
			working day	municipality	1.00	450.00		450.00	450.00	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	110.00		110.00	110.00
				rural	1.00	910.00		910.00	910.00	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00
		65-74	weekend	rural	1.00	250.00		250.00	250.00	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00
			working day	city	1.00	250.00		250.00	250.00	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	110.00		110.00	110.00

Figure 17: Summary statistics for activities in **Cluster 0**.



					working					studying				washing_dishes				watching_TV			
					count	mean	std	min	max	count	mean	std	min	max	count	mean	std	min	max	count	mean	std	min	max
cluster	sex	age	day	place																			
1	female	20-24	weekend	city	2.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	2.00	5.00	7.07	0.00	10.00	2.00	0.00	0.00	0.00	0.00
				municipality	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	120.00		120.00	120.00
			working day	city	3.00	10.00	17.32	0.00	30.00	3.00	0.00	0.00	0.00	0.00	3.00	0.00	0.00	0.00	0.00	3.00	113.33	98.66	0.00	180.00
				municipality	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	60.00		60.00	60.00
		25-34	weekend	city	10.00	11.10	35.10	0.00	111.00	10.00	17.00	53.76	0.00	170.00	10.00	4.00	6.99	0.00	20.00	10.00	40.00	59.44	0.00	150.00
				municipality	3.00	0.00	0.00	0.00	0.00	3.00	0.00	0.00	0.00	0.00	3.00	3.33	5.77	0.00	10.00	3.00	93.33	15.28	80.00	110.00
				rural	3.00	13.33	23.09	0.00	40.00	3.00	0.00	0.00	0.00	0.00	3.00	13.33	11.55	0.00	20.00	3.00	66.67	61.10	0.00	120.00
			working day	city	7.00	38.57	70.81	0.00	180.00	7.00	0.00	0.00	0.00	0.00	7.00	7.14	9.51	0.00	20.00	7.00	91.43	60.12	0.00	180.00
				municipality	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00
				rural	2.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	2.00	10.00	0.00	10.00	10.00	2.00	65.00	7.07	60.00	70.00
			weekend	city	7.00	10.00	26.46	0.00	70.00	7.00	0.00	0.00	0.00	0.00	7.00	8.57	10.69	0.00	20.00	7.00	92.86	71.58	0.00	170.00
				municipality	3.00	0.00	0.00	0.00	0.00	3.00	0.00	0.00	0.00	0.00	3.00	13.33	5.77	10.00	20.00	3.00	46.67	64.29	0.00	120.00
				rural	2.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	2.00	50.00	70.71	0.00	100.00
		35-44	working day	city	12.00	6.67	23.09	0.00	80.00	12.00	0.00	0.00	0.00	0.00	12.00	2.50	4.52	0.00	10.00	12.00	86.75	65.29	0.00	190.00
				municipality	2.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	2.00	70.00	28.28	50.00	90.00
				rural	3.00	0.00	0.00	0.00	0.00	3.00	0.00	0.00	0.00	0.00	3.00	6.67	5.77	0.00	10.00	3.00	77.33	51.59	20.00	120.00
			weekend	city	7.00	0.00	0.00	0.00	0.00	7.00	0.00	0.00	0.00	0.00	7.00	5.71	7.87	0.00	20.00	7.00	107.14	58.51	30.00	190.00
				municipality	6.00	19.00	46.54	0.00	114.00	6.00	0.00	0.00	0.00	0.00	6.00	8.33	9.83	0.00	20.00	6.00	133.33	50.46	40.00	180.00
				rural	2.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	2.00	5.00	7.07	0.00	10.00	2.00	65.00	49.50	30.00	100.00
		45-54	working day	city	14.00	17.86	42.82	0.00	150.00	14.00	0.00	0.00	0.00	0.00	14.00	8.57	9.49	0.00	20.00	14.00	89.29	64.98	0.00	180.00
				municipality	4.00	0.00	0.00	0.00	0.00	4.00	0.00	0.00	0.00	0.00	4.00	10.00	8.16	0.00	20.00	4.00	97.50	75.00	0.00	170.00
				rural	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	8.00		8.00	8.00	1.00	30.00		30.00	30.00
			weekend	city	11.00	3.64	12.06	0.00	40.00	11.00	0.00	0.00	0.00	0.00	11.00	7.27	10.09	0.00	20.00	11.00	80.91	56.12	0.00	150.00
				municipality	5.00	0.00	0.00	0.00	0.00	5.00	0.00	0.00	0.00	0.00	5.00	8.00	8.37	0.00	20.00	5.00	80.00	74.16	0.00	170.00
				rural	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	20.00		20.00	20.00	1.00	0.00		0.00	0.00
		55-64	working day	city	15.00	0.00	0.00	0.00	0.00	15.00	0.00	0.00	0.00	0.00	15.00	8.67	8.34	0.00	20.00	15.00	102.00	74.18	0.00	200.00
				municipality	3.00	0.00	0.00	0.00	0.00	3.00	0.00	0.00	0.00	0.00	3.00	3.33	5.77	0.00	10.00	3.00	93.33	83.27	0.00	160.00
				rural	2.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	2.00	10.00	0.00	10.00	10.00	2.00	65.00	35.36	40.00	90.00
		75+	weekend	city	3.00	0.00	0.00	0.00	0.00	3.00	0.00	0.00	0.00	0.00	3.00	3.33	5.77	0.00	10.00	3.00	120.00	105.83	0.00	200.00
				municipality	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	100.00		100.00	100.00
			working day	city	3.00	0.00	0.00	0.00	0.00	3.00	0.00	0.00	0.00	0.00	3.00	3.33	5.77	0.00	10.00	3.00	126.67	110.15	0.00	200.00
	male	20-24	weekend	city	5.00	0.00	0.00	0.00	0.00	5.00	26.00	58.14	0.00	130.00	5.00	0.00	0.00	0.00	0.00	5.00	60.00	82.16	0.00	150.00
				municipality	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00
			working day	city	4.00	18.50	37.00	0.00	74.00	4.00	0.00	0.00	0.00	0.00	4.00	2.50	5.00	0.00	10.00	4.00	72.50	42.72	10.00	100.00
				city	11.00	0.00	0.00	0.00	0.00	11.00	9.09	30.15	0.00	100.00	11.00	3.64	8.09	0.00	20.00	11.00	89.09	63.63	0.00	190.00
			weekend	municipality	2.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	2.00	5.00	7.07	0.00	10.00	2.00	165.00	35.36	140.00	190.00
				rural	2.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	2.00	120.00	0.00	120.00	120.00
		25-34	working day	city	2.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	2.00	10.00	14.14	0.00	20.00	2.00	70.00	70.71	20.00	120.00
				municipality	3.00	10.00	17.32	0.00	30.00	3.00	0.00	0.00	0.00	0.00	3.00	6.67	11.55	0.00	20.00	3.00	26.67	46.19	0.00	80.00
				rural	3.00	0.00	0.00	0.00	0.00	3.00	0.00	0.00	0.00	0.00	3.00	3.33	5.77	0.00	10.00	3.00	70.00	112.69	0.00	200.00
			weekend	city	7.00	0.00	0.00	0.00	0.00	7.00	0.00	0.00	0.00	0.00	7.00	0.00	0.00	0.00	0.00	7.00	105.71	71.85	0.00	190.00
				rural	2.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	2.00	6.50	4.95	3.00	10.00	2.00	100.00	42.43	70.00	130.00
		35-44	working day	city	11.00	0.00	0.00	0.00	0.00	11.00	0.00	0.00	0.00	0.00	11.00	3.64	5.05	0.00	10.00	11.00	103.64	71.87	0.00	190.00
				municipality	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	20.00		20.00	20.00	1.00	185.00		185.00	185.00
			weekend	city	7.00	10.00	26.46	0.00	70.00	7.00	0.00	0.00	0.00	0.00	7.00	0.00	0.00	0.00	0.00	7.00	159.14	33.84	90.00	184.00
				municipality	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	190.00		190.00	190.00
				rural	2.00	25.00	35.36	0.00	50.00	2.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	2.00	80.00	14.14	70.00	90.00
		45-54	working day	city	15.00	18.00	49.31	0.00	170.00	15.00	0.00	0.00	0.00	0.00	15.00	1.33	3.52	0.00	10.00	15.00	109.33	65.08	0.00	190.00
				municipality	2.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	2.00	85.00	120.21	0.00	170.00
				rural	3.00	0.00	0.00	0.00	0.00	3.00	0.00	0.00	0.00	0.00	3.00	3.33	5.77	0.00	10.00	3.00	153.33	25.17	130.00	180.00
			weekend	city	13.00	8.46	21.54	0.00	70.00	13.00	0.00	0.00	0.00	0.00	13.00	2.31	5.99	0.00	20.00	13.00	117.69	77.04	0.00	200.00
				municipality	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	70.00		70.00	70.00
				rural	2.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	2.00	75.00	106.07	0.00	150.00
		55-64	working day	city	10.00	11.00	34.79	0.00	110.00	10.00	0.00	0.00	0.00	0.00	10.00	1.00	3.16	0.00	10.00	10.00	75.00	75.46	0.00	180.00
				municipality	4.00	10.00	20.00	0.00	40.00	4.00	0.00	0.00	0.00	0.00	4.00	7.50	9.57	0.00	20.00	4.00	122.50	85.78	0.00	200.00
				rural	2.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	2.00	5.00	7.07	0.00	10.00	2.00	90.00	70.71	40.00	140.00
			weekend	city	11.00	0.00	0.00	0.00	0.00	11.00	0.00	0.00	0.00	0.00	11.00	1.82	6.03	0.00	20.00	11.00	119.09	43.92	60.00	180.00
				municipality	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	10.00		10.00	10.00	1.00	130.00		130.00	130.00
		65-74		rural	3.00	0.00	0.00	0.00	0.00	3.00	0.00	0.00	0.00	0.00	3.00	0.00	0.00	0.00	0.00	3.00	173.33	25.17	150.00	200.00
			working day	city	11.00	0.00	0.00	0.00	0.00	11.00	0.00	0.00	0.00	0.00	11.00	2.09	4.01	0.00	10.00	11.00	111.82	64.16	0.00	190.00
				rural	4.00	0.00	0.00	0.00	0.00	4.00	0.00	0.00	0.00	0.00	4.00	2.50	5.00	0.00	10.00	4.00	112.50	49.24	50.00	170.00
				city	5.00	0.00	0.00	0.00	0.00	5.00	0.00	0.00	0.00	0.00	5.00	4.00	8.94	0.00	20.00	5.00	130.00	72.80	0.00	170.00
			weekend	municipality	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	200.00		200.00	200.00
		75+		rural	2.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	2.00	140.00	0.00	140.00	140.00
			working day	city	3.00	0.00	0.00	0.00	0.00	3.00	0.00	0.00	0.00	0.00	3.00	0.00	0.00	0.00	0.00	3.00	143.33	32.15	120.00	180.00
				rural	6.00	0.00	0.00	0.00	0.00	6.00	0.00	0.00	0.00	0.00	6.00	6.67	8.16	0.00	20.00	6.00	100.00	43.36	30.00	160.00

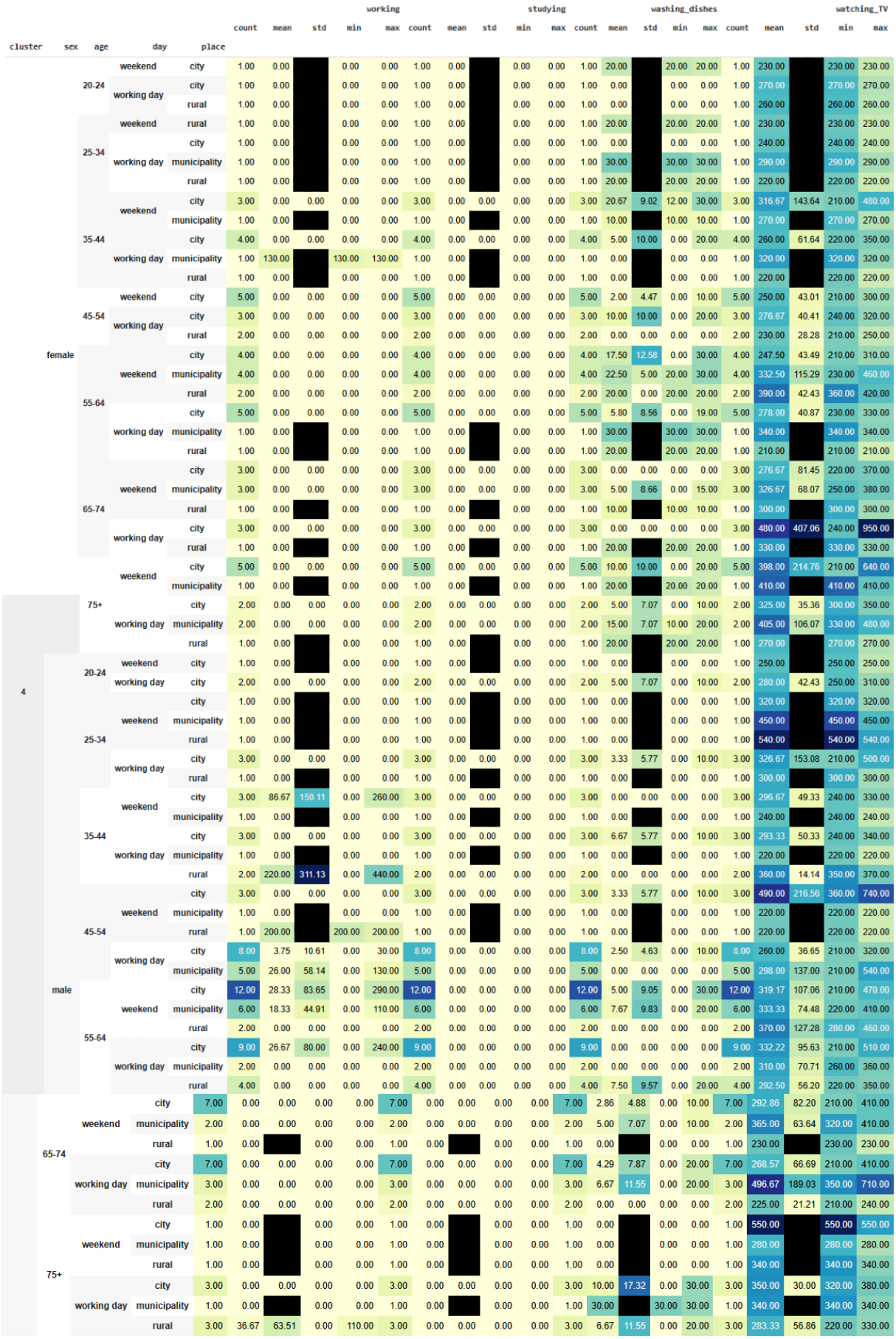
Figure 18: Summary statistics for activities in **Cluster 1**.

					working					studying					washing_dishes					watching_TV			
					count	mean	std	min	max	count	mean	std	min	max	count	mean	std	min	max	count	mean	std	min	max
cluster	sex	age	day	place																			
2	female	20-24	working day	rural	2.00	30.00	42.43	0.00	60.00	2.00	305.00	63.64	260.00	350.00	2.00	5.00	7.07	0.00	10.00	2.00	5.00	7.07	0.00	10.00
	male	20-24	working day	city	1.00	0.00		0.00	0.00	1.00	430.00		430.00	430.00	1.00	0.00		0.00	0.00	1.00	60.00		60.00	60.00
		25-34	weekend	city	1.00	0.00		0.00	0.00	1.00	350.00		350.00	350.00	1.00	0.00		0.00	0.00	1.00	50.00		50.00	50.00

Figure 19: Summary statistics for activities in **Cluster 2**.

					working					studying					washing_dishes					watching_TV			
					count	mean	std	min	max	count	mean	std	min	max	count	mean	std	min	max	count	mean	std	min	max
cluster	sex	age	day	place																			
3	female	20-24	weekend	city	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	60.00		60.00	60.00	1.00	0.00		0.00	0.00
			working day	city	2.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	2.00	60.00	14.14	50.00	70.00	2.00	210.00	141.42	110.00	310.00
		25-34		city	2.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	2.00	55.00	21.21	40.00	70.00	2.00	55.00	77.78	0.00	110.00
				municipality	3.00	0.00	0.00	0.00	0.00	3.00	0.00	0.00	0.00	0.00	3.00	43.33	15.28	30.00	60.00	3.00	40.00	34.64	0.00	60.00
				rural	2.00	120.00	169.71	0.00	240.00	2.00	0.00	0.00	0.00	0.00	2.00	30.00	0.00	30.00	30.00	2.00	80.00	113.14	0.00	160.00
			weekend	city	4.00	0.00	0.00	0.00	0.00	4.00	0.00	0.00	0.00	0.00	4.00	32.50	5.00	30.00	40.00	4.00	47.50	55.00	0.00	100.00
				rural	2.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	2.00	40.00	0.00	40.00	40.00	2.00	75.00	21.21	60.00	90.00
		35-44	working day	city	5.00	0.00	0.00	0.00	0.00	5.00	0.00	0.00	0.00	0.00	5.00	48.00	17.89	30.00	70.00	5.00	32.00	20.49	0.00	50.00
				municipality	3.00	0.00	0.00	0.00	0.00	3.00	0.00	0.00	0.00	0.00	3.00	40.00	10.00	30.00	50.00	3.00	150.00	81.85	60.00	220.00
				city	5.00	6.00	13.42	0.00	30.00	5.00	0.00	0.00	0.00	0.00	5.00	48.00	14.83	30.00	70.00	5.00	104.00	101.64	0.00	230.00
			weekend	municipality	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	50.00		50.00	50.00	1.00	0.00		0.00	0.00
		45-54		rural	2.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	2.00	45.00	7.07	40.00	50.00	2.00	25.00	7.07	20.00	30.00
			working day	city	9.00	24.44	73.33	0.00	220.00	9.00	0.00	0.00	0.00	0.00	9.00	42.22	17.87	30.00	80.00	9.00	73.33	45.00	0.00	120.00
				municipality	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	40.00		40.00	40.00	1.00	140.00		140.00	140.00
			weekend	city	7.00	65.71	173.86	0.00	460.00	7.00	0.00	0.00	0.00	0.00	7.00	54.29	18.13	40.00	90.00	7.00	128.57	126.02	0.00	290.00
		55-64		city	12.00	0.00	0.00	0.00	0.00	12.00	0.00	0.00	0.00	0.00	12.00	40.00	11.28	30.00	70.00	12.00	148.33	111.99	0.00	400.00
			working day	municipality	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	50.00		50.00	50.00	1.00	290.00		290.00	290.00
				rural	5.00	0.00	0.00	0.00	0.00	5.00	0.00	0.00	0.00	0.00	5.00	52.00	23.87	30.00	90.00	5.00	156.00	37.82	120.00	210.00
				city	5.00	0.00	0.00	0.00	0.00	5.00	0.00	0.00	0.00	0.00	5.00	38.00	8.37	30.00	50.00	5.00	118.00	70.50	60.00	240.00
			weekend	municipality	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	50.00		50.00	50.00	1.00	160.00		160.00	160.00
				rural	4.00	0.00	0.00	0.00	0.00	4.00	0.00	0.00	0.00	0.00	4.00	47.50	9.57	40.00	60.00	4.00	135.00	92.56	0.00	200.00
		65-74		city	8.00	0.00	0.00	0.00	0.00	8.00	0.00	0.00	0.00	0.00	8.00	45.00	15.12	30.00	70.00	8.00	211.25	129.55	60.00	420.00
			working day	municipality	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	40.00		40.00	40.00	1.00	220.00		220.00	220.00
				rural	7.00	0.00	0.00	0.00	0.00	7.00	0.00	0.00	0.00	0.00	7.00	40.00	8.16	30.00	50.00	7.00	144.29	86.00	0.00	280.00
			weekend	city	4.00	0.00	0.00	0.00	0.00	4.00	0.00	0.00	0.00	0.00	4.00	37.50	9.57	30.00	50.00	4.00	154.25	87.06	50.00	247.00
		75+		rural	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	80.00		80.00	80.00	1.00	120.00		120.00	120.00
			working day	city	8.00	0.00	0.00	0.00	0.00	8.00	0.00	0.00	0.00	0.00	8.00	50.00	21.38	30.00	80.00	8.00	230.12	52.89	160.00	330.00
	male	25-34	weekend	rural	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	30.00		30.00	30.00	1.00	0.00		0.00	0.00
			working day	city	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	30.00		30.00	30.00	1.00	0.00		0.00	0.00
			weekend	city	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	30.00		30.00	30.00	1.00	0.00		0.00	0.00
		35-44	working day	city	2.00	85.00	120.21	0.00	170.00	2.00	0.00	0.00	0.00	0.00	2.00	35.00	7.07	30.00	40.00	2.00	80.00	56.57	40.00	120.00
				rural	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	70.00		70.00	70.00	1.00	260.00		260.00	260.00
		45-54	working day	city	3.00	0.00	0.00	0.00	0.00	3.00	0.00	0.00	0.00	0.00	3.00	30.00	0.00	30.00	30.00	3.00	170.00	30.00	140.00	200.00
		55-64	working day	municipality	2.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	2.00	30.00	0.00	30.00	30.00	2.00	155.00	49.50	120.00	190.00
			weekend	city	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	40.00		40.00	40.00	1.00	120.00		120.00	120.00
		65-74		municipality	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	30.00		30.00	30.00	1.00	0.00		0.00	0.00
			working day	rural	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	30.00		30.00	30.00	1.00	50.00		50.00	50.00
		75+	weekend	city	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	30.00		30.00	30.00	1.00	270.00		270.00	270.00
			working day	city	2.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	2.00	50.00	0.00	50.00	50.00	2.00	165.00	63.64	120.00	210.00

Figure 20: Summary statistics for activities in **Cluster 3**.



cluster	sex	age	day	place	working					studying				washing_dishes					watching_TV			
					count	mean	std	min	max	count	mean	std	min	max	count	mean	std	min	max	count	mean	std	min	max
4	female	20-24	weekend	city	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	20.00		20.00	20.00	1.00	230.00		230.00	230.00
			working day	city	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	270.00		270.00	270.00
				rural	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	260.00		260.00	260.00
		25-34	weekend	rural	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	20.00		20.00	20.00	1.00	230.00		230.00	230.00
			working day	city	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	240.00		240.00	240.00
				municipality	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	30.00		30.00	30.00	1.00	290.00		290.00	290.00
				rural	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	20.00		20.00	20.00	1.00	220.00		220.00	220.00
			weekend	city	3.00	0.00	0.00	0.00	0.00	3.00	0.00	0.00	0.00	0.00	3.00	20.67	9.02	12.00	30.00	3.00	316.67	143.64	210.00	480.00
				municipality	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	10.00		10.00	10.00	1.00	270.00		270.00	270.00
		35-44		city	4.00	0.00	0.00	0.00	0.00	4.00	0.00	0.00	0.00	0.00	4.00	5.00	10.00	0.00	20.00	4.00	260.00	61.64	220.00	350.00
			working day	municipality	1.00	130.00		130.00	130.00	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	320.00		320.00	320.00
				rural	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	220.00		220.00	220.00
			weekend	city	5.00	0.00	0.00	0.00	0.00	5.00	0.00	0.00	0.00	0.00	5.00	2.00	4.47	0.00	10.00	5.00	250.00	43.01	210.00	300.00
		45-54	working day	city	3.00	0.00	0.00	0.00	0.00	3.00	0.00	0.00	0.00	0.00	3.00	10.00	10.00	0.00	20.00	3.00	276.67	40.41	240.00	320.00
				rural	2.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	2.00	230.00	28.28	210.00	250.00
			weekend	city	4.00	0.00	0.00	0.00	0.00	4.00	0.00	0.00	0.00	0.00	4.00	17.50	12.58	0.00	30.00	4.00	247.50	43.49	210.00	310.00
				municipality	4.00	0.00	0.00	0.00	0.00	4.00	0.00	0.00	0.00	0.00	4.00	22.50	5.00	20.00	30.00	4.00	332.50	115.29	230.00	460.00
				rural	2.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	2.00	20.00	0.00	20.00	20.00	2.00	390.00	42.43	360.00	420.00
		55-64	working day	city	5.00	0.00	0.00	0.00	0.00	5.00	0.00	0.00	0.00	0.00	5.00	5.80	8.56	0.00	19.00	5.00	278.00	40.87	230.00	330.00
				municipality	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	30.00		30.00	30.00	1.00	340.00		340.00	340.00
				rural	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	20.00		20.00	20.00	1.00	210.00		210.00	210.00
			weekend	city	3.00	0.00	0.00	0.00	0.00	3.00	0.00	0.00	0.00	0.00	3.00	0.00	0.00	0.00	0.00	3.00	276.67	81.45	220.00	370.00
				municipality	3.00	0.00	0.00	0.00	0.00	3.00	0.00	0.00	0.00	0.00	3.00	5.00	8.66	0.00	15.00	3.00	326.67	68.07	250.00	380.00
		65-74		rural	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	10.00		10.00	10.00	1.00	300.00		300.00	300.00
			working day	city	3.00	0.00	0.00	0.00	0.00	3.00	0.00	0.00	0.00	0.00	3.00	0.00	0.00	0.00	0.00	3.00	480.00	407.06	240.00	950.00
				rural	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	20.00		20.00	20.00	1.00	330.00		330.00	330.00
			weekend	city	5.00	0.00	0.00	0.00	0.00	5.00	0.00	0.00	0.00	0.00	5.00	10.00	10.00	0.00	20.00	5.00	398.00	214.76	210.00	640.00
				municipality	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	20.00		20.00	20.00	1.00	410.00		410.00	410.00
		75+		city	2.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	2.00	5.00	7.07	0.00	10.00	2.00	325.00	35.36	300.00	350.00
			working day	municipality	2.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	2.00	15.00	7.07	10.00	20.00	2.00	405.00	106.07	330.00	480.00
				rural	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	20.00		20.00	20.00	1.00	270.00		270.00	270.00
	male	20-24	weekend	city	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	250.00		250.00	250.00
			working day	city	2.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	2.00	5.00	7.07	0.00	10.00	2.00	280.00	42.43	250.00	310.00
				city	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	320.00		320.00	320.00
			weekend	municipality	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	450.00		450.00	450.00
		25-34		rural	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	540.00		540.00	540.00
			working day	city	3.00	0.00	0.00	0.00	0.00	3.00	0.00	0.00	0.00	0.00	3.00	3.33	5.77	0.00	10.00	3.00	326.67	153.08	210.00	500.00
				rural	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	300.00		300.00	300.00
			weekend	city	3.00	86.67	150.11	0.00	260.00	3.00	0.00	0.00	0.00	0.00	3.00	0.00	0.00	0.00	0.00	3.00	296.67	49.33	240.00	330.00
				municipality	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	240.00		240.00	240.00
		35-44		city	3.00	0.00	0.00	0.00	0.00	3.00	0.00	0.00	0.00	0.00	3.00	6.67	5.77	0.00	10.00	3.00	293.33	50.33	240.00	340.00
			working day	municipality	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	220.00		220.00	220.00
				rural	2.00	220.00	311.13	0.00	440.00	2.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	2.00	360.00	14.14	350.00	370.00
			weekend	city	3.00	0.00	0.00	0.00	0.00	3.00	0.00	0.00	0.00	0.00	3.00	3.33	5.77	0.00	10.00	3.00	490.00	216.56	360.00	740.00
				municipality	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	220.00		220.00	220.00
		45-54		rural	1.00	200.00		200.00	200.00	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	220.00		220.00	220.00
			working day	city	8.00	3.75	10.61	0.00	30.00	8.00	0.00	0.00	0.00	0.00	8.00	2.50	4.63	0.00	10.00	8.00	260.00	36.65	210.00	320.00
				municipality	5.00	26.00	58.14	0.00	130.00	5.00	0.00	0.00	0.00	0.00	5.00	0.00	0.00	0.00	0.00	5.00	298.00	137.00	210.00	540.00
				city	12.00	28.33	83.65	0.00	290.00	12.00	0.00	0.00	0.00	0.00	12.00	5.00	9.05	0.00	30.00	12.00	319.17	107.06	210.00	470.00
			weekend	municipality	6.00	18.33	44.91	0.00	110.00	6.00	0.00	0.00	0.00	0.00	6.00	7.67	9.83	0.00	20.00	6.00	333.33	74.48	220.00	410.00
				rural	2.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	2.00	370.00	127.28	280.00	460.00
		55-64		city	9.00	26.67	80.00	0.00	240.00	9.00	0.00	0.00	0.00	0.00	9.00	0.00	0.00	0.00	0.00	9.00	332.22	95.63	210.00	510.00
			working day	municipality	2.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	2.00	310.00	70.71	260.00	360.00
				rural	4.00	0.00	0.00	0.00	0.00	4.00	0.00	0.00	0.00	0.00	4.00	7.50	9.57	0.00	20.00	4.00	292.50	56.20	220.00	350.00
				city	7.00	0.00	0.00	0.00	0.00	7.00	0.00	0.00	0.00	0.00	7.00	2.86	4.88	0.00	10.00	7.00	292.86	82.20	210.00	410.00
			weekend	municipality	2.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	2.00	5.00	7.07	0.00	10.00	2.00	365.00	63.64	320.00	410.00
				rural	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	230.00		230.00	230.00
		65-74		city	7.00	0.00	0.00	0.00	0.00	7.00	0.00	0.00	0.00	0.00	7.00	4.29	7.87	0.00	20.00	7.00	268.57	66.69	210.00	410.00
			working day	municipality	3.00	0.00	0.00	0.00	0.00	3.00	0.00	0.00	0.00	0.00	3.00	6.67	11.55	0.00	20.00	3.00	496.67	189.03	350.00	710.00
				rural	2.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	2.00	225.00	21.21	210.00	240.00
				city	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	550.00		550.00	550.00
			weekend	municipality	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	280.00		280.00	280.00
				rural	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	340.00		340.00	340.00
		75+		city	3.00	0.00	0.00	0.00	0.00	3.00	0.00	0.00	0.00	0.00	3.00	10.00	17.32	0.00	30.00	3.00	350.00	30.00	320.00	380.00
			working day	municipality	1.00	0.00		0.00	0.00	1.00	0.00		0.00	0.00	1.00	30.00		30.00	30.00	1.00	340.00		340.00	340.00
				rural	3.00	36.67	63.51	0.00	110.00	3.00	0.00	0.00	0.00	0.00	3.00	6.67	11.55	0.00	20.00	3.00	283.33	56.86	220.00	330.00

Figure 21: Summary statistics for activities in **Cluster 4**.

In the code below, we will represent the cluster centers with their average values of numerical variables.

```
# Plot cluster centers for numerical columns
numerical_summary.plot(kind='bar', figsize=(10, 6))
plt.title('Average Values of Numerical Variables per Cluster')
plt.xlabel('Cluster')
plt.ylabel('Average Value')
plt.legend(title='Variables')
plt.show()
```

Listing 35: Cluster Centers with Average Value for Numerical Variables

As can be observed from figure 22, clusters are relatively distinct when it comes to numerical variables values.

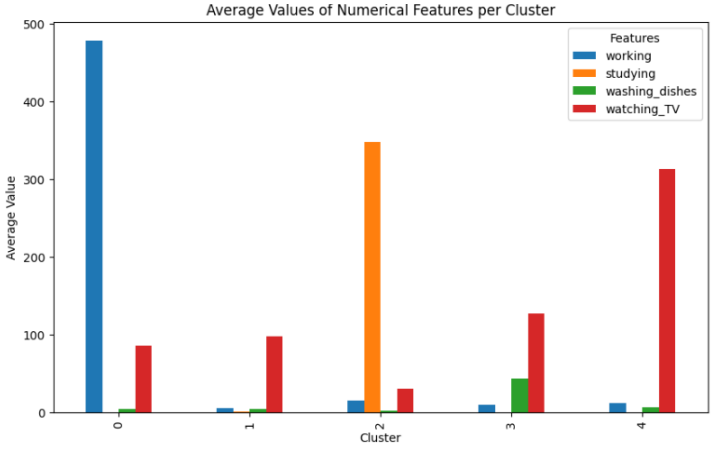


Figure 22: Cluster Centers with Average Value for Numerical Variables

- Cluster 0 has extremely higher average value for `working` column compared to the other clusters.
- Cluster 2 is the only cluster with `studying` variable included, suggesting that all individuals that study were grouped in one cluster.
- Cluster 3 has significantly higher average value for the `washing_dishes` column.
- Cluster 4 has significantly higher average value for `watching_TV` column.

In order to examine how categorical variables were distributed among the clusters, we employed the code given below:

```
# Create a plot for each categorical attribute to show cluster composition
fig, axes = plt.subplots(2, 3, figsize=(18, 10))
axes = axes.ravel()

for i, col in enumerate(categorical_columns):
    sns.countplot(data=k_means_data, x='cluster', hue=col, ax=axes[i])
    axes[i].set_title(f'Distribution of {col.capitalize()} in Each Cluster'
        )
    axes[i].set_xlabel('Cluster')
    axes[i].set_ylabel('Count')
    axes[i].legend(title=col.capitalize())
```

```
11
12 plt.tight_layout()
13 plt.show()
```

Listing 36: Cluster Composition for Categorical Variables

In the figure below, cluster composition is visually represented.

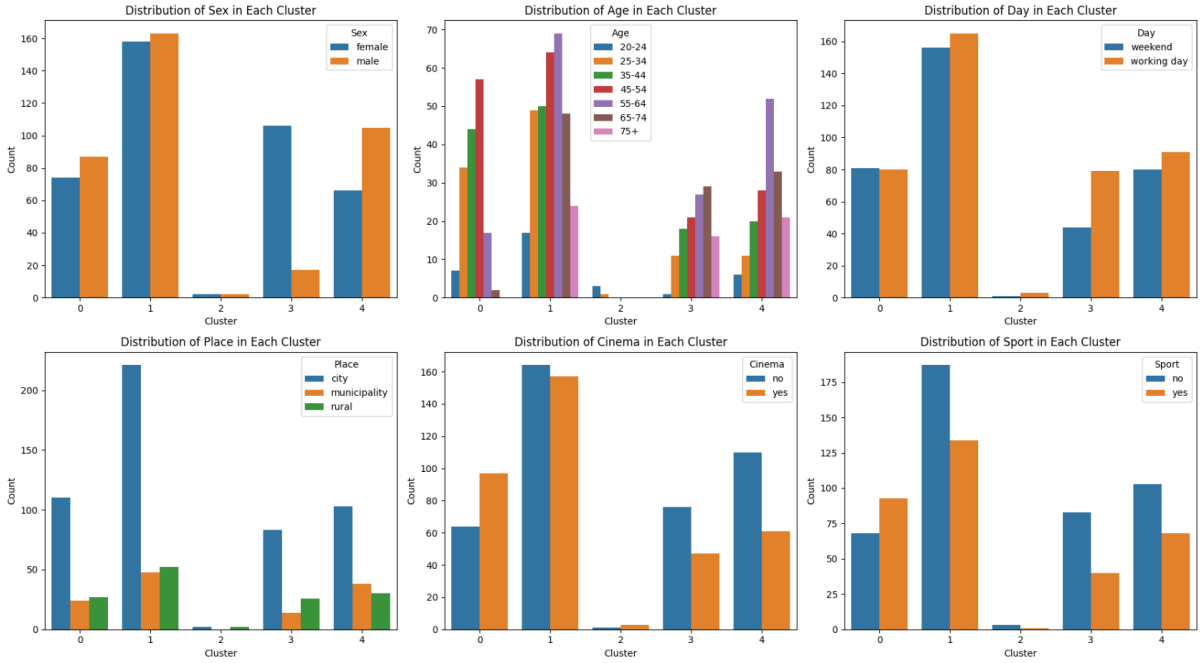


Figure 23: Cluster Composition for Categorical Variables

Some observations can be made:

- `Sex` variable is similarly present in clusters 1 and 2, whereas there is a significant difference in presence for cluster 3.
- `Age` variable is quite spread out. Cluster 0 does not include the 75+ age group, and cluster 2 includes only two age groups, 20-24 and 25-34. The rest of the clusters include all age groups.
- `Day` variable is similarly present in clusters 0, 1, 2, and 4, whereas there is a significant difference in presence for cluster 3.
- `Place` variable is quite spread out in all of the clusters. Cluster 2 does not contain municipality.
- `Cinema` variable is quite spread out in all of the clusters. The presence is similar only in clusters 1 and 2.
- `Sport` variable is quite spread out in all of the clusters.

Lastly, visualization of the clusters was done using 2 methods. First one was using Principal Component Analysis (PCA), visualizing the first two PCA components; the second one was using t-SNE to reduce to two dimensions.
PCA code and output are given below. Explained variance by PCA components: [0.32617119 0.2610285] - sum 0.5871996924536714.

```python
# Perform PCA on numerical columns to reduce to 2 components
pca = PCA(n_components=2)
pca_components = pca.fit_transform(k_means_scaled_data)

# Add PCA components to the DataFrame for plotting
k_means_data['PCA1'] = pca_components[:, 0]
k_means_data['PCA2'] = pca_components[:, 1]

explained_variance = pca.explained_variance_ratio_
print(f"Explained variance by PCA components: {explained_variance} - sum {
    explained_variance[0] + explained_variance[1]}")

# Plot the clusters in PCA space
plt.figure(figsize=(18, 12))
sns.scatterplot(data=k_means_data, x='PCA1', y='PCA2', hue='cluster',
    palette='tab10', s=60)
plt.title('Clusters in PCA-Reduced Space')
plt.xlabel(f'PCA Component 1 ({explained_variance[0]*100:.1f}%)')
plt.ylabel(f'PCA Component 2 ({explained_variance[1]*100:.1f}%)')
plt.legend(title='Cluster')
plt.show()
```

Listing 37: Principal Component Analysis for 2D Representation of Clusters

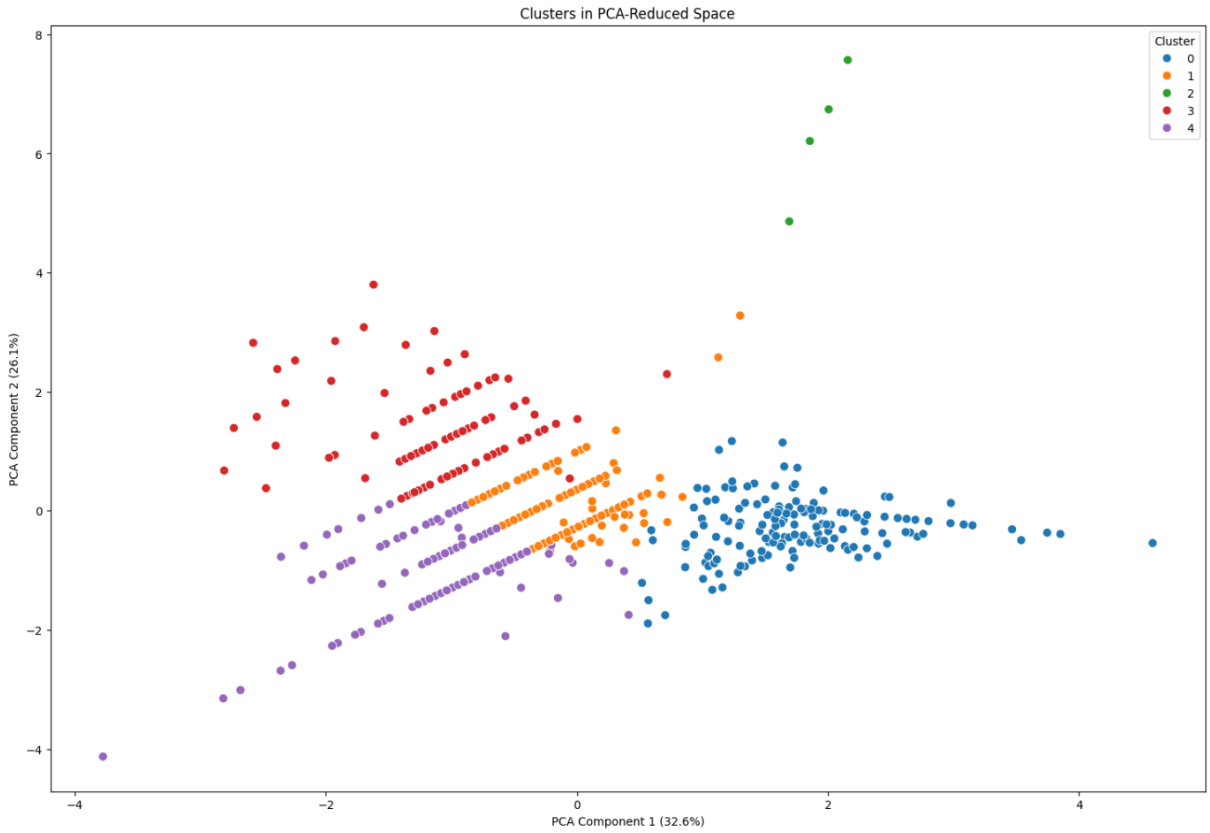


Figure 24: Principal Component Analysis for 2D Representation of Clusters with Explained Variance in Axis Names

t-SNE (t-Distributed Stochastic Neighbor Embedding) is a dimensionality reduction technique commonly used for visualizing high-dimensional data in two or three dimensions. It's

particularly effective at preserving the local structure of the data, making it a popular choice for visualizing clusters or patterns. t-SNE code and output are given below. More about this method can be read at link.

```
# Using t-SNE to reduce to 2D
tsne = TSNE(n_components=2, random_state=42)
tsne_components = tsne.fit_transform(k_means_scaled_data)

# Add t-SNE components to the DataFrame for plotting
k_means_data['t-SNE1'] = tsne_components[:, 0]
k_means_data['t-SNE2'] = tsne_components[:, 1]

# Plot t-SNE result
plt.figure(figsize=(18, 12))
sns.scatterplot(data=k_means_data, x='t-SNE1', y='t-SNE2', hue='cluster',
    palette='tab10', s=60)
plt.title('Clusters in t-SNE Reduced Space')
plt.xlabel('t-SNE Component 1')
plt.ylabel('t-SNE Component 2')
plt.legend(title='Cluster')
plt.show()
```

Listing 38: t-SNE for 2D Representation of Clusters

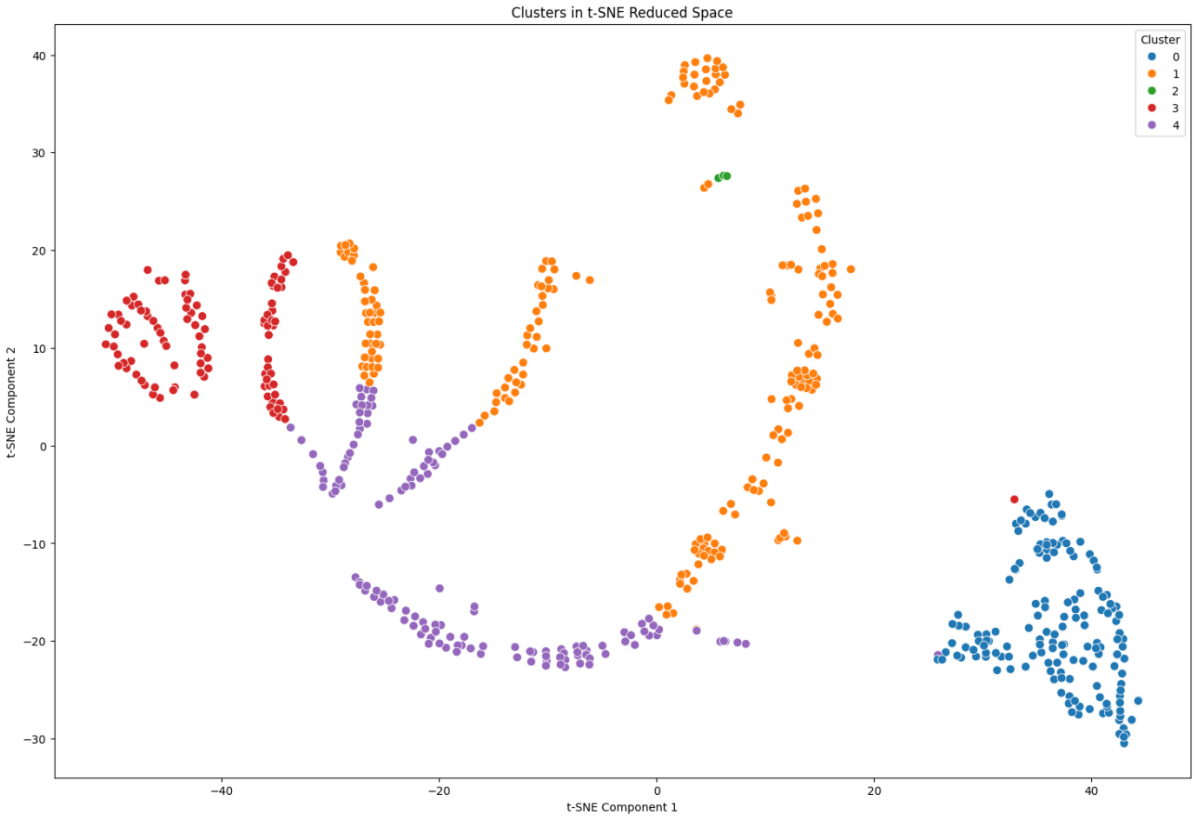


Figure 25: T-SNE for 2D Representation of Clusters

As it can be seen, although clusters have some overlap, they are still relatively well separated and groups are clearly defined.

- Cluster 0 was named the *Workaholic* cluster, given that the average `working` value in this cluster was significantly higher compared to the other clusters.

- Cluster 1 did not have any highly significant values in order to get a name.
- Cluster 2 was named the *Student* cluster, given that it is the only one that contains individuals which have `studying` time higher than 0.
- Cluster 3 was named the *Stay at Home* cluster, given that it had the highest `washing_dishes` value.
- Cluster 4 was named the *Lazy* cluster, given that the `watching_tv` time is significantly higher compared to the rest of the clusters, and `working_time` and `washing_dishes` was low.

6 Statistical Testing

Finally, statistical testing was performed to determine whether observed differences in activities across demographic groups are statistically significant. This allows us to validate whether these differences are meaningful or could have occurred by chance, providing a robust foundation for interpreting the patterns in the data.

6.1 Normality Testing

Before conducting statistical tests, we first checked whether the numerical variables follow a normal distribution using the Shapiro-Wilk test. The null hypothesis (H0) for this test is that the data is normally distributed. The Python code used for the normality test is as follows:

```
for col in numerical_columns:
    print(f"\n{col.capitalize()}")
    print_normality_test_results(df[col], f"{col} is normally distributed")
```

Listing 39: Normality Test for Numerical Variables

The results of the normality tests are summarized below:

- `Working`:
  - Statistics = 0.576, p-value = 0.000
  - H0: `working` is normally distributed.
  - Conclusion: The sample does not follow a Gaussian distribution (H0 rejected).
- `Studying`:
  - Statistics = 0.060, p-value = 0.000
  - H0: `studying` is normally distributed.
  - Conclusion: The sample does not follow a Gaussian distribution (H0 rejected).
- `Washing Dishes`:
  - Statistics = 0.703, p-value = 0.000
  - H0: `washing_dishes` is normally distributed.
  - Conclusion: The sample does not follow a Gaussian distribution (H0 rejected).
- `Watching TV`:
  - Statistics = 0.907, p-value = 0.000
  - H0: `watching_TV` is normally distributed.
  - Conclusion: The sample does not follow a Gaussian distribution (H0 rejected).

6.2 Correlation Analysis

Since none of the variables were normally distributed, non-parametric statistical tests were selected for statistical comparison and correlation analysis. To explore the relationships between the activities , we computed their pairwise correlations using Spearman's rank correlation method. This method was chosen since the variables were not normally distributed. The Python code used for generating the correlation heatmap is as follows:

```
# Correlation heatmap
plt.figure(figsize=(12, 10))
sns.heatmap(df[numerical_columns].corr(method='spearman'), annot=True, cmap
    ='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```

Listing 40: Correlation Heatmap

The resulting heatmap is displayed below:

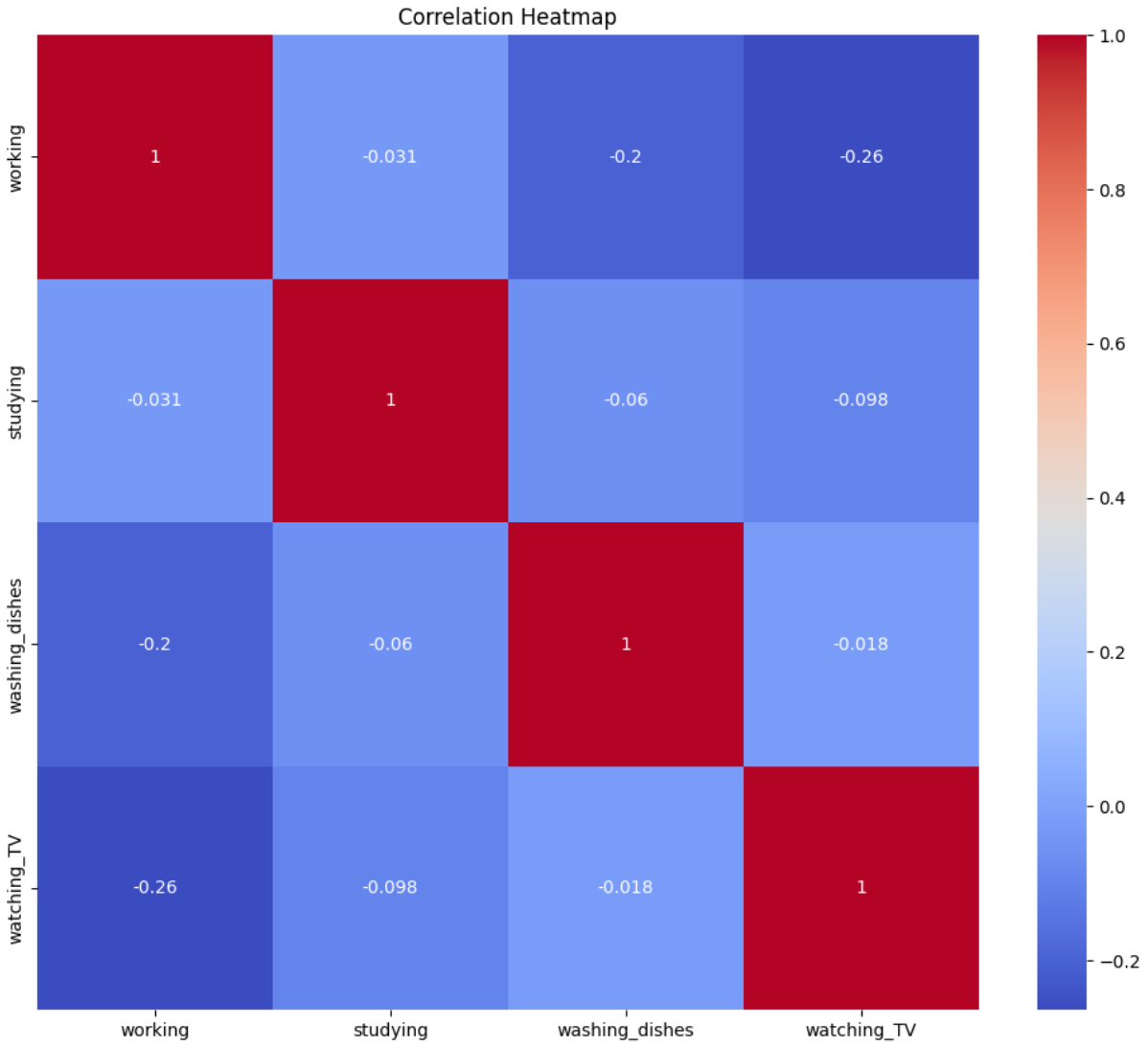


Figure 26: Spearman Correlation Heatmap for Numerical Variables

From the heatmap, we observe the following:

- The strongest correlation is between `working` and `watching_TV` (−0.26), indicating that individuals who work more tend to watch less TV.
- A moderate negative correlation exists between `working` and `washing_dishes` (−0.20), suggesting that individuals dedicating more time to household tasks are less likely to work.

Overall, the correlations are generally weak, suggesting limited dependence among the activities. To verify the presence of correlations, we performed Spearman's rank correlation tests between all pairs of numerical variables. The null hypothesis (`H0`) for each test states that the two variables are not correlated. The Python code used for the tests is as follows:

```
# Correlation tests - spearman since the values are not normally
    distributed
for col in numerical_columns:
    print(f"\n{col.capitalize()}")
    for col2 in numerical_columns:
        if col != col2:
            stat, p = spearmanr(df[col], df[col2])
            print_test_results(stat, p, f"{col} and {col2} are not
                correlated - spearman")
```

Listing 41: Spearman Correlation Tests

The results of the tests are summarized below:

- `Working`:
  - `Working` and `washing_dishes`: $\rho = -0.203$, $p = 0.000$ (Reject `H0`). Individuals who work more tend to spend less time on household activities like washing dishes.
  - `Working` and `watching_TV`: $\rho = -0.264$, $p = 0.000$ (Reject `H0`). A significant negative correlation exists, showing that working more is associated with less time spent watching TV.
- `Studying`:
  - `Studying` and `watching_TV`: $\rho = -0.098$, $p = 0.006$ (Reject `H0`). A weak negative correlation indicates that individuals who study more tend to watch slightly less TV.
- `Washing Dishes`:
  - No statistically significant correlations with other variables.
- `Watching TV`:
  - Correlations with both `working` and `studying` were confirmed, as described above.

These tests confirm the previously observed correlations, particularly the significant negative relationships between `working` and both `washing_dishes` and `watching_TV`, while most other relationships remain negligible.

To further visualize the relationships between activities, we created a pairplot. This scatter plot matrix provides an intuitive view of pairwise relationships while also showing the distributions of each variable along the diagonal. The Python code used is as follows:

```
sns.pairplot(df, vars=numerical_columns)
```

Listing 42: Pairplot of Numerical Variables

The resulting pairplot is shown below:

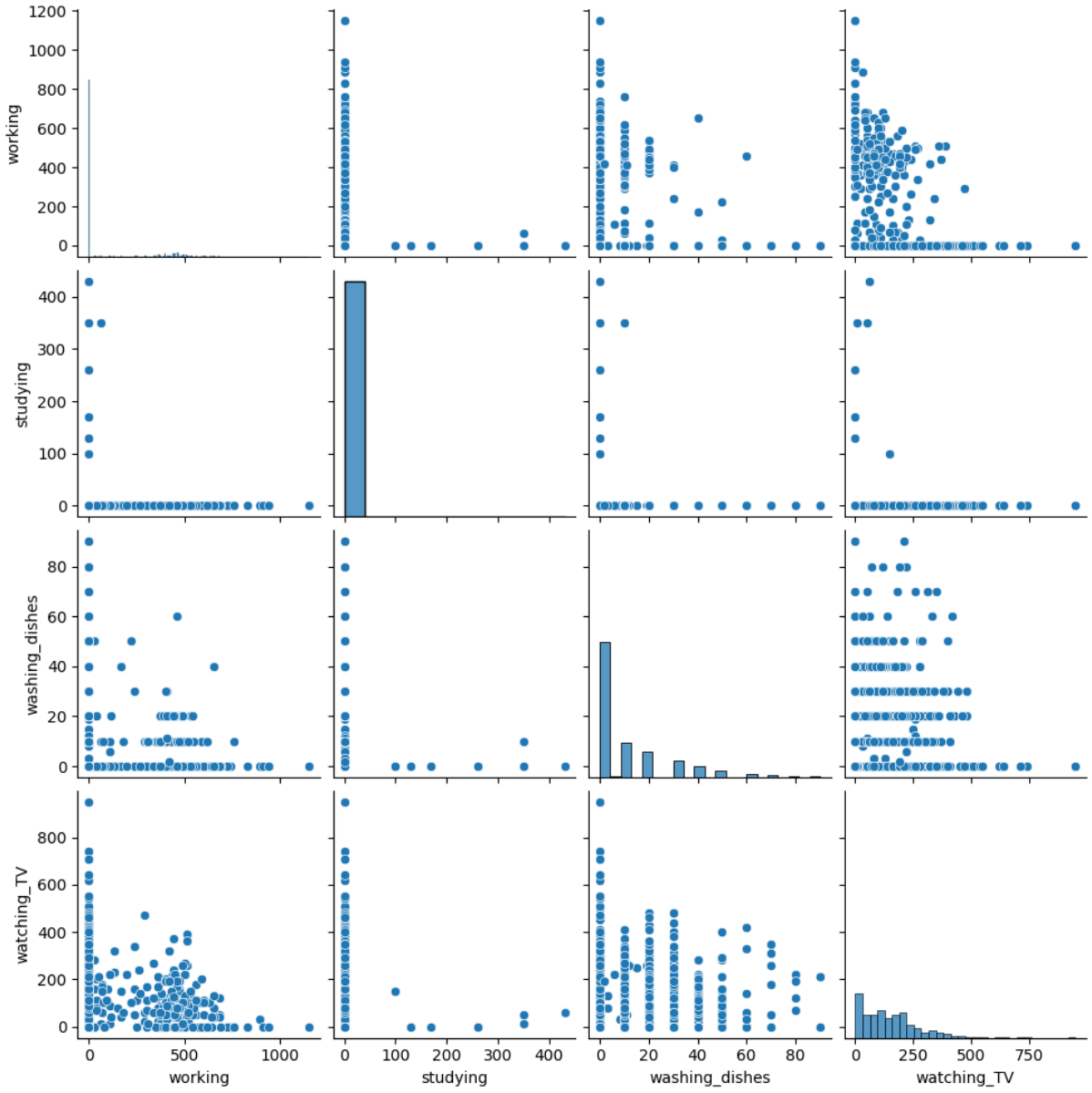


Figure 27: Pairplot Showing Pairwise Relationships Between Numerical Variables

This visualization complements the earlier statistical analyses and confirms that the relationships between variables are generally weak.

6.3 Statistical Comparison

All of the statistical tests used from this point below include multiple correction, given the fact that multiple independent test are performed.

6.3.1 Mann-Whitney U Test

To further investigate whether gender influences the time spent on different activities, we conducted the Mann-Whitney U test for each activity. The Mann-Whitney U test was chosen

because there are two genders, not all activities are normally distributed, as previously shown, and the data is not paired. The Python code for the analysis is shown below:

```
# Mann-Whitney U test for statistical testing between different genders and numerical columns
# Collect p-values for multiple tests
p_values = []
test_results = []

for col in numerical_columns:
  female_column = df[df['sex'] == 'female'][col]
  male_column = df[df['sex'] == 'male'][col]

  statistic, p_value = mannwhitneyu(female_column, male_column, alternative='two-sided')
  p_values.append(p_value)
  test_results.append((col, statistic, p_value))

# Apply multiple testing correction (Benjamini-Hochberg)
adjusted_results = multipletests(p_values, method='fdr_bh')
adjusted_p_values = adjusted_results[1]

for (col, stat, p), adj_p in zip(test_results, adjusted_p_values):
  print_test_results(stat, p, f"{col.capitalize()} time between females and males is the same", adj_p)
  print("")
```

Listing 43: Mann-Whitney U Test for Gender Differences

The results of the Mann-Whitney U tests are summarized below:

- `Working`:
  - $Statistics = 70204.000$, $p = 0.018$, $p\_adj = 0.024$
  - `H0`: Working time between males and females is the same.
  - Conclusion: Reject `H0`. There is a statistically significant difference in working time between genders.
- `Studying`:
  - $Statistics = 75671.500$, $p = 0.626$, $p\_adj = 0.626$
  - `H0`: Studying time between males and females is the same.
  - Conclusion: Fail to reject `H0`. There is no statistically significant difference in studying time between genders.
- `Washing Dishes`:
  - $Statistics = 110232.500$, $p = 0.000$, $p\_adj = 0.000$
  - `H0`: Washing dishes time between males and females is the same.
  - Conclusion: Reject `H0`. There is a statistically significant difference in washing dishes time between genders.
- `Watching TV`:

- *Statistics* = 64519.000, p = 0.000, p_adj = 0.001
- H0: Watching TV time between males and females is the same.
- Conclusion: Reject H0. There is a statistically significant difference in watching TV time between genders.

These results indicate that males and females differ significantly in terms of working, washing_dishes, and watching_TV. However, there is no significant difference in studying time between genders. The significant differences in washing_dishes and watching_TV times highlight gender-based differences in household and leisure activities. These findings in overall match the trends observed in Figure 10.

Furthermore, to assess whether the day type (working day vs. weekend) influences the time spent on different activities, we performed the Mann-Whitney U test for each activity. As before, the Mann-Whitney U test was chosen because there are only two day types, none of the activities are normally distributed, and the data is not paired. The Python code for this analysis is as follows:

```
# Mann-Whitney U test for statistical testing between different days and
    numerical columns
# Collect p-values for multiple tests
p_values = []
test_results = []

for col in numerical_columns:
  working_day_column = df[df['day'] == 'working day'][col]
  weekend_column = df[df['day'] == 'weekend'][col]

  statistic, p_value = mannwhitneyu(working_day_column, weekend_column,
      alternative='two-sided')
  p_values.append(p_value)
  test_results.append((col, statistic, p_value))

# Apply multiple testing correction (Benjamini-Hochberg)
adjusted_results = multipletests(p_values, method='fdr_bh')
adjusted_p_values = adjusted_results[1]

for (col, stat, p), adj_p in zip(test_results, adjusted_p_values):
  print_test_results(stat, p, f"{col.capitalize()} time between working
      days and weekends is the same", adj_p)
  print("")
```

Listing 44: Mann-Whitney U Test for Day Differences

The results of the Mann-Whitney U tests are summarized below:

- Working:
  - *Statistics* = 73746.000, p = 0.427, p_adj = 0.570
  - H0: Working time between working days and weekends is the same.
  - Conclusion: Fail to reject H0. There is no statistically significant difference in working time between the day types.
- Studying:
  - *Statistics* = 75369.500, p = 0.574, p_adj = 0.574

- H0: Studying time between working days and weekends is the same.
- Conclusion: Fail to reject H0. There is no statistically significant difference in studying time between the day types.

- `Washing Dishes`:
  - $Statistics = 82465.500$, $p = 0.016$, $p\_adj = 0.065$
  - H0: Washing dishes time between working days and weekends is the same.
  - Conclusion: Fail to reject H0. There is no statistically significant difference in washing dishes time between the day types. **This is the only result which was changed by using multiple correction.**

- `Watching TV`:
  - $Statistics = 78188.000$, $p = 0.419$, $p\_adj = 0.570$
  - H0: Watching TV time between working days and weekends is the same.
  - Conclusion: Fail to reject H0. There is no statistically significant difference in watching TV time between the day types.

These results align with the patterns shown in Figure 12, where time spent on `working`, `studying`, and `watching_TV` is similar between working days and weekends. A significant difference in the distribution of `washing_dishes` time is apparent in the figure; however, according to adjusted p value, there is no significant difference in washig dishes time across day types either.

6.3.2 Kruskal-Wallis Test

To evaluate whether the place of residence influences the time spent on different activities, we performed the Kruskal-Wallis test for each activity. This test was chosen as it is suitable for comparing the medians of more than two groups when the data is not normally distributed and unpaired. The Python code for this analysis is as follows:

```
# Perform Kruskal-Wallis test for different places and numerical columns
# Collect p-values for multiple tests
p_values = []
test_results = []

for col in numerical_columns:
  city_column = df[df['place'] == 'city'][col]
  municipality_column = df[df['place'] == 'municipality'][col]
  rural_column = df[df['place'] == 'rural'][col]

  statistic, p_value = kruskal(city_column, municipality_column,
     rural_column)
  p_values.append(p_value)
  test_results.append((col, statistic, p_value))

# Apply multiple testing correction (Benjamini-Hochberg)
adjusted_results = multipletests(p_values, method='fdr_bh')
adjusted_p_values = adjusted_results[1]

for (col, stat, p), adj_p in zip(test_results, adjusted_p_values):
```

```
  print_test_results(stat, p, f"{col.capitalize()} time median between
      different places is the same", adj_p)
  print("")
```

Listing 45: Kruskal-Wallis Test for Place Differences

The results of the Kruskal-Wallis tests are summarized below:

- `Working`:
  - $Statistics = 0.254, p = 0.881, p\_adj = 0.881$
  - `H0`: The median working time between different places is the same.
  - Conclusion: Fail to reject `H0`. There is no statistically significant difference in working time across places.
- `Studying`:
  - $Statistics = 1.638, p = 0.441, p\_adj = 0.588$
  - `H0`: The median studying time between different places is the same.
  - Conclusion: Fail to reject `H0`. There is no statistically significant difference in studying time across places.
- `Washing Dishes`:
  - $Statistics = 2.895, p = 0.235, p\_adj = 0.470$
  - `H0`: The median washing dishes time between different places is the same.
  - Conclusion: Fail to reject `H0`. There is no statistically significant difference in washing dishes time across places.
- `Watching TV`:
  - $Statistics = 4.193, p = 0.123, p\_adj = 0.470$
  - `H0`: The median watching TV time between different places is the same.
  - Conclusion: Fail to reject `H0`. There is no statistically significant difference in watching TV time across places.

These results align with the distributions shown in Figure 13. These findings suggest that the place of residence does not play a significant role in influencing the time spent on these activities.

To examine whether different age groups influence the time spent on activities, we performed the Kruskal-Wallis test for each activity. Again, this test was chosen since there were multiple groups and the data was unpaired and not normally distributed. The Python code used for this analysis is shown below:

```
# Perform Kruskal-Wallis test for different age groups and numerical
    columns
# Collect p-values for multiple tests
p_values = []
test_results = []

for col in numerical_columns:
  age_20_24_column = df[df['age'] == '20-24'][col]
  age_25_34_column = df[df['age'] == '25-34'][col]
```

```
    age_35_44_column = df[df['age'] == '35-44'][col]
    age_45_54_column = df[df['age'] == '45-54'][col]
    age_55_64_column = df[df['age'] == '55-64'][col]
    age_65_74_column = df[df['age'] == '65-74'][col]
    age_75_column = df[df['age'] == '75+'][col]

    statistic, p_value = kruskal(age_20_24_column, age_25_34_column,
        age_35_44_column, age_45_54_column,age_55_64_column,age_65_74_column,
        age_75_column)
    p_values.append(p_value)
    test_results.append((col, statistic, p_value))

# Apply multiple testing correction (Benjamini-Hochberg)
adjusted_results = multipletests(p_values, method='fdr_bh')
adjusted_p_values = adjusted_results[1]

for (col, stat, p), adj_p in zip(test_results, adjusted_p_values):
    print_test_results(stat, p, f"{col.capitalize()} time median between
        different age groups is the same", adj_p)
    print("")
```

Listing 46: Kruskal-Wallis Test for Age Group Differences

The results of the Kruskal-Wallis tests are summarized below:

- `Working`:
  - $Statistics = 100.411, p = 0.000, p\_adj = 0.000$
  - `H0`: The median working time between different age groups is the same.
  - Conclusion: Reject `H0`. There is a statistically significant difference in working time across age groups.
- `Studying`:
  - $Statistics = 55.451, p = 0.000, p\_adj = 0.000$
  - `H0`: The median studying time between different age groups is the same.
  - Conclusion: Reject `H0`. There is a statistically significant difference in studying time across age groups.
- `Washing Dishes`:
  - $Statistics = 17.342, p = 0.008, p\_adj = 0.008$
  - `H0`: The median washing dishes time between different age groups is the same.
  - Conclusion: Reject `H0`. There is a statistically significant difference in washing dishes time across age groups.
- `Watching TV`:
  - $Statistics = 88.753, p = 0.000, p\_adj = 0.000$
  - `H0`: The median watching TV time between different age groups is the same.
  - Conclusion: Reject `H0`. There is a statistically significant difference in watching TV time across age groups.

These results align with the distributions shown in Figure 11, where significant differences in activity times are visually evident across age groups. These findings confirm that age groups play a significant role in determining activity patterns, with notable differences across most activities.

6.3.3 Wilcoxon Signed-rank Test

Building on the previous analyses, we sought to determine whether the distributions of different activities were statistically similar within people. Since the data for activities is paired and the variables are not normally distributed, we performed the Wilcoxon signed-rank test. The Python code used for this analysis is shown below:

```
# Perform the Wilcoxon signed-rank test because data in rows belongs to the same family
# Collect p-values for multiple tests
p_values = []
test_results = []

for i in range(len(numerical_columns)):
  for j in range(i + 1, len(numerical_columns)):
    data1 = df[numerical_columns[i]]
    data2 = df[numerical_columns[j]]

    statistic, p_value = wilcoxon(data1, data2)
    p_values.append(p_value)
    test_results.append((numerical_columns[i], numerical_columns[j], statistic, p_value))

# Apply multiple testing correction (Benjamini-Hochberg)
adjusted_results = multipletests(p_values, method='fdr_bh')
adjusted_p_values = adjusted_results[1]

for (col1, col2, stat, p), adj_p in zip(test_results, adjusted_p_values):
    print_test_results(stat, p, f"{numerical_columns[i].capitalize()} and {numerical_columns[j].capitalize()} come from the same distribution", adj_p)
    print("")
```

Listing 47: Wilcoxon Signed-Rank Test for Paired Activity Data

The results of the Wilcoxon signed-rank tests are summarized below:

- `Working` vs. `Studying`:
  - $Statistics = 321.500$, $p = 0.000$, $p\_adj = 0.000$
  - `H0`: `Working` and `Studying` come from the same distribution.
  - Conclusion: Reject `H0`. There is a statistically significant difference between these activities.
- `Working` vs. `Washing Dishes`:
  - $Statistics = 40511.500$, $p = 0.000$, $p\_adj = 0.000$
  - `H0`: `Working` and `Washing Dishes` come from the same distribution.
  - Conclusion: Reject `H0`. There is a statistically significant difference between these activities.

- `Working` vs. `Watching TV`:
  - $Statistics = 92032.000$, $p = 0.000$, $p\_adj = 0.000$
  - `H0`: `Working` and `Watching TV` come from the same distribution.
  - Conclusion: Reject `H0`. There is a statistically significant difference between these activities.
- `Studying` vs. `Washing Dishes`:
  - $Statistics = 2394.000$, $p = 0.000$, $p\_adj = 0.000$
  - `H0`: `Studying` and `Washing Dishes` come from the same distribution.
  - Conclusion: Reject `H0`. There is a statistically significant difference between these activities.
- `Studying` vs. `Watching TV`:
  - $Statistics = 3041.000$, $p = 0.000$, $p\_adj = 0.000$
  - `H0`: `Studying` and `Watching TV` come from the same distribution.
  - Conclusion: Reject `H0`. There is a statistically significant difference between these activities.
- `Washing Dishes` vs. `Watching TV`:
  - $Statistics = 3762.000$, $p = 0.000$, $p\_adj = 0.000$
  - `H0`: `Washing Dishes` and `Watching TV` come from the same distribution.
  - Conclusion: Reject `H0`. There is a statistically significant difference between these activities.

These results confirm that all activity pairs differ significantly in their distributions within households.

6.3.4 Chi-squared test

Lastly, we conducted the Chi-squared test to evaluate whether pairs of categorical variables are independent. This test was chosen as it is suitable for assessing relationships between categorical variables, based on their contingency tables. The Python code for this analysis is shown below:

```
# Chi squared test between pairs of categorical features
categorical_cols = ['sex', 'age', 'day', 'place', 'cinema', 'sport']

# Collect p-values for multiple tests
p_values = []
test_results = []

for i in range(len(categorical_cols)):
  for j in range(i + 1, len(categorical_cols)):
    col1 = categorical_cols[i]
    col2 = categorical_cols[j]
    contingency_table = pd.crosstab(df[col1], df[col2])
    chi2, p, dof, expected = chi2_contingency(contingency_table)
```

```
    p_values.append(p)
    test_results.append((col1, col2, chi2, dof, p, contingency_table))

# Apply multiple testing correction (Benjamini-Hochberg)
adjusted_results = multipletests(p_values, method='fdr_bh')
adjusted_p_values = adjusted_results[1]

for (col1, col2, chi2, dof, p, contingency_table), adj_p in zip(
    test_results, adjusted_p_values):
  print(f"Chi-squared test for {col1} vs {col2}:")
  print(f"Degrees of freedom: {dof}")
  print("Contingency Table:")
  print(contingency_table)
  print("")
  print_test_results(chi2, p, f"{col1} and {col2} are independent of each
      other", adj_p)
  print("")
```

Listing 48: Chi-squared Test for Categorical Variables

The results of the Chi-squared tests are summarized below:

- **Sex vs. Age**:
  - Degrees of Freedom: 6
  - Contingency Table:

age	20-24	25-34	35-44	45-54	55-64	65-74	75+
sex							
female	19	53	74	89	85	55	31
male	15	53	58	81	80	57	30

  - $Statistics = 1.680, p = 0.947, p\_adj = 0.947$
  - `H0`: `Sex` and `Age` are independent of each other.
  - Conclusion: Fail to reject H_0. There is no statistically significant relationship between `Sex` and `Age`.
- **Sex vs. Day**:
  - Degrees of Freedom: 1
  - Contingency Table:

day	weekend	working day
sex		
female	183	223
male	179	195

  - $Statistics = 0.501, p = 0.479, p\_adj = 0.599$
  - `H0`: `Sex` and `Day` are independent of each other.
  - Conclusion: Fail to reject H_0. There is no statistically significant relationship between `Sex` and `Day`.

- **Sex vs. Place**:
  - Degrees of Freedom: 2
  - Contingency Table:

```
place   city  municipality  rural
sex
female   272            67     67
male     247            57     70
```

  - $Statistics = 0.765, p = 0.682, p\_adj = 0.787$
  - `H0`: `Sex` and `Place` are independent of each other.
  - Conclusion: Fail to reject H_0. There is no statistically significant relationship between `Sex` and `Place`.
- **Sex vs. Cinema**:
  - Degrees of Freedom: 1
  - Contingency Table:

```
cinema   no  yes
sex
female  199  207
male    216  158
```

  - $Statistics = 5.626, p = 0.018, p\_adj = 0.044$
  - `H0`: `Sex` and `Cinema` are independent of each other.
  - Conclusion: Reject H_0. There is a statistically significant relationship between `Sex` and `Cinema`.
- **Sex vs. Sport**:
  - Degrees of Freedom: 1
  - Contingency Table:

```
sport    no  yes
sex
female  259  147
male    185  189
```

  - $Statistics = 15.719, p = 0.000, p\_adj = 0.000$
  - `H0`: `Sex` and `Sport` are independent of each other.
  - Conclusion: Reject H_0. There is a statistically significant relationship between `Sex` and `Sport`.
- **Age vs. Day**:
  - Degrees of Freedom: 6

- Contingency Table:

day	weekend	working day
age		
20-24	15	19
25-34	54	52
35-44	61	71
45-54	72	98
55-64	78	87
65-74	55	57
75+	27	34

- $Statistics = 2.565, p = 0.861, p\_adj = 0.923$
- H0: Age and Day are independent of each other.
- Conclusion: Fail to reject H_0. There is no statistically significant relationship between Age and Day.

- **Age vs. Place**:
  - Degrees of Freedom: 12
  - Contingency Table:

place	city	municipality	rural
age			
20-24	26	4	4
25-34	66	20	20
35-44	92	20	20
45-54	116	28	26
55-64	110	32	23
65-74	69	13	30
75+	40	7	14

  - $Statistics = 15.190, p = 0.231, p\_adj = 0.433$
  - H0: Age and Place are independent of each other.
  - Conclusion: Fail to reject H_0. There is no statistically significant relationship between Age and Place.

- **Age vs. Cinema**:
  - Degrees of Freedom: 6
  - Contingency Table:

cinema	no	yes
age		
20-24	7	27
25-34	38	68
35-44	47	85
45-54	82	88

```
55-64   103   62
65-74    88   24
75+      50   11
```

  - $Statistics = 100.307$, $p = 0.000$, $p\_adj = 0.000$
  - H0: Age and Cinema are independent of each other.
  - Conclusion: Reject H_0. There is a statistically significant relationship between Age and Cinema.

- **Age vs. Sport**:
  - Degrees of Freedom: 6
  - Contingency Table:

```
sport   no  yes
age
20-24   20   14
25-34   57   49
35-44   53   79
45-54   84   86
55-64  100   65
65-74   85   27
75+     45   16
```

  - $Statistics = 43.943$, $p = 0.000$, $p\_adj = 0.000$
  - H0: Age and Sport are independent of each other.
  - Conclusion: Reject H_0. There is a statistically significant relationship between Age and Sport.

- **Day vs. Place**:
  - Degrees of Freedom: 2
  - Contingency Table:

```
place        city  municipality  rural
day
weekend       242            63     57
working day   277            61     80
```

  - $Statistics = 2.245$, $p = 0.325$, $p\_adj = 0.488$
  - H0: Day and Place are independent of each other.
  - Conclusion: Fail to reject H_0. There is no statistically significant relationship between Day and Place.

- **Day vs. Cinema**:
  - Degrees of Freedom: 1

- Contingency Table:

cinema	no	yes
day		
weekend	200	162
working day	215	203

- $Statistics = 0.985, p = 0.321, p\_adj = 0.488$
- H0: Day and Cinema are independent of each other.
- Conclusion: Fail to reject H_0. There is no statistically significant relationship between Day and Cinema.

- **Day vs. Sport**:
  - Degrees of Freedom: 1
  - Contingency Table:

sport	no	yes
day		
weekend	212	150
working day	232	186

  - $Statistics = 0.622, p = 0.430, p\_adj = 0.587$
  - H0: Day and Sport are independent of each other.
  - Conclusion: Fail to reject H_0. There is no statistically significant relationship between Day and Sport.

- **Place vs. Cinema**:
  - Degrees of Freedom: 2
  - Contingency Table:

cinema	no	yes
place		
city	250	269
municipality	74	50
rural	91	46

  - $Statistics = 16.986, p = 0.000, p\_adj = 0.001$
  - H0: Place and Cinema are independent of each other.
  - Conclusion: Reject H_0. There is a statistically significant relationship between Place and Cinema.

- **Place vs. Sport**:
  - Degrees of Freedom: 2
  - Contingency Table:

sport	no	yes
place		
city	304	215
municipality	61	63
rural	79	58

  - $Statistics = 3.629, p = 0.163, p\_adj = 0.349$
  - H0: Place and Sport are independent of each other.
  - Conclusion: Fail to reject H_0. There is no statistically significant relationship between Place and Sport.

- **Cinema vs. Sport**:
  - Degrees of Freedom: 1
  - Contingency Table:

sport	no	yes
cinema		
no	280	135
yes	164	201

  - $Statistics = 39.317, p = 0.000, p\_adj = 0.000$
  - H0: Cinema and Sport are independent of each other.
  - Conclusion: Reject H_0. There is a statistically significant relationship between Cinema and Sport.

In conclusion, the following relationships were found to be statistically significant, indicating that these variables are related:

- Sex vs. Cinema
- Sex vs. Sport
- Age vs. Cinema
- Age vs. Sport
- Place vs. Cinema
- Cinema vs. Sport

These findings suggest that all the demographic factors in the dataframe significantly influence participation in Cinema and Sport activities.

7 Conclusions

In this study, we utilized the given data to analyze the habits of households and individuals. Through systematic data preparation, descriptive statistics, and statistical testing, we were able to address the requested questions. The answers to these questions are summarized below:

1. *Characterize the individuals that are present in the data. Are there groups of similar persons?*
   The answer to this is explored throughout the whole report, in the sections 4, 5, 6. K-Means section (Section 5) gives the overview on groups of similar persons.

2. *Estimate how much time on average households spend daily on each activity.*
   The answer to this question was already given in the section 4, in the table 4. For clarity, the information is restated below:

   - **Activity Average Time (minutes):**
     - `working`: 165.99
     - `studying`: 3.62
     - `washing_dishes`: 17.41
     - `watching_TV`: 232.30

3. *With respect to which activities do living environments differ?*
   To answer this question, we firstly examined the distributions of activities across different places to gain initial understanding, which can be seen in Figure 13. Then, we conducted Kruskal-Wallis test, as described in Subsection 6.3.2. The results indicated that there are no statistically significant differences in any of the activities across living environments.

4. *With respect to which activities do working days and weekends differ?*
   Similar to the previous question, we first examined the distributions of activities in working days and weekends, which can be seen in Figure 12. Then, we conducted Mann-Whitney U test, as described in Subsection 6.3.1. Initially, the results suggested that the only activity significantly influenced by day type was `washing dishes`. However, after applying a multiple comparison correction to account for the risk of false positives, we concluded that there are no statistically significant differences between working days and weekends.

5. *Which activities are associated with each other?*
   To detect which activities are associated with each other, we conducted a Spearman's rank correlation analysis in the section 6.2. In summary, the analysis revealed that there are statistically significant associations between:

   - `Working` and `Watching TV` (negative).
   - `Working` and `Washing Dishes` (negative).
   - `Studying` and `Watching TV` (negative, weak).

In conclusion, we successfully answered all the questions posed in this study, providing clear and well-supported analysis based on the given data. The findings offer meaningful insights into the habits of individuals and households and pave the way for future research and applications. The full implementation of this study is available in the *Project Notebook* provided.