Descriptive statistics with Python

Dhafer Malouche

Contents

1 Descriptive Statistics with Python **1**
1.1 Type of variables . . . 2
1.1.1 Discrete Variable . . . 2
1.1.2 Continuous Variable . . . 6
1.2 Measure relationship between two variables . . . 12
1.2.1 Between two continuous variables . . . 12
1.2.2 Between two disrete variables . . . 21

1 Descriptive Statistics with Python

This chapter aims to learn how to perform your first data analysis using Python. We will first talk about the types of variables found in any collected dataset. We then show the kind of statistics that should be calculated: mean, median, mode, to provide a good description of the data. We finish the chapter by showing how to study the pairwise relationships between variables. We will then show how to compute correlation coefficients and draw the correlation's heat map. We will also present the χ^2-test to determine the nature of the relationship between two categorical variables

Learning outcomes:

- Types of variables
- Discrete variables, frequency, proportions, percentages
- Measuring central tendency for continuous variables: Mean, Median, Quantiles
- Measuring dispersion for continuous variables: Range, IQR, Variance, STD
- Relationship between two continuous variables: Correlation coefficient and matrix, heat map.
- Relationship between two discrete variables: Independence models, χ^2-test, mosaic plot.

Table of Contents

1 Type of variables

1.1 Discrete Variable

1.2 Continuous Variable

1.2.1 Measuring central tendancy

1.2.2 Measuring dispersion

2 Measure relationship between two variables

2.1 Between two continuous variables

2.2 Between two disrete variables

1.1 Type of variables

1.1.1 Discrete Variable

[2]:
```
import pandas as pd
```

[3]:
```
df=pd.read_csv("data1.csv",index_col='EmployeeNumber')
```

[6]:
```
df
```

[6]:
```
                Age Attrition     BusinessTravel  DailyRate  \
EmployeeNumber
1                41       Yes      Travel_Rarely       1102
2                49        No  Travel_Frequently        279
4                37       Yes      Travel_Rarely       1373
5                33        No  Travel_Frequently       1392
7                27        No      Travel_Rarely        591
...             ...       ...                ...        ...
2061             36        No  Travel_Frequently        884
2062             39        No      Travel_Rarely        613
2064             27        No      Travel_Rarely        155
2065             49        No  Travel_Frequently       1023
2068             34        No      Travel_Rarely        628

                            Department  DistanceFromHome  Education  \
EmployeeNumber
1                                Sales                 1          2
2               Research & Development                 8          1
4               Research & Development                 2          2
5               Research & Development                 3          4
7               Research & Development                 2          1
...                                ...               ...        ...
2061            Research & Development                23          2
2062            Research & Development                 6          1
2064            Research & Development                 4          3
2065                             Sales                 2          3
2068            Research & Development                 8          3

               EducationField  EmployeeCount  EnvironmentSatisfaction  ...  \
EmployeeNumber                                                         ...
1               Life Sciences              1                        2  ...
2               Life Sciences              1                        3  ...
4                       Other              1                        4  ...
```

```
5                Life Sciences              1                         4  ...
7                      Medical              1                         1  ...
...                        ...            ...                       ...  ...
2061                   Medical              1                         3  ...
2062                   Medical              1                         4  ...
2064             Life Sciences              1                         2  ...
2065                   Medical              1                         4  ...
2068                   Medical              1                         2  ...

               RelationshipSatisfaction  StandardHours  StockOptionLevel  \
EmployeeNumber
1                                     1             80                 0
2                                     4             80                 1
4                                     2             80                 0
5                                     3             80                 0
7                                     4             80                 1
...                                 ...            ...               ...
2061                                  3             80                 1
2062                                  1             80                 1
2064                                  2             80                 1
2065                                  4             80                 0
2068                                  1             80                 0

                TotalWorkingYears TrainingTimesLastYear  WorkLifeBalance  \
EmployeeNumber
1                               8                     0                1
2                              10                     3                3
4                               7                     3                3
5                               8                     3                3
7                               6                     3                3
...                           ...                   ...              ...
2061                           17                     3                3
2062                            9                     5                3
2064                            6                     0                3
2065                           17                     3                2
2068                            6                     3                4

               YearsAtCompany  YearsInCurrentRole  YearsSinceLastPromotion  \
EmployeeNumber
1                           6                   4                        0
2                          10                   7                        1
4                           0                   0                        0
5                           8                   7                        3
7                           2                   2                        2
...                       ...                 ...                      ...
2061                        5                   2                        0
2062                        7                   7                        1
```

```
2064                           6                    2                         0
2065                           9                    6                         0
2068                           4                    3                         1

                YearsWithCurrManager
EmployeeNumber
1                                  5
2                                  7
4                                  0
5                                  0
7                                  2
...                              ...
2061                               3
2062                               7
2064                               3
2065                               8
2068                               2

[1470 rows x 34 columns]
```

[5]:
```
df.columns
```

[5]:
```
Index(['Age', 'Attrition', 'BusinessTravel', 'DailyRate', 'Department',
       'DistanceFromHome', 'Education', 'EducationField', 'EmployeeCount',
       'EnvironmentSatisfaction', 'Gender', 'HourlyRate', 'JobInvolvement',
       'JobLevel', 'JobRole', 'JobSatisfaction', 'MaritalStatus',
       'MonthlyIncome', 'MonthlyRate', 'NumCompaniesWorked', 'Over18',
       'OverTime', 'PercentSalaryHike', 'PerformanceRating',
       'RelationshipSatisfaction', 'StandardHours', 'StockOptionLevel',
       'TotalWorkingYears', 'TrainingTimesLastYear', 'WorkLifeBalance',
       'YearsAtCompany', 'YearsInCurrentRole', 'YearsSinceLastPromotion',
       'YearsWithCurrManager'],
      dtype='object')
```

[6]:
```
df['JobSatisfaction']
```

[6]:
```
EmployeeNumber
1       4
2       2
4       3
5       3
7       2
       ..
2061    4
2062    1
2064    2
2065    2
```

```
2068    3
Name: JobSatisfaction, Length: 1470, dtype: int64
```

[7]:
```
df['Attrition'].value_counts()
```

[7]:
```
No     1233
Yes     237
Name: Attrition, dtype: int64
```

[8]:
```
df['Gender'].value_counts()
```

[8]:
```
Male      882
Female    588
Name: Gender, dtype: int64
```

[8]:
```
df['JobSatisfaction'].value_counts()
```

[8]:
```
4    459
3    442
1    289
2    280
Name: JobSatisfaction, dtype: int64
```

[9]:
```
df['Age'].value_counts()
```

[9]:
```
35    78
34    77
36    69
31    69
29    68
32    61
30    60
33    58
38    58
40    57
37    50
27    48
28    48
42    46
39    42
45    41
41    40
26    39
44    33
46    33
43    32
50    30
25    26
```

```
24    26
49    24
47    24
55    22
51    19
53    19
48    19
54    18
52    18
22    16
56    14
23    14
58    14
21    13
20    11
59    10
19     9
18     8
60     5
57     4
Name: Age, dtype: int64
```

[11]:
```
df['Attrition'].value_counts()
```

[11]:
```
No     1233
Yes     237
Name: Attrition, dtype: int64
```

[10]:
```
df['Attrition'].value_counts(normalize=True)
```

[10]:
```
No     0.838776
Yes    0.161224
Name: Attrition, dtype: float64
```

1.1.2 Continuous Variable

Measuring central tendancy **Mean**: $\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$

[14]:
```
df['Age'].mean()
```

[14]:
```
36.923809523809524
```

[15]:
```
df['Education'].value_counts()
```

[15]:
```
3    572
4    398
2    282
```

```
1     170
5      48
Name: Education, dtype: int64
```

[ ]:

[13]:
```
df['Education'].mean()
```

[13]:
```
2.912925170068027
```

The **mode** is the highest-occuring item in a group of observations

[16]:
```
df['Age'].mode()
```

[16]:
```
0    35
dtype: int64
```

[17]:
```
df['Age'][0:10]
```

[17]:
```
EmployeeNumber
1      41
2      49
4      37
5      33
7      27
8      32
10     59
11     30
12     38
13     36
Name: Age, dtype: int64
```

[18]:
```
df['Age'][0:10].mode()
```

[18]:
```
0    27
1    30
2    32
3    33
4    36
5    37
6    38
7    41
8    49
9    59
dtype: int64
```

The **median** is the midpoint or middle value in a group of observations. It is also called the 50th percentile.

[16]:
```
df['Age'].median()
```

[16]:
```
36.0
```

The **median** is also called the **50% quantile** or the **2nd quantile**

[17]:
```
df['Age'].quantile(.5)
```

[17]:
```
36.0
```

we can compute the **1st quantile** or the **25% quantile**

[18]:
```
df['Age'].quantile(.25)
```

[18]:
```
30.0
```

and the the **3rd quantile** or the **75% quantile**

[16]:
```
df['Age'].quantile(.75)
```

[16]:
```
43.0
```

[19]:
```
df['Age'].quantile(.1)
```

[19]:
```
26.0
```

[20]:
```
df['Age'].quantile(.2)
```

[20]:
```
29.0
```

And we can compute **the five numbers summary**. It's composed of the `min`, `1st quantile`, `median`, `3rd quantile`, and `max`

[21]:
```
df['Age'].quantile([0,.25,.5,.75,1])
```

[21]:
```
0.00    18.0
0.25    30.0
0.50    36.0
0.75    43.0
1.00    60.0
Name: Age, dtype: float64
```

Measuring dispersion **Range** is the difference between the maximum and the minimum

[22]:
```
df['Age'].max()
```

[22]:
```
60
```

[23]:
```
df['Age'].min()
```

[23]: 18

[24]:
```
df['Age'].max()-df['Age'].min()
```

[24]: 42

The **Inter-quantile range**:

[26]:
```
df['Age'].quantile(.75)-df['Age'].quantile(.25)
```

[26]: 13.0

The **Variance** measures the deviation from the mean

$$\sigma^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2$$

[27]:
```
df['Age'].var()
```

[27]: 83.45504878602227

[28]:
```
xbar=df['Age'].mean()
```

[29]:
```
y=(df['Age']-xbar)**2
```

[28]:
```
4**2
```

[28]: 16

[30]:
```
y[0:4]
```

[30]:
```
EmployeeNumber
1      16.615329
2     145.834376
4       0.005805
5      15.396281
Name: Age, dtype: float64
```

[31]:
```
n=len(df['Age'])
```

[32]:
```
n
```

[32]: 1470

[33]:
```
sum(y)
```

[33]: 122595.46666666672

[34]:
```
sum(y)/n
```

[34]: 83.39827664399097

[35]:
```
sum(y)/(n-1)
```

[35]: 83.45504878602227

The **standard deviation** is the square root of the variance

[35]:
```
df['Age'].std()
```

[35]: 9.135373489136734

[34]:
```
df['Age'].var()**(.5)
```

[34]: 9.135373489136734

We can also compute all these measures using one `Python` function

[36]:
```
df['Age'].describe()
```

[36]:
```
count    1470.000000
mean       36.923810
std         9.135373
min        18.000000
25%        30.000000
50%        36.000000
75%        43.000000
max        60.000000
Name: Age, dtype: float64
```

`describe` can be used on the whole dataset.

[37]:
```
df.describe()
```

[37]:
```
               Age    DailyRate  DistanceFromHome    Education  EmployeeCount  \
count  1470.000000  1470.000000       1470.000000  1470.000000         1470.0
mean     36.923810   802.485714          9.192517     2.912925            1.0
std       9.135373   403.509100          8.106864     1.024165            0.0
min      18.000000   102.000000          1.000000     1.000000            1.0
25%      30.000000   465.000000          2.000000     2.000000            1.0
50%      36.000000   802.000000          7.000000     3.000000            1.0
75%      43.000000  1157.000000         14.000000     4.000000            1.0
max      60.000000  1499.000000         29.000000     5.000000            1.0

       EnvironmentSatisfaction   HourlyRate  JobInvolvement     JobLevel  \
count              1470.000000  1470.000000     1470.000000  1470.000000
mean                  2.721769    65.891156        2.729932     2.063946
std                   1.093082    20.329428        0.711561     1.106940
min                   1.000000    30.000000        1.000000     1.000000
```

```
25%                    2.000000    48.000000         2.000000     1.000000
50%                    3.000000    66.000000         3.000000     2.000000
75%                    4.000000    83.750000         3.000000     3.000000
max                    4.000000   100.000000         4.000000     5.000000

       JobSatisfaction  ...  RelationshipSatisfaction  StandardHours  \
count      1470.000000  ...               1470.000000         1470.0
mean          2.728571  ...                  2.712245           80.0
std           1.102846  ...                  1.081209            0.0
min           1.000000  ...                  1.000000           80.0
25%           2.000000  ...                  2.000000           80.0
50%           3.000000  ...                  3.000000           80.0
75%           4.000000  ...                  4.000000           80.0
max           4.000000  ...                  4.000000           80.0

       StockOptionLevel  TotalWorkingYears  TrainingTimesLastYear  \
count       1470.000000        1470.000000            1470.000000
mean           0.793878          11.279592               2.799320
std            0.852077           7.780782               1.289271
min            0.000000           0.000000               0.000000
25%            0.000000           6.000000               2.000000
50%            1.000000          10.000000               3.000000
75%            1.000000          15.000000               3.000000
max            3.000000          40.000000               6.000000

       WorkLifeBalance  YearsAtCompany  YearsInCurrentRole  \
count      1470.000000     1470.000000         1470.000000
mean          2.761224        7.008163            4.229252
std           0.706476        6.126525            3.623137
min           1.000000        0.000000            0.000000
25%           2.000000        3.000000            2.000000
50%           3.000000        5.000000            3.000000
75%           3.000000        9.000000            7.000000
max           4.000000       40.000000           18.000000

       YearsSinceLastPromotion  YearsWithCurrManager
count              1470.000000           1470.000000
mean                  2.187755              4.123129
std                   3.222430              3.568136
min                   0.000000              0.000000
25%                   0.000000              2.000000
50%                   1.000000              3.000000
75%                   3.000000              7.000000
max                  15.000000             17.000000

[8 rows x 25 columns]
```

1.2 Measure relationship between two variables

In statistics, we're always to find the variables that are related to each other. After the one-dimensional description of the variables (called also `flat sorting` of the data) we explore also the pairwise relationship between the variables.

1.2.1 Between two continuous variables

To determine the relationship between two continuous variables, we use the **correlation coefficient**. It's often denoted by ρ. It's a number belonging to $[0, 1]$ and can be interpreted as follows

- If ρ is close to zero, we conclude that there's no evidence of a linear relationship between the two variables
- If ρ is close to +1, there's probably a *positive* relationship between the two variables
- If ρ is close to -1, there's probably a *negative* relationship between the two variables

Before computing ρ we should first draw a plot call the **scatter plot**:

[38]:
```
df.columns
```

[38]:
```
Index(['Age', 'Attrition', 'BusinessTravel', 'DailyRate', 'Department',
       'DistanceFromHome', 'Education', 'EducationField', 'EmployeeCount',
       'EnvironmentSatisfaction', 'Gender', 'HourlyRate', 'JobInvolvement',
       'JobLevel', 'JobRole', 'JobSatisfaction', 'MaritalStatus',
       'MonthlyIncome', 'MonthlyRate', 'NumCompaniesWorked', 'Over18',
       'OverTime', 'PercentSalaryHike', 'PerformanceRating',
       'RelationshipSatisfaction', 'StandardHours', 'StockOptionLevel',
       'TotalWorkingYears', 'TrainingTimesLastYear', 'WorkLifeBalance',
       'YearsAtCompany', 'YearsInCurrentRole', 'YearsSinceLastPromotion',
       'YearsWithCurrManager'],
      dtype='object')
```

We will check the relationship between `HourlyRate` and `YearsAtCompany`. We will first draw the scatter plot. We need then to install `matplotlib` library.

[39]:
```
!pip install matplotlib
```

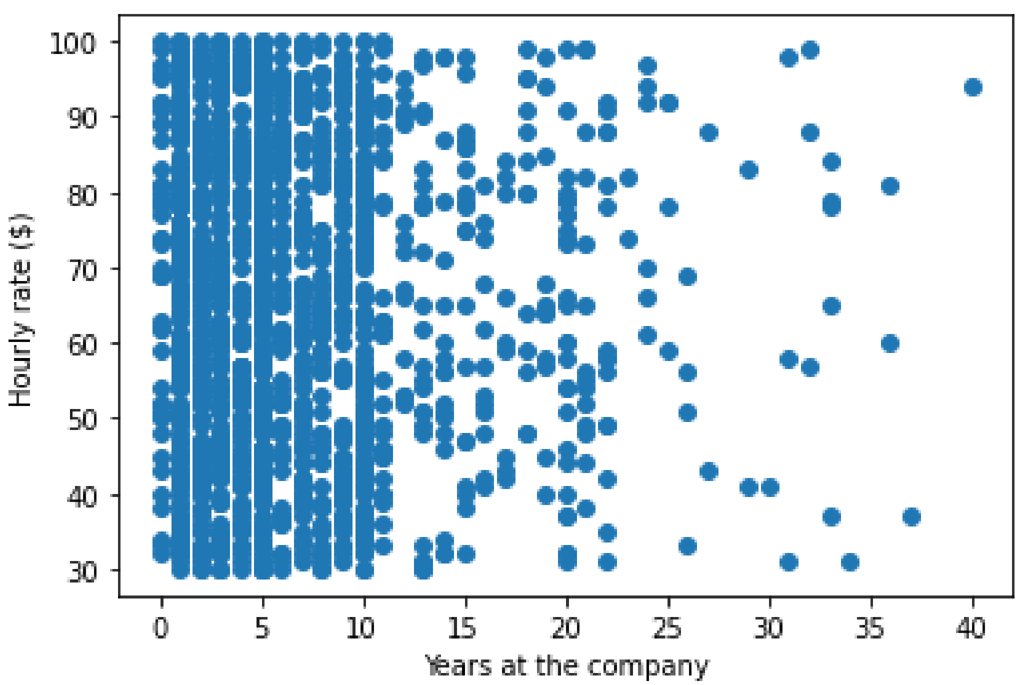
```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: matplotlib in
c:\users\dhafe\appdata\roaming\python\python310\site-packages (3.5.1)
Requirement already satisfied: pyparsing>=2.2.1 in
c:\users\dhafe\appdata\roaming\python\python310\site-packages (from matplotlib)
(3.0.7)
Requirement already satisfied: cycler>=0.10 in
c:\users\dhafe\appdata\roaming\python\python310\site-packages (from matplotlib)
(0.11.0)
Requirement already satisfied: python-dateutil>=2.7 in
c:\users\dhafe\appdata\roaming\python\python310\site-packages (from matplotlib)
(2.8.2)
Requirement already satisfied: pillow>=6.2.0 in
```

```
c:\users\dhafe\appdata\roaming\python\python310\site-packages (from matplotlib)
(9.0.0)
Requirement already satisfied: fonttools>=4.22.0 in
c:\users\dhafe\appdata\roaming\python\python310\site-packages (from matplotlib)
(4.29.0)
Requirement already satisfied: kiwisolver>=1.0.1 in
c:\users\dhafe\appdata\roaming\python\python310\site-packages (from matplotlib)
(1.3.2)
Requirement already satisfied: packaging>=20.0 in
c:\users\dhafe\appdata\roaming\python\python310\site-packages (from matplotlib)
(21.3)
Requirement already satisfied: numpy>=1.17 in
c:\users\dhafe\appdata\roaming\python\python310\site-packages (from matplotlib)
(1.22.1)
Requirement already satisfied: six>=1.5 in
c:\users\dhafe\appdata\roaming\python\python310\site-packages (from python-
dateutil>=2.7->matplotlib) (1.16.0)
```

We import then `matplotlib`

[40]:
```python
import matplotlib.pyplot as plt
```

[42]:
```python
plt.scatter(df['YearsAtCompany'],df['HourlyRate'])
plt.ylabel("Hourly rate ($)")
plt.xlabel("Years at the company")
plt.show()
```



The correlation coefficient between these two variables is displayed in the following matrix

[43]:
```
df[['YearsAtCompany','HourlyRate']].corr(method='pearson')
```

[43]:

	YearsAtCompany	HourlyRate
YearsAtCompany	1.000000	-0.019582
HourlyRate	-0.019582	1.000000

and ρ can be extracted as follows

[53]:
```
x=df[['YearsAtCompany','HourlyRate']].corr(method='pearson')
```

[48]:
```
import numpy as np
```

[51]:
```
x=np.array(x)
```

[52]:
```
x[0,1]
```

[52]: -0.01958161620912128

Interpreation: there's no evidence of a linear relationship between the variables HourlyRate and YearsAtCompany

We can also compute at the same all the pairwise correlations between the variables of the data. **ONLY CORRELATIONS BETWEEN CONTINUOUS VARIABLES HAVE A STATISTICAL INTERPRETATION**.

[54]:
```
df.corr()
```

[54]:

	Age	DailyRate	DistanceFromHome	Education	\
Age	1.000000	0.010661	-0.001686	0.208034	
DailyRate	0.010661	1.000000	-0.004985	-0.016806	
DistanceFromHome	-0.001686	-0.004985	1.000000	0.021042	
Education	0.208034	-0.016806	0.021042	1.000000	
EmployeeCount	NaN	NaN	NaN	NaN	
EnvironmentSatisfaction	0.010146	0.018355	-0.016075	-0.027128	
HourlyRate	0.024287	0.023381	0.031131	0.016775	
JobInvolvement	0.029820	0.046135	0.008783	0.042438	
JobLevel	0.509604	0.002966	0.005303	0.101589	
JobSatisfaction	-0.004892	0.030571	-0.003669	-0.011296	
MonthlyIncome	0.497855	0.007707	-0.017014	0.094961	
MonthlyRate	0.028051	-0.032182	0.027473	-0.026084	
NumCompaniesWorked	0.299635	0.038153	-0.029251	0.126317	
PercentSalaryHike	0.003634	0.022704	0.040235	-0.011111	
PerformanceRating	0.001904	0.000473	0.027110	-0.024539	
RelationshipSatisfaction	0.053535	0.007846	0.006557	-0.009118	
StandardHours	NaN	NaN	NaN	NaN	
StockOptionLevel	0.037510	0.042143	0.044872	0.018422	

```
TotalWorkingYears          0.680381   0.014515           0.004628   0.148280
TrainingTimesLastYear     -0.019621   0.002453          -0.036942  -0.025100
WorkLifeBalance           -0.021490  -0.037848          -0.026556   0.009819
YearsAtCompany             0.311309  -0.034055           0.009508   0.069114
YearsInCurrentRole         0.212901   0.009932           0.018845   0.060236
YearsSinceLastPromotion    0.216513  -0.033229           0.010029   0.054254
YearsWithCurrManager       0.202089  -0.026363           0.014406   0.069065

                          EmployeeCount  EnvironmentSatisfaction  HourlyRate  \
Age                                 NaN                 0.010146    0.024287
DailyRate                           NaN                 0.018355    0.023381
DistanceFromHome                    NaN                -0.016075    0.031131
Education                           NaN                -0.027128    0.016775
EmployeeCount                       NaN                      NaN         NaN
EnvironmentSatisfaction             NaN                 1.000000   -0.049857
HourlyRate                          NaN                -0.049857    1.000000
JobInvolvement                      NaN                -0.008278    0.042861
JobLevel                            NaN                 0.001212   -0.027853
JobSatisfaction                     NaN                -0.006784   -0.071335
MonthlyIncome                       NaN                -0.006259   -0.015794
MonthlyRate                         NaN                 0.037600   -0.015297
NumCompaniesWorked                  NaN                 0.012594    0.022157
PercentSalaryHike                   NaN                -0.031701   -0.009062
PerformanceRating                   NaN                -0.029548   -0.002172
RelationshipSatisfaction            NaN                 0.007665    0.001330
StandardHours                       NaN                      NaN         NaN
StockOptionLevel                    NaN                 0.003432    0.050263
TotalWorkingYears                   NaN                -0.002693   -0.002334
TrainingTimesLastYear               NaN                -0.019359   -0.008548
WorkLifeBalance                     NaN                 0.027627   -0.004607
YearsAtCompany                      NaN                 0.001458   -0.019582
YearsInCurrentRole                  NaN                 0.018007   -0.024106
YearsSinceLastPromotion             NaN                 0.016194   -0.026716
YearsWithCurrManager                NaN                -0.004999   -0.020123

                          JobInvolvement  JobLevel  JobSatisfaction  ...  \
Age                             0.029820  0.509604        -0.004892  ...
DailyRate                       0.046135  0.002966         0.030571  ...
DistanceFromHome                0.008783  0.005303        -0.003669  ...
Education                       0.042438  0.101589        -0.011296  ...
EmployeeCount                        NaN       NaN              NaN  ...
EnvironmentSatisfaction        -0.008278  0.001212        -0.006784  ...
HourlyRate                      0.042861 -0.027853        -0.071335  ...
JobInvolvement                  1.000000 -0.012630        -0.021476  ...
JobLevel                       -0.012630  1.000000        -0.001944  ...
JobSatisfaction                -0.021476 -0.001944         1.000000  ...
MonthlyIncome                  -0.015271  0.950300        -0.007157  ...
```

```
MonthlyRate                -0.016322  0.039563          0.000644  ...
NumCompaniesWorked          0.015012  0.142501         -0.055699  ...
PercentSalaryHike          -0.017205 -0.034730          0.020002  ...
PerformanceRating          -0.029071 -0.021222          0.002297  ...
RelationshipSatisfaction    0.034297  0.021642         -0.012454  ...
StandardHours                    NaN       NaN               NaN  ...
StockOptionLevel            0.021523  0.013984          0.010690  ...
TotalWorkingYears          -0.005533  0.782208         -0.020185  ...
TrainingTimesLastYear      -0.015338 -0.018191         -0.005779  ...
WorkLifeBalance            -0.014617  0.037818         -0.019459  ...
YearsAtCompany             -0.021355  0.534739         -0.003803  ...
YearsInCurrentRole          0.008717  0.389447         -0.002305  ...
YearsSinceLastPromotion    -0.024184  0.353885         -0.018214  ...
YearsWithCurrManager        0.025976  0.375281         -0.027656  ...

                          RelationshipSatisfaction  StandardHours  \
Age                                       0.053535            NaN
DailyRate                                 0.007846            NaN
DistanceFromHome                          0.006557            NaN
Education                                -0.009118            NaN
EmployeeCount                                  NaN            NaN
EnvironmentSatisfaction                   0.007665            NaN
HourlyRate                                0.001330            NaN
JobInvolvement                            0.034297            NaN
JobLevel                                  0.021642            NaN
JobSatisfaction                          -0.012454            NaN
MonthlyIncome                             0.025873            NaN
MonthlyRate                              -0.004085            NaN
NumCompaniesWorked                        0.052733            NaN
PercentSalaryHike                        -0.040490            NaN
PerformanceRating                        -0.031351            NaN
RelationshipSatisfaction                  1.000000            NaN
StandardHours                                  NaN            NaN
StockOptionLevel                         -0.045952            NaN
TotalWorkingYears                         0.024054            NaN
TrainingTimesLastYear                     0.002497            NaN
WorkLifeBalance                           0.019604            NaN
YearsAtCompany                            0.019367            NaN
YearsInCurrentRole                       -0.015123            NaN
YearsSinceLastPromotion                   0.033493            NaN
YearsWithCurrManager                     -0.000867            NaN

                          StockOptionLevel  TotalWorkingYears  \
Age                               0.037510           0.680381
DailyRate                         0.042143           0.014515
DistanceFromHome                  0.044872           0.004628
Education                         0.018422           0.148280
```

EmployeeCount	NaN	NaN
EnvironmentSatisfaction	0.003432	-0.002693
HourlyRate	0.050263	-0.002334
JobInvolvement	0.021523	-0.005533
JobLevel	0.013984	0.782208
JobSatisfaction	0.010690	-0.020185
MonthlyIncome	0.005408	0.772893
MonthlyRate	-0.034323	0.026442
NumCompaniesWorked	0.030075	0.237639
PercentSalaryHike	0.007528	-0.020608
PerformanceRating	0.003506	0.006744
RelationshipSatisfaction	-0.045952	0.024054
StandardHours	NaN	NaN
StockOptionLevel	1.000000	0.010136
TotalWorkingYears	0.010136	1.000000
TrainingTimesLastYear	0.011274	-0.035662
WorkLifeBalance	0.004129	0.001008
YearsAtCompany	0.015058	0.628133
YearsInCurrentRole	0.050818	0.460365
YearsSinceLastPromotion	0.014352	0.404858
YearsWithCurrManager	0.024698	0.459188

	TrainingTimesLastYear	WorkLifeBalance \
Age	-0.019621	-0.021490
DailyRate	0.002453	-0.037848
DistanceFromHome	-0.036942	-0.026556
Education	-0.025100	0.009819
EmployeeCount	NaN	NaN
EnvironmentSatisfaction	-0.019359	0.027627
HourlyRate	-0.008548	-0.004607
JobInvolvement	-0.015338	-0.014617
JobLevel	-0.018191	0.037818
JobSatisfaction	-0.005779	-0.019459
MonthlyIncome	-0.021736	0.030683
MonthlyRate	0.001467	0.007963
NumCompaniesWorked	-0.066054	-0.008366
PercentSalaryHike	-0.005221	-0.003280
PerformanceRating	-0.015579	0.002572
RelationshipSatisfaction	0.002497	0.019604
StandardHours	NaN	NaN
StockOptionLevel	0.011274	0.004129
TotalWorkingYears	-0.035662	0.001008
TrainingTimesLastYear	1.000000	0.028072
WorkLifeBalance	0.028072	1.000000
YearsAtCompany	0.003569	0.012089
YearsInCurrentRole	-0.005738	0.049856
YearsSinceLastPromotion	-0.002067	0.008941

YearsWithCurrManager	-0.004096	0.002759

	YearsAtCompany	YearsInCurrentRole \
Age	0.311309	0.212901
DailyRate	-0.034055	0.009932
DistanceFromHome	0.009508	0.018845
Education	0.069114	0.060236
EmployeeCount	NaN	NaN
EnvironmentSatisfaction	0.001458	0.018007
HourlyRate	-0.019582	-0.024106
JobInvolvement	-0.021355	0.008717
JobLevel	0.534739	0.389447
JobSatisfaction	-0.003803	-0.002305
MonthlyIncome	0.514285	0.363818
MonthlyRate	-0.023655	-0.012815
NumCompaniesWorked	-0.118421	-0.090754
PercentSalaryHike	-0.035991	-0.001520
PerformanceRating	0.003435	0.034986
RelationshipSatisfaction	0.019367	-0.015123
StandardHours	NaN	NaN
StockOptionLevel	0.015058	0.050818
TotalWorkingYears	0.628133	0.460365
TrainingTimesLastYear	0.003569	-0.005738
WorkLifeBalance	0.012089	0.049856
YearsAtCompany	1.000000	0.758754
YearsInCurrentRole	0.758754	1.000000
YearsSinceLastPromotion	0.618409	0.548056
YearsWithCurrManager	0.769212	0.714365

	YearsSinceLastPromotion	YearsWithCurrManager
Age	0.216513	0.202089
DailyRate	-0.033229	-0.026363
DistanceFromHome	0.010029	0.014406
Education	0.054254	0.069065
EmployeeCount	NaN	NaN
EnvironmentSatisfaction	0.016194	-0.004999
HourlyRate	-0.026716	-0.020123
JobInvolvement	-0.024184	0.025976
JobLevel	0.353885	0.375281
JobSatisfaction	-0.018214	-0.027656
MonthlyIncome	0.344978	0.344079
MonthlyRate	0.001567	-0.036746
NumCompaniesWorked	-0.036814	-0.110319
PercentSalaryHike	-0.022154	-0.011985
PerformanceRating	0.017896	0.022827
RelationshipSatisfaction	0.033493	-0.000867
StandardHours	NaN	NaN

```
StockOptionLevel                        0.014352                 0.024698
TotalWorkingYears                       0.404858                 0.459188
TrainingTimesLastYear                  -0.002067                -0.004096
WorkLifeBalance                         0.008941                 0.002759
YearsAtCompany                          0.618409                 0.769212
YearsInCurrentRole                      0.548056                 0.714365
YearsSinceLastPromotion                 1.000000                 0.510224
YearsWithCurrManager                    0.510224                 1.000000

[25 rows x 25 columns]
```

Problem: Check the relationship between professional experience variables. We will be only interested in the following variables: `YearsAtCompany`, `YearsInCurrentRole`, `YearsSinceLastPromotion`, and `YearsWithCurrManager`

Solution: We will first compute the correlation matrix rounded with 2 digits

[60]:
```
cormat=df[['YearsAtCompany','YearsInCurrentRole','YearsSinceLastPromotion','YearsWithCurrManager
 ↪corr().round(2)
cormat
```

[60]:
```
                         YearsAtCompany  YearsInCurrentRole  \
YearsAtCompany                     1.00                0.76
YearsInCurrentRole                 0.76                1.00
YearsSinceLastPromotion            0.62                0.55
YearsWithCurrManager               0.77                0.71

                         YearsSinceLastPromotion  YearsWithCurrManager
YearsAtCompany                              0.62                  0.77
YearsInCurrentRole                          0.55                  0.71
YearsSinceLastPromotion                     1.00                  0.51
YearsWithCurrManager                        0.51                  1.00
```

It's very common to represent the correlation matrix with a agraph called a **heat map**. To make this visualization we will need to install first `seaborn`

[61]:
```
!pip install seaborn
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: seaborn in
c:\users\dhafe\appdata\roaming\python\python310\site-packages (0.11.2)
Requirement already satisfied: pandas>=0.23 in
c:\users\dhafe\appdata\roaming\python\python310\site-packages (from seaborn)
(1.4.0)
Requirement already satisfied: numpy>=1.15 in
c:\users\dhafe\appdata\roaming\python\python310\site-packages (from seaborn)
(1.22.1)
Requirement already satisfied: matplotlib>=2.2 in
c:\users\dhafe\appdata\roaming\python\python310\site-packages (from seaborn)
```

```
(3.5.1)
Requirement already satisfied: scipy>=1.0 in
c:\users\dhafe\appdata\roaming\python\python310\site-packages (from seaborn)
(1.7.3)
Requirement already satisfied: packaging>=20.0 in
c:\users\dhafe\appdata\roaming\python\python310\site-packages (from
matplotlib>=2.2->seaborn) (21.3)
Requirement already satisfied: python-dateutil>=2.7 in
c:\users\dhafe\appdata\roaming\python\python310\site-packages (from
matplotlib>=2.2->seaborn) (2.8.2)
Requirement already satisfied: kiwisolver>=1.0.1 in
c:\users\dhafe\appdata\roaming\python\python310\site-packages (from
matplotlib>=2.2->seaborn) (1.3.2)
Requirement already satisfied: fonttools>=4.22.0 in
c:\users\dhafe\appdata\roaming\python\python310\site-packages (from
matplotlib>=2.2->seaborn) (4.29.0)
Requirement already satisfied: pillow>=6.2.0 in
c:\users\dhafe\appdata\roaming\python\python310\site-packages (from
matplotlib>=2.2->seaborn) (9.0.0)
Requirement already satisfied: cycler>=0.10 in
c:\users\dhafe\appdata\roaming\python\python310\site-packages (from
matplotlib>=2.2->seaborn) (0.11.0)
Requirement already satisfied: pyparsing>=2.2.1 in
c:\users\dhafe\appdata\roaming\python\python310\site-packages (from
matplotlib>=2.2->seaborn) (3.0.7)
Requirement already satisfied: pytz>=2020.1 in
c:\users\dhafe\appdata\roaming\python\python310\site-packages (from
pandas>=0.23->seaborn) (2021.3)
Requirement already satisfied: six>=1.5 in
c:\users\dhafe\appdata\roaming\python\python310\site-packages (from python-
dateutil>=2.7->matplotlib>=2.2->seaborn) (1.16.0)
```

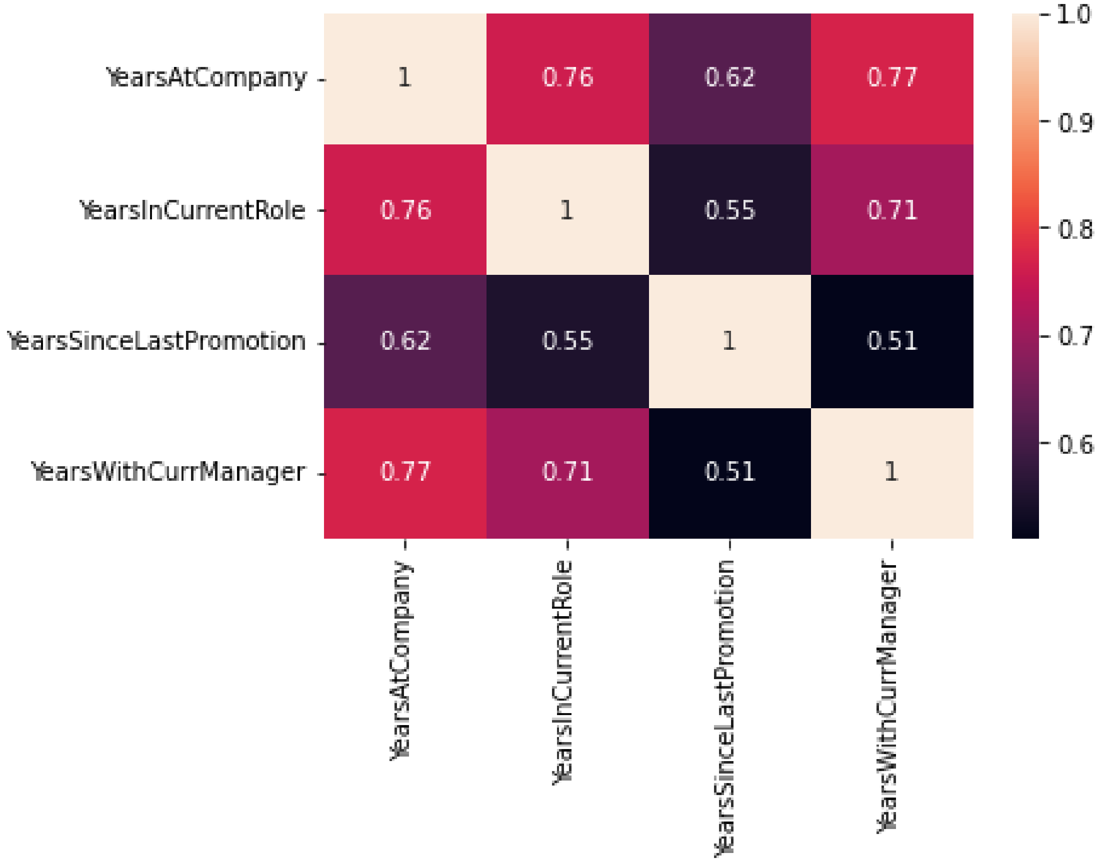
We import then `seaborn` library with the `matplotlib` library

[63]:
```
import seaborn as sns
import matplotlib.pyplot as plt
```

We draw then the correlation matrix `cormat` created above as a heat map

[64]:
```
sns.heatmap(cormat, annot=True)
```

[64]: <AxesSubplot:>



1.2.2 Between two disrete variables

The relationship between two discrete variables is measured using **contingency tables**.

[4]:
```
df.columns
```

[4]:
```
Index(['Age', 'Attrition', 'BusinessTravel', 'DailyRate', 'Department',
       'DistanceFromHome', 'Education', 'EducationField', 'EmployeeCount',
       'EnvironmentSatisfaction', 'Gender', 'HourlyRate', 'JobInvolvement',
       'JobLevel', 'JobRole', 'JobSatisfaction', 'MaritalStatus',
       'MonthlyIncome', 'MonthlyRate', 'NumCompaniesWorked', 'Over18',
       'OverTime', 'PercentSalaryHike', 'PerformanceRating',
       'RelationshipSatisfaction', 'StandardHours', 'StockOptionLevel',
       'TotalWorkingYears', 'TrainingTimesLastYear', 'WorkLifeBalance',
       'YearsAtCompany', 'YearsInCurrentRole', 'YearsSinceLastPromotion',
       'YearsWithCurrManager'],
      dtype='object')
```

[5]:
```
df['Attrition'].value_counts()
```

[5]:
```
No     1233
Yes     237
Name: Attrition, dtype: int64
```

[6]:
```
df['Gender'].value_counts()
```

[6]:
```
Male      882
Female    588
Name: Gender, dtype: int64
```

A **contingency table** is a multi-way table that describes a data set in which each observation belongs to one category for each of several variables. For example, if there are two variables, one with r levels and one with k levels, then we have a contingency table. The table can be described in terms of the number of observations that fall into a given cell of the table, e.g. T_{ij} is the number of observations that have level i for the first variable and level j for the second variable.

The contingency table of the variables `Attrition` and `Gender` can be computed using `crosstab` function from `Pandas` library

[7]:
```
pd.crosstab(df['Attrition'], df['Gender'])
```

[7]:
```
Gender     Female  Male
Attrition
No            501   732
Yes            87   150
```

We can add margins to the contingency table

[78]:
```
pd.crosstab(df['Attrition'], df['Gender'],margins=True)
```

[78]:
```
Gender     Female  Male   All
Attrition
No            501   732  1233
Yes            87   150   237
All           588   882  1470
```

We will now explore the library `statsmodels` that supports a variety of approaches for analyzing contingency tables, including methods for assessing independence

In a probabilistic way, the lack of relationship between two discrete variables can be expressed using two independent variables:

Two random discrete random variables A and B are independent if for all i, j

$$\underbrace{\mathbb{P}(A = i, B = j)}_{P_{ij}} = \underbrace{\mathbb{P}(A = i)}_{P_{i+}} \times \underbrace{\mathbb{P}(B = j)}_{P_{+j}}$$

[10]:
```
!pip install statsmodels
```

```
Defaulting to user installation because normal site-packages is not writeable
Collecting statsmodels
```

```
  Downloading statsmodels-0.13.2-cp310-cp310-win_amd64.whl (9.1 MB)
     ---------------------------------------- 9.1/9.1 MB 30.7 MB/s eta 0:00:00
Requirement already satisfied: scipy>=1.3 in
c:\users\dhafe\appdata\roaming\python\python310\site-packages (from statsmodels)
(1.7.3)
Collecting patsy>=0.5.2
  Downloading patsy-0.5.2-py2.py3-none-any.whl (233 kB)
     ------------------------------------- 233.7/233.7 KB 14.9 MB/s eta 0:00:00
Requirement already satisfied: pandas>=0.25 in
c:\users\dhafe\appdata\roaming\python\python310\site-packages (from statsmodels)
(1.4.0)
Requirement already satisfied: packaging>=21.3 in
c:\users\dhafe\appdata\roaming\python\python310\site-packages (from statsmodels)
(21.3)
Requirement already satisfied: numpy>=1.17 in
c:\users\dhafe\appdata\roaming\python\python310\site-packages (from statsmodels)
(1.22.1)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in
c:\users\dhafe\appdata\roaming\python\python310\site-packages (from
packaging>=21.3->statsmodels) (3.0.7)
Requirement already satisfied: pytz>=2020.1 in
c:\users\dhafe\appdata\roaming\python\python310\site-packages (from
pandas>=0.25->statsmodels) (2021.3)
Requirement already satisfied: python-dateutil>=2.8.1 in
c:\users\dhafe\appdata\roaming\python\python310\site-packages (from
pandas>=0.25->statsmodels) (2.8.2)
Requirement already satisfied: six in
c:\users\dhafe\appdata\roaming\python\python310\site-packages (from
patsy>=0.5.2->statsmodels) (1.16.0)
Installing collected packages: patsy, statsmodels
Successfully installed patsy-0.5.2 statsmodels-0.13.2
```

We import then the necessary libraries

[11]:
```
 import numpy as np
```

[12]:
```
import pandas as pd
```

[13]:
```
import statsmodels.api as sm
```

[14]:
```
tab = pd.crosstab(df['Attrition'], df['Gender'])
```

[15]:
```
tab
```

[15]:

Gender	Female	Male
Attrition		
No	501	732
Yes	87	150

[21]:
```
tab.loc[:, ['Female'] ]
```

[21]:
```
Gender     Female
Attrition
No            501
Yes            87
```

[25]:
```
table = sm.stats.Table(tab)
```

[27]:
```
print(table)
```

```
A 2x2 contingency table with counts:
[[501. 732.]
 [ 87. 150.]]
```

[28]:
```
table.table
```

[28]:
```
array([[501., 732.],
       [ 87., 150.]])
```

[29]:
```
table.fittedvalues
```

[29]:
```
Gender     Female   Male
Attrition
No          493.2  739.8
Yes          94.8  142.2
```

[39]:
```
tab.loc[['No'],['Female']]
```

[39]:
```
Gender     Female
Attrition
No            501
```

Estimating marginal probabilities of the variable gender

[40]:
```
pG=df['Gender'].value_counts(normalize=True)
```

[41]:
```
pG
```

[41]:
```
Male      0.6
Female    0.4
Name: Gender, dtype: float64
```

[42]:
```
pG[0]
```

[42]:
```
0.6
```

[43]:
```
pG[1]
```

[43]: 0.4

[44]:
```
pA=df['Attrition'].value_counts(normalize=True)
```

[45]:
```
pA
```

[45]:
```
No     0.838776
Yes    0.161224
Name: Attrition, dtype: float64
```

[48]:
```
n=df.shape[0]
```

[49]:
```
n
```

[49]: 1470

Computing the fitting contingency table

[50]:
```
tabhat=n*np.array([[pA[0]*pG[0],pA[0]*pG[1]],
              [pA[1]*pG[0],pA[1]*pG[1]]])
```

[51]:
```
print(tabhat)
```

```
[[739.8 493.2]
 [142.2  94.8]]
```

[53]:
```
tabhat.transpose()
```

[53]:
```
array([[739.8, 142.2],
       [493.2,  94.8]])
```

[54]:
```
table.fittedvalues
```

[54]:
```
Gender     Female   Male
Attrition
No          493.2  739.8
Yes          94.8  142.2
```

The coefficients in the previous table are called the **expected value**: E_{ij} is the expected value for the cell in the i^{th} column and j^{th} row. E_{ij} can be computed as follows:

$$E_{ij} = \frac{T_{i+} \times T_{+j}}{N}$$

where $T_{i+} = \sum_j T_{ij}$, $T_{+j} = \sum_i T_{ij}$, and N is the sample size.

We compute then the **Pearson residuals**:

$$r_{ij} = \frac{T_{ij} - E_{ij}}{\sqrt{E_{ij}}}$$

When the variables are independents, the pearson residuals are expected to be close to zero and with a modulus non higher than 2.

[55]:
```
table.resid_pearson
```

[55]:
```
Gender        Female      Male
Attrition
No          0.351223 -0.286772
Yes        -0.801107  0.654101
```

To decide about the relationship of the variables (independence or no), we compute the χ^2 statistics and the corresponding *p-value*.

The variables `Attrition` and `Gender` are both nominal variables, we consider the test measuring the association between *nominal* variables

[61]:
```
rslt = table.test_nominal_association()
```

The χ^2-statistics of the test

[62]:
```
print(rslt.statistic)
```

```
1.2752163602205142
```

It's given by $\sum_{ij} r_{ij}^2$

[64]:
```
0.35122**2+(-0.286772)**2+(-0.801107)**2+0.654101**2
```

[64]:
```
1.2752142120340002
```

The corresponding degree of freedoms: It's equal $((T_{i+} - 1) \times (T_{+j} - 1))$

[65]:
```
print(rslt.df)
```

```
1
```

We consider the *p-value*

[59]:
```
print(rslt.pvalue)
```

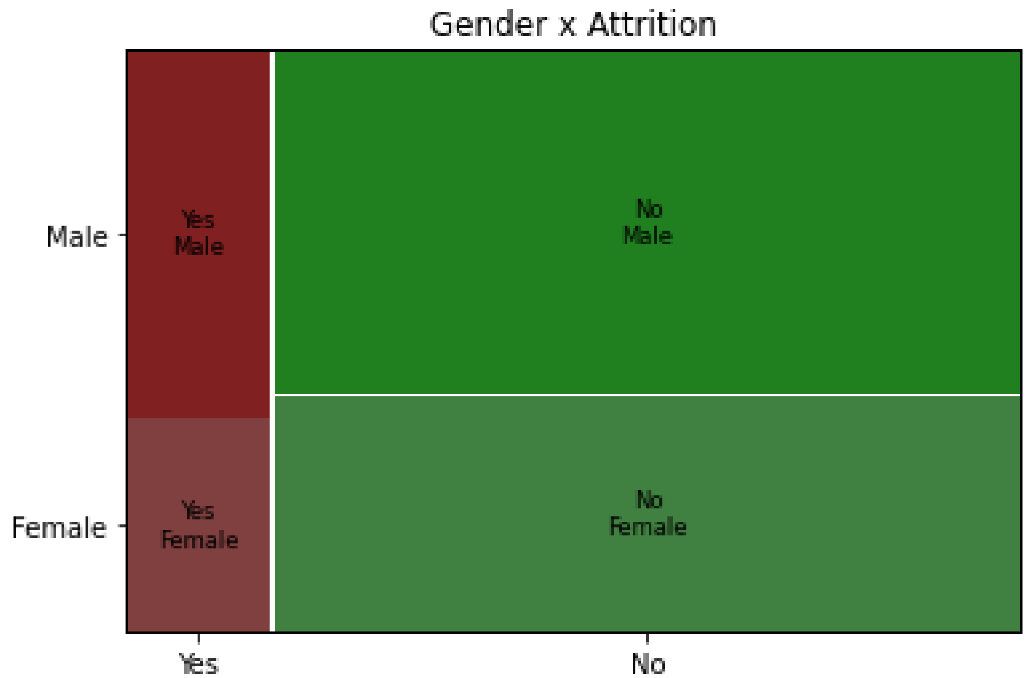
[59]:
```
0.3355102040816326
```

Compared to 0.05, the *p-value* is higher than 0.05, we can decide that there's no relationship between the variables `Attrition` and `Gender`.

To finish this analysis we show draw the `mosaic` plot implemented in `statsmodels` library

[69]:
```
import matplotlib.pyplot as plt
from statsmodels.graphics.mosaicplot import mosaic
```

[74]:
```
mosaic(df, ['Attrition', 'Gender'], title=' Gender x Attrition ')
plt.show()
```


Gender x Attrition
Male
Female
Yes
Male
No
Male
Yes
Female
No
Female
Yes
No

```
[ ]:
```