DATA VISUALIZATION PROJECT

TEAM MEMBERS: **ADITYA CHITLANGIA (16BCE1143)**

DHRUV GARG (16BCE1190)

Dataset: Young people survey, taken from Kaggle.com

Tools used: Python and R

About the dataset:

The survey is based on the young people (age group 15-30) of Slovakia. The data-file contains 1010 rows and 150 columns. The rows have text and numerical values, and also contain missing values.

OBJECTIVE: To study the the correlation between various parameters, and study in detail about patterns in music, rural vs city and male vs female characteristics.

Imports and tools

```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
import seaborn as sns
import math
from collections import Counter, OrderedDict
import xgboost as xgb
from sklearn.svm import SVC
from sklearn.linear_model import LinearRegression
from sklearn.neural_network import MLPRegressor
from sklearn.preprocessing import Imputer
import warnings
warnings.filterwarnings("ignore", category=DeprecationWarning)
```

Import the dataset

```
df = pd.read_csv("responses.csv")
```

	Music	Slow songs or fast songs	Dance	Folk	Country	Classical music	Musical	Pop	Rock	Metal or Hardrock	...	Age	Height	Weight	Number of siblings	Gender
0	5.0	3.0	2.0	1.0	2.0	2.0	1.0	5.0	5.0	1.0	...	20.0	163.0	48.0	1.0	female
1	4.0	4.0	2.0	1.0	1.0	1.0	2.0	3.0	5.0	4.0	...	19.0	163.0	58.0	2.0	female
2	5.0	5.0	2.0	2.0	3.0	4.0	5.0	3.0	5.0	3.0	...	20.0	176.0	67.0	2.0	female
3	5.0	3.0	2.0	1.0	1.0	1.0	1.0	2.0	2.0	1.0	...	22.0	172.0	59.0	1.0	female
4	5.0	3.0	4.0	3.0	2.0	4.0	3.0	5.0	3.0	1.0	...	20.0	170.0	59.0	1.0	female

MAKING THE DATA MORE USABLE

Segmenting the dataset into categories like: music, movies, phobias, interests, health, personal, demo(demographics) and spending.

```
music     = df.iloc[:,0:19]
movies    = df.iloc[:,19:31]
phobias   = df.iloc[:,63:73]
interests = df.iloc[:,31:63]
health    = df.iloc[:,73:76]
personal  = df.iloc[:, 76:133]
demo      = df.iloc[:,140:150]
spending  = df.iloc[:,133:140]
```

We noticed that certain columns hold categorical (text) data rather than numerical (integer) data. For the sake of visualization, we wanted to convert each of these categorical inputs into a corresponding numerical input.

It's also worth pointing out that many of these columns contain NaN ("not a number") values, which might give us a bit of trouble down the line. So we will be filling these NaN values by the most frequent value in the column.

Processing 1: Converting categorical data to numerical data

In the “Personal” subframe from the dataset:

```
# Convert “Internet usage” column containing categorical data
# get unique column values for Internet usage column
personal["Internet usage"].unique()
```

Output:

```
array(['few hours a day', 'most of the day', 'less than an hour a day',
       'no time at all'], dtype=object)
```

Now we must replace these distinct strings by distinct numbers.

```
# convert "Internet usage" column to integers
for i in personal["Internet usage"]:
    if i == "no time at all":
        personal.replace(i, 1.0, inplace=True)
    elif i == "less than an hour a day":
        personal.replace(i, 2.0, inplace=True)
    elif i == "few hours a day":
        personal.replace(i, 3.0, inplace=True)
    elif i == "most of the day":
        personal.replace(i, 4.0, inplace=True)
```

This replacement of categorical data to numerical data is done for all columns in the Personal subframe.

Processing 2: [Data imputation] Now we must replace the string nan with numpy NaN in the Personal sub frame, and replace the NaN values with the mode value of that column.

```
 # replace string nans with numpy compatible nan value
personal = personal.replace("nan", np.nan)
personnal = personal.replace("NaN", np.nan)
```

```
# replace nans with column mode
imp = Imputer(missing_values='NaN', strategy='most_frequent', axis=0)
imp.fit(personal)
personal_data = imp.transform(personal)
```

Note: The categorical (string) data to numerical (integer) data [Process 1] and data imputation is done for all the 8 sub-frames of the dataset. For the sake of simplicity, all the code has not been put into the word document.

FEATURE ENGINEERING

While the majority of the columns in the demographics subframe (and dataset as a whole) feature discrete inputs, the first four columns here (Age, Height, Weight, Number of Siblings) hold continuous values. We can easily calculate an individual's BMI (Body Mass Index) from their height and weight. BMI tends to be a far more indicative factor than height and weight alone, so we'll drop height and weight in favour of BMI.

```
# calculate bmi
hw = list(zip(demo["Height"], demo["Weight"]))
bmi = []
for height, weight in hw:
        if (str(height) == "nan" or str(weight) == "nan"):
            bmi.append("nan")
        else:
            result = weight/((height/100)**2)
            bmi.append(float(str(round(result, 2))))
```

```
# add "BMI" column to demo
```

```
demo["BMI"] = bmi

# convert BMI to bins
for i in demo["BMI"]:
    if str(i) != "nan":
        if (16.0 <= i < 18.5):
            demo["BMI"].replace(i, 1.0, inplace=True)
        elif (18.5 <= i < 25.0):
            demo["BMI"].replace(i, 2.0, inplace=True)
        elif (25.0 <= i < 30.0):
            demo["BMI"].replace(i, 3.0, inplace=True)
        elif (i >= 30.0):
            demo["BMI"].replace(i, 4.0, inplace=True)
```

DATA EXPLORATION AND VISUALIZATION

At the beginning of the exploration, we visualize the correlation between various variables.

PLOT 1

```
correlations = music.corr()
mask = np.zeros_like(correlations, dtype=np.bool)
mask[np.triu_indices_from(mask)] = True
f, ax = plt.subplots(figsize=(11, 8))
cmap = sns.diverging_palette(220, 10, as_cmap=True)
_ = sns.heatmap(correlations, mask=mask, cmap=cmap, vmax=.3, center=0,
                square=True, linewidths=.5, cbar_kws={"shrink": .5})
```

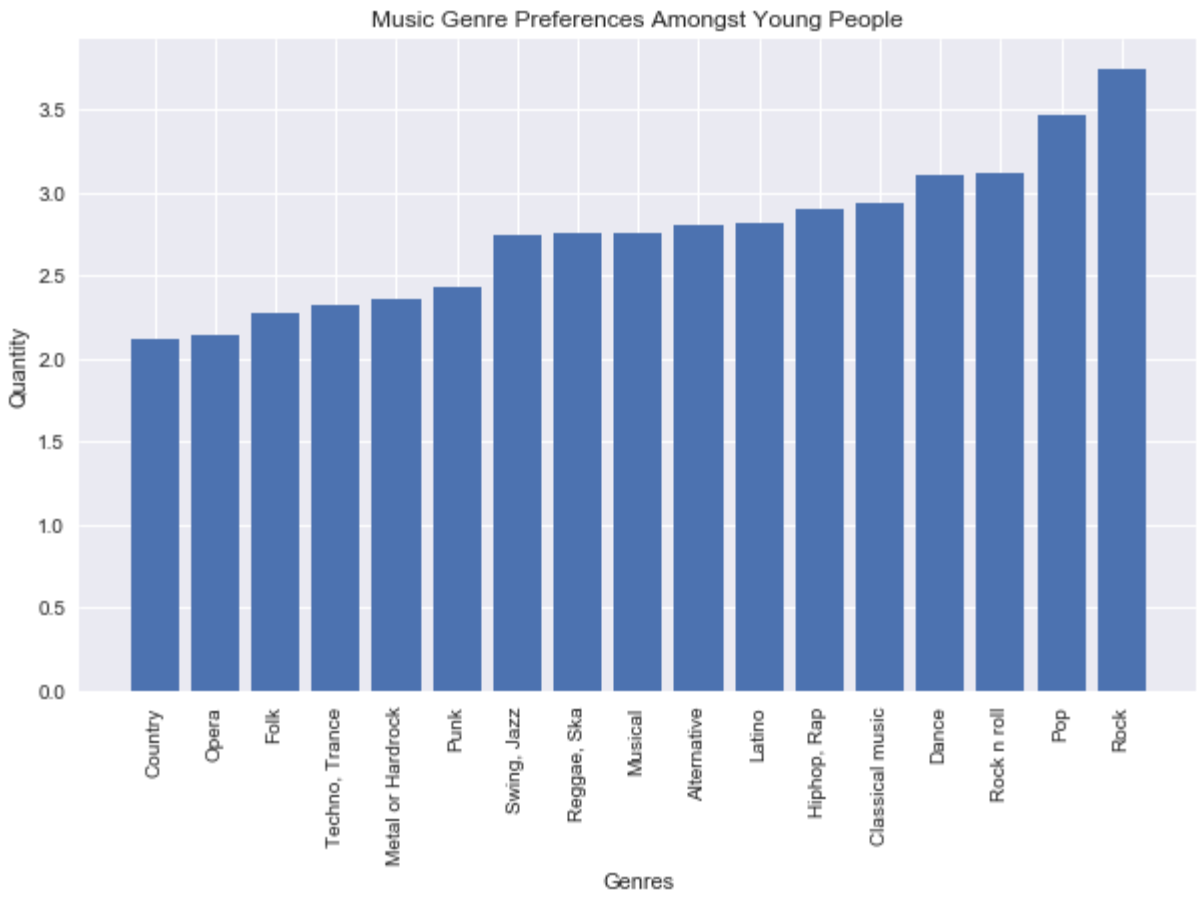
INFERENCE:

Country and classical music has direct correlation with folk music. Also, rock music has inverse correlation with pop music.

PLOT 2 (Genre distribution)

```
genres = music.columns.tolist()
genre_pop = get_pop(music, genres)

fig, ax = plt.subplots(figsize=(10,6))

# d = plt.figure(figsize=(10, 5))
plt.bar(range(len(genre_pop)), genre_pop.values(), align='center')
plt.xticks(range(len(genre_pop)), genre_pop.keys(), rotation="vertical")
```

```
ax.set_title("Music Genre Preferences Amongst Young People")
ax.set_xlabel("Genres")
ax.set_ylabel("Quantity")
plt.show()
```



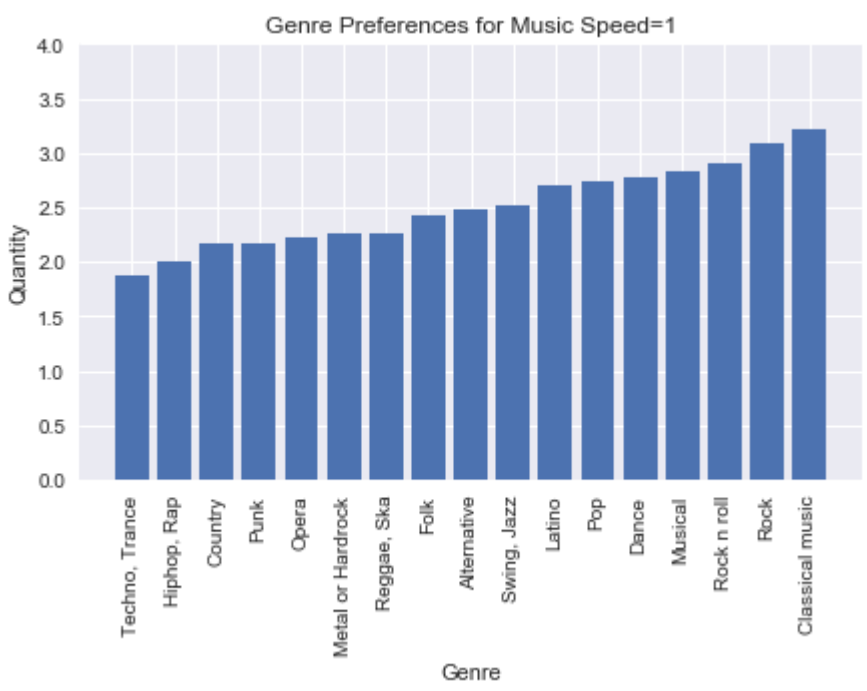
INFERENCE:

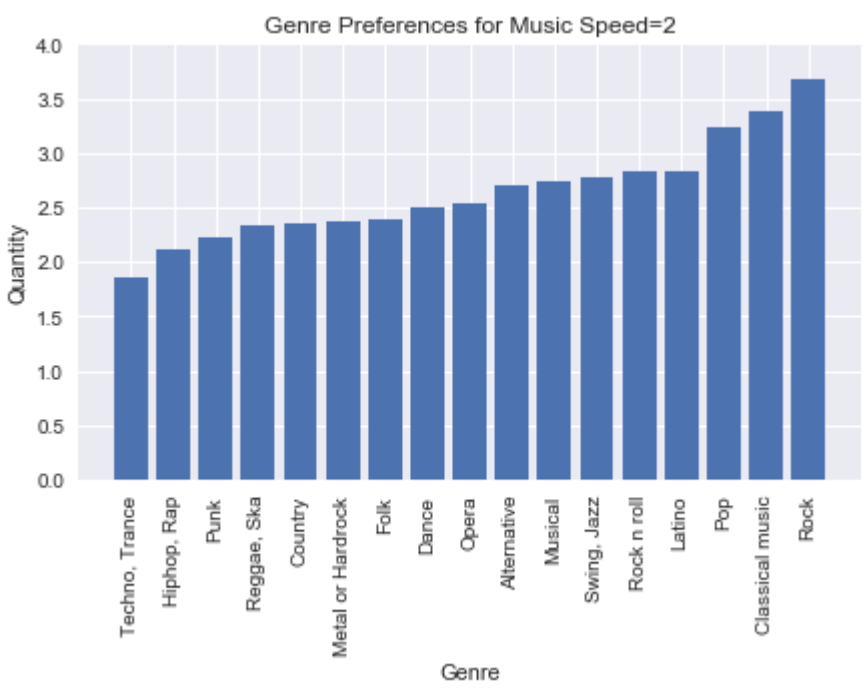
Rock and pop music are the most preferred genre among young people while country and opera music are least preferred music.

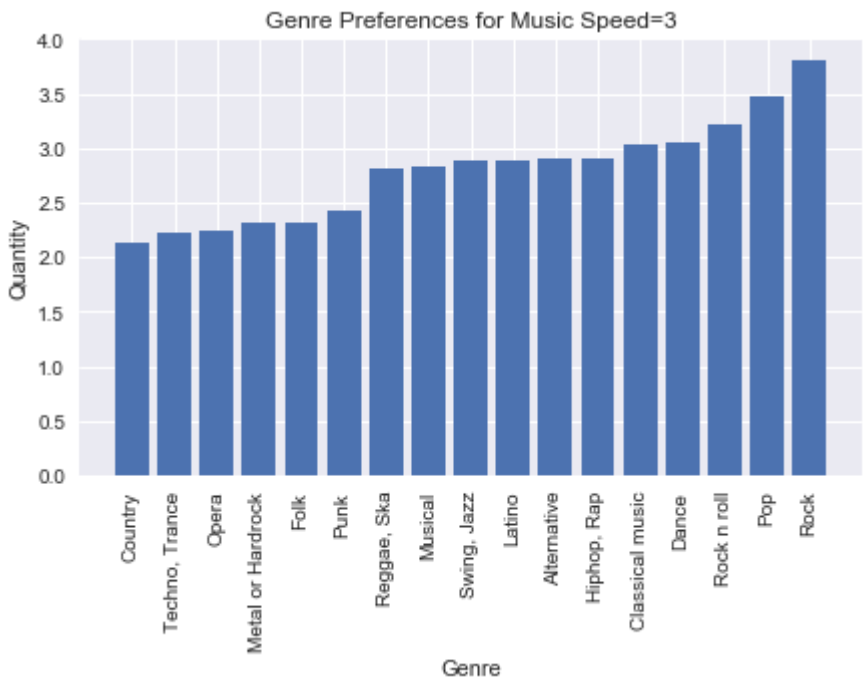
PLOT 3 (Genre distribution against self-perceived preference for music speed)

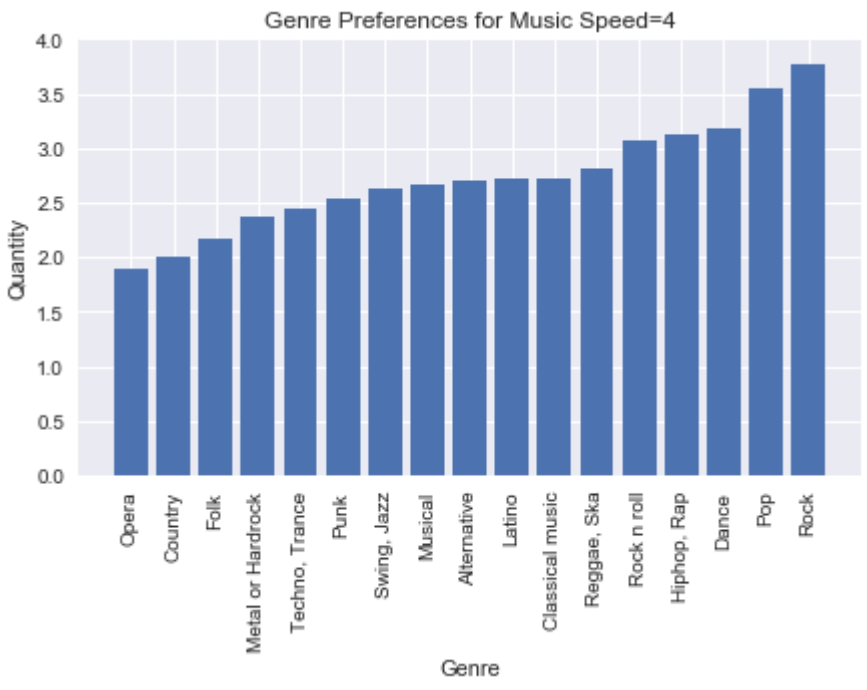
```
df1 = music[music["Slow songs or fast songs"]==1.0]
df2 = music[music["Slow songs or fast songs"]==2.0]
df3 = music[music["Slow songs or fast songs"]==3.0]
df4 = music[music["Slow songs or fast songs"]==4.0]
df5 = music[music["Slow songs or fast songs"]==5.0]

pref1 = get_pop(df1, genres=genres)
pref2 = get_pop(df2, genres=genres)
pref3 = get_pop(df3, genres=genres)
pref4 = get_pop(df4, genres=genres)
pref5 = get_pop(df5, genres=genres)

fig, ax = plt.subplots(figsize=(7,4))
plt.subplot()
axes = plt.gca()
axes.set_ylim([0,4])
plt.bar((range(len(pref1))), pref1.values(), align='center')
plt.xticks(range(len(pref1)), pref1.keys(), rotation="vertical")
plt.title("Genre Preferences for Music Speed=1")
plt.xlabel("Genre")
plt.ylabel("Quantity")
plt.show()

fig, ax = plt.subplots(figsize=(7,4))
plt.subplot()
axes = plt.gca()
axes.set_ylim([0,4])
plt.bar((range(len(pref2))), pref2.values(), align='center')
plt.xticks(range(len(pref2)), pref2.keys(), rotation="vertical")
plt.title("Genre Preferences for Music Speed=2")
plt.xlabel("Genre")
plt.ylabel("Quantity")
plt.show()

fig, ax = plt.subplots(figsize=(7,4))
plt.subplot()
```

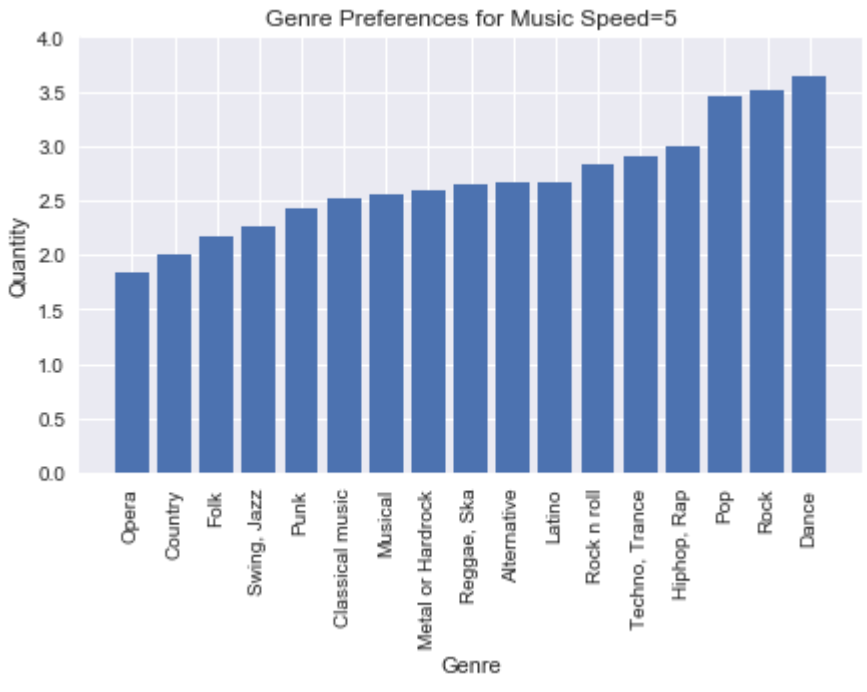
```
axes = plt.gca()
axes.set_ylim([0,4])
plt.bar((range(len(pref3))), pref3.values(), align='center')
plt.xticks(range(len(pref3)), pref3.keys(), rotation="vertical")
plt.title("Genre Preferences for Music Speed=3")
plt.xlabel("Genre")
plt.ylabel("Quantity")
plt.show()

fig, ax = plt.subplots(figsize=(7,4))
plt.subplot()
axes = plt.gca()
axes.set_ylim([0,4])
plt.bar((range(len(pref4))), pref4.values(), align='center')
plt.xticks(range(len(pref4)), pref4.keys(), rotation="vertical")
plt.title("Genre Preferences for Music Speed=4")
plt.xlabel("Genre")
plt.ylabel("Quantity")
plt.show()

fig, ax = plt.subplots(figsize=(7,4))
plt.subplot()
axes = plt.gca()
axes.set_ylim([0,4])
plt.bar((range(len(pref5))), pref5.values(), align='center')
plt.xticks(range(len(pref5)), pref5.keys(), rotation="vertical")
plt.title("Genre Preferences for Music Speed=5")
plt.xlabel("Genre")
plt.ylabel("Quantity")
plt.show()
```











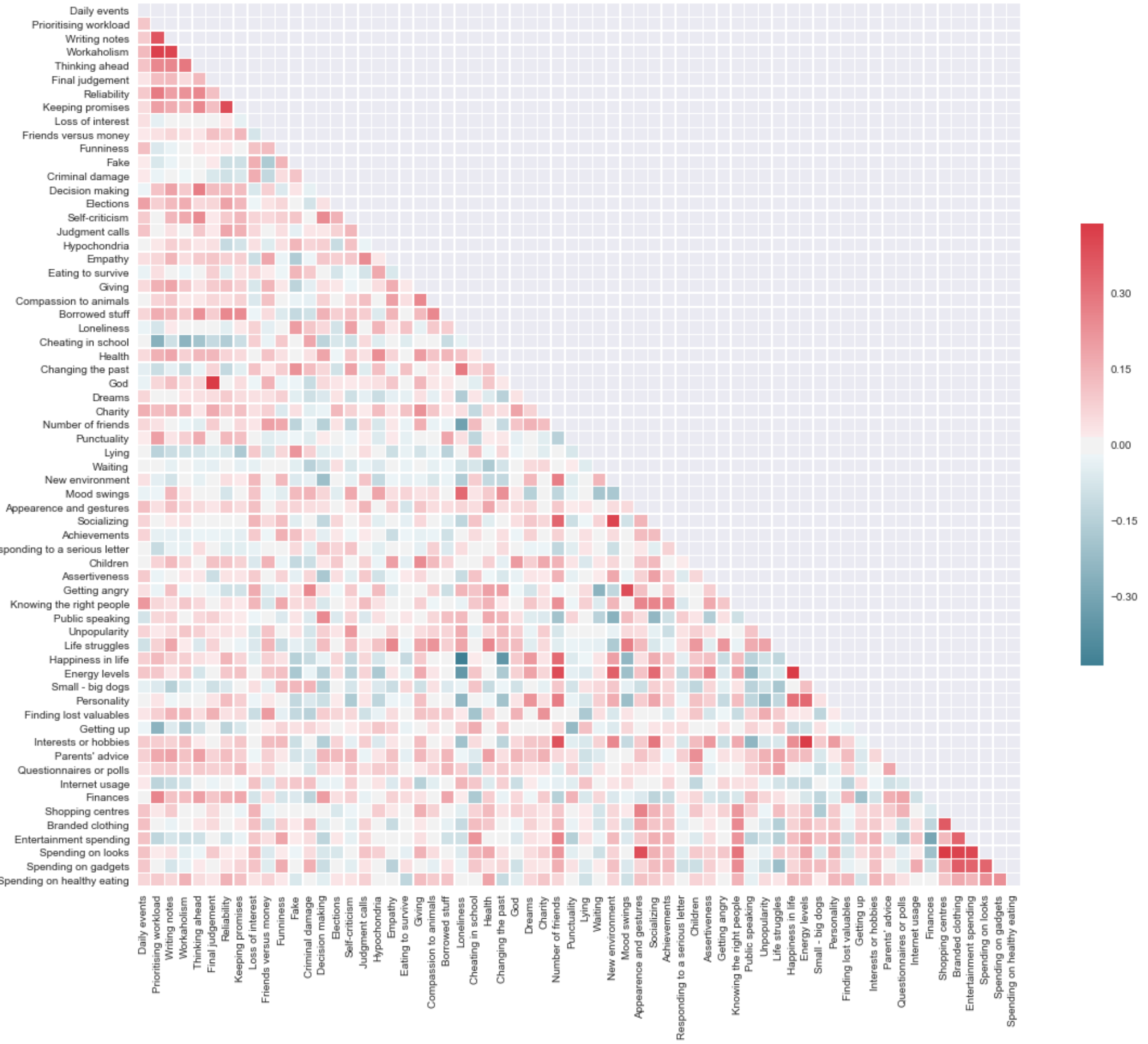
INFERENCE:

With slow music speed young people prefer classical music while for fast music they prefer rock and dance music.

PLOT 4 (Some more correlation plots) – Collinearity 1: spending vs personal values

```
correlations = df.corr()
```

```
correlations = personal.join(spending).corr()
mask = np.zeros_like(correlations, dtype=np.bool)
mask[np.triu_indices_from(mask)] = True
f, ax = plt.subplots(figsize=(20, 15))
cmap = sns.diverging_palette(220, 10, as_cmap=True)
_ = sns.heatmap(correlations, mask=mask, cmap=cmap, vmax=.3, center=0,
                square=True, linewidths=.5, cbar_kws={"shrink": .5})
```



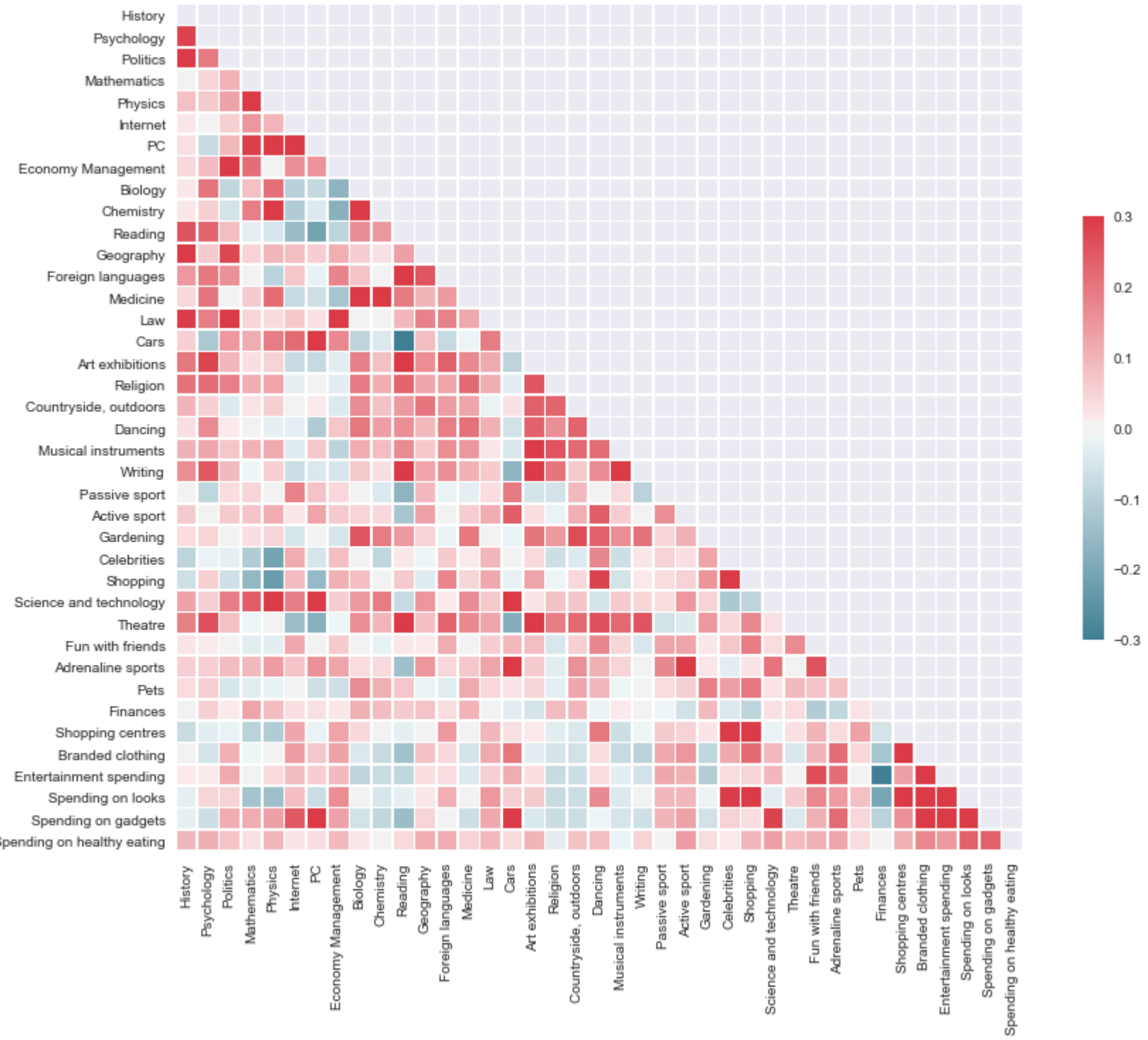
INFERENCE:

The correlation plot shows that among the young people, there is a direct correlation between keeping promises and reliability, appearance and spending on looks, happiness in life and energy levels, shopping centres and spending on looks and God and Final Judgement.

The correlation plot also shows that among young people, there is an inverse correlation between cheating in school and prioritising workload, loneliness and happiness in life/energy, finances and entertainment spending/spending on looks, and getting angry and waiting.

PLOT 5 (Some more correlation plots) – Collinearity 1: spending vs interests

```
correlations = interests.join(spending).corr()
mask = np.zeros_like(correlations, dtype=np.bool)
mask[np.triu_indices_from(mask)] = True
f, ax = plt.subplots(figsize=(15, 11))
cmap = sns.diverging_palette(220, 10, as_cmap=True)
_ = sns.heatmap(correlations, mask=mask, cmap=cmap, vmax=.3, center=0,
                square=True, linewidths=.5, cbar_kws={"shrink": .5})
```

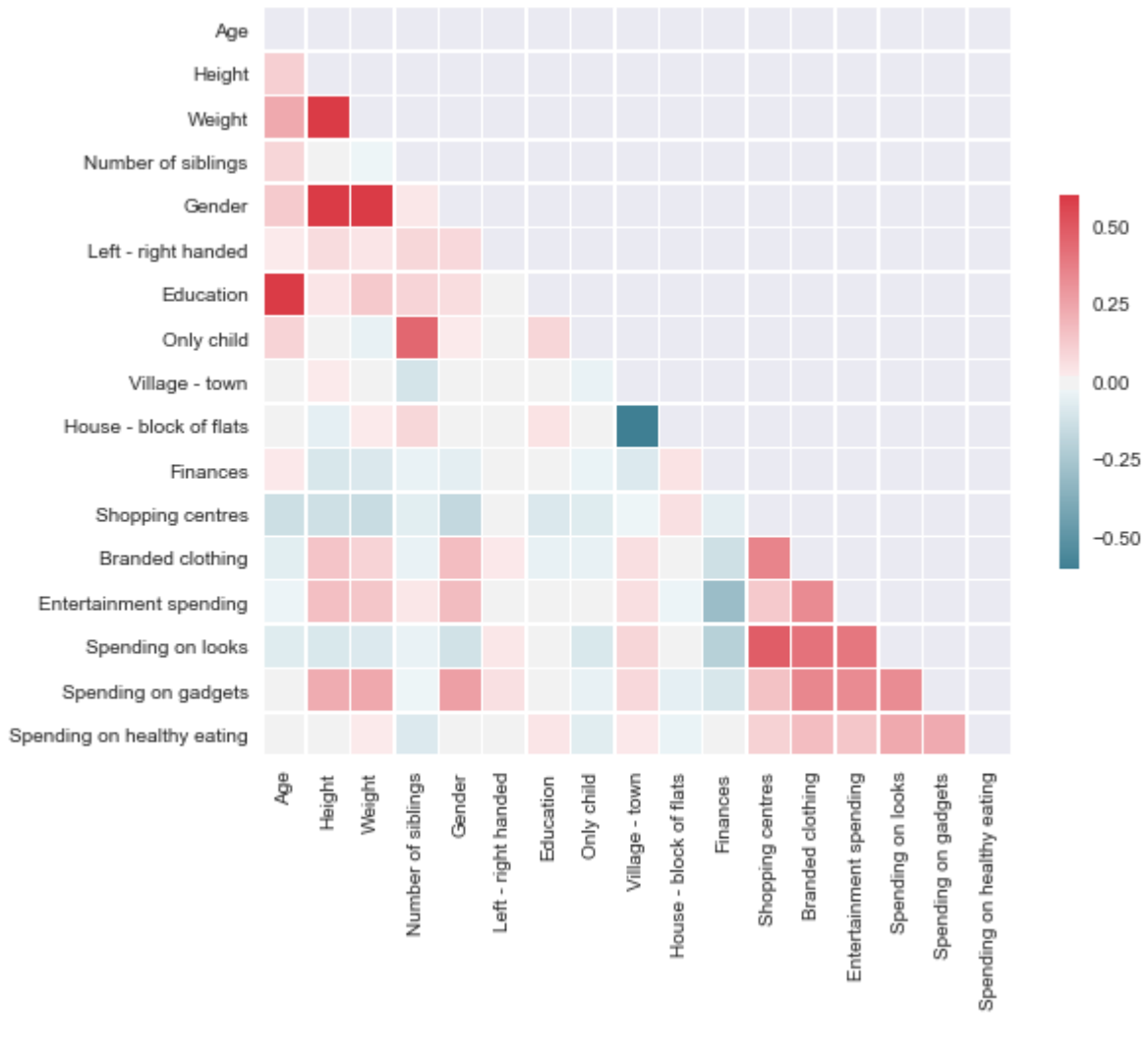


The correlation plot shows that there is a direct correlation between PCs and science and technology, cars and adrenaline sports, pshychology and writing, celebrities and spending on looks, and art exhibitions and music.

The correlation plot also shows that there is inverse correlation between entertainment spending and finances, PCs and reading, reading and active sports, celebrities and science and technology, and cars and theatre.

PLOT 6 (Some more correlation plots) – Collinearity 2: spending vs demographic information

```
correlations = demo.join(spending).corr()
mask = np.zeros_like(correlations, dtype=np.bool)
mask[np.triu_indices_from(mask)] = True
f, ax = plt.subplots(figsize=(9, 7))
cmap = sns.diverging_palette(220, 10, as_cmap=True)
_ = sns.heatmap(correlations, mask=mask, cmap=cmap, vmax=.3, center=0,
                square=True, linewidths=.5, cbar_kws={"shrink": .5})
```



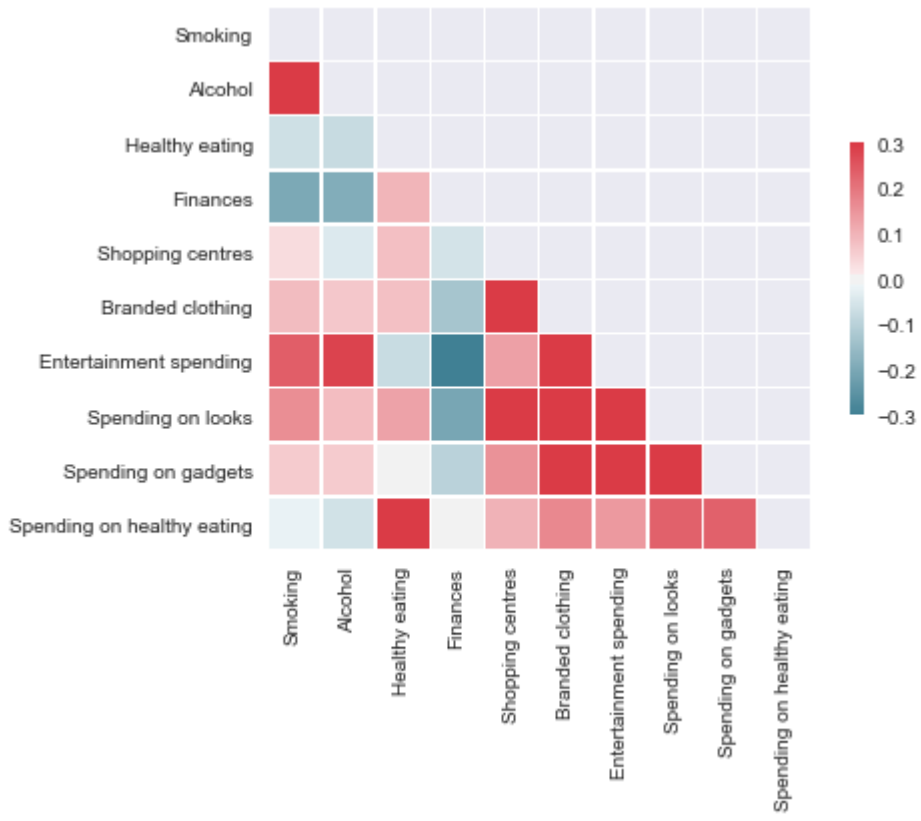
INFERENCE:

The correlation plot shows that among the young people, there is direct correlation between shopping centres and spending on looks, age and education and branded clothing and spending on gadgets.

The correlation plot also shows that there is inverse correlation between people finances and entertainment spending and between finances an spending on looks.

PLOT 7 (Some more correlation plots)

```
correlations = health.join(spending).corr()
mask = np.zeros_like(correlations, dtype=np.bool)
mask[np.triu_indices_from(mask)] = True
f, ax = plt.subplots(figsize=(7, 5))
cmap = sns.diverging_palette(220, 10, as_cmap=True)
_ = sns.heatmap(correlations, mask=mask, cmap=cmap, vmax=.3, center=0,
                square=True, linewidths=.5, cbar_kws={"shrink": .5})
```



INFERENCE:

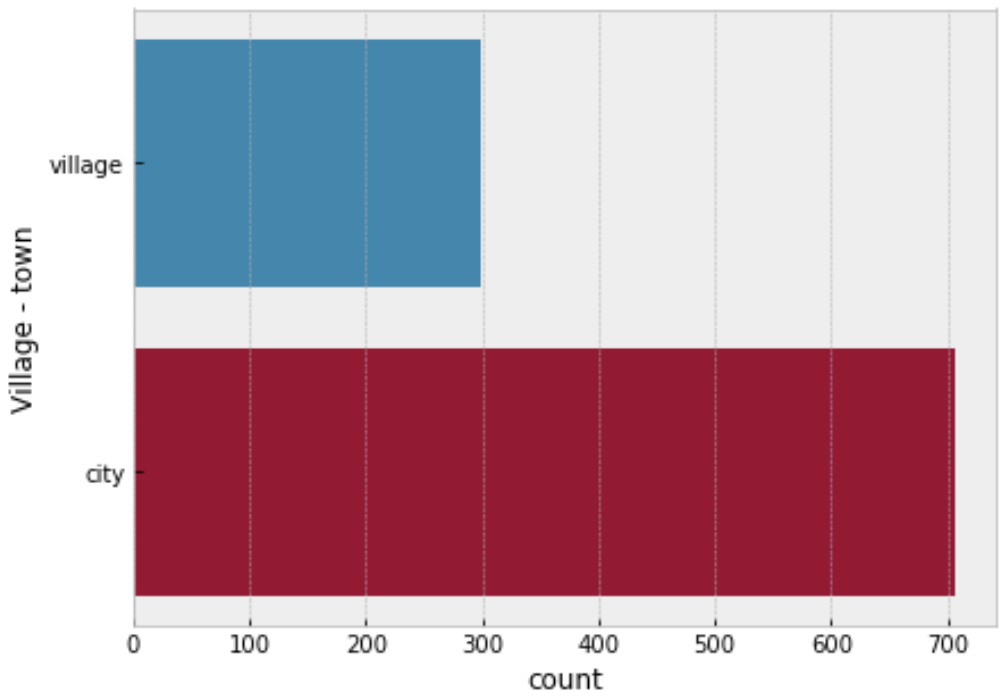
The correlation plot shows that there is a direct correlation between alcohol and smoking, entertainment spending and alcohol/smoking, and branded clothing and spending on gadgets.

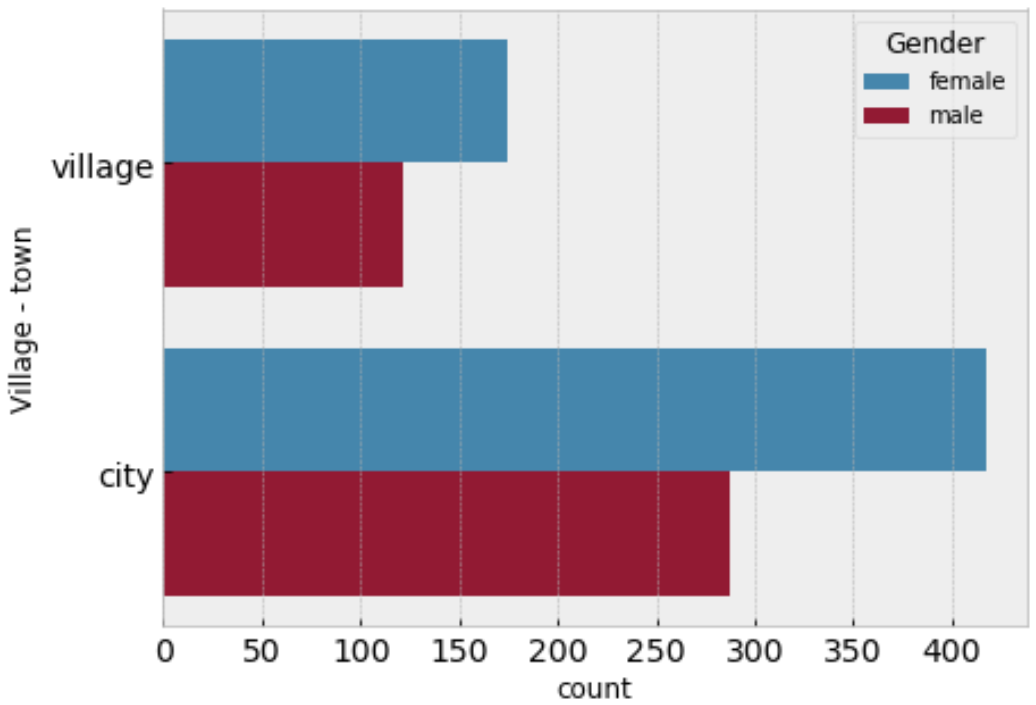
There is an inverse correlation between finances and entertainment spending and finances and smoking/alcohol.

PLOT 8 Rural vs city analysis – population

```
var_of_interest = 'Village - town'
mapping = {var_of_interest: {'city': 0, 'village': 1}}
young.dropna(subset=[var_of_interest], inplace=True)
# to be able to use hue parameter for better comparison in seaborn
young["all"] = ""
fig, ax = plt.subplots(nrows=1, ncols=2,figsize=(15,5))
sns.countplot(y = var_of_interest, data = young, ax = ax[0])
sns.countplot(y = var_of_interest, hue = 'Gender', data = young, ax = ax[1])
_ = plt.xticks(fontsize=14)
```

```
_ = plt.yticks(fontsize=14)
```



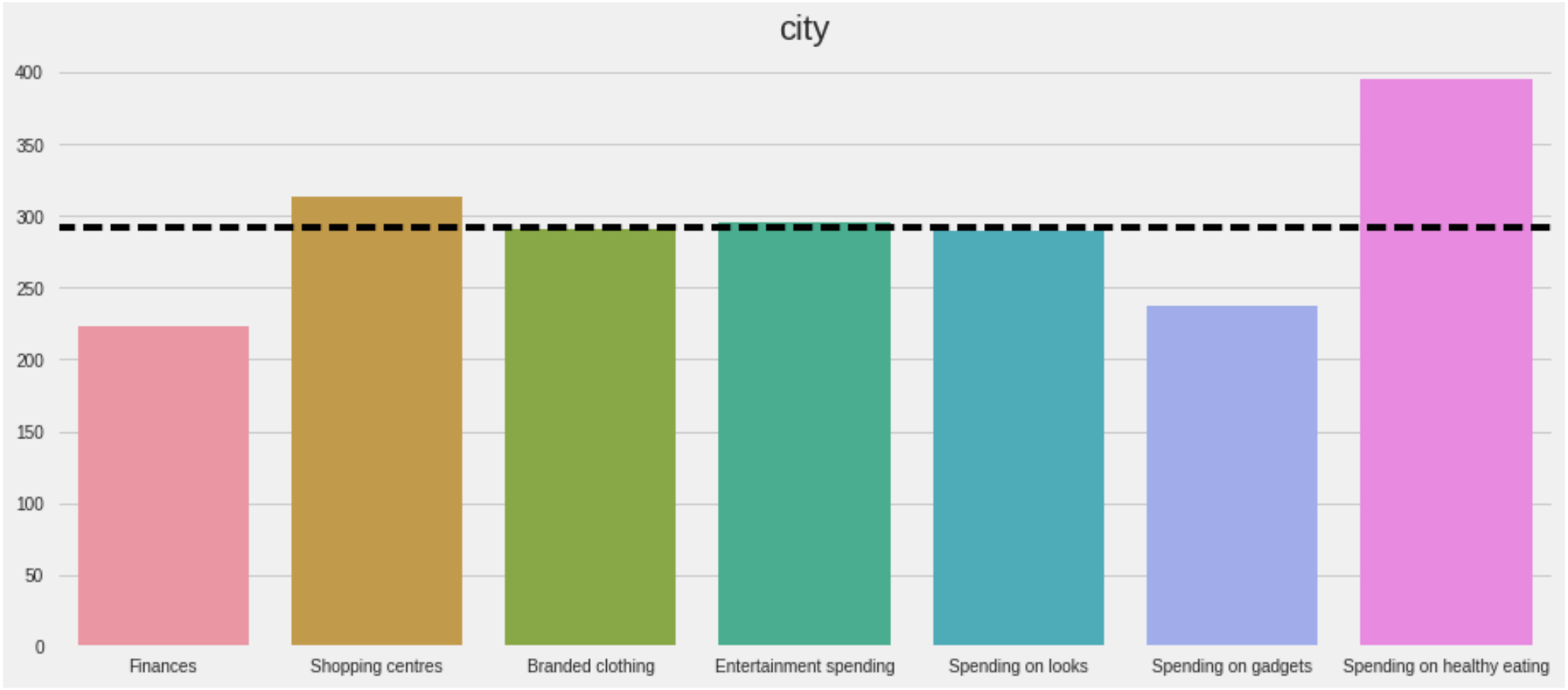


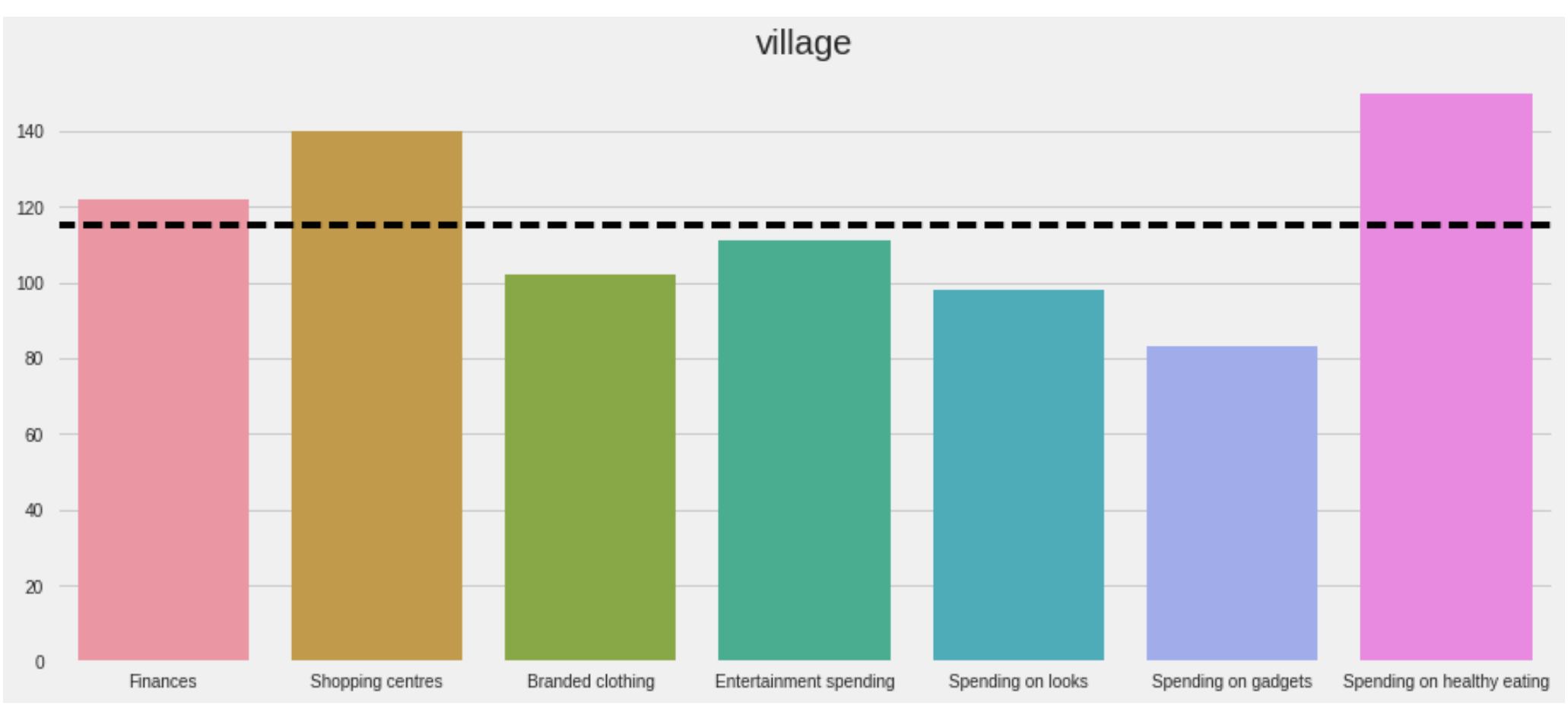
INFERENCE:

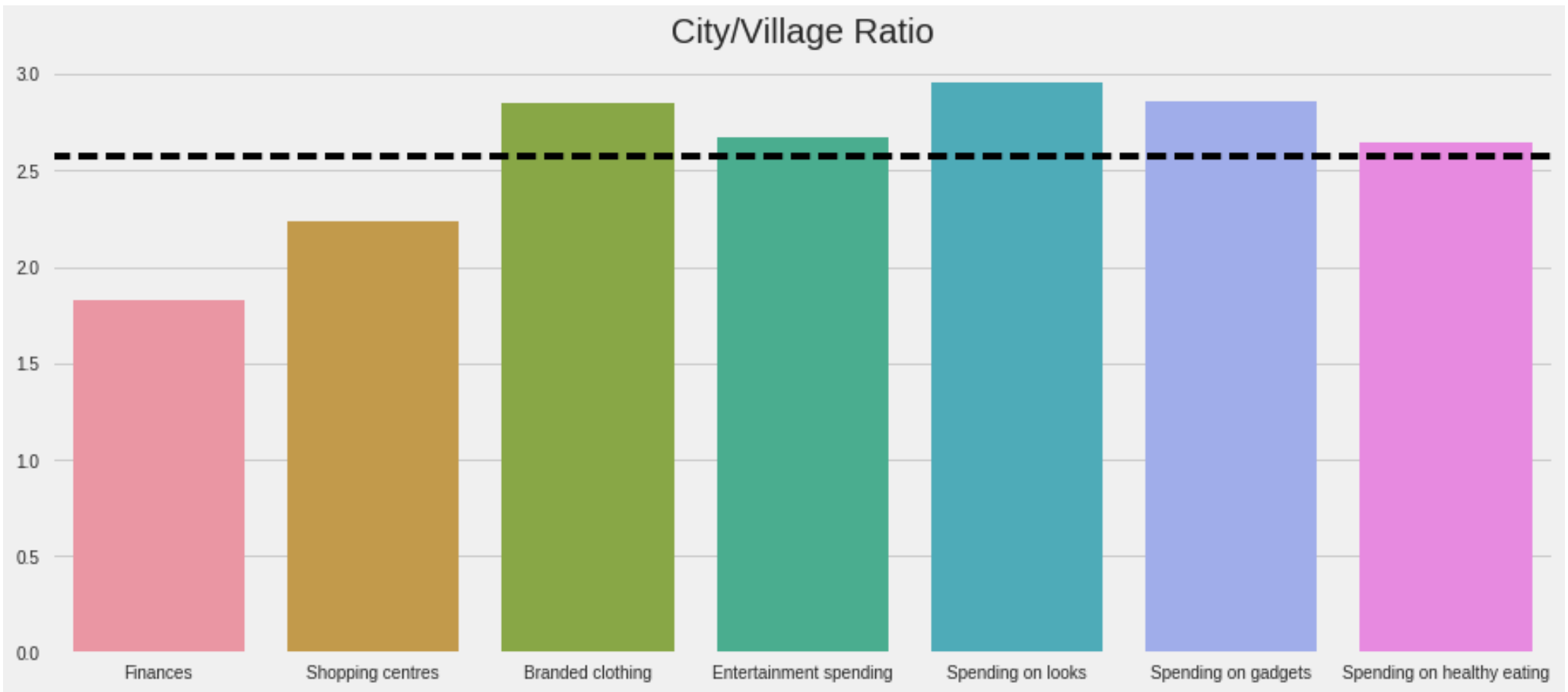
The bar plot clearly shows that the dataset is skewed – people living in the city are more than double than those living in the villages. Also, in both cities and villages surveyed, females are greater in number than males.

PLOT 9 Rural vs city analysis – shopping

```
temp = []
for i in spending.columns:
    temp.append(like[i].groupby('Village - town').count()['Age'][0])
plt.figure(figsize=(14,6))
plt.title('{}'.format(like[i].groupby('Village - town').count()['Age'].index[0]))
plt.axhline(y=np.mean(temp), color='k', lw=4, ls='dashed')
sns.barplot(spending.columns,temp)

temp = []
for i in spending.columns:
    temp.append(like[i].groupby('Village - town').count()['Age'][1])
plt.figure(figsize=(14,6))
plt.title('{}'.format(like[i].groupby('Village - town').count()['Age'].index[1]))
plt.axhline(y=np.mean(temp), color='k', lw=4, ls='dashed')
sns.barplot(spending.columns,temp)

temp = []
for i in spending.columns:
    temp.append(like[i].groupby('Village - town').count()['Age'][0]/
                like[i].groupby('Village - town').count()['Age'][1])
plt.figure(figsize=(14,6))
plt.title('City/Village Ratio')
plt.axhline(y=np.mean(temp), color='k', lw=4, ls='dashed')
sns.barplot(spending.columns,temp)
```







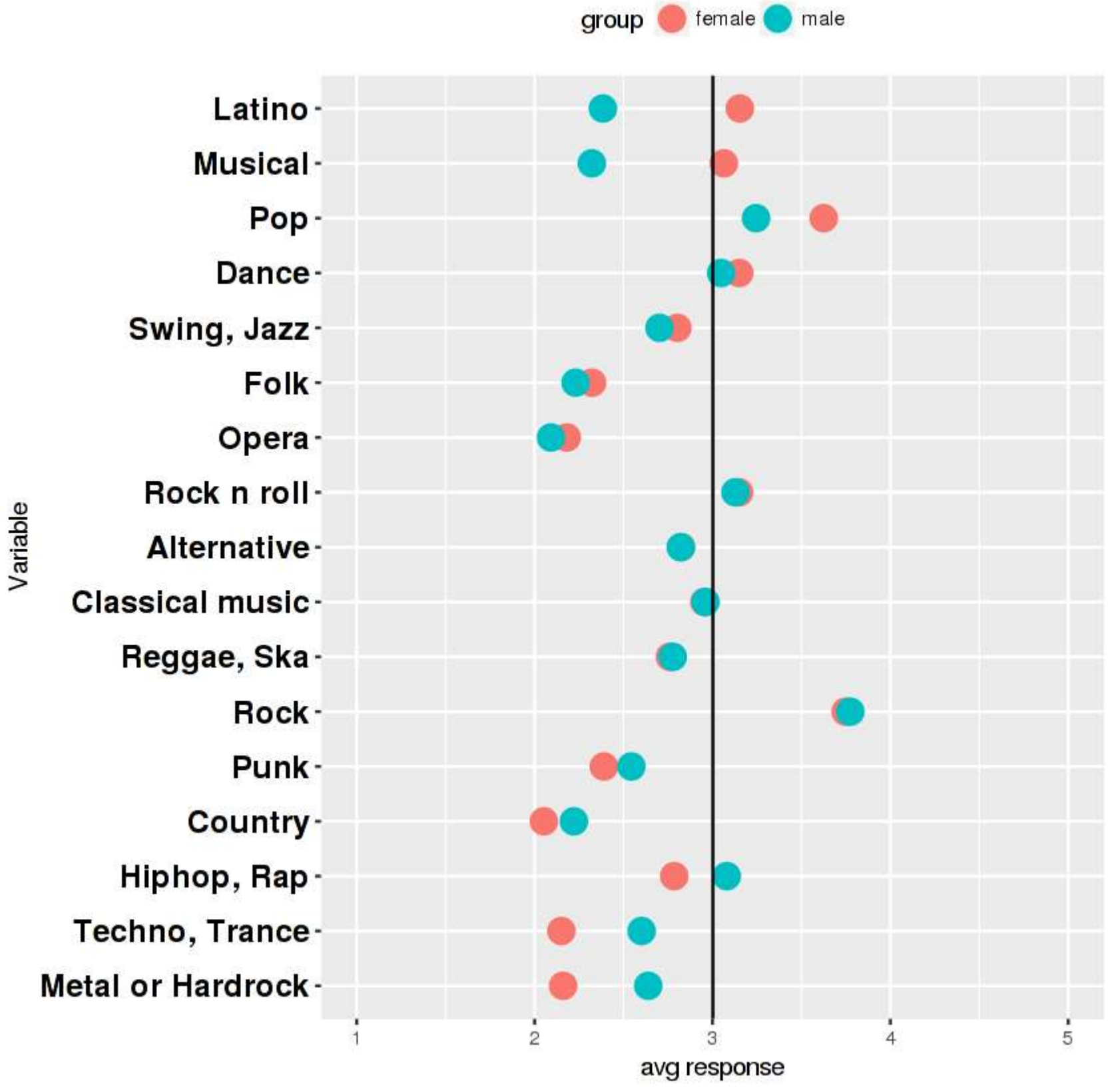
INFERENCE:

The bar plots show the spending patterns in the cities and villages. In the villages, people spend more on maintaining finances and inside shopping centres, while they spend lesser on gadgets, looks and branded clothing when compared to the people from the city.

PLOT 10 Gender analysis – Music preferences

```
library(magrittr)
suppressMessages(library(dplyr))
library(readr)
library(ggplot2)
suppressMessages(library(tidyr))

# Helper functions
analyze_group_differences <- function(df, group, start, end) {
  avgs_by_group <-  df %>%
    dplyr::rename_(group = group) %>%
    select_("group", paste0("`", start, "`:`", end,"`")) %>%
    dplyr::group_by(group) %>%
    dplyr::summarise_all(mean, na.rm = TRUE) %>%
    na.omit

  vars_by_difference <- avgs_by_group %>%
    dplyr::select(-group) %>%
    apply(2, function(x) x[1] - x[2]) %>%
    sort %>%
    names

  avgs_by_group %>%
    tidyr::gather(Variable, `avg response`, -group) %>%
    ggplot(aes(x = Variable, y = `avg response`, group = group, colour = group)) +
    geom_point(size = 5) +
    scale_x_discrete(limits = vars_by_difference) +
    ylim(1, 5) +
    geom_hline(yintercept = 3) +
    coord_flip() +
    theme(axis.text.y = element_text(face="bold", color="black", size=14),
          legend.position="top")

}

# Reading data
df <- readr::read_delim("../input/responses.csv", delim = ",")
dim(df)
```

```
# Gender differences in music preferences
df %>% analyze_group_differences("Gender", "Dance", "Opera")
```

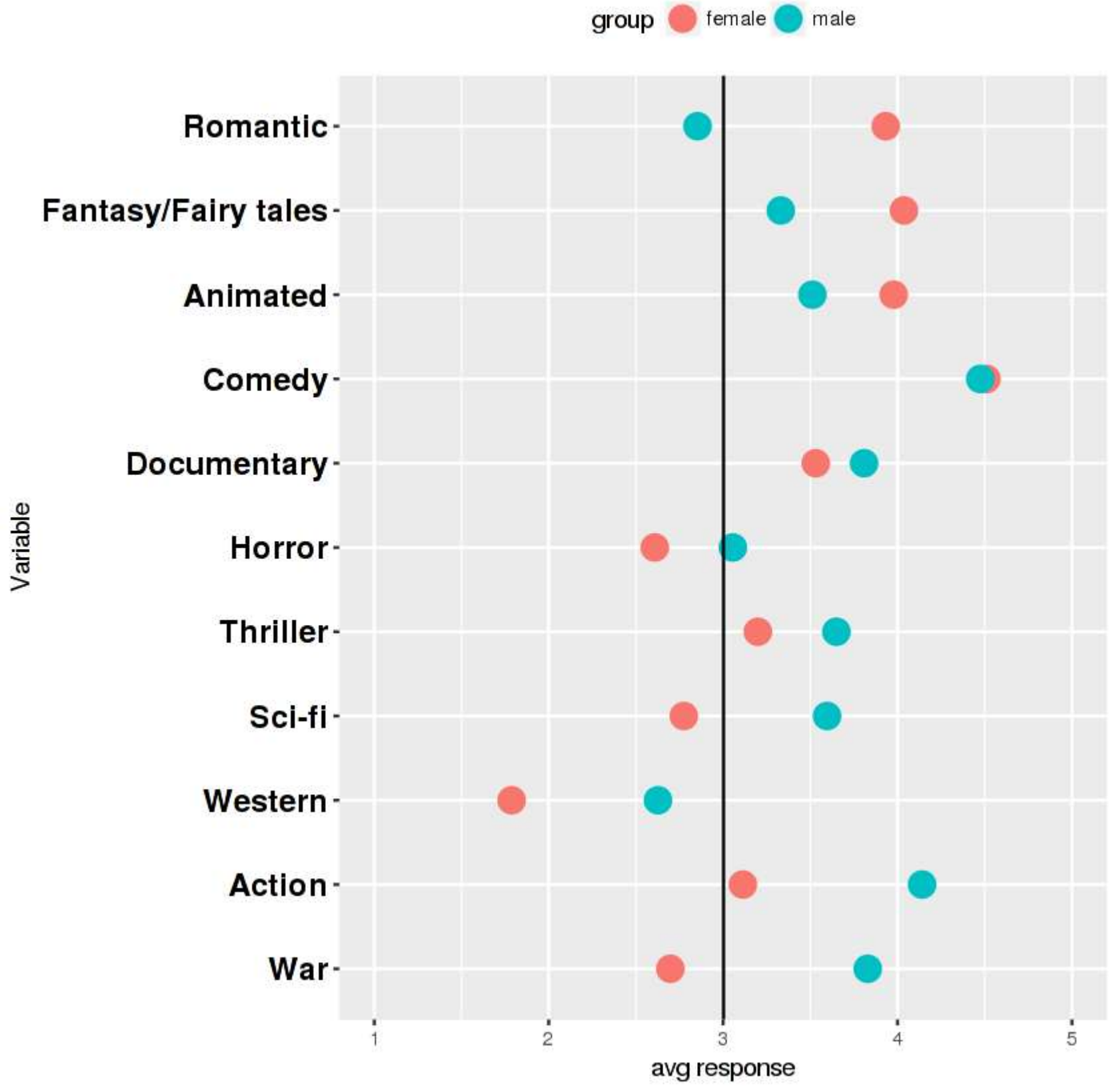


INFERENCE:

From the gender analysis plot on music, females prefer latino, instrumental music and pop music while males prefer metal/hard rock, rap and techno/trance. Interestingly, both males and females equally like rock music.

PLOT 11 Gender analysis – Music preferences

```
# Gender differences in movie preferences
df %>% analyze_group_differences("Gender", "Horror", "Action")
```

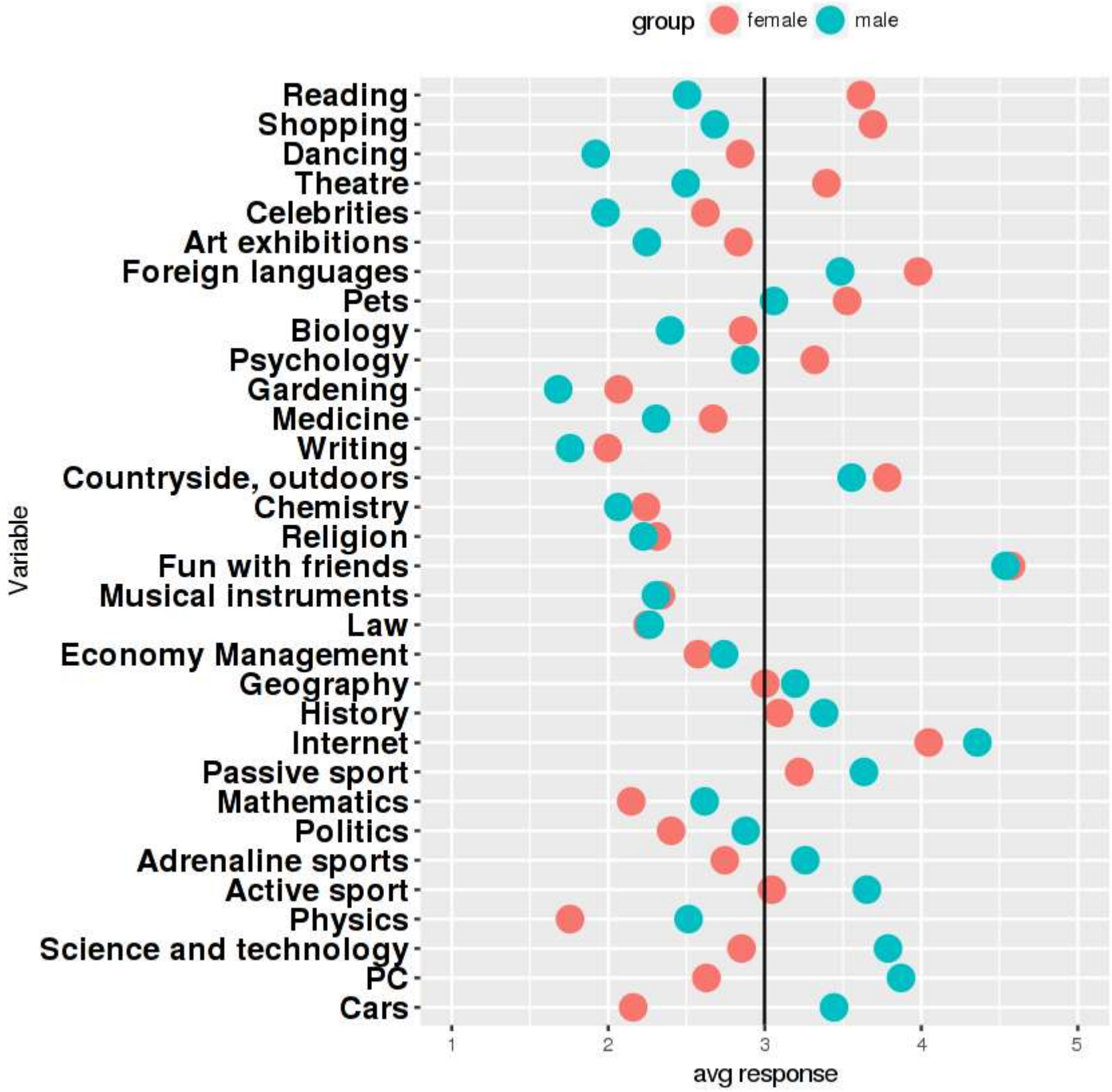


INFERENCE:

From the gender analysis plot on movies, we can see that females prefer romantic, fantasy/fairy tales as movie themes while men prefer thriller, sci-fi, action and ware as movie themes. Both males and females equally like comedy moves.

PLOT 12 Gender analysis – interests

```
# Gender differences in interests
df %>% analyze_group_differences("Gender", "History", "Pets")
```

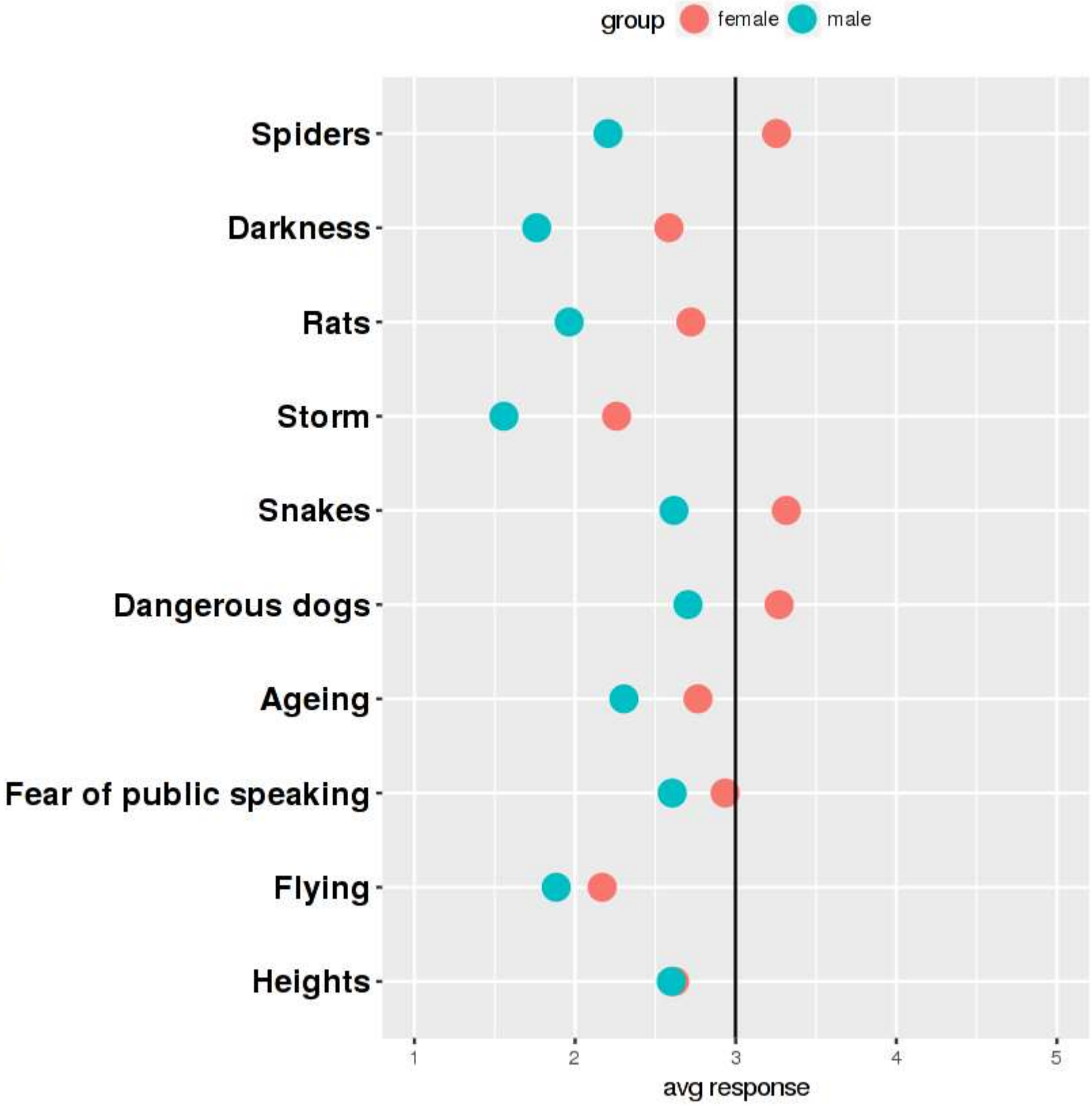


INFERENCE:

From the gender analysis plots on interest, it can be seen that women are interested in reading, shopping, theatre, dancing, pshychology and foreign languages while men are interested in PCs, science and technology, physics, adrenaline sports and politics. Both men and women are equally interested in having fun with friends.

PLOT 13 Gender analysis – phobias

```
# Gender differences in phobias
df %>% analyze_group_differences("Gender", "Flying", "Fear of public speaking")
```



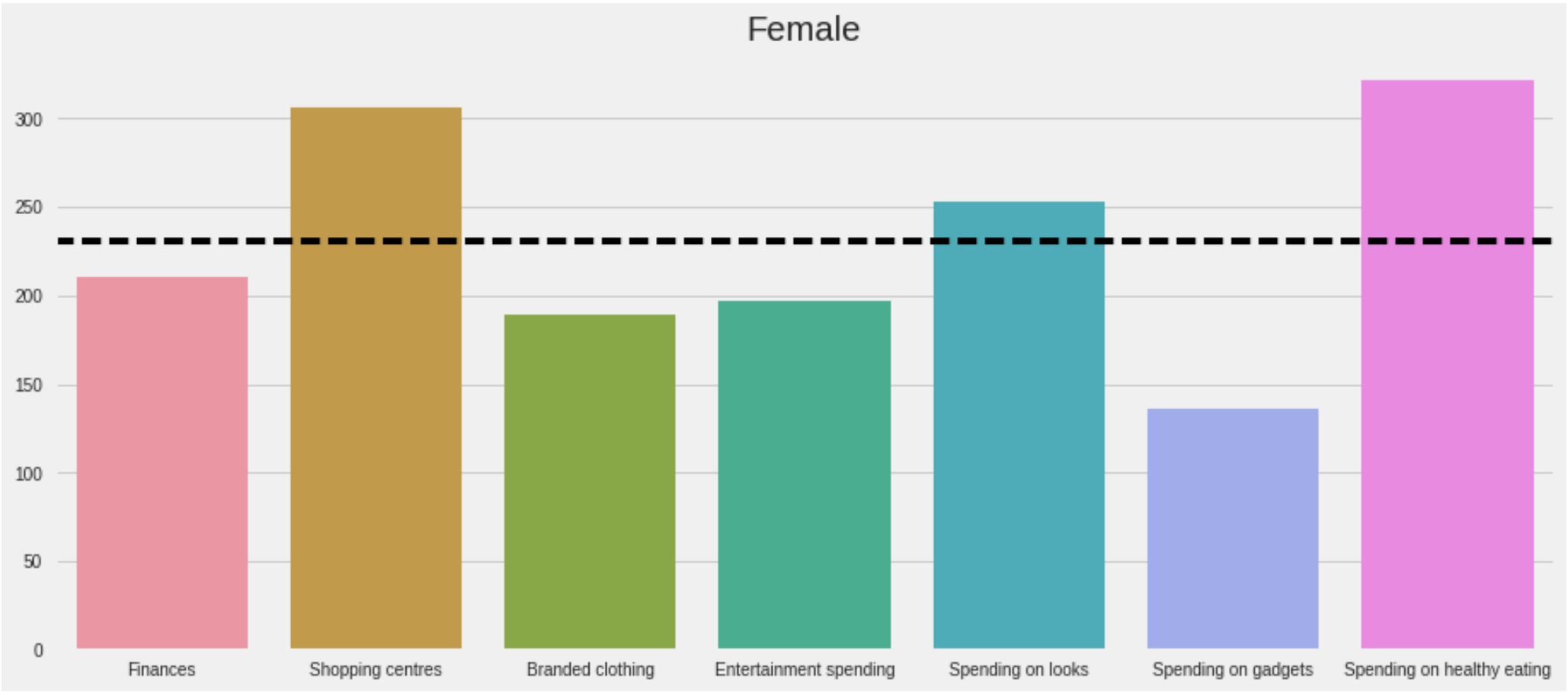
INFERENCE:

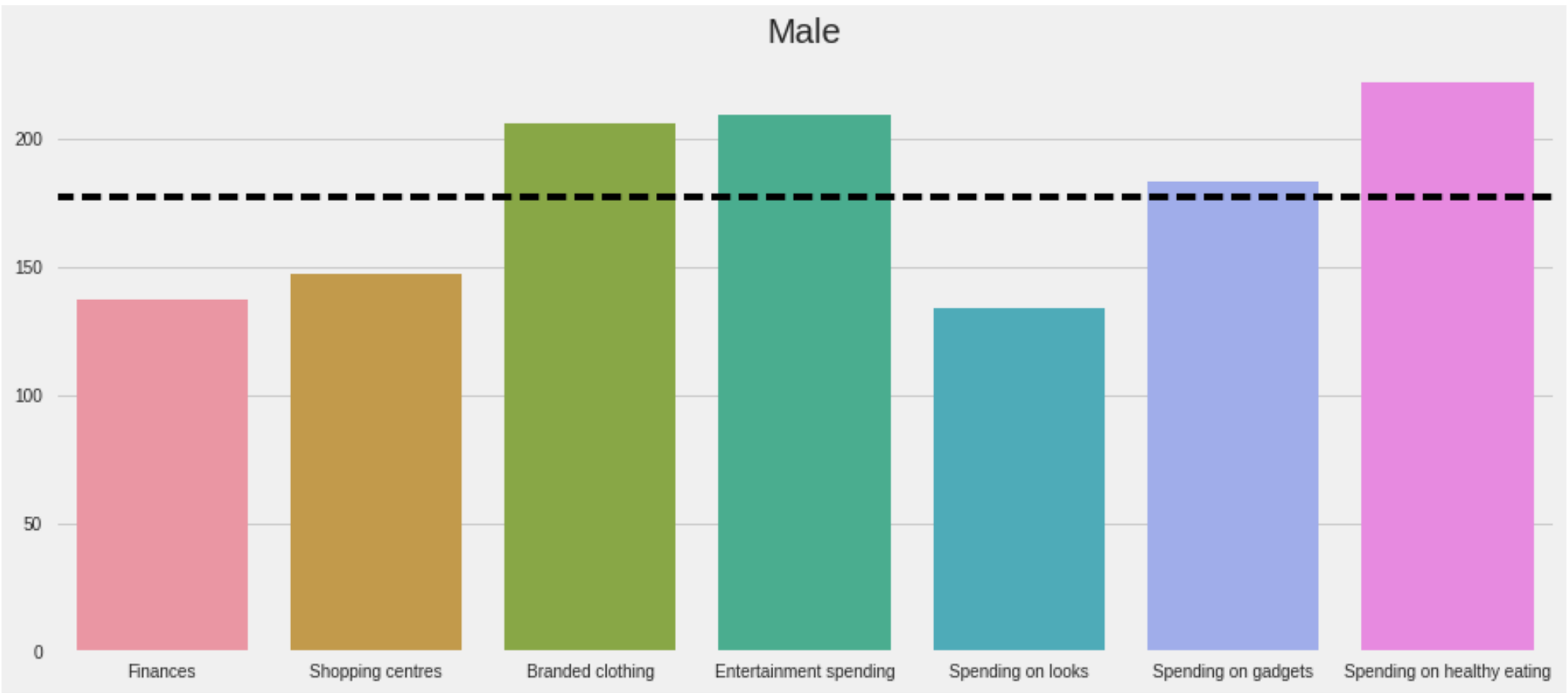
From the plot on gender analysis on phobias, we can see that there is no such thing that men fear more than women. Women fear spiders and snakes the most. Also, both men and women are equally scared of heights.

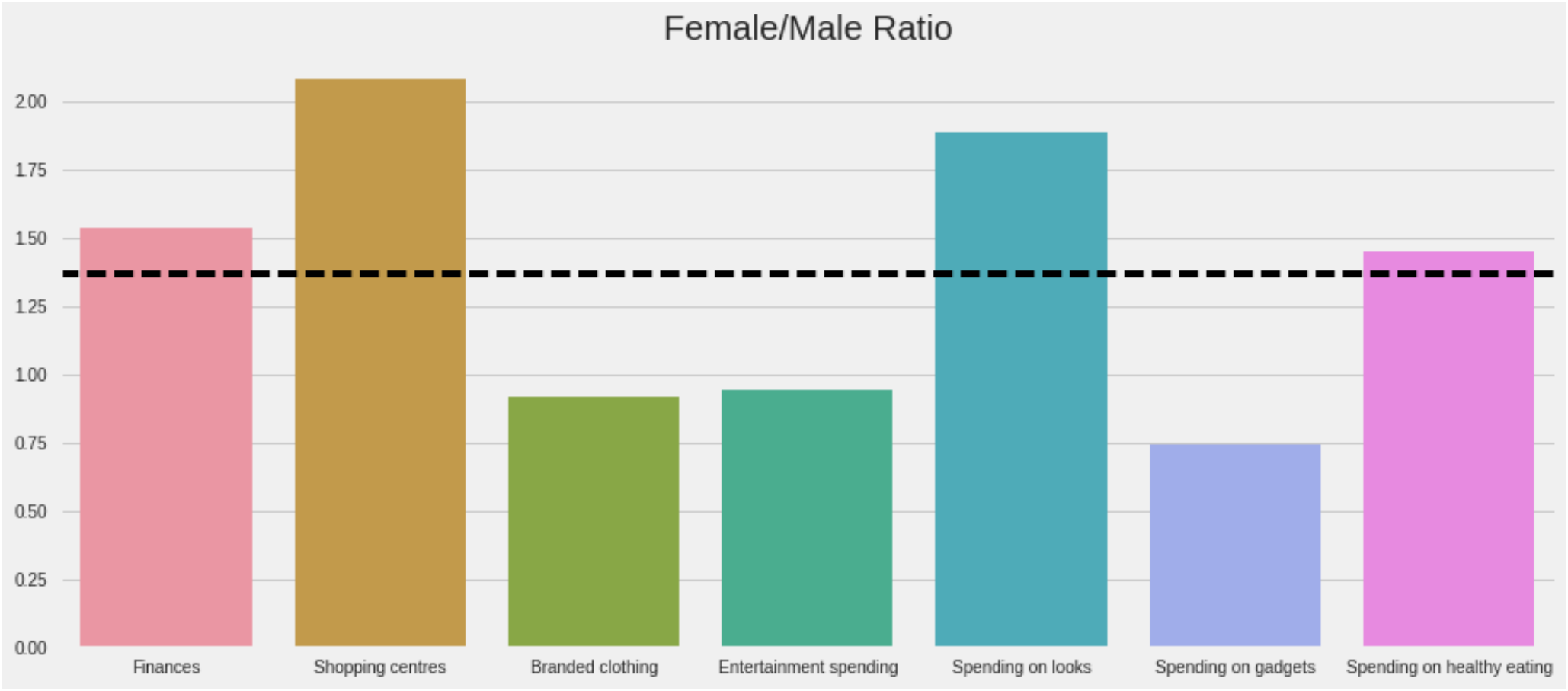
PLOT 14 Gender analysis – shopping

```
like = {}
for i in spending.columns:
    df_temp = demographics[spending[i]>=4]
    like[i] = df_temp

temp = []
for i in spending.columns:
    temp.append(like[i].groupby('Gender').count()['Age'][0])
plt.figure(figsize=(14,6))
plt.title('Female')
plt.axhline(y=np.mean(temp), color='k', lw=4, ls='dashed')
sns.barplot(spending.columns,temp)
    #print('{}'.format(i),like[i].groupby('Gender').count()['Age'])
```

```
    #print('{}'.format(i),notlike[i].groupby('Gender').count()['Age'])
#demographics
temp = []
for i in spending.columns:
    temp.append(like[i].groupby('Gender').count()['Age'][1])
plt.figure(figsize=(14,6))
plt.title('Male')
plt.axhline(y=np.mean(temp), color='k', lw=4, ls='dashed')
sns.barplot(spending.columns,temp)

temp = []
for i in spending.columns:
    temp.append(like[i].groupby('Gender').count()['Age'][0]/like[i].groupby('Gender').count()['Age'][1])
plt.figure(figsize=(14,6))
plt.title('Female/Male Ratio')
plt.axhline(y=np.mean(temp), color='k', lw=4, ls='dashed')
sns.barplot(spending.columns,temp)
```







INFERENCE:

It can be observed from the gender analysis plot on spending that women spend more on shopping centres and looks when compared to men. Women also are noted to spend lesser on finances and gadgets when compared to men. Interestingly, men are seen to spend more on branded clothing than women. Men spend more on entertainment when compared to women.