Report on Time Use Survey

Tutor: Juho Heimonen

Course Name: Statistical Data Analysis

Course Code: TKO_7093-3004

Group Number: Project_013

Presented by:

Nouman Bashir (2507543)

Dominic Amoateng Sabeng (2510400)

Saif Ur Rehman (2414361)

Submission date:

27th October, 2025

University of Turku, Department of Computing

Introduction

The Time Use Survey in Finland examined how Finnish individuals allocate time to activities such as working, sleeping, reading, dining in restaurants, and visiting libraries. The survey aimed to gather insights into the habits of individuals and households across Finland. Data were collected from respondents in cities, municipalities, and rural areas using an online data collection tool. This report seeks to:

a. Characterise respondents using demographic and activity variables
b. Perform clustering analysis on the activity variables (working, sleeping and reading)
c. Perform PCA on the activity variables (working, sleeping and reading)
d. Estimate the average daily time Finnish households spend on activity variables (working, sleeping and reading)
e. Determine differences in the average daily time spent on the activity variables based on the living environment in Finland
f. Determine the difference in the average daily time spent on the activity variables based on the day of the week Finnish people perform the activity
g. Determine the difference in the average daily time spent on the activity variables based on the sex of respondents.
h. Determine the association between the demographic variables (sex, age group, living environment, day of week) and the activity variables (dining at the restaurant and visiting the library)
i. Analyse the relationships among the activity variables (working, sleeping and reading) in Finland

Methods

The data for this report were sourced from Statistics Finland's dataset, "Teaching Use Data of the Time Use Survey," licensed under the Creative Commons Attribution 4.0 International license. The survey included 780 respondents. This secondary dataset comprised 11 variables: 6 demographic variables (household ID, member ID, day of the week, sex, age group, living environment) and 5 activity variables (working, sleeping, reading, dining at restaurants, visiting libraries). The activity variables covered activities performed by respondents over the past 12 months. Working, sleeping, and reading were measured in minutes, while dining at restaurants and

visiting libraries were recorded as binary (yes/no) responses, indicating whether respondents engaged in these activities.

Data Preparation

a. Data type

The data had both categorical and numerical variables. Table 1 describes the data type of variables in the dataset.

Table 1: Data type of variables in the dataset

Code	Variable name	Data type
khode	Household ID	Numerical variable
jasen	Member ID (within household)	Numerical variable
pvknro	Day of week	Binary (categorical variable)
sp	Sex	Binary (categorical variable)
IKAL1	Age group	Categorical variable
ASALUE	Living environment	Categorical variable
A1 (minutes)	Working	Numerical variable
A2 (minutes)	Sleeping	Numerical variable
A3 (minutes)	Reading	Numerical variable
A4	Dining at restaurant	Binary (categorical) variable
A5	Visiting library	Binary (categorical) variable

b. Data Cleaning

Initial characterization of the dataset revealed inconsistencies in the activity variables. Some data points were entered in “hours and minutes” format, as “?”, as minutes, or as zeros (e.g., 10:30, “?”, 120, 0). Data points in hours and minutes were converted to minutes for all activity variables. For the activity variables “dining at restaurants” and “visiting libraries,” “?” entries were replaced with the mode of the respective variable’s data points. Minutes entries were assigned “yes” to indicate the activity was performed, with the rationale that the activity occurred but the duration was rather recorded instead of indicating yes. For the sleeping variable, 10 “?” entries were imputed using the median of the data points, stratified by the demographic variables’ “sex” and “age group,” as sleep duration was found to vary by age group. The median imputation was done since everyone sleeps, hence “?” entries could not be treated as

missing values or discarded entirely from the dataset. For the working and reading variables, “?” entries were replaced with zeros to indicate that respondents did not engage in these activities. Zero minutes for working were assumed to represent students or retired individuals who do not work, while zeros for reading were assigned to individuals who do not read.

Descriptive Statistics

Baseline descriptive statistics were calculated for all variables in the dataset. Table 1 summarizes the baseline characteristics of respondents. Of the 780 respondents, 406 were male (52.05%), and 374 were female (47.95%). Most respondents were aged 45-54 years (21.79%), followed by 55-64 years (21.15%). Additionally, 4.35% were 24 years or younger, 13.59% were 25-34 years, 14.35% were 65-74 years, and 7.82% were 75 years or older. Furthermore, 418 respondents (53.59%) reported working from Monday to Friday. Regarding living environment, 519 respondents (66.53%) lived in cities, 124 (15.89%) in municipalities, and 137 (17.56%) in rural areas.

Interestingly, the data revealed that about 53.6% of the Finnish people dined out at the restaurants while 68.3% of them visited the library. However, it was revealed the number of people who visit the library but do not read was about 7.89%. The average daily time spent by Finnish was estimated to be 77.98 minutes for working, 529.17 for sleeping and 56.65 for reading. These values were believed not to be a true reflection of these activities due to some outliers in the data. Moreover, the data points from the active variables were not normally distributed hence using the mean in this context was deemed problematic and hence not trustworthy.

Table 2: Baseline descriptive statistics of the variables

Demographic variables	Statistic
Sex – no. (%)	
Male	374 (47.95)
Female	406 (52.05)
Age group (years) – no. (%)	
20 – 24	34 (04.35)
25 – 34	106 (13.59)
35 – 44	132 (16.92)
45 – 54	170 (21.79)

55 – 64	165 (21.15)
65 – 74	112 (14.35)
75 and above	61 (07.82)
Day of week – no. (%)	
Working day	418 (53.59)
Weekend	362 (46.41)
Living environment – no. (%)	
City	519 (66.53)
Municipality	124 (15.89)
Rural area	137 (17.56)
Activity Variables	
Working - mean (s.d)	77.98 (160.34)
Sleeping - mean (s.d)	529.17 (103.29)
Reading - mean (s.d)	56.65 (75.46)
Dining at a restaurant – no. (%)	
Yes	418 (53.59)
No	362 (46.41)
Visiting the library – no. (%)	
Yes	533 (62.95)
No	247 (37.05)

It can be observed from Figure 1 that most of the respondents (male and female) live in the city whereas there is an almost even distribution of male and female residents Finnish people in both municipalities and rural areas.

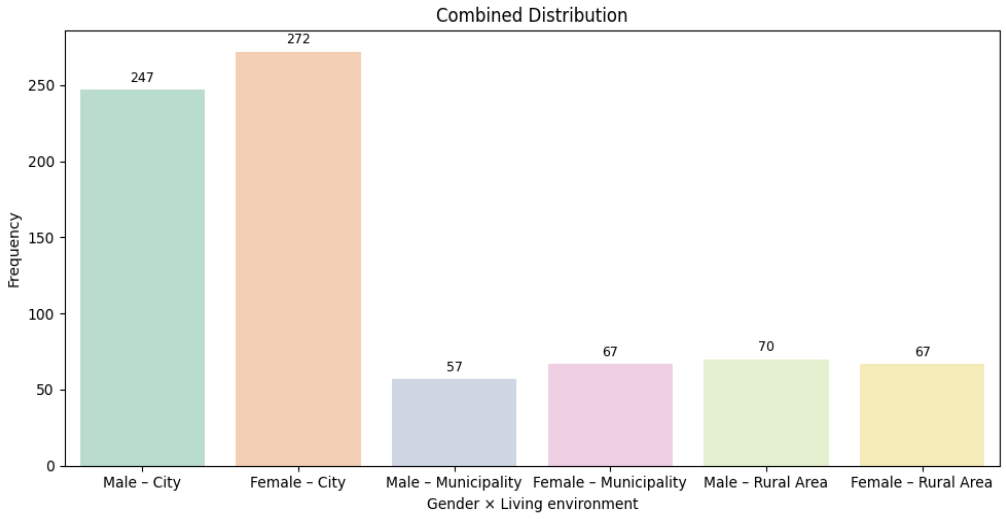

Figure 1: Distribution of Respondent by sex and living environment

Cluster Analysis of the Activity Variables

The activity variables (working, sleeping, reading) were subjected to cluster analysis to group the Finnish population based on time allocation patterns. Table 3 and Figure 2 present the results of clustering and the average minutes spent on these activities. The analysis identified two distinct clusters, as illustrated in Figure 2. From Table 3, individuals in Cluster 1 allocate a substantial amount of time to working, whereas those in Cluster 2 spend little to no time working. On average, Cluster 2 individuals spend more time sleeping and reading compared to their Cluster 1.

Generally, Cluster 1 represents a working group characterized by high working hours and lower time spent sleeping and reading whereas, Cluster 2 represents a non-working and leisure group, characterized by minimal or no work hours, increased sleep, and greater time devoted to reading. Cluster 2 we believe might be students, the retired and old aged Finnish population not working and having more time for sleep and reading.

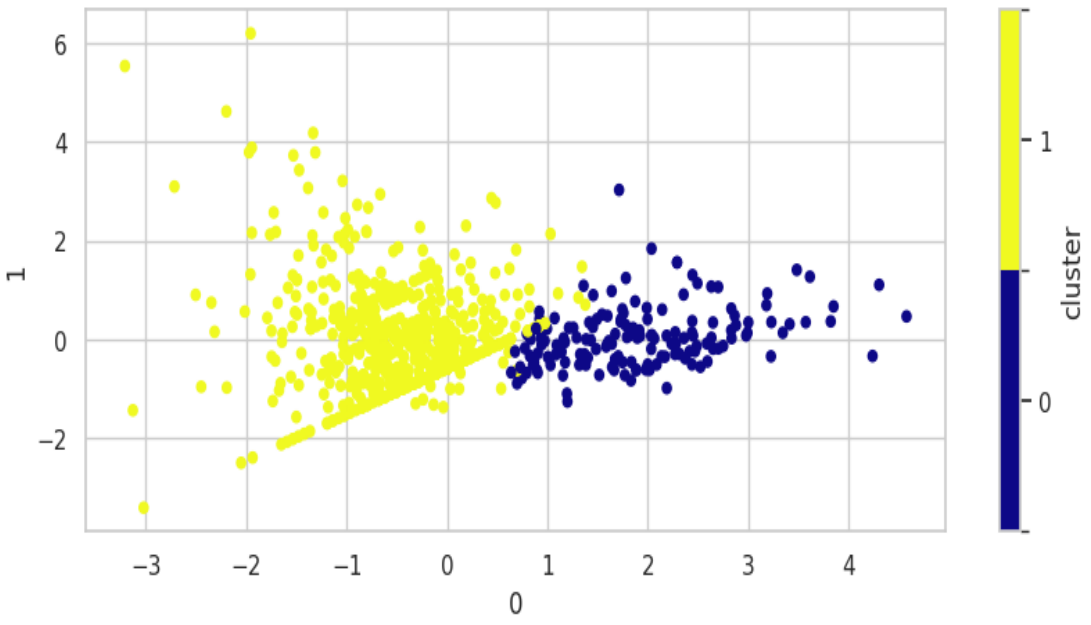

Figure 2: Clustering analysis based on the activity variables

Table 3: Average time spent on activity variables based on clusters

Activity variables	Cluster 1	Cluster 2
Working	345 Minutes (~6 hours)	0 Minutes
Sleeping	460 Minutes (~7.5 hours)	540 Minutes (9 hours)
Reading	10 Minutes	40 Minutes

Principal component analysis on Activity variables

Principal component analysis was conducted on the activity variables: working, reading and sleeping. From the PCA, it was revealed that about 47% of the total variation in the data is explained by the first principal component, 33% variation is explained by the second component while the third component explains 21% of the variation. Table 4 presents the PCA scores. From Table 4, it can be not that for the first component working and sleeping are quite prominent than reading though sleeping relates negatively to working. Furthermore, from the second component, reading and sleeping are prominent and also relates opposite or negatively to one another.

Table 4: PCA scores for activity variables

Components	Working	Sleeping	Reading
0	0.705129	-0.589727	-0.393720
1	0.004743	-0.551321	0.834280
2	0.709063	0.590142	0.385955

Estimation of the average daily time Finnish households spend on activity variables

Analysis from the data cleaning showed that some Finnish were not working and not reading at all. Table 5 present the information on Finnish people that were not working and reading as well as those that were reading. From Table 5, majority of the Finnish population were not working with only a quarter of them working whereas 65.5% of them were people who read with the remaining 34.5% not reading.

Table 5: Distribution of Respondents based working and reading

Activity variables	Frequency	Percentage
Working		
Not Working	595	75.0
Working	195	25.0
Reading		
Not reading	269	34.5
Reading	511	65.5

The estimated daily average of the Finnish people who were working and reading was based on the sub population who were working and reading using the median score. The median was used due to outliers found in the data. Table 6 present the average estimated time of activity variables undertaken by the Finnish people. From Table 6, Finnish working people work on the average 225 minutes making 4.25 hours whereas the Finnish reading people also read on the average 60 minutes (1 hour). For sleeping, Finnish people sleep for 530 minutes (8.83 hours).

Table 6: Average Time of activity variable using central tendency median

Activity variables	**Average time in minutes**	**Average time in hours**
Working	255	4.25
Sleeping	530	8.83
Reading	60	1.0

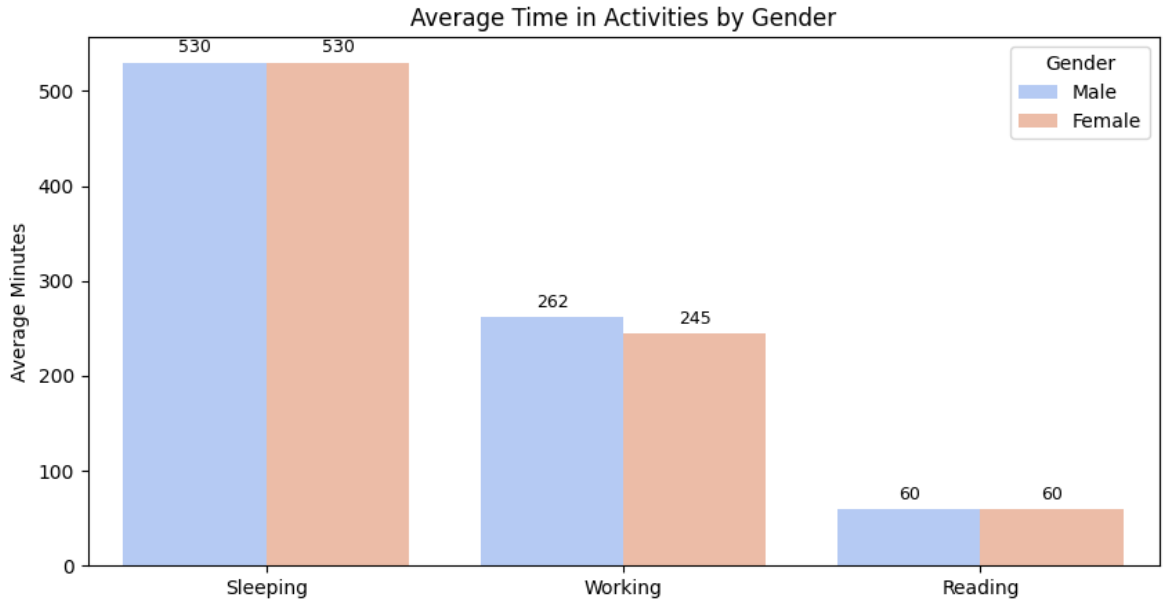

Figure 3: Distribution of activity variables based on Gender

Figure 3 illustrates that on average sleeping and reading variables have the same Average irrespective of the Gender, but we see a difference in working for gender as male tends to work more than female.

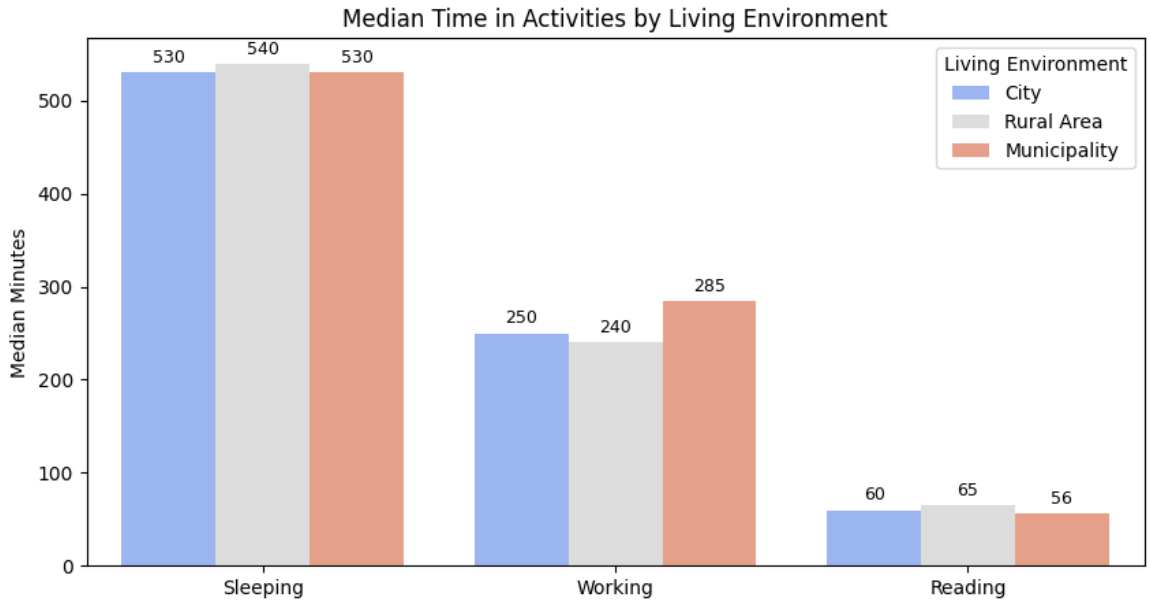

Figure 4: Distribution of activity variables based on living environment

Figure 4 illustrates that people work more in municipality and work less in rural areas. In reading and sleeping the rural area has spent more time on average.

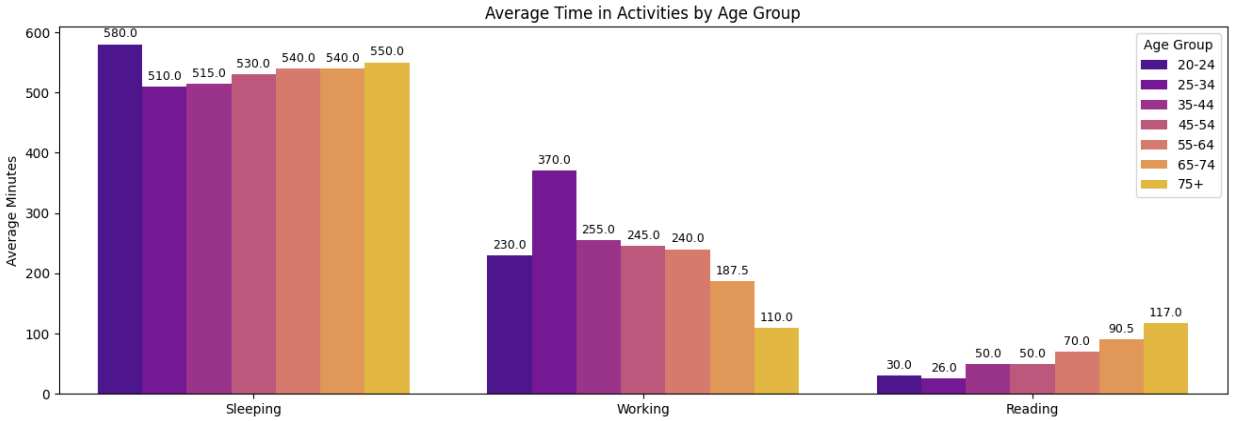

Figure 5: Distribution of activity variable based on age groups

From Figure 5, We see there is a trend in Reading as the age passes, people read more and the oldest ones read around **2** hours on average. For Working, the age group 25-34 works more than all the other groups for almost **6** hours on average and the trend declines after that for every age group. For Sleeping, The younger ones (20-24 years) sleep for almost **10** hours on average.

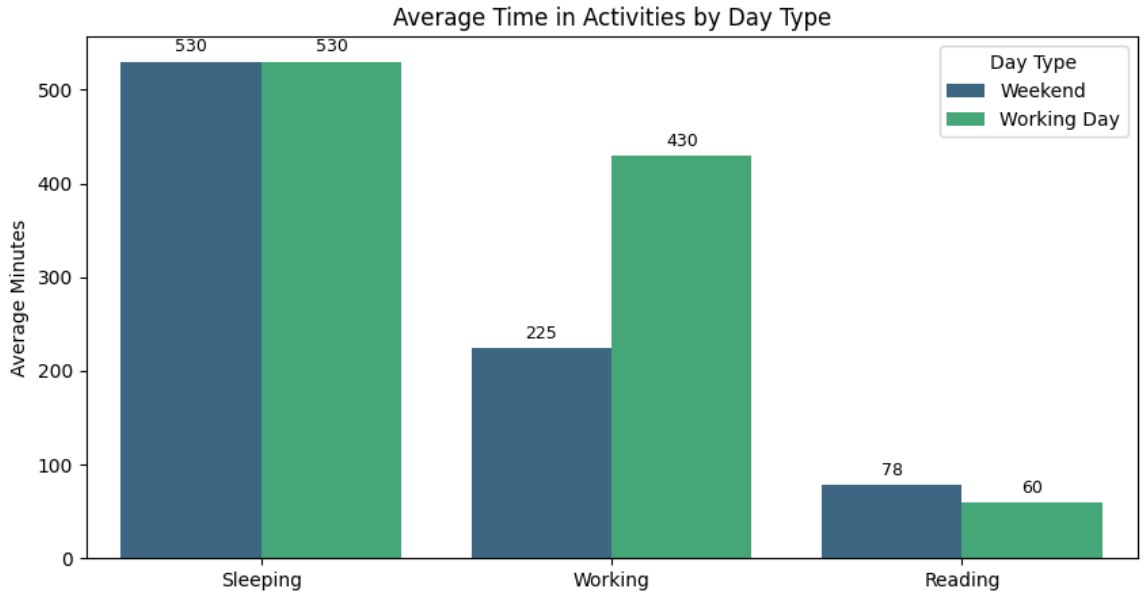

Figure 6: Distribution of activity variables based on Day of week

It is an interesting fact that Finns work on weekends for **225** minutes (almost 4 hours) on average, and **430** minutes (more than 7 hours) on working days. There is no

difference found between the sleeping in this group as illustrated in Figure 6. Finns tend to read more on weekends instead of working days.

Differences in Activity Variables based on Living Environment in Finland

To test the difference in the activity variables, a normality test was conducted on the variables based on the living environment. The null hypothesis of normally distributed data was tested against the alternative hypothesis of non-normally distributed data using the Shapiro-Wilk test. Table 7 displays the results of the analysis, with p-values less than 0.05 indicating significance of the test, hence rejecting the null hypothesis of normally distributed data. From Table 7 we can conclude that the activity variables working, sleeping and reading based on the living environment of respondents are not normally distributed. The non-parametric test, the Kruskal-Wallis test, for more than two independent samples, will be conducted to test the difference in the activity variables.

Table 7: Normality test of activity variables based on living environment

Source of variation	Groups	W-statistic	p-value	Decision
Working	City	0.565	< .001	Not normal
	Municipality	0.553	< .001	Not normal
	Rural	0.537	< .001	Not normal
Sleeping	City	0.17423	< .001	Not normal
	Municipality	0.17423	0.0077	Not normal
	Rural	0.17423	0.6851	Normal
Reading	City	9902.0	< .001	Not normal
	Municipality	9902.0	< .001	Not normal
	Rural area	9902.0	< .001	Not normal

The Kruskal-Wallis test was conducted with the null hypothesis of no significant difference in the minutes of activity variables performed by respondents based on their living environment against the alternative hypothesis of a significant difference in the minutes of activity variables performed. The results of the test for the activity variables are displayed in Table 8. The results in Table 8 indicate that there is no significant

difference in the working and reading minutes of respondents in the city, municipality and rural areas. However, there was a significant difference in the sleeping time of respondents in the city, municipality and rural areas. The pairwise comparisons showed that there was a significant difference in the sleeping minutes between rural area respondents and both city and municipality respondents, whereas there was no significant difference between city and municipality respondents.

Table 8: Difference in activity variables based on the living environment of respondents

Test	Source of variation	Groups	Statistic	df	P-value	Decision
Kruskal-Wallis	Working	Living environment	0.1742	2	0.9166	Not significant
Kruskal-Wallis	Reading	Living environment	3.6951	2	0.1576	Not significant
Kruskal-Wallis	Sleeping	Living environment	10.8915	2	0.0043	Significant
Post-hoc test						
Mann-Whitney U	Sleeping	City vs rural areas	29113.5	1	0.001096	Significant
Mann-Whitney U	Sleeping	City vs municipality	31439.5	1	0.691099	Not significant
Mann-Whitney U	Sleeping	Rural area vs municipality	9902.0	1	0.020705	Significant

Differences in Activity Variables based on Day of Week the activity is performed

To test the difference in the activity variables, a normality test was conducted on the variables based on the day of the week activities are performed. The null hypothesis

of normally distributed data was tested against the alternative hypothesis of non-normally distributed data using the Shapiro-Wilk test. Table 9 displays the results of the analysis, with p-values less than 0.05 indicating significance of the test, hence rejecting the null hypothesis of normally distributed data. The results from Table 9, indicate that the activity variables working, sleeping and reading based on the day of the week that respondents performed these activities are not normally distributed. The non-parametric test, Mann-Whitney U test, for two independent samples, would be conducted to test the difference in the activity variables.

Table 9: Normality test of activity variables based on living environment

Source of variation	Groups	W-statistic	P-value	Decision
Working	Weekend	3.79207	< .001	Not normal
	Working day	3.79207	< .001	Not normal
Sleeping	Weekend	1700.0	< .001	Not normal
	Working day	1700.0	< .001	Not normal
Reading	Weekend	73498.5	< .001	Not normal
	Working day	73498.5	< .001	Not normal

The Mann-Whitney U test was conducted with the null hypothesis of no significant difference in the minutes of activity variables performed by respondents based on day of week against the alternative hypothesis of a significant difference in the minutes of activity variables performed. The results of the test for the activity variables are displayed in Table 10. The results in Table 10 indicate that there is a significant difference in the minutes of working and reading by respondents on the working day and the weekend. However, there was no significant difference in the sleeping time of respondents performing this activity on a working day and on the weekend.

Table 10: Difference in activity variables based on the day of the week activity was performed by respondents.

Test	Source of variation	Groups	Statistic	df	P-value	Decision
Mann-Whitney U	Working	Day of week	1700.0	1	< .001	Significant

Mann-Whitney U	Sleeping	Day of week	73498.5	1	0.4912	Not significant
Mann-Whitney U	Reading	Day of week	38785.50	1	< .001	Significant

Differences in Activity Variables based on the Sex of Respondent in Finland

To test the difference in the activity variables, a normality test was conducted on the variables based on the sex of respondents. The null hypothesis of normally distributed data was tested against the alternative hypothesis of non-normally distributed data using the Shapiro-Wilk test. Table *11* displays the results of the analysis, with p-values less than 0.05 indicating significance of the test, hence rejecting the null hypothesis of normally distributed data. The results from Table 11, indicate that the activity variables working, sleeping and reading based on the sex of respondents are not normally distributed. The non-parametric test, Mann-Whitney U test, for two independent samples, would be conducted to test the difference in the activity variables.

Table 11: Normality test of activity variables based on living environment

Source of variation	Groups	W-statistic	p-value	Decision
Working	Female	0.195762	0.018689	Not normal
	Male	0.195762	0.010305	Not normal
Sleeping	Female	4341.0	< .001	Not normal
	Male	4341.0	< .001	Not normal
Reading	Female	75697.0	< .001	Not normal
	Male	75697.0	< .001	Not normal

The Mann-Whitney U test was conducted with the null hypothesis of no significant difference in the minutes of activity variables performed by respondents based on their sex against the alternative hypothesis of a significant difference in the minutes of activity variables performed. The results of the test for the activity variables are displayed in Table 12. The results in Table 12 indicate that there is no significant difference in the minutes of the performance of the activity variables based on the respondent's sex. This shows that there is no difference in the working, sleeping and reading time of males and females in Finland.

Table 12: Difference in activity variables based on the day of the week the activity was performed by respondents.

Test	Source of variation	Groups	Statistic	df	P-value	Decision
Mann-Whitney U	Working	Sex	4341.0	1	0.4588	Not significant
Mann-Whitney U	Sleeping	Sex	75697.0	1	0.9430	Not significant
Mann-Whitney U	Reading	Sex	32726.5	1	0.7922	Not significant

Association of activity variables and Demographic variables

The test of association was conducted on categorical variables Dine-out and Visiting-library. **Table 13** illustrates that going to the library is dependent on age-group, so there is a significant association between them $X^2(6) = 58.5917, p = 0.0000$. The age-group and living-environment have significant association with dine-out as well $X^2(6) = 90.5619, p = 0.0000$ and $X^2(2) = 16.4043, p = 0.0003$.

Table 13: Association Results

	Statistic	df	p-value	Association
Dine out				
Gender	1.2974	1	0.2547	Not Significant
Age Group	90.5619	6	0.0000	Significant
Living Environment	16.4043	2	0.0003	Significant
Day Type	1.4956	1	0.2213	Not Significant
Visiting Library				
Gender	2.9073	1	0.0882	Not Significant
Age Group	58.5917	6	0.0000	Significant
Living Environment	3.7921	2	0.1502	Not Significant
Day Type	0.1958	1	0.6582	Not Significant

Analysis of the Relationship between Activity Variables

To determine the relationship between the activity variables (working, sleeping, reading) we adopted the spearman correlation because it first ranks the values and repeated zeros will not affect the relationship, also the main reason to use this method is for the outliers. A Spearman correlation analysis showed a significant negative relationship between working and sleeping time ($\rho = -0.34$, $p < 0.001$), indicating that individuals who work more tend to sleep less. A weaker but significant negative relationship was also found between working and reading time ($\rho = -0.21$, $p < 0.001$), so, people who work more tend to read slightly less, though not as strongly as the work-sleep link. No significant relationship was observed between sleeping and reading ($\rho = 0.02$, $p = 0.52$).

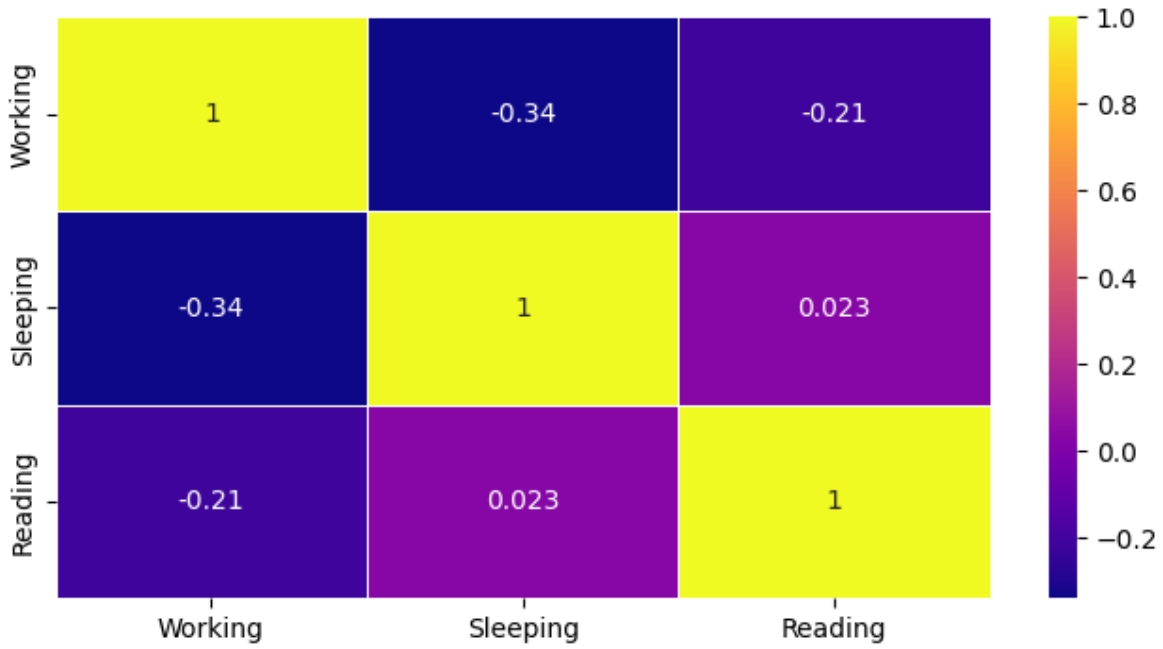

Figure 7: Correlation Matrix of Activities

Appendix:

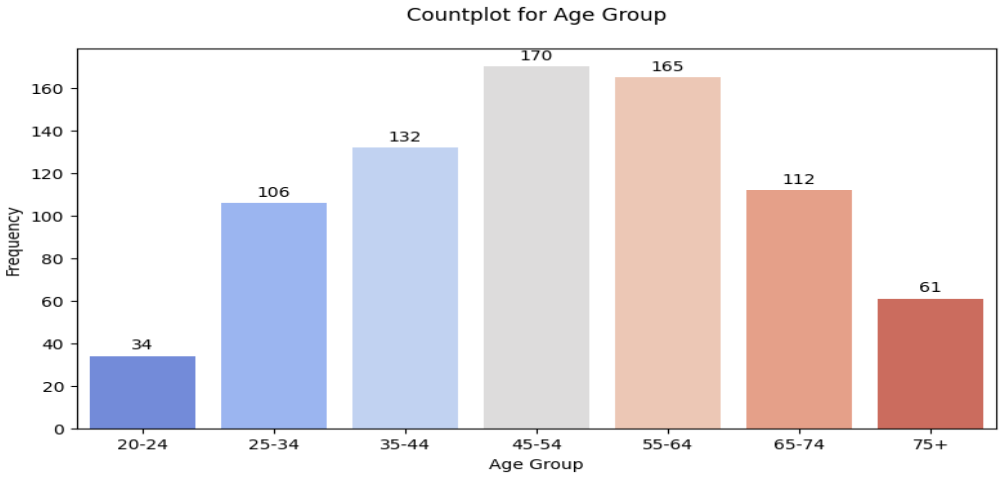

Figure 1: Distribution of respondents by Age groups

Code:

```
# **Installation**
!pip install scikit-posthocs
# **Libraries to import**
import pandas as pd
import scikit_posthocs as sp
import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from scipy.stats import chi2_contingency, kruskal, mannwhitneyu
import scipy.stats
import itertools
from scipy.stats import shapiro
import scipy.stats as S
import seaborn as sns
import matplotlib.pyplot as plt
import sklearn.cluster as cluster
from sklearn.cluster import KMeans
import sklearn.decomposition as SD
from statsmodels.stats.multitest import multipletests
# **Data Reading**
df = pd.read_csv('/content/habits.data', sep=';')
df.describe()
# **Data Preparation and Cleaning**
# function to convert ? to 0 and to convert hh:mm to minutes
def standardize_time_column(value):
    if isinstance(value, str):
        clean_str = value.strip()
```

```
        if clean_str in ('?'):
            return 0
        if ":" in clean_str:
                hours, minutes = map(int, clean_str.split(":"))
                return float((hours * 60) + minutes)
        else:
            return float(clean_str)
    return None
df.info()
def covert_sp_male_female(v): # function to convert as Male and Female
    if v == 1:
        return 'Male'
    else:
        return 'Female'
df['Gender'] = df['sp'].apply(covert_sp_male_female) # new variable Gender
df.groupby('Gender').count()
male_percentage = (df['Gender'] == 'Male').mean() * 100
female_percentage = (df['Gender'] == 'Female').mean() * 100
print(f'Male % = {male_percentage.round(2)}')
print(f'Female % = {female_percentage.round(2)}')
def covert_living_env(v): # function to convert living environment values to strings
    if v == 1:
        return 'City'
    elif v==2:
        return 'Municipality'
    else:
        return 'Rural Area'
df['Living Environment'] = df['ASALUE'].apply(covert_living_env)
```

```
df.groupby('Living Environment').count()
envir=['City','Municipality','Rural Area']
print('Living Environment:\n')
for i in envir:
  print(f'{i}: {(df["Living Environment"] == i).mean()*100:.2f}%')
def covert_day_of_week(v): #function to convert Day Type
    if v == 1:
        return 'Working Day'
    else:
        return 'Weekend'
df['Day Type'] = df['pvknro'].apply(covert_day_of_week)
df.groupby('Day Type').count()
daytype=['Working Day','Weekend']
print('Day Type:\n')
for i in daytype:
  print(f'{i}: {(df["Day Type"] == i).mean()*100:.2f}%')
  def convert_age_group(v): # to convert age group
    if v == 3:
      return '20-24'
    elif v == 4:
      return '25-34'
    elif v == 5:
      return '35-44'
    elif v == 6:
      return '45-54'
    elif v == 7:
      return '55-64'
    elif v == 8:
```

```
      return '65-74'
    else:
      return '75+'
df['Age Group']=df['IKAL1'].apply(convert_age_group)
age_groups = ['20-24', '25-34', '35-44', '45-54', '55-64', '65-74', '75+']

for group in age_groups:
    percentage = (df['Age Group'] == group).mean() * 100
    print(f"{group}: {percentage:.2f}%")

df.head(4)
df['Working'] = df['A1'].apply(standardize_time_column)
#to convert ? to NAN and to convert hh:mm to minutes

def convert_a2_to_minutes(value):
    if isinstance(value, str):
        clean_str = value.strip()
        if clean_str in ('?'):
            return None # Return None for '?' to be filled later
        if ":" in clean_str:
                hours, minutes = map(int, clean_str.split(":"))
                return float((hours * 60) + minutes)
        else:
                return float(clean_str)
    return None
# Apply the conversion function to create a new column with numeric minutes and
NaNs
df['Sleeping'] = df['A2'].apply(convert_a2_to_minutes)
```

```
# Perform group-wise median imputation on the new numeric column
df['Sleeping'] = df.groupby(['IKAL1', 'sp'])['Sleeping'].transform(lambda x: x.fillna(x.median()))
#df['A2_Minutes'] = df['A2_Minutes'].astype(int) # convert float into int for the decimals of the mean values
#df['A2_Minutes'] = df['A2_Minutes'].astype(float) # convert back into float from int because others are in float
df['Reading'] = df['A3'].apply(standardize_time_column)
df.describe()
#df.to_csv('cleaned_habits.csv', index=False)
df.columns[9:11]
for i in df.columns[9:11]: # to convert A4 and A5 to numeric data type
    df[i+'_New'] = pd.to_numeric(df[i], errors='coerce')
df.head(10)
df.describe()
#df['A4'].info()
#df['A4'].mode()[0]
# to convert 0s and NANs of A4 and A5 using mode()

for i in df.columns[18:20]:
  print('Mode() of '+i+' is ',df[i].mode()[0])
  df[i] = df[i].fillna(df[i].mode()[0])
  df.loc[df[i] == 0, i] = 2
  df.loc[df[i] > 2, i] = 1
#for i in df['A4_New']:
 # print(i)
df.head(5)
def convert_a4_5_to_yes_no(v):
  if v == 1:
```

```
    return 'Yes'
  else:
    return 'No'
df['A4_New_'] = df['A4_New'].apply(convert_a4_5_to_yes_no)
df['A5_New_'] = df['A5_New'].apply(convert_a4_5_to_yes_no)
df2 = df.drop(columns=df.columns[0:15])
df2.drop(columns=['A4_New_','A5_New_'],inplace=True)
df2.drop(columns=['A4_New','A5_New'],inplace=True)
df2.head(1) # df2 only contains the continuous activities
# **Graphs** (Descriptive Analysis)
#for i in df.columns[6:11]:
 # print(df[i])
#df = df.drop(columns=df.columns[6:11])
df.head(2)
df.describe()
#df.isna().sum()
#df.to_csv('updated_habits.csv',index=False)
df.head(1)
order = ["20-24", "25-34", "35-44", "45-54","55-64","65-74","75+"]
ax=sns.countplot(df,x='Age Group', palette='coolwarm', order=order)
for container in ax.containers:
    ax.bar_label(container, fmt='%d', label_type='edge', padding=3, fontsize=10, color='black')
plt.show()
# '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+'
```

According to the above graph, we came to conclusion that the age between 35-64 years has 59.87% in the data distribution.

```
age_order = ["20-24", "25-34", "35-44", "45-54", "55-64", "65-74", "75+"] # to maintain the order for age group

cols = ['Age Group', 'Gender', 'Living Environment'] #columns used


for col in cols:

    plt.figure(figsize=(6, 4))

    if col == 'Age Group':

        ax = sns.countplot(data=df, x=col, palette='coolwarm', order=age_order) #this plot has been ordered using age group

    else:

        ax = sns.countplot(data=df, x=col, palette='coolwarm',hue='Gender')


    for container in ax.containers:

        ax.bar_label(container, fmt='%d', label_type='edge', padding=3, fontsize=10, color='black')

    plt.title(f"Countplot for {col}\n")

    plt.xlabel(col)

    plt.ylabel("Frequency")

    plt.tight_layout()

    plt.show()

# calculate the percentages distribution


def plot_countplot_with_percent(df, col, order=None, palette='coolwarm'):

    ax = sns.countplot(data=df, x=col, order=order, palette=palette)

    total = len(df)

    for container in ax.containers:

        ax.bar_label(container,

                     labels=[f'{(v.get_height()/total)*100:.1f}%' for v in container],

                     label_type='edge', padding=3, fontsize=9, color='black')
```

```
    plt.title(f"Percentage Distribution of {col}")
    plt.xlabel(col)
    plt.ylabel("Frequency")
    plt.tight_layout()
    plt.show()

for c in ['Age Group', 'Gender', 'Living Environment']:
    plot_countplot_with_percent(df, col=c, order=age_order if c == 'Age Group' else
None)

def plot_countplot_with_percent(df, col, order=None, palette='coolwarm'):
    ax = sns.countplot(data=df, x=col, order=order, palette=palette, hue='Gender')
    total = len(df)
    for container in ax.containers:
        ax.bar_label(container,
                     labels=[f'{(v.get_height()/total)*100:.1f}%' for v in container],
                     label_type='edge', padding=3, fontsize=9, color='black')
    plt.title(f"Percentage Distribution of {col}\n")
    plt.xlabel(col)
    plt.ylabel("Count")
    plt.tight_layout()
    plt.show()

for c in ['Age Group', 'Gender', 'Living Environment']:
    plot_countplot_with_percent(df, col=c, order=age_order if c == 'Age Group' else
None)
count=df.groupby(['Gender','Living Environment']).size().reset_index(name='count')
print(count)
ordre=['City','Municipality','Rural Area']
```

```
count["Gender & Living Environment"] = count["Gender"] + " – " + count["Living Environment"]


sp_values = count['Gender'].unique().tolist()
order_labels = []
for area in ordre:
    for sp in sp_values:
        order_labels.append(f"{sp} – {area}")


# Filter order_labels to only those present in count (prevents missing combos)
order_labels = [lbl for lbl in order_labels if lbl in count['Gender & Living Environment'].values]


plt.figure(figsize=(10,5))
ax3 = sns.barplot(x=count["Gender & Living Environment"], y=count["count"], palette=plt.cm.Pastel2.colors, order=order_labels)


total = count["count"].sum()
for container in ax3.containers:
    ax3.bar_label(container,fmt='%d',
                  label_type='edge', padding=3, fontsize=9, color='black')
plt.title("Combined Distribution")
plt.ylabel("Frequency")
plt.tight_layout()
plt.show()
#df.describe()
#!rm -rf /data/charts
#ai_avg= (df['A1_Minutes'].sum())/780
#df = df.drop(columns=['time spend on daily activity'])
```

#averages=df.describe().iloc[1:2,6:9]

#print(ai_avg)

#print(df.columns[6:9])

#averages

Estimate how much time on average (using mean) **Finnish households spend daily on each activity.**

average_time_on_daily_activity=df.describe().iloc[1:2,6:9].round(2).iloc[0].tolist() # daily activity average for A1, A2, and A3

average_time_on_daily_activity # average time in minutes spent on daily basis by a household (mean)

average_time_per_hour_daily = (np.array(average_time_on_daily_activity) / 60).round(2) # we have change the list to array here for our usage

print(average_time_per_hour_daily) #time spent in hours on daily basis by every household

The Aveargе time (hours) spent by the Finnish Households for the activities Working, Sleeping, and Reading are 1.3, 8.82, and 0.94 respectively. (In this analysis 0s are not eliminated)

Remove rows where A1_Minutes is 0

df0 = df[df['Working'] != 0]

df0.reset_index(drop=True, inplace=True)

df0.info()

sns.boxplot(df0['Working'])

average_time_spent_A1_without_0=df0.describe().iloc[1:2,6:7].round(2).iloc[0].tolist()

print(average_time_spent_A1_without_0)

print((np.array(average_time_spent_A1_without_0) / 60).round(2))

By Removing the outliers

sns.boxplot(df0['Working'])

plt.show()

#df0.head(10)

```
average_time_spent_A1_without_0=df0.describe().iloc[1:2,6:7].round(2).iloc[0].tolist()

average_time_spent_A1_without_0 #Minutes for A1

(np.array(average_time_spent_A1_without_0) / 60).round(2) #hours for A1

sns.boxplot(df['Sleeping'])

sns.histplot(df['Sleeping'],kde=True)

df.describe()

df9=df[(df['Reading']!=0)] # remove 0s

df9.info()

sns.boxplot(df9['Reading'])

# **Estimate how much time on average** (using median) **Finnish households spend daily on each activity.** (median was used because of the outliers)

df0.info()

print(df9['Reading'].median()) # after removing 0s

# 60 minutes

print(df9['Reading'].median()/60) # reading in hours

print(df['Sleeping'].median())

print(round(df['Sleeping'].median()/60, 2)) # sleeping in hours

print(df0['Working'].median()) # after removing 0s

print(df0['Working'].median()/60) # working in hours

df9.describe()

#Time spent in minutes for Dine-In and Visiting_Library


columns = ['A4_New', 'A5_New']

values_to_check = [1.0, 2.0]

all_counts = [[df[col].value_counts().get(v, 0) for v in values_to_check] for col in columns]

print(all_counts)

#average time spent in hours for Dine-In and Visiting Library
```

```
for a in all_counts:
  percentages = [(count / 780) * 100 for count in a]
  print([round(p, 2) for p in percentages])
```

About 53.6% persons have dines out and 68.3% persons visited the library for the past year

Those who visit ibrary but didn't read

```
lib_visit=df['A5_New'].sum()
no_read=df[(df['A5_New'] == 2.0) & (df['Reading']==0)].shape[0]
print(f"Visiting Library but not Reading: {((no_read/lib_visit)*100):.2f}%")
```

Average time in activities with demograhpic variables. **Plot using median() and also df, df0, and df9** (instead of only original df)

```
df.columns[15:18]
df_G_A = df.melt(id_vars='Gender',
                 value_vars=df.columns[15:18],
                 var_name='Activity',
                 value_name='Value')

plt.figure(figsize=(10,5))
ax3=sns.barplot(x='Activity', y='Value', hue='Gender', data=df_G_A, palette='coolwarm',ci=None)

for container in ax3.containers:
    ax3.bar_label(container,fmt='%d', label_type='edge', padding=3, fontsize=9, color='black')

plt.title("Average Time in Activities by Gender")
plt.xlabel("")
plt.ylabel("Average Minutes")
```

```
plt.show()
df_sleeping = df[['Gender', 'Sleeping']].rename(columns={'Sleeping':'Value'})
df_sleeping['Activity'] = 'Sleeping'

df_working = df0[['Gender', 'Working']].rename(columns={'Working':'Value'})
df_working['Activity'] = 'Working'

df_reading = df9[['Gender', 'Reading']].rename(columns={'Reading':'Value'})
df_reading['Activity'] = 'Reading'

df_all = pd.concat([df_sleeping, df_working, df_reading], ignore_index=True)

# Plot using median
plt.figure(figsize=(10,5))
ax = sns.barplot(x='Activity', y='Value', hue='Gender',
data=df_all,estimator=np.median, palette='coolwarm', errorbar=None)

# Adding bar labels for each bar
for container in ax.containers:
    for bar in container:
        height = bar.get_height()  # get the median computed by seaborn
        ax.annotate(f'{int(height)}',xy=(bar.get_x() + bar.get_width()/2,
height),xytext=(0, 3),  # 3 points vertical offset
                    textcoords='offset points',
                    ha='center', va='bottom', fontsize=9, color='black')

plt.title("Average Time in Activities by Gender")
plt.xlabel("")
plt.ylabel("Average Minutes")
```

```
plt.show()
df_LE_A = df.melt(id_vars='Living Environment',
                  value_vars=df.columns[15:18],
                  var_name='Activity',
                  value_name='Value')

activity_order = df_LE_A['Activity'].sort_values().unique()
plt.figure(figsize=(10,5))

ax3=sns.barplot(x='Activity', y='Value', hue='Living Environment', data=df_LE_A,
palette='coolwarm',ci=None, order=activity_order)

for container in ax3.containers:
    ax3.bar_label(container,fmt='%d', label_type='edge', padding=3, fontsize=9,
color='black')

plt.title("Average Time in Activities by Living Environment")
plt.xlabel("")
plt.ylabel("Average Minutes")
plt.show()
df_sleeping = df[['Living Environment',
'Sleeping']].rename(columns={'Sleeping':'Value'})
df_sleeping['Activity'] = 'Sleeping'

df_working = df0[['Living Environment',
'Working']].rename(columns={'Working':'Value'})
df_working['Activity'] = 'Working'

df_reading = df9[['Living Environment',
'Reading']].rename(columns={'Reading':'Value'})
```

```
df_reading['Activity'] = 'Reading'


df_all = pd.concat([df_sleeping, df_working, df_reading], ignore_index=True)


# Plot using median
plt.figure(figsize=(10,5))
ax = sns.barplot(x='Activity', y='Value', hue='Living Environment',
data=df_all,estimator=np.median, palette='coolwarm', errorbar=None)


# Adding bar labels for each bar
for container in ax.containers:
    for bar in container:
        height = bar.get_height()  # get the median computed by seaborn
        ax.annotate(f'{int(height)}',xy=(bar.get_x() + bar.get_width()/2,
height),xytext=(0, 3),  # 3 points vertical offset
                    textcoords='offset points',
                    ha='center', va='bottom', fontsize=9, color='black')


plt.title("Average Time in Activities by Living Environment")
plt.xlabel("")
plt.ylabel("Average Minutes")
plt.show()
# age_order = ["20-24", "25-34", "35-44", "45-54", "55-64", "65-74", "75+"]
# activity_order = df.columns[15:18].tolist()  # or ['A','B','C'] etc.


# df_AG_A = df.melt(id_vars='Age Group',
#                   value_vars=df.columns[15:18],
#                   var_name='Activity',
#                   value_name='Value')
```

```
# agg = (df_AG_A
#       .groupby(['Activity', 'Age Group'], observed=True)['Value']
#       .mean() # using mean
#       .reset_index())

# plt.figure(figsize=(16,5))

# ax3 = sns.barplot(
#     data=agg,
#     x='Activity',
#     y='Value',
#     hue='Age Group',
#     order=activity_order,
#     hue_order=[a for a in age_order if a in df['Age Group'].unique()],  # only use present groups
#     ci=None,
#     palette='plasma',
#     dodge=True
# )

# # Annotate bars with one decimal place
# for container in ax3.containers:
#     labels = [f'{bar.get_height():.1f}' for bar in container]
#     ax3.bar_label(container, labels=labels, label_type='edge', padding=3, fontsize=9, color='black')

# plt.title("Average Time in Activities by Age Group")
# plt.xlabel("")
```

```
# plt.ylabel("Average Minutes")
# plt.show()
age_order = ["20-24", "25-34", "35-44", "45-54", "55-64", "65-74", "75+"]
activity_order = ['Sleeping', 'Working', 'Reading']

df_sleeping_AG = df[['Age Group', 'Sleeping']].rename(columns={'Sleeping': 'Value'})
df_sleeping_AG['Activity'] = 'Sleeping'

df_working_AG = df0[['Age Group', 'Working']].rename(columns={'Working': 'Value'})
df_working_AG['Activity'] = 'Working'

df_reading_AG = df9[['Age Group', 'Reading']].rename(columns={'Reading': 'Value'})
df_reading_AG['Activity'] = 'Reading'

df_AG_all = pd.concat([df_sleeping_AG, df_working_AG, df_reading_AG], ignore_index=True)

agg = (df_AG_all
       .groupby(['Activity', 'Age Group'], observed=True)['Value']
       .median()
       .reset_index())

# Plot
plt.figure(figsize=(16,5))

ax3 = sns.barplot(
```

```
    data=agg,
    x='Activity',
    y='Value',
    hue='Age Group',
    order=activity_order,
    hue_order=[a for a in age_order if a in agg['Age Group'].unique()],
    ci=None,
    palette='plasma',
    dodge=True
)

# Annotate bars with one decimal place
for container in ax3.containers:
    labels = [f'{bar.get_height():.1f}' for bar in container]
    ax3.bar_label(container, labels=labels, label_type='edge', padding=3, fontsize=9, color='black')

plt.title("Average Time in Activities by Age Group")
plt.xlabel("")
plt.ylabel("Average Minutes")
plt.show()
df_sleeping = df[['Day Type', 'Sleeping']].rename(columns={'Sleeping':'Value'})
df_sleeping['Activity'] = 'Sleeping'

df_working = df0[['Day Type', 'Working']].rename(columns={'Working':'Value'})
df_working['Activity'] = 'Working'

df_reading = df9[['Day Type', 'Reading']].rename(columns={'Reading':'Value'})
```

```
df_reading['Activity'] = 'Reading'


df_all = pd.concat([df_sleeping, df_working, df_reading], ignore_index=True)


# Plot using median
plt.figure(figsize=(10,4))
ax = sns.barplot(x='Activity', y='Value', hue='Day Type',
data=df_all,estimator=np.median, palette='viridis', errorbar=None)


# Add bar labels for each bar
for container in ax.containers:
    for bar in container:
        height = bar.get_height()  # get the median computed by seaborn
        ax.annotate(f'{int(height)}',xy=(bar.get_x() + bar.get_width()/2,
height),xytext=(0, 3),  # 3 points vertical offset
                    textcoords='offset points',
                    ha='center', va='bottom', fontsize=9, color='black')


plt.title("Average Time in Activities by Day Type")
plt.xlabel("")
plt.ylabel("Average Minutes")
plt.show()
#df.describe()
# **Correlation among the Activities**
plt.figure(figsize=(8, 4))
sns.heatmap(df[['Working','Sleeping','Reading']].corr(method='spearman'),
annot=True, cmap='plasma', linewidths=0.5)
plt.show()
#spearman is perfect in this case
```

```
from scipy.stats import spearmanr
cols = ['Working', 'Sleeping', 'Reading']

for i in range(len(cols)):
    for j in range(i+1, len(cols)):
        col1, col2 = cols[i], cols[j]
        corr, p_value = spearmanr(df[col1], df[col2])
        print(f"{col1} vs {col2}: correlation = {corr:.3f}, p-value = {p_value:.4f}")
# **PCA and Clusters**
scaler=StandardScaler()
scaler.fit(df2)
scaled_data = scaler.transform(df2)
pca = SD.PCA(n_components=3).fit(scaled_data)
pca.fit(scaled_data)
x_pca = pca.transform(scaled_data)
scaled_data.shape
x_pca.shape
x_pca_df = pd.DataFrame(x_pca, index=df2.index)

df_extended = pd.concat([df2, x_pca_df], axis=1)
scores = pca.transform(scaled_data)[:, :2]
df_extended = pd.DataFrame(scores, columns=['PC1','PC2'])
pca.explained_variance_ratio_ # explained variance
pd.DataFrame(pca.components_, columns=df2.columns) #pca results
kmeans = cluster.KMeans(n_clusters=2, random_state=42).fit(scaled_data) #
Added random_state for reproducibility
predictions = kmeans.predict(scaled_data)
```

```
df_extended['cluster'] = pd.Categorical(predictions)


df_extended.plot.scatter(x='PC1', y='PC2', c='cluster', colormap='plasma',
figsize=(10,4))

df_merge=df.copy()

df_merge['cluster'] = df_extended.cluster

df_merge.groupby('cluster')['Working'].median()

df_merge=df.copy()

df_merge['cluster'] = df_extended.cluster

df_merge.groupby('cluster')['Sleeping'].median()

df_merge=df.copy()

df_merge['cluster'] = df_extended.cluster

df_merge.groupby('cluster')['Reading'].median()

# **Determine difference in activity variables based on Living Environment in
Finland**

#Normality tests on working based on living evironment


results = {}


for group, subset in df.groupby('Living Environment'):

    S, p = shapiro(subset['Working'])

    results[group] = {'W-statistic': S, 'p-value': p}


results_df = pd.DataFrame(results).T

print(results_df)

groups = [subset['Working'].values for name, subset in df.groupby('Living
Environment')]

stat, p = kruskal(*groups) # * is sued because the values are in arrays list

print(f"Kruskal-Wallis H-statistic = {stat:.4f}, p-value = {p:.4f}")
```

```
# We failed to reject the Null hypothesis as there is no difference in the working minutes based on the Living Environment.

results = {}


for group, subset in df.groupby('Living Environment'):
    S, p = shapiro(subset['Sleeping'])
    results[group] = {'W-statistic': stat, 'p-value': p}


results_df = pd.DataFrame(results).T
print(results_df)
groups = [subset['Sleeping'].values for name, subset in df.groupby('Living Environment')]
stat, p = kruskal(*groups)
print(f"Kruskal-Wallis H-statistic = {stat:.4f}, p-value = {p:.4f}")



# # We reject the Null hypothesis as there is a difference in the sleeping minutes based on the Living Environment.
groups = df['Living Environment'].unique()


pairs = list(itertools.combinations(groups, 2))


results = []


for g1, g2 in pairs:
    data1 = df.loc[df['Living Environment'] == g1, 'Sleeping']
    data2 = df.loc[df['Living Environment'] == g2, 'Sleeping']
```

```
    stat, p = scipy.stats.mannwhitneyu(data1, data2, alternative='two-sided')
    results.append({'Group 1': g1, 'Group 2': g2, 'U-statistic': stat, 'p-value': p})

results_df = pd.DataFrame(results)
#results_df['p-adj'] = multipletests(results_df['p-value'], method='bonferroni')[1] # to adjust the p-value so that the false-positive value also handles
#results_df = results_df.sort_values('p-adj')
print(results_df)
results = {}

for group, subset in df9.groupby('Living Environment'): #df9 where we remove 0s
    S, p = shapiro(subset['Reading'])
    results[group] = {'W-statistic': stat, 'p-value': p}

results_df = pd.DataFrame(results).T
print(results_df)

# We reject the Null hypothesis of normally distributed.
## kruskal-wallis test of reading on living environment

groups = [subset['Reading'].values for name, subset in df9.groupby('Living Environment')]
stat, p = kruskal(*groups)
print(f"Kruskal-Wallis H-statistic = {stat:.4f}, p-value = {p:.4f}")

# # We failed to reject the Null hypothesis as there is no difference in the reading minutes based on the Living Environment.
contingency_table = pd.crosstab(df['Living Environment'], df['A4_New'])
```

```
stat, p, dof, expected = chi2_contingency(contingency_table)

print("Chi-square Test of Independence:\n")
print(f"Chi2 Statistic = {stat:.4f}")
print(f"Degrees of Freedom = {dof}")
print(f"P-value = {p:.4f}")

#there is association between them and we reject Ho.
contingency_table = pd.crosstab(df['Living Environment'], df['A5_New'])

stat, p, dof, expected = chi2_contingency(contingency_table)

print("Chi-square Test of Independence:\n")
print(f"Chi2 Statistic = {stat:.4f}")
print(f"Degrees of Freedom = {dof}")
print(f"P-value = {p:.4f}")

#there is no association between them and we failed to reject Ho.
# **Determine difference in activity variables based on Day Type in Finland**
df.head(1)
results = {}

for group, subset in df0.groupby('Day Type'): #df0 where we remove 0s
    S, p = shapiro(subset['Working'])
    results[group] = {'W-statistic': stat, 'p-value': p}

results_df = pd.DataFrame(results).T
print(results_df)
```

```
# # We reject the Null hypothesis of normally distributed.
groups = [subset['Working'].values for name, subset in df0.groupby('Day Type')]
stat, p = mannwhitneyu(*groups)
print(f"MannWhitneyU = {stat:.4f}, p-value = {p:.4f}")

# it's significant, we reject the Null Hypothesis
results = {}

for group, subset in df.groupby('Day Type'): #df0 where we remove 0s
    S, p = shapiro(subset['Sleeping'])
    results[group] = {'W-statistic': stat, 'p-value': p}

results_df = pd.DataFrame(results).T
print(results_df)

# # We reject the Null hypothesis of normally distributed.
groups = [subset['Sleeping'].values for name, subset in df.groupby('Day Type')]
stat, p = mannwhitneyu(*groups)
print(f"MannWhitneyU = {stat:.4f}, p-value = {p:.4f}")

# it's not significant, we failed to reject the Null Hypothesis
results = {}

for group, subset in df9.groupby('Day Type'): #df0 where we remove 0s
    S, p = shapiro(subset['Reading'])
    results[group] = {'W-statistic': stat, 'p-value': p}
```

```
results_df = pd.DataFrame(results).T
print(results_df)

# # We reject the Null hypothesis of normally distributed.
groups = [subset['Reading'].values for name, subset in df9.groupby('Day Type')]
stat, p = mannwhitneyu(*groups)
print(f"MannWhitneyU = {stat:.4f}, p-value = {p:.4f}")

# it's significant, we reject the Null Hypothesis
contingency_table = pd.crosstab(df['Day Type'], df['A4_New'])

stat, p, dof, expected = chi2_contingency(contingency_table)

print("Chi-square Test of Independence:\n")
print(f"Chi2 Statistic = {stat:.4f}")
print(f"Degrees of Freedom = {dof}")
print(f"P-value = {p:.4f}")

#we failed to reject Ho so there is no association
contingency_table = pd.crosstab(df['Day Type'], df['A5_New'])

stat, p, dof, expected = chi2_contingency(contingency_table)

print("Chi-square Test of Independence:\n")
print(f"Chi2 Statistic = {stat:.4f}")
print(f"Degrees of Freedom = {dof}")
print(f"P-value = {p:.4f}")
```

```
#we failed to reject Ho  so there is no association
# **Determine difference in activity variables based on Gender in Finland**
results = {}
for group, subset in df0.groupby('Gender'): #df0 where we remove 0s
    S, p = shapiro(subset['Working'])
    results[group] = {'W-statistic': stat, 'p-value': p}

results_df = pd.DataFrame(results).T
print(results_df)

# # We reject the Null hypothesis of normally distributed.
groups = [subset['Working'].values for name, subset in df0.groupby('Gender')]
stat, p = mannwhitneyu(*groups)
print(f"MannWhitneyU = {stat:.4f}, p-value = {p:.4f}")

# it's not significant, we failed to reject the Null Hypothesis
for group, subset in df.groupby('Gender'): #df0 where we remove 0s
    S, p = shapiro(subset['Sleeping'])
    results[group] = {'W-statistic': stat, 'p-value': p}

results_df = pd.DataFrame(results).T
print(results_df)

# # We reject the Null hypothesis of normally distributed.
groups = [subset['Sleeping'].values for name, subset in df.groupby('Gender')]
stat, p = mannwhitneyu(*groups)
print(f"MannWhitneyU = {stat:.4f}, p-value = {p:.4f}")
```

```
# it's not significant, we failed to reject the Null Hypothesis
for group, subset in df9.groupby('Gender'): #df0 where we remove 0s
    S, p = shapiro(subset['Reading'])
    results[group] = {'W-statistic': stat, 'p-value': p}

results_df = pd.DataFrame(results).T
print(results_df)

# # We reject the Null hypothesis of normally distributed.
groups = [subset['Reading'].values for name, subset in df9.groupby('Gender')]
stat, p = mannwhitneyu(*groups)
print(f"MannWhitneyU = {stat:.4f}, p-value = {p:.4f}")

# it's not significant, we failed to reject the Null Hypothesis
contingency_table = pd.crosstab(df['Gender'], df['A4_New'])

# Run Chi-Square Test of Independence
stat, p, dof, expected = chi2_contingency(contingency_table)

print("Chi-square Test of Independence:\n")
print(f"Chi2 Statistic = {stat:.4f}")
print(f"Degrees of Freedom = {dof}")
print(f"P-value = {p:.4f}")

#we failed to reject Ho so there is no association
contingency_table = pd.crosstab(df['Gender'], df['A5_New'])

stat, p, dof, expected = chi2_contingency(contingency_table)
```

```
print("Chi-square Test of Independence:\n")
print(f"Chi2 Statistic = {stat:.4f}")
print(f"Degrees of Freedom = {dof}")
print(f"P-value = {p:.4f}")

#we failed to reject Ho so there is no association
# **Determine difference in activity variables based on Age Group in Finland**
results={}

#Normality tests on working based on living evironment
for group, subset in df0.groupby('Age Group'): #df0 where we remove 0s
    S, p = shapiro(subset['Working'])
    results[group] = {'W-statistic': S, 'p-value': p}

results_df = pd.DataFrame(results).T
print(results_df)
groups = [subset['Working'].values for name, subset in df0.groupby('Age Group')]
stat, p = kruskal(*groups)
print(f"Kruskal-Wallis H-statistic = {stat:.4f}, p-value = {p:.4f}")

# # We failed to reject the Null hypothesis as there is no difference in the working minutes based on the Age Group.
results={}
#Normality tests on working based on living evironment
for group, subset in df.groupby('Age Group'):
    S, p = shapiro(subset['Sleeping'])
    results[group] = {'W-statistic': S, 'p-value': p}
```

```
results_df = pd.DataFrame(results).T
print(results_df)
groups = [subset['Sleeping'].values for name, subset in df.groupby('Age Group')]
stat, p = kruskal(*groups)
print(f"Kruskal-Wallis H-statistic = {stat:.4f}, p-value = {p:.4f}")

# # We reject the Null hypothesis as there is significant difference in the sleeping minutes based on the Age Group.
groups = df['Age Group'].unique()

# Create all pairwise combinations
pairs = list(itertools.combinations(groups, 2))

results = []

for g1, g2 in pairs:
    data1 = df.loc[df['Age Group'] == g1, 'Sleeping']
    data2 = df.loc[df['Age Group'] == g2, 'Sleeping']

    stat, p = scipy.stats.mannwhitneyu(data1, data2, alternative='two-sided')
    results.append({'Group 1': g1, 'Group 2': g2, 'U-statistic': stat, 'p-value': p})

# Convert to DataFrame for nice output
results_df = pd.DataFrame(results)
print(results_df)

if p < 0.05:
```

```
    for i in results_df.index:
        if results_df.loc[i, 'p-value'] < 0.05:
            print(results_df.loc[i, 'Group 1'], results_df.loc[i, 'Group 2'])
results={}

#Normality tests on working based on living evironment
for group, subset in df9.groupby('Age Group'): #df9 where we remove 0s
    S, p = shapiro(subset['Reading'])
    results[group] = {'W-statistic': S, 'p-value': p}

results_df = pd.DataFrame(results).T
print(results_df)
groups = [subset['Reading'].values for name, subset in df9.groupby('Age Group')]
stat, p = kruskal(*groups)
print(f"Kruskal-Wallis H-statistic = {stat:.4f}, p-value = {p:.4f}")

# # We reject the Null hypothesis as there is signficant difference in the sleeping minutes based on the Age Group
groups = df9['Age Group'].unique()

pairs = list(itertools.combinations(groups, 2))

results = []

for g1, g2 in pairs:
    data1 = df9.loc[df9['Age Group'] == g1, 'Reading']
    data2 = df9.loc[df9['Age Group'] == g2, 'Reading']
```

```
    stat, p = scipy.stats.mannwhitneyu(data1, data2, alternative='two-sided')
    results.append({'Group 1': g1, 'Group 2': g2, 'U-statistic': stat, 'p-value': p})

# Convert to DataFrame for nice output
results_df = pd.DataFrame(results)
print(results_df)
for i in results_df.index:
        if results_df.loc[i, 'p-value'] < 0.05:
            print(results_df.loc[i, 'Group 1'], results_df.loc[i, 'Group 2']) # groups those are significant
contingency_table = pd.crosstab(df['Age Group'], df['A4_New'])

stat, p, dof, expected = chi2_contingency(contingency_table)

print("Chi-square Test of Independence:\n")
print(f"Chi2 Statistic = {stat:.4f}")
print(f"Degrees of Freedom = {dof}")
print(f"P-value = {p:.4f}")
contingency_table = pd.crosstab(df['Age Group'], df['A5_New'])

stat, p, dof, expected = chi2_contingency(contingency_table)

print("Chi-square Test of Independence:\n")
print(f"Chi2 Statistic = {stat:.4f}")
print(f"Degrees of Freedom = {dof}")
print(f"P-value = {p:.4f}")
```