Narine Marutyan HW 1

```
library(reticulate)
library(ggplot2)
library(ggthemes)
library(lattice)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

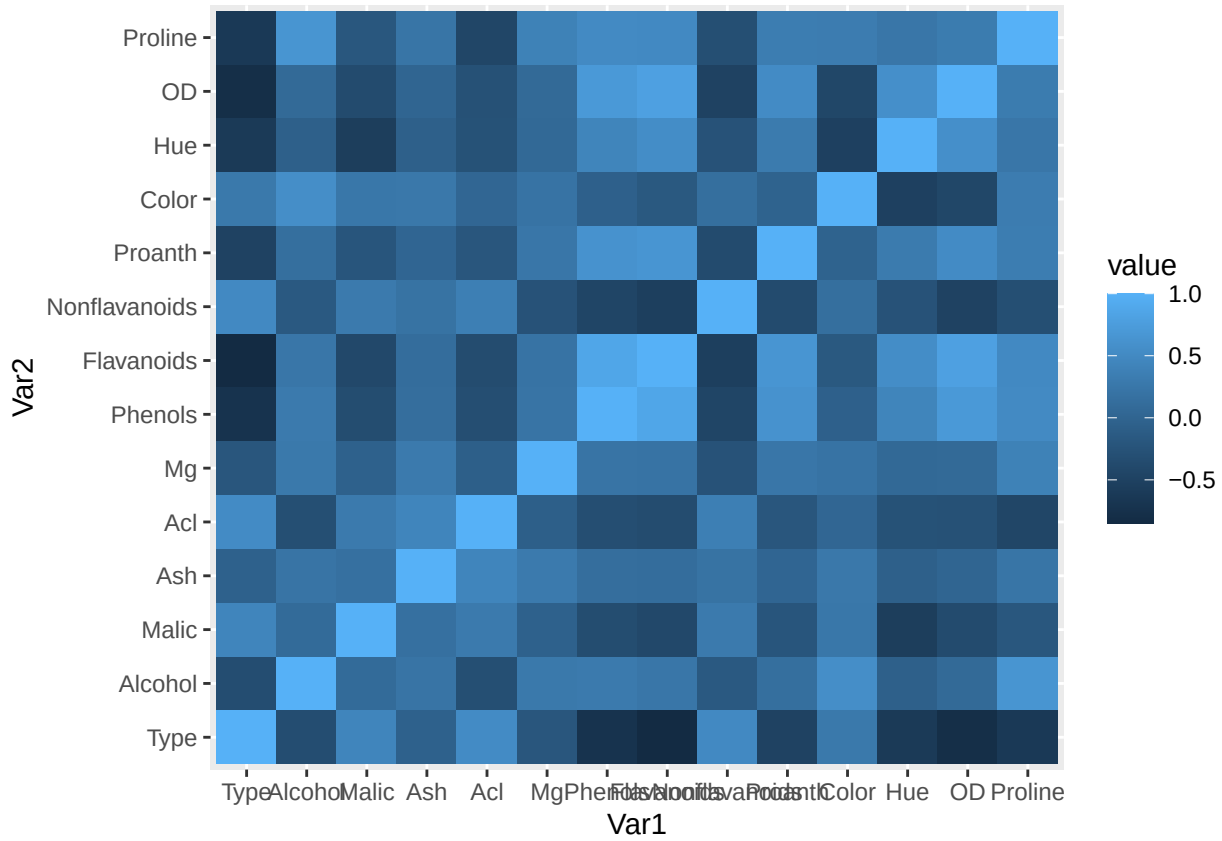
```
library(reshape2)

# ---------------------------------------------------------------------
# Task 2
# ---------------------------------------------------------------------
# Read the data and check the structure. Convert Type to factor.
# Check if there are any missing values.

wine <- read.csv('wine.csv')
str(wine)
```

```
## 'data.frame':    178 obs. of  14 variables:
##  $ Type         : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Alcohol      : num  14.2 13.2 13.2 14.4 13.2 ...
##  $ Malic        : num  1.71 1.78 2.36 1.95 2.59 1.76 1.87 2.15 1.64 1.35 ...
##  $ Ash          : num  2.43 2.14 2.67 2.5 2.87 2.45 2.45 2.61 2.17 2.27 ...
##  $ Acl          : num  15.6 11.2 18.6 16.8 21 15.2 14.6 17.6 14 16 ...
##  $ Mg           : int  127 100 101 113 118 112 96 121 97 98 ...
##  $ Phenols      : num  2.8 2.65 2.8 3.85 2.8 3.27 2.5 2.6 2.8 2.98 ...
##  $ Flavanoids   : num  3.06 2.76 3.24 3.49 2.69 3.39 2.52 2.51 2.98 3.15 ...
##  $ Nonflavanoids: num  0.28 0.26 0.3 0.24 0.39 0.34 0.3 0.31 0.29 0.22 ...
##  $ Proanth      : num  2.29 1.28 2.81 2.18 1.82 1.97 1.98 1.25 1.98 1.85 ...
##  $ Color        : num  5.64 4.38 5.68 7.8 4.32 6.75 5.25 5.05 5.2 7.22 ...
##  $ Hue          : num  1.04 1.05 1.03 0.86 1.04 1.05 1.02 1.06 1.08 1.01 ...
##  $ OD           : num  3.92 3.4 3.17 3.45 2.93 2.85 3.58 3.58 2.85 3.55 ...
##  $ Proline      : int  1065 1050 1185 1480 735 1450 1290 1295 1045 1045 ...
```

```
sum(is.na(wine))
```

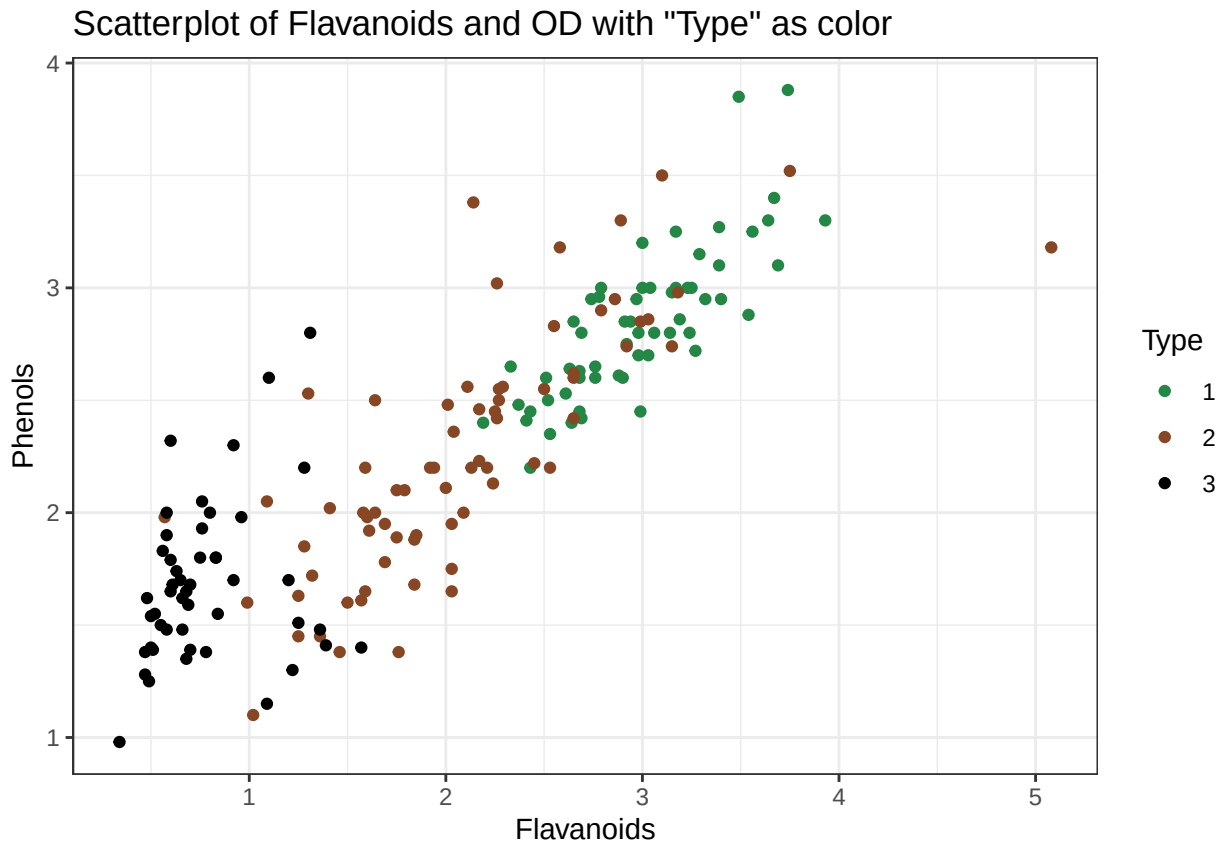
```
## [1] 0
```

```
# ------------------------------------------------------------------
# Task 2.1
# ------------------------------------------------------------------
# Select all numeric variables from your data. Calculate correlation
# between variables and create correlation heatmap.

only_numeric <- select_if(wine, is.numeric)
c <- round(cor(only_numeric),2)
melted_c <- reshape2::melt(c)
ggplot(data = melted_c,aes(x=Var1,y=Var2,fill=value)) + geom_tile()
```

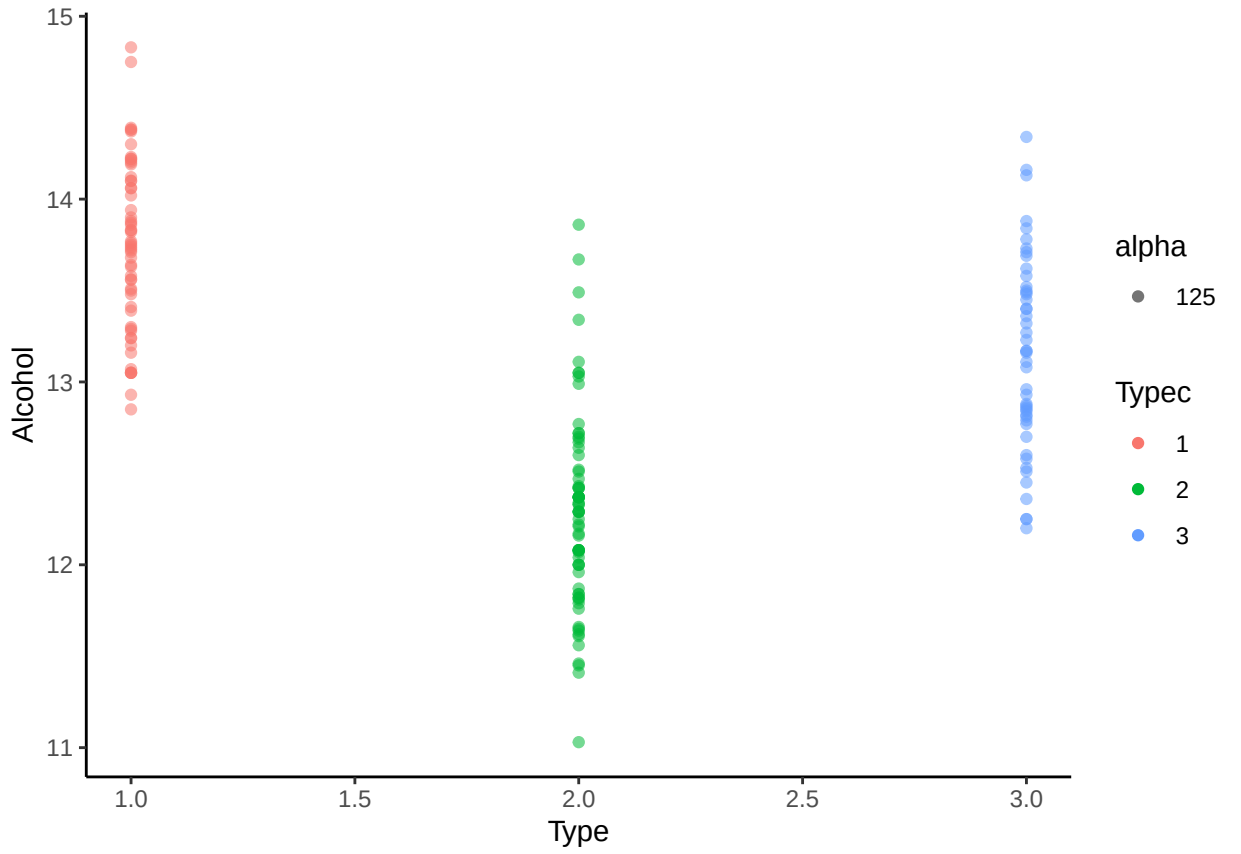



```
# From the picture it is apparent that the lighter the shade
# the higher the correlation coefficent (but let`s not take
# 1 since only same things are correlated that much) . It is
# quite apparent that flavanoids and Phenols are very much
# positively correlated

# ------------------------------------------------------------------
# Task 2.2
# ------------------------------------------------------------------
```

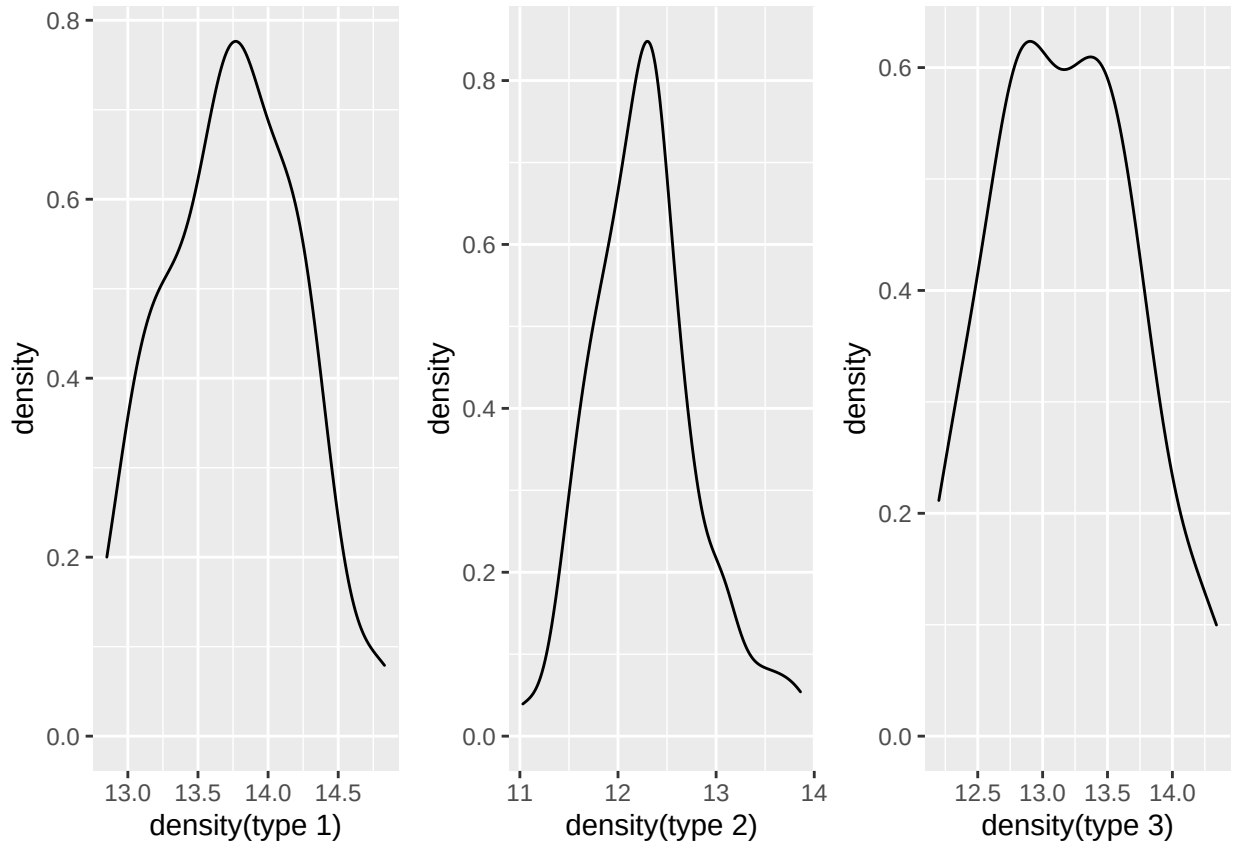
```
# Using correlation matrix from task 1 extract two highly correlated
# variables. Create scatter plot of that two variables. Use type as
# color parameter. Define following colors for your graph:
# #248745, #874724, #000000. Use theme bw. Add appropriate title.

High <- 0.865
Typec <- as.character(wine$Type)

ggplot(data = wine, aes(x = Flavanoids, y = Phenols, color = Typec)) +
geom_point() +
scale_color_manual(values = c('#248745', '#874724', '#000000')) +
theme_bw() +
ggtitle('Scatterplot of Flavanoids and OD with "Type" as color') +
labs(color='Type')
```




```
# ------------------------------------------------------------------
# Task 2.3
# ------------------------------------------------------------------
# Using visualization techniques check if there is distributional
# difference between alcohol and type of wine. Use theme_classic
# for your graph(s). Using alpha add some transparency. Comment
# on your findings.

dist_diff <- ggplot(data = wine, aes(x = Type, y = Alcohol, color = Typec, alpha = 125)) + geom_point()
dist_diff
```

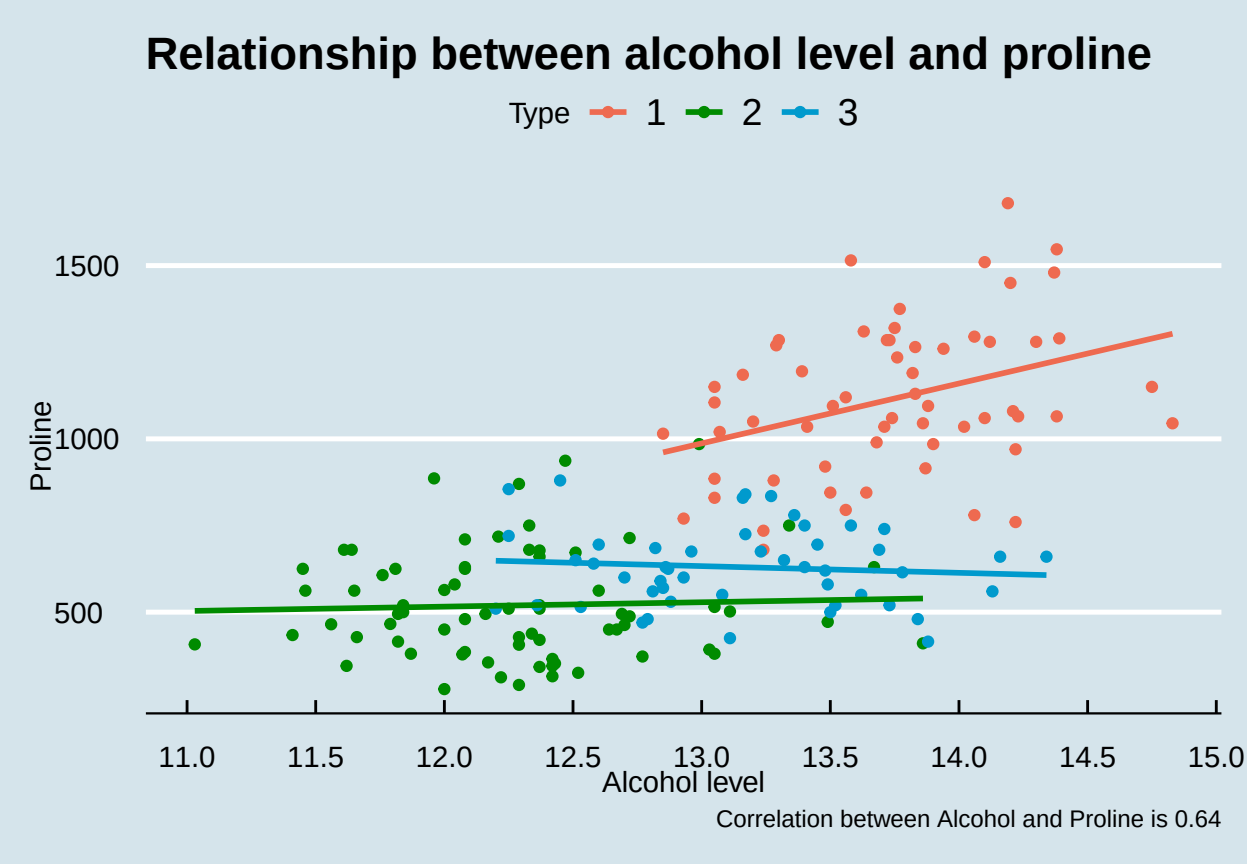



```
# it is apparent that Type 1 has the highest alcohol percentage,
# whereas Type 2 has the lowest and perhaps it has the widest
# range of percentages. Type 3, on the other hand, is somewhere
# between Type 1 and Type 2

wine_type1 <- subset(wine, Type == 1)
wine_type2 <- subset(wine, Type == 2)
wine_type3 <- subset(wine, Type == 3)
first <-ggplot(wine_type1, aes(x = Alcohol)) + geom_density() + xlab('density(type 1)')
second <- ggplot(wine_type2, aes(x = Alcohol)) + geom_density() + xlab('density(type 2)')
third <- ggplot(wine_type3, aes(x = Alcohol)) + geom_density() + xlab('density(type 3)')
gridExtra::grid.arrange(
  first,
  second,
  third,
  ncol = 3,
  nrow = 1
)
```

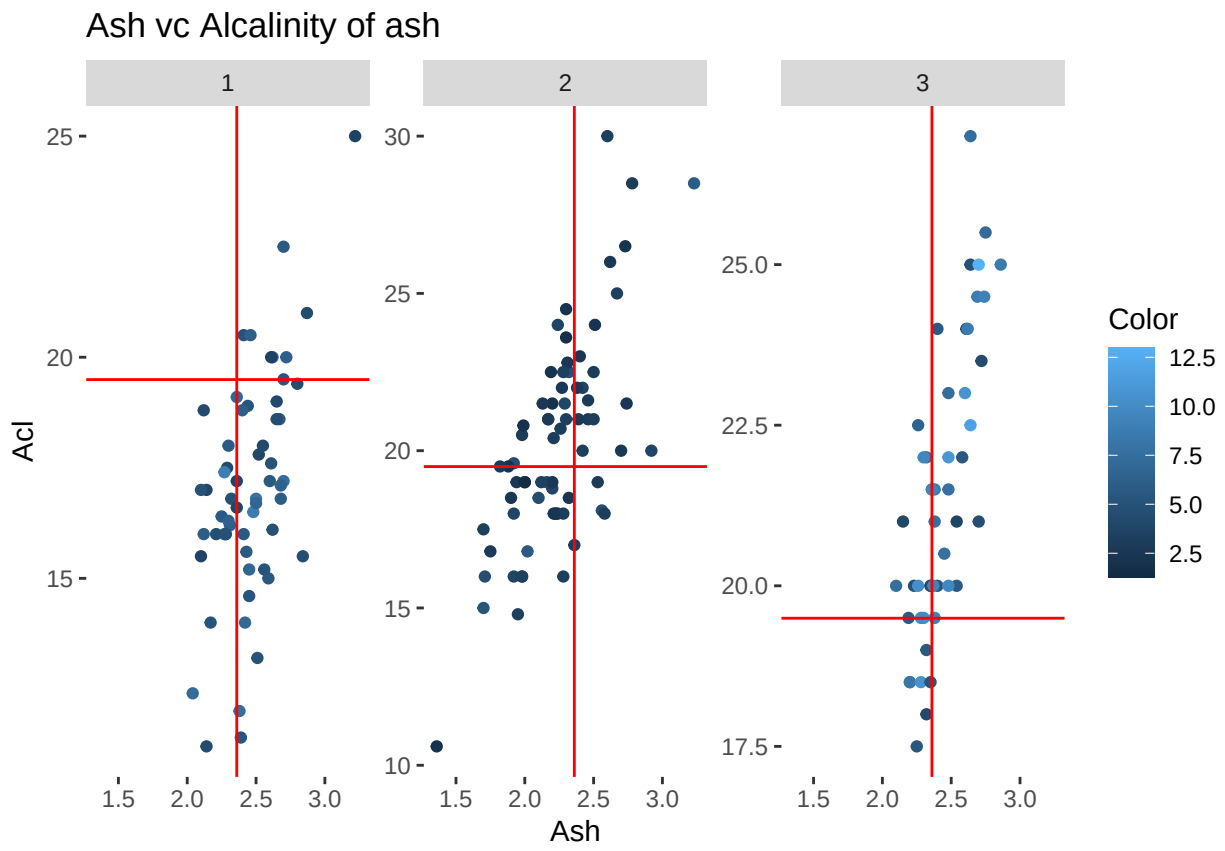



```
# here are my density plots which show some insights about distributions
# type 3 has two peaks, whereas type 1 and type 2 have single peak points
# also they are all a little right skewed and look kind of like bell
# shaped distribution

# ------------------------------------------------------------------
# Task 2.4
# ------------------------------------------------------------------
# Reproduce
footnote <- paste0('Correlation between Alcohol and Proline is ', round(cor(wine$Proline, wine$Alcohol)

corr_proalc <- ggplot(data = wine, aes(x = Alcohol, y = Proline, color = Typec)) +
geom_point() + geom_smooth(method = 'lm', alpha = 0) + xlab('Alcohol level') + ggtitle('Relationship be
scale_x_continuous(breaks = seq(11, 15, 0.5)) +
scale_color_manual(values = c('#EE6A50', '#008B00', '#009ACD')) +
theme_economist() + labs(color='Type', caption = footnote)

corr_proalc
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
# ------------------------------------------------------------------
# BONUS
# ------------------------------------------------------------------
ggplot(data = wine, aes(x = Ash, y = Acl, color = Color)) +
  geom_point() +
  labs(title = "Ash vc Alcalinity of ash") +
  theme(panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    panel.border = element_blank(),
    panel.background = element_blank(),
    axis.line = element_blank()) +
  facet_wrap(~Type, scales = 'free_y') +
  geom_hline(yintercept = mean(wine$Acl), color="red") +
  geom_vline(xintercept = median(wine$Ash), color="red")
```



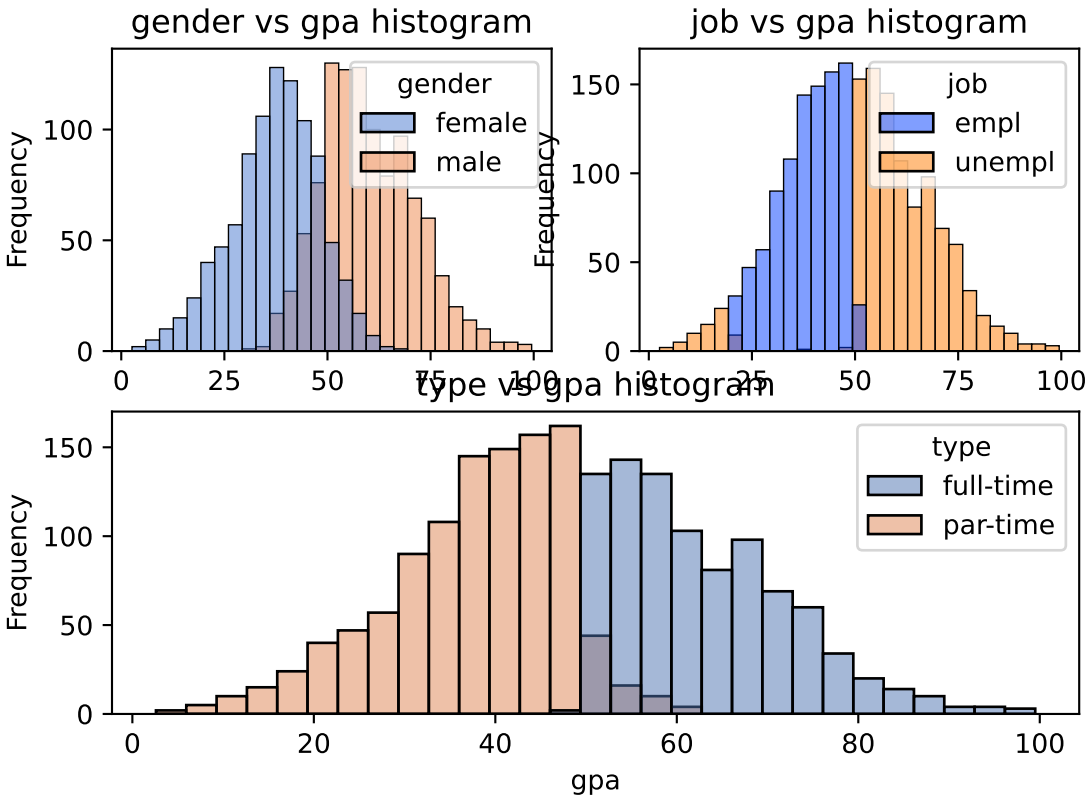

```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib as mpl
import pandas as pd
import seaborn as sns
```

```
# ------------------------------------------------------------------
# Task 3
# ------------------------------------------------------------------
# Read the data and check if there are any missing values. Convert
# all object variables to categorical.

df = pd.read_csv('gpafactors_2.csv')
df.isnull().values.any()
```

```
## False
```

```
df['gender'] = df.gender.astype('category')
df['job'] = df.job.astype('category')
df['type'] = df.type.astype('category')
df = df.rename(columns={'marital.status': 'marital'})
df['marital.status'] = df.marital.astype('category')
df['imp'] = df.imp.astype('category')

# ------------------------------------------------------------------
```

```
# Task 3.1
# ------------------------------------------------------------------
# Create histograms with binary variables vs grade point average of
# students. Use your own colors for each graph with appropriate title.

plt.close()
plt.subplot(2,2,1)
sns.histplot(data = df, x = 'gpa', hue = 'gender', palette = 'muted')
plt.title('gender vs gpa histogram')
plt.ylabel('Frequency')

plt.subplot(2,2,2)
sns.histplot(data = df, x = 'gpa', hue = 'job', palette = 'bright')
plt.title('job vs gpa histogram')
plt.ylabel('Frequency')

plt.subplot(2,1,2)
sns.histplot(data = df, x = 'gpa', hue = 'type', palette = 'deep')
plt.title('type vs gpa histogram')
plt.ylabel('Frequency')
plt.show()

# ------------------------------------------------------------------
# Task 3.2
# ------------------------------------------------------------------
# Using np.unique() and list() functions extract unique values of
# gender and job columns. Visualize relationship between age and
# gpa for each possible combination (create subplots). Set
# different colors and approrpiate titles for each subplot.
```

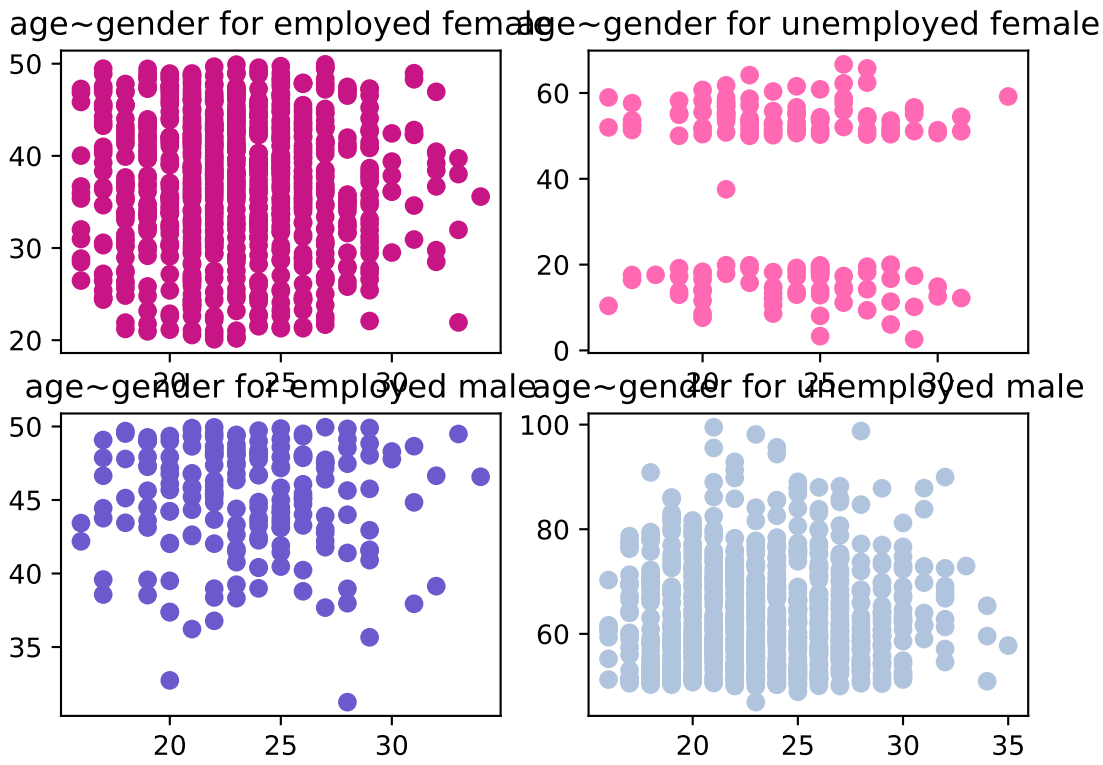



```
a = np.unique(df['gender']).tolist()
b = np.unique(df['job']).tolist()
list = [(x,y) for x in a for y in b]
list
```

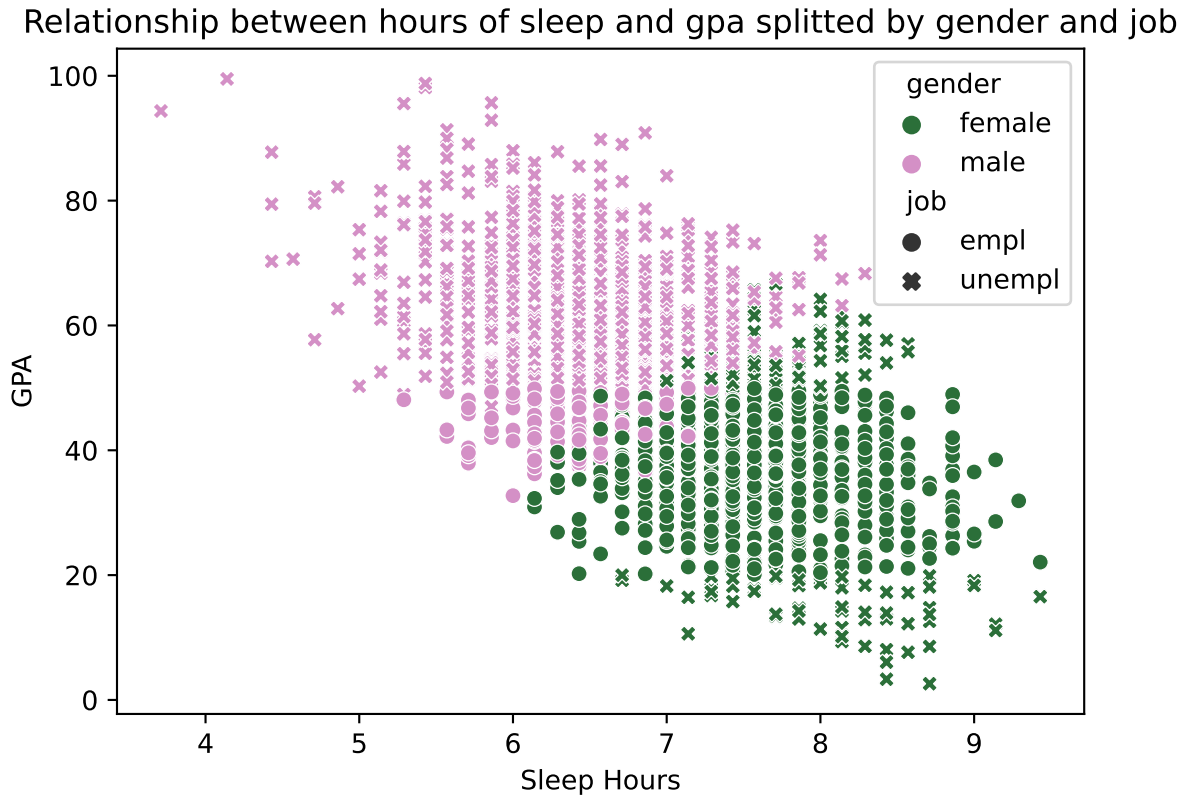
```
## [('female', 'empl'), ('female', 'unempl'), ('male', 'empl'), ('male', 'unempl')]
```

```
plt.close()
df1 = df.loc[(df.gender == list[0][0]) & (df.job == list[0][1])]
plt.subplot(2,2,1)
plt.scatter(data = df1, x = 'age', y = 'gpa', c = 'mediumvioletred')
plt.title('age~gender for employed female')

df2 = df.loc[(df.gender == list[1][0]) & (df.job == list[1][1])]
plt.subplot(2,2,2)
plt.scatter(data = df2, x = 'age', y = 'gpa', c = 'hotpink')
plt.title('age~gender for unemployed female')

df3 = df.loc[(df.gender == list[2][0]) & (df.job == list[2][1])]
plt.subplot(2,2,3)
plt.scatter(data = df3, x = 'age', y = 'gpa', c = 'slateblue')
plt.title('age~gender for employed male')

df4 = df.loc[(df.gender == list[3][0]) & (df.job == list[3][1])]
plt.subplot(2,2,4)
```

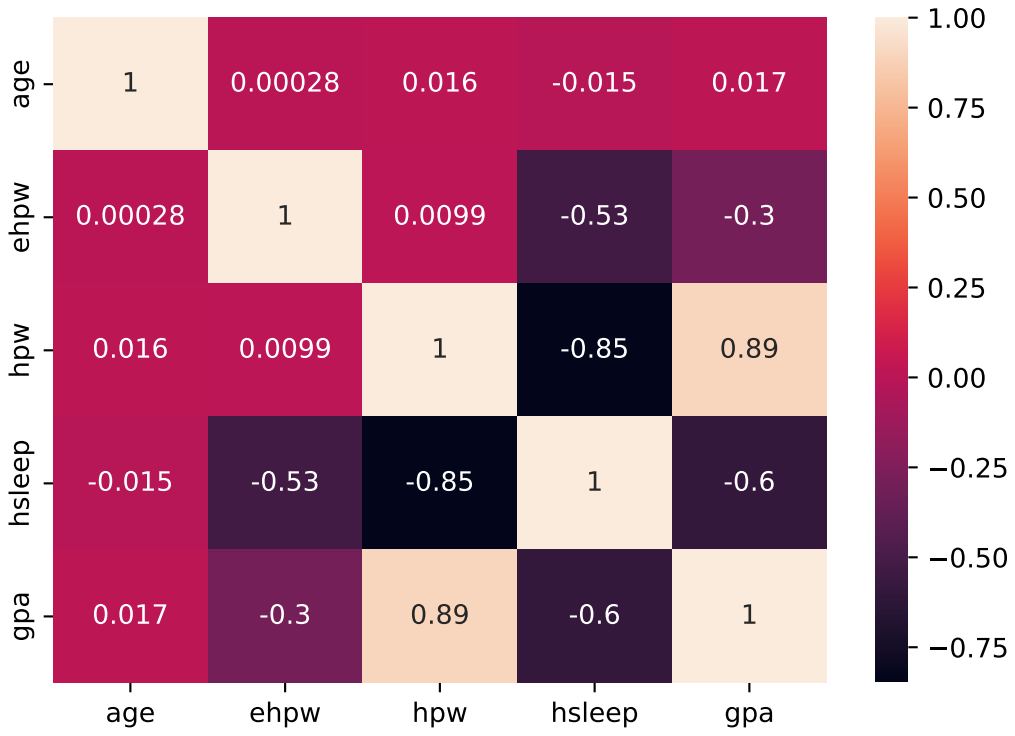
```
plt.scatter(data = df4, x = 'age', y = 'gpa', c = 'lightsteelblue')
plt.title('age~gender for unemployed male')
plt.show()

# ------------------------------------------------------------------
# Task 3.3
# ------------------------------------------------------------------
# Reproduce
```




```
plt.close()
sns.scatterplot(x = 'hsleep', y = 'gpa', data=df, hue = 'gender', palette = 'cubehelix', style = 'job')
plt.title('Relationship between hours of sleep and gpa splitted by gender and job')
plt.xlabel('Sleep Hours')
plt.ylabel('GPA')
plt.show()

# ------------------------------------------------------------------
# Task 3.4
# ------------------------------------------------------------------
# Create correlation heatmap and show correlation values on the
# graph. Comment on the results.
```




```
plt.close()
sns.heatmap(df.corr(), annot = True)
plt.show()

# with the values on it, it is clearly visible that the most
# positively correlated pair is gpa~hsleep. The most
# negatively correlated pair is hsleep~hpw. There are some pairs
# that are almost not correlated like ehpw~age and etc.

# ------------------------------------------------------------------
# Task 3.5
# ------------------------------------------------------------------
# look at my 3.3 graph there are almost NO women from 20-43
# who are not employed. That`s so weird.
```



Task 4

---

https://www.youtube.com/watch?v=1FZ7DbQwVcw

https://www.youtube.com/watch?v=o_1aF54DO60