mafs

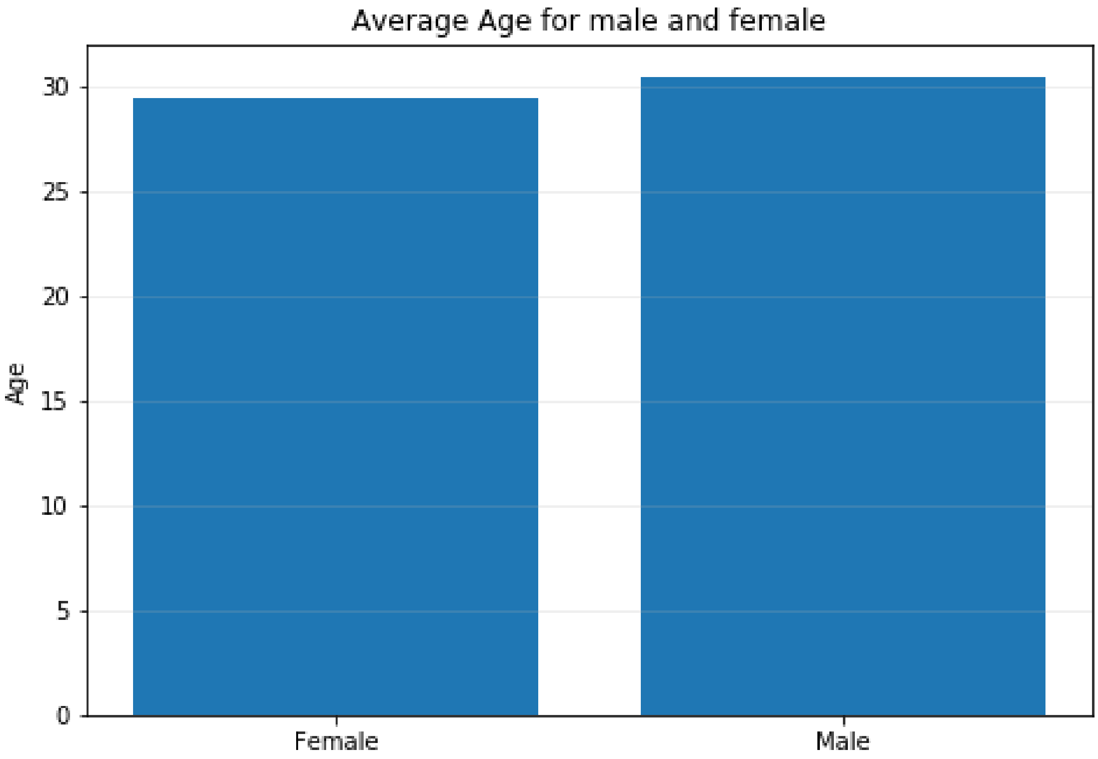
July 6, 2020

Import libraries and load data

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

file = pd.read_csv("mafs.csv")
file1 = file.copy(deep = True)
```

[24]:
```
# Age variation using a histogram
plt.figure()
A=plt.hist(file1["Age"], edgecolor="red")
plt.xlabel("Age")
plt.ylabel("Frequency")
plt.title("Age variation among couples")
```

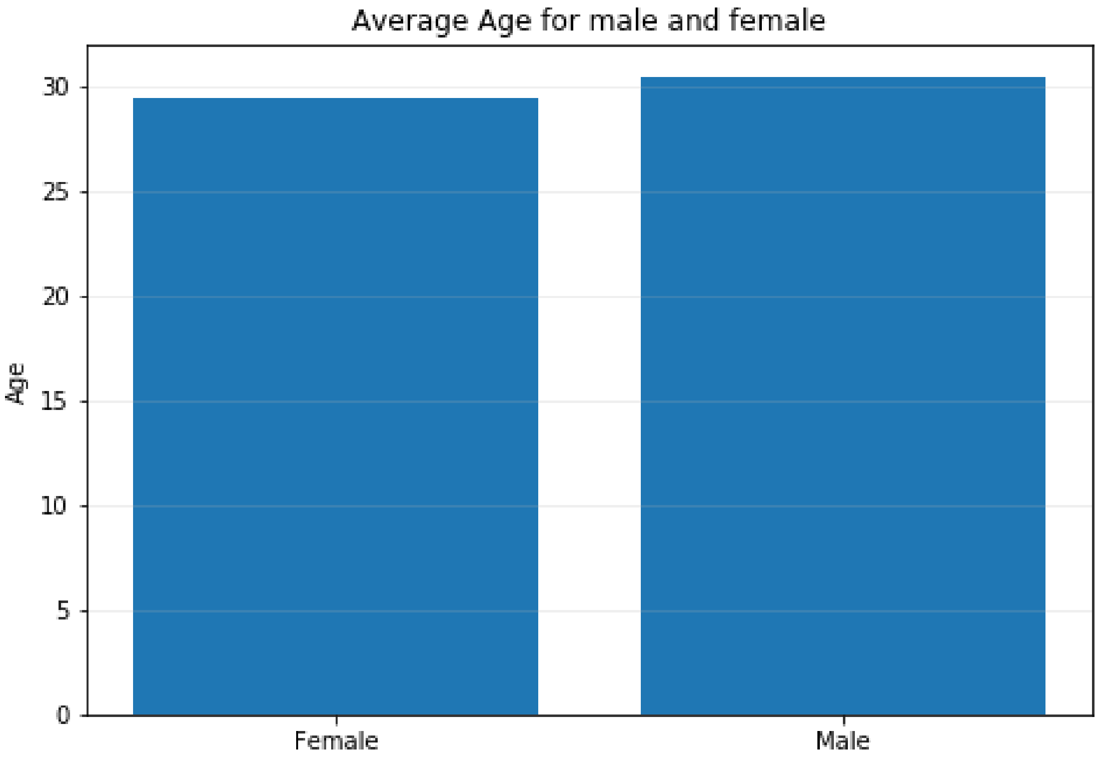
[24]:
```
Text(0.5,1,'Age variation among couples')
```



What's the average age of male and female?

[30]:
```
F_Age = []
M_Age = []

for i in file1.index:
    if file1['Gender'][i]=='F':
        F_Age.append(file1['Age'][i])
    else:
        M_Age.append(file1['Age'][i])

print(round(sum(F_Age)/len(F_Age),2))
print(round(sum(M_Age)/len(M_Age),2))

# Plot average age of F&M
fig = plt.figure()
ax = fig.add_axes([0,0,1,1])
x_ax = ['Female','Male']
y_ax = [29.44,30.5]
ax.bar(x_ax,y_ax)
plt.ylabel('Age')
plt.grid(axis='y', alpha=0.25)
plt.title('Average Age for male and female')
plt.show()
```

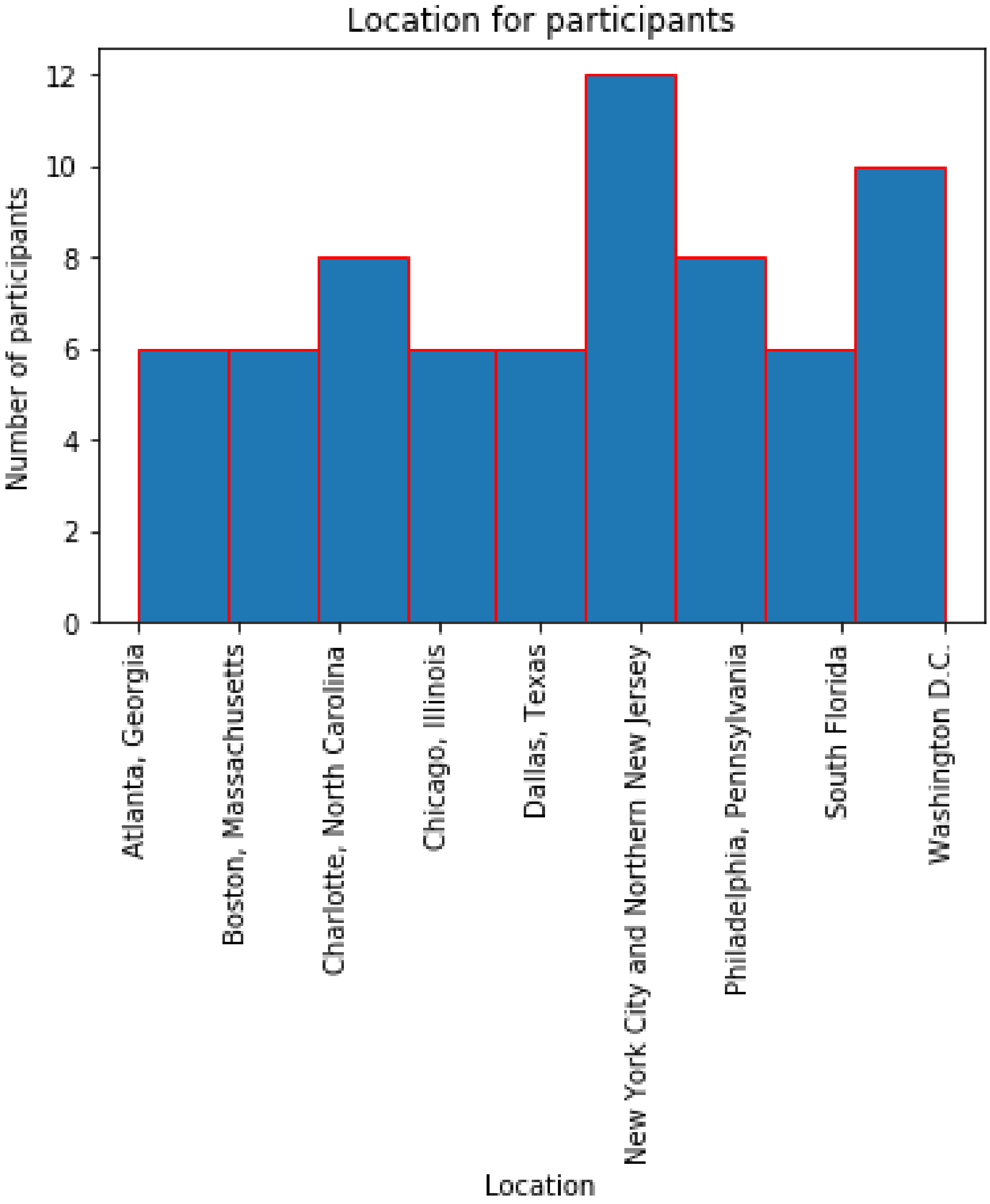
```
29.44
30.5
```



Where are they from ?

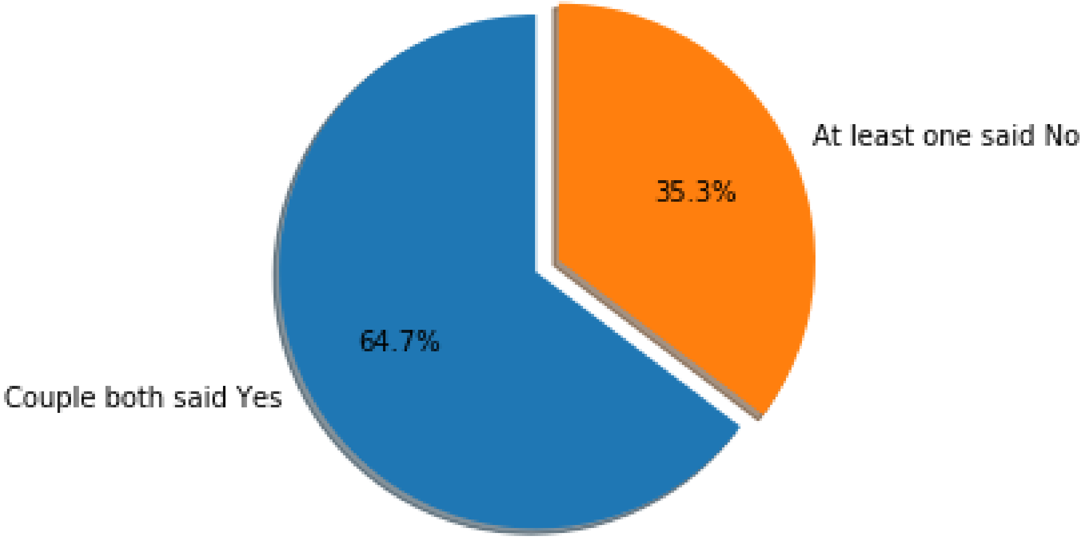
[102]:
```
# location
plt.figure()
plt.hist(file1["Location"], bins=9,edgecolor="red")
plt.xticks(rotation=90)
plt.xlabel("Location")
plt.ylabel("Number of participants")
plt.title("Location for participants")
```

[102]:
```
Text(0.5,1,'Location for participants')
```



How many couples said yes(accept marriage or dating) to each other?

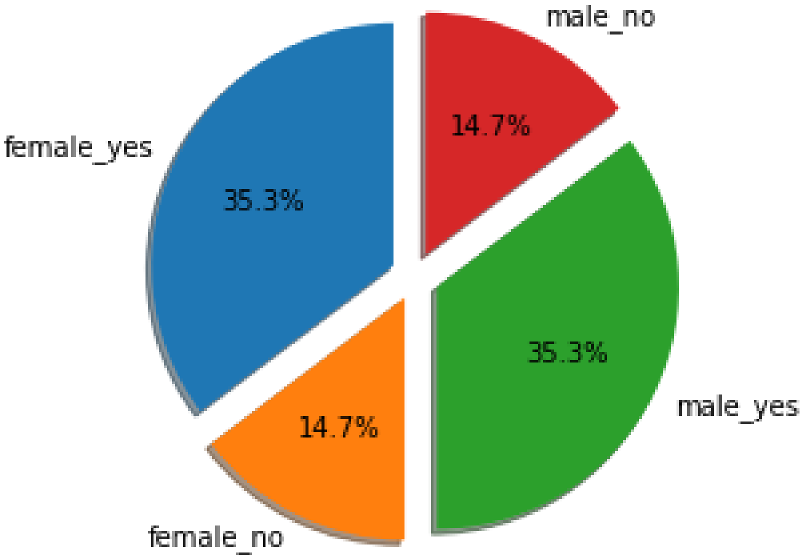
```
[88]: F_dec = []
M_dec = []
st = []
for i in file1.index:
    if file1['Gender'][i]=='F':
        F_dec.append(file1['Decision'][i])
        st.append(file1['Status'][i])
    else:
        M_dec.append(file1['Decision'][i])

dec_count=0
fe_yes=0
mal_yes=0
final_dec=[]
# both say yes
for i in range(len(F_dec)):
    if F_dec[i] =='Yes' and M_dec[i] =='Yes':
        dec_count+=1
        final_dec.append('Yes')
    else:
        final_dec.append('No')
    if F_dec[i] =='Yes':
        fe_yes+=1
    if M_dec[i]=='Yes':
        mal_yes+=1


# print('both_yes:{}'.format(dec_count))#22 yes
# print('fe_yes:{}'.format(fe_yes)) #24
# print('mal_yes:{}'.format(mal_yes))#24
# print(len(F_dec))#34

# Status count (married or divorced)
st_count=0
for i in st:
    if i =='Married':
        st_count+=1
#print(st_count) #married 9
#print(F_dec)
#print(M_dec)
#print(st)

#how many couples said yes (accept marriage or dating)?
labels= 'Couple both said Yes','At least one said No'
sizes = [dec_count,len(F_dec)-dec_count]
explode = (0, 0.1)

fig1, ax1 = plt.subplots()
ax1.pie(sizes, explode=explode, labels=labels, autopct='%1.1f%%',
        shadow=True, startangle=90)
ax1.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.
plt.show()
```



Does female say yes more than male? or another way?

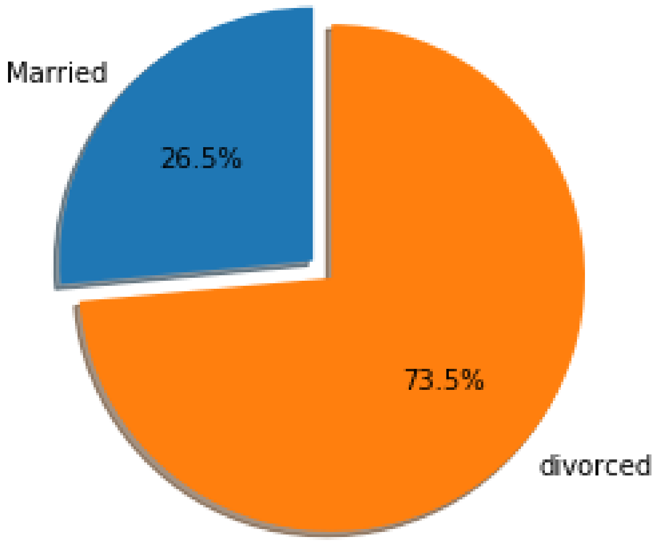
[87]:

```
# female_yes, male_yes
labels= 'female_yes','female_no','male_yes', 'male_no'
sizes = [fe_yes,len(F_dec)-fe_yes, mal_yes,len(M_dec)-mal_yes]
explode = (0.1, 0.1,0.1,0.1)

fig1, ax1 = plt.subplots()
ax1.pie(sizes, explode=explode, labels=labels, autopct='%1.1f%%',
        shadow=True, startangle=90)
ax1.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.

plt.show()
```



How many couples married and still married?

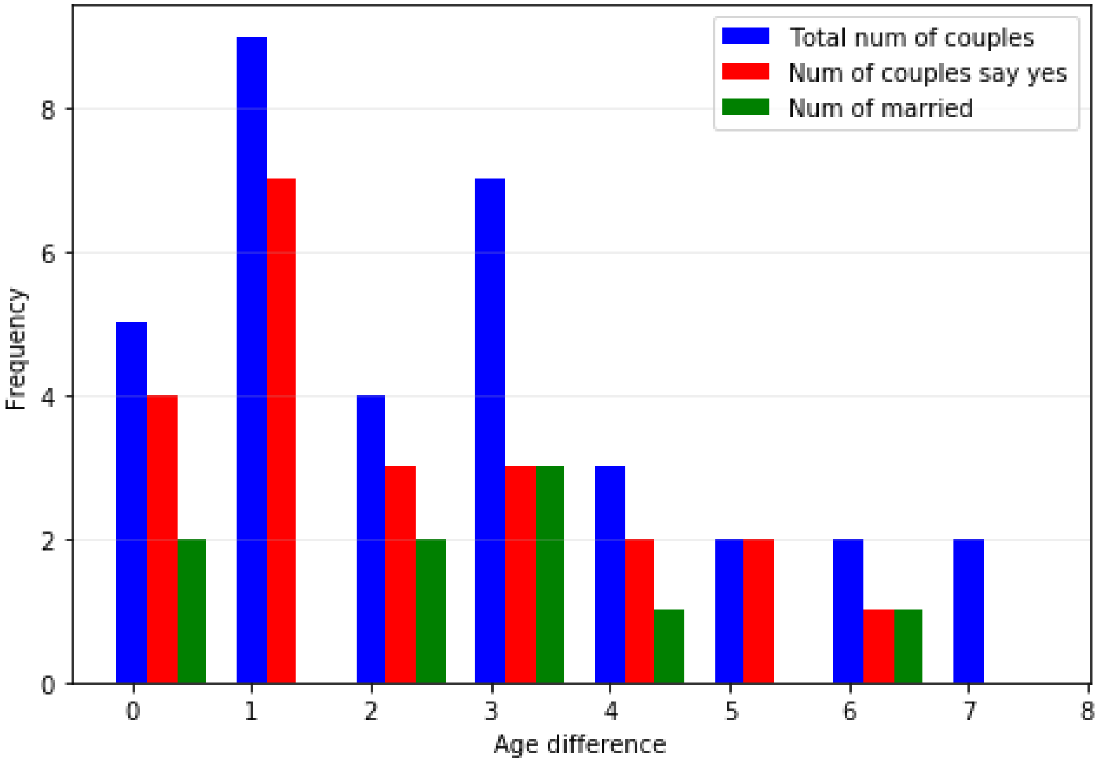
[82]:
```
# how many couples still married?
labels= 'Married','divorced'
sizes = [st_count,len(F_dec)-st_count]
explode = (0, 0.1)

fig1, ax1 = plt.subplots()
ax1.pie(sizes, explode=explode, labels=labels, autopct='%1.1f%%',
        shadow=True, startangle=90)
ax1.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.

plt.show()
```



Does age different is the key factor for their relationship?

[96]:

```
# age difference between couples
#print(F_Age)
#print(M_Age)
age_diff=[]
for i in range(len(F_Age)):
    age_diff.append(abs(F_Age[i]-M_Age[i]))
#print(age_diff)

# calculate nums of couples for each age difference
age_dec={}
age_dif={}
for age, dec in zip(age_diff, final_dec):
    age_dif[age]=age_dif.get(age,0)+1
    if dec =='Yes':
        age_dec[age]= age_dec.get(age,0)+1

#print(age_dec)
#print(age_dif)

#age difference for married couple
f_age=[]
m_age=[]
for i in file1.index:
    if file1['Status'][i]=='Married':
        if file1['Gender'][i] == 'F':
            f_age.append(file1['Age'][i])
        else:
            m_age.append(file1['Age'][i])

married_age_diff_dic={}
for i in range(len(f_age)):
    married_age_diff_dic[abs(f_age[i]-m_age[i])]=married_age_diff_dic.
 ↪get(abs(f_age[i]-m_age[i]),0)+1

#print(married_age_diff_dic)
num_yes = [4,7,3,3,2,2,1,0]
num_age_diff=[5,9,4,7,3,2,2,2]
num_married =[2,0,2,3,1,0,1,0]

# Does age difference effect couples?
x=np.arange(8)
fig = plt.figure()
ax = fig.add_axes([0,0,1,1])
ax.bar(x + 0.00, num_age_diff, color='b',width=0.25)
ax.bar(x + 0.25, num_yes, color='r', width = 0.25)
ax.bar(x + 0.5, num_married, color='g', width = 0.25)
plt.legend(['Total num of couples', 'Num of couples say yes','Num of married'])
plt.grid(axis='y', alpha=0.25)
plt.xlabel('Age difference')
plt.ylabel('Frequency')
plt.show()
```


Total num of couples
Num of couples say yes
Num of married
Frequency
0
2
4
6
8
0
1
2
3
4
5
6
7
8
Age difference

```
[ ]:
```