

# 20230419\_Wirtschaftsinformatik\_MV2

April 19, 2023

```
[1]: # Case Study WI MV2
```

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

```
mv = pd.read_csv('/Users/h4/Desktop/20230419_Case_Study_Wirtschaftsinformatik.  
↪csv', header = 0)
```

```
mv.head()
```

```
[5]:
```

	Age	Attrition	BusinessTravel	DailyRate	Department	\
0	41	Yes	Travel_Rarely	1102		Sales
1	49	No	Travel_Frequently	279	Research & Development	
2	37	Yes	Travel_Rarely	1373	Research & Development	
3	33	No	Travel_Frequently	1392	Research & Development	
4	27	No	Travel_Rarely	591	Research & Development	

	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	\
0		1	2 Life Sciences		1	1
1		8	1 Life Sciences		1	2
2		2	2 Other		1	4
3		3	4 Life Sciences		1	5
4		2	1 Medical		1	7

	...	RelationshipSatisfaction	StandardHours	StockOptionLevel	\
0	...		1	80	0
1	...		4	80	1
2	...		2	80	0
3	...		3	80	0
4	...		4	80	1

	TotalWorkingYears	TrainingTimesLastYear	WorkLifeBalance	YearsAtCompany	\
0		8	0	1	6
1		10	3	3	10
2		7	3	3	0
3		8	3	3	8
4		6	3	3	2

	YearsInCurrentRole	YearsSinceLastPromotion	YearsWithCurrManager
0	4	0	5
1	7	1	7
2	0	0	0
3	7	3	0
4	2	2	2

[5 rows x 35 columns]

```
[7]: import numpy as np

np.shape(mv) # die form des Datasets wird geliefert # 1470 Zeilen und 35 Spalten
```

[7]: (1470, 35)

```
[8]: # plot in Pannels (2x2) also 4 Plots in einer Graphik

import matplotlib.pyplot as plt

import seaborn as sns

fig, ax = plt.subplots(2,2, figsize=(10,10))

# wir definieren 2 Variabeln "fig" und "ax": "fig" ist das Bild, "ax" ist ein
↳ index für den Pannel

plt.suptitle('Darstellung von wichtigen Merkmalen', fontsize = 20)
# Superior Title ist das Haupttitel des Bildes mit Schriftgröße 20

sns.distplot(mv['Age'], ax = ax[0,0])
ax[0][0].set_title('Altersverteilung')
# das erste Bild auf der Position 0,0 (oben links)

sns.distplot(mv['TotalWorkingYears'], ax = ax[0,1])
ax[0][1].set_title('Verteilung der Arbeitnehmerdauer')
# das erste Bild auf der Position 0,1 (oben rechts)

sns.distplot(mv['YearsAtCompany'], ax = ax[1,0])
ax[1][0].set_title('Verteilung der Zugehörigkeit')
# das erste Bild auf der Position 1,0 (unten links)

sns.distplot(mv['YearsInCurrentRole'], ax = ax[1,1])
ax[1][1].set_title('Verteilung der Jahre in der Rolle')
# das erste Bild auf der Position 1,1 (unten rechts)

plt.show()
```

```
/Users/h4/anaconda3/lib/python3.9/site-packages/seaborn/distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
```

```
warnings.warn(msg, FutureWarning)
```

```
/Users/h4/anaconda3/lib/python3.9/site-packages/seaborn/distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
```

```
warnings.warn(msg, FutureWarning)
```

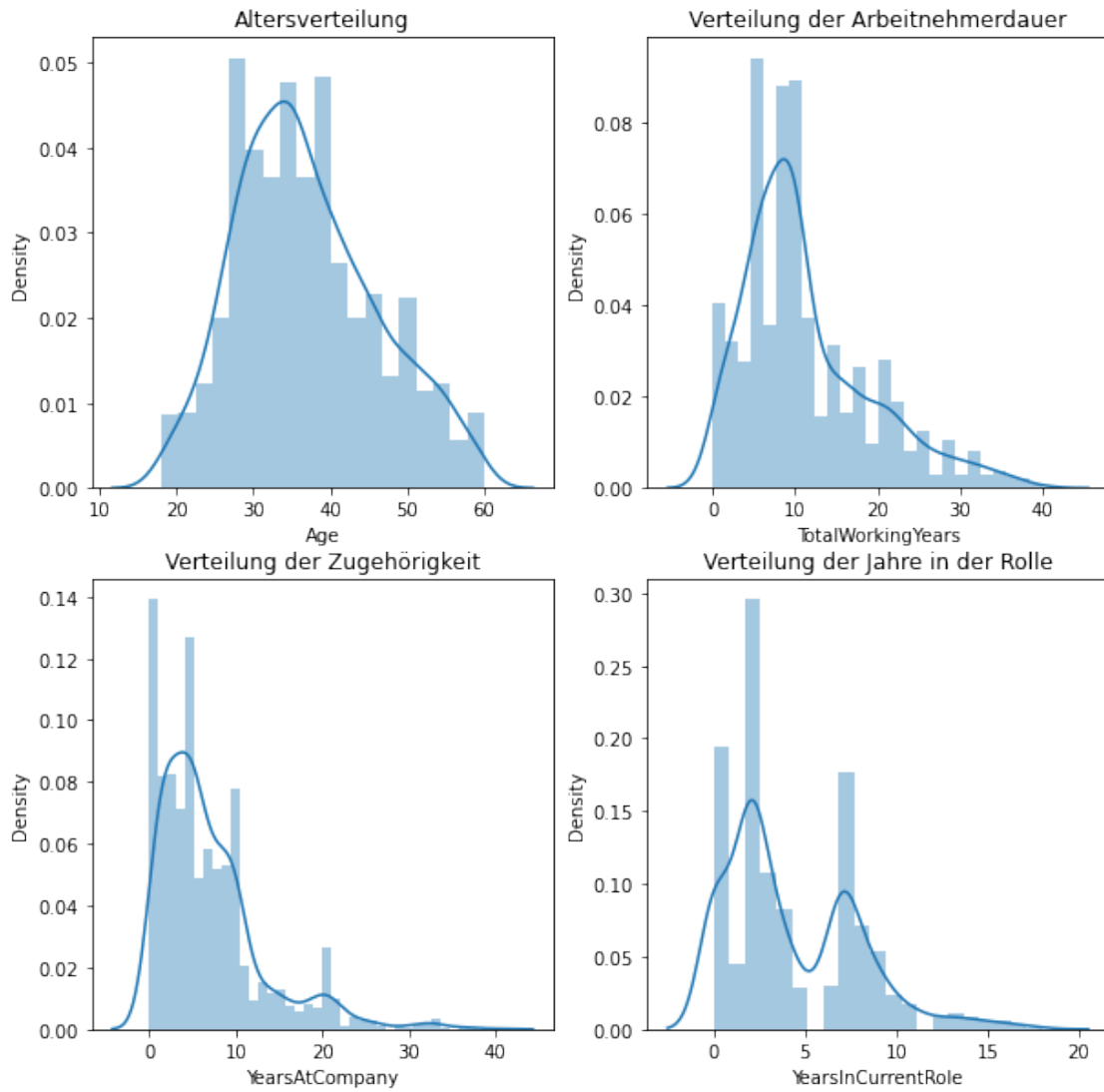
```
/Users/h4/anaconda3/lib/python3.9/site-packages/seaborn/distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
```

```
warnings.warn(msg, FutureWarning)
```

```
/Users/h4/anaconda3/lib/python3.9/site-packages/seaborn/distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
```

```
warnings.warn(msg, FutureWarning)
```

## Darstellung von wichtigen Merkmalen



0.0.1 die meisten Mitarbeiter sind zw. 35 und 40 Jahre alt.

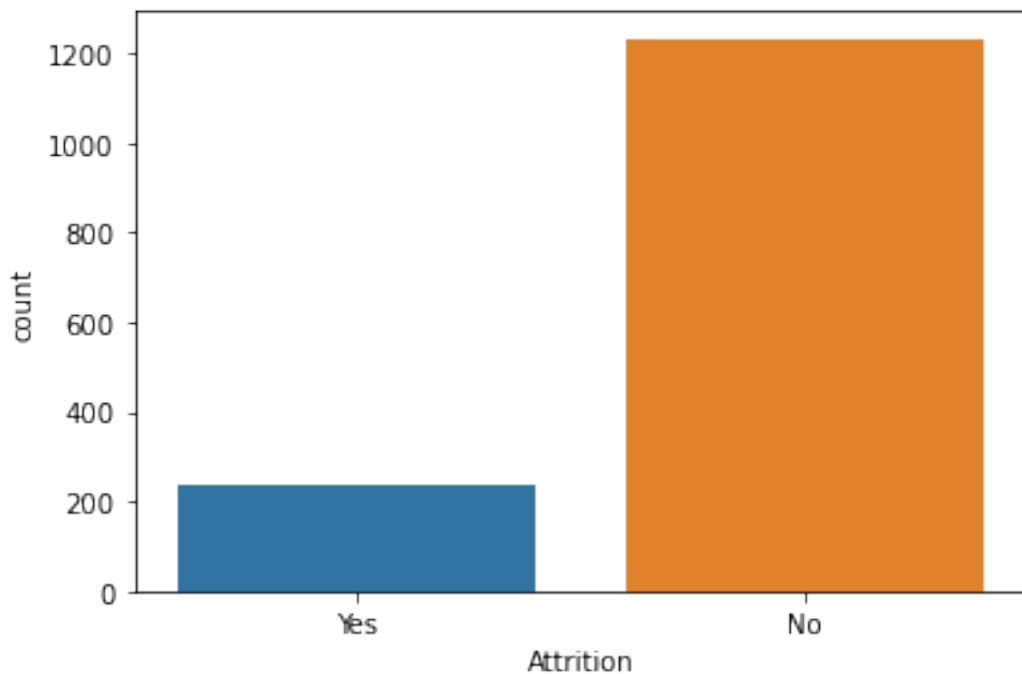
0.0.2 mittlere Arbeitsnehmerdauer ist 8 bis 10 Jahre.

0.0.3 idR bleiben die Mitarbeiter 2 bis 4 Jahre im Unternehmen

0.0.4 idR bleiben die Mitarbeiter 2 bis 3 Jahre in der Rolle

```
[13]: import warnings
      warnings.filterwarnings('ignore')
```

```
[14]: sns.countplot(mv['Attrition'])
      plt.show()
```



0.0.5 ca. 250 Mitarbeiter haben die Firma verlassen.

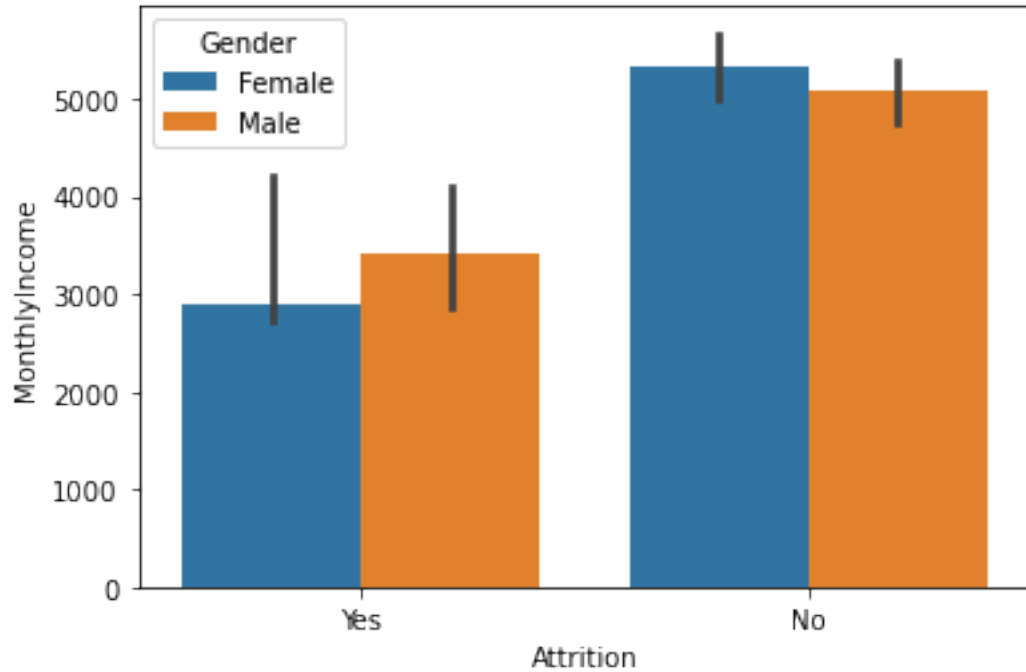
0.0.6 die Firma konnte ca. 1200 Mitarbeiter behalten

```
[11]: # Darstellung der Mitarbeiter "Attrition" abhängig von dem "Einkommen" und nach
      ↪ "Geschlecht"

      from numpy import median
```

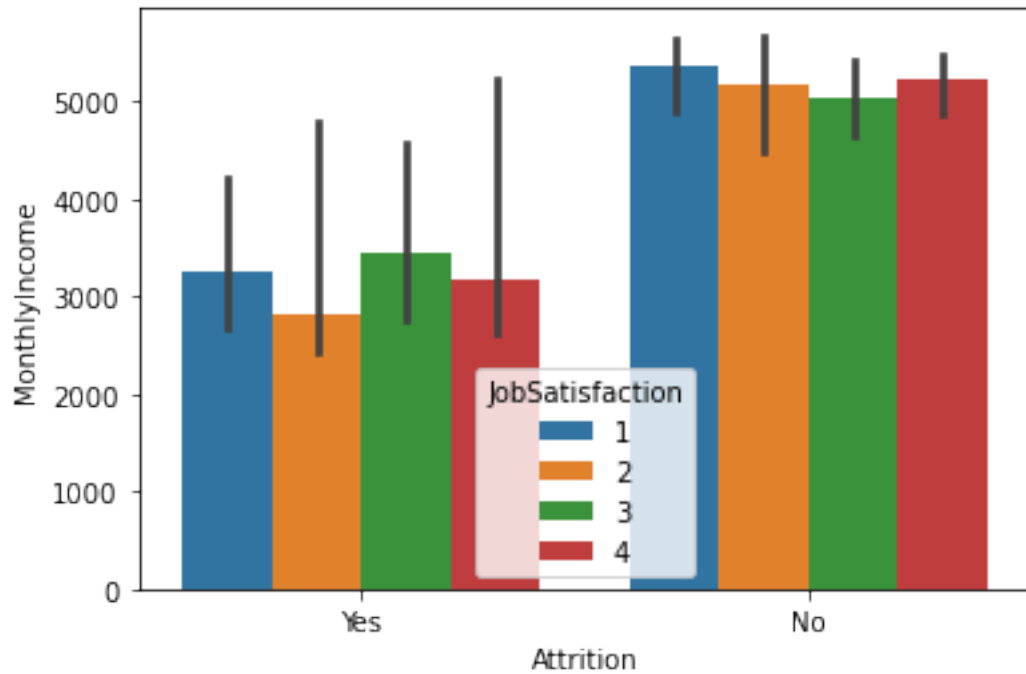
```
sns.barplot(x = 'Attrition', y = 'MonthlyIncome', hue = 'Gender', data = mv,
            estimator = median)
```

[11]: <AxesSubplot:xlabel='Attrition', ylabel='MonthlyIncome'>



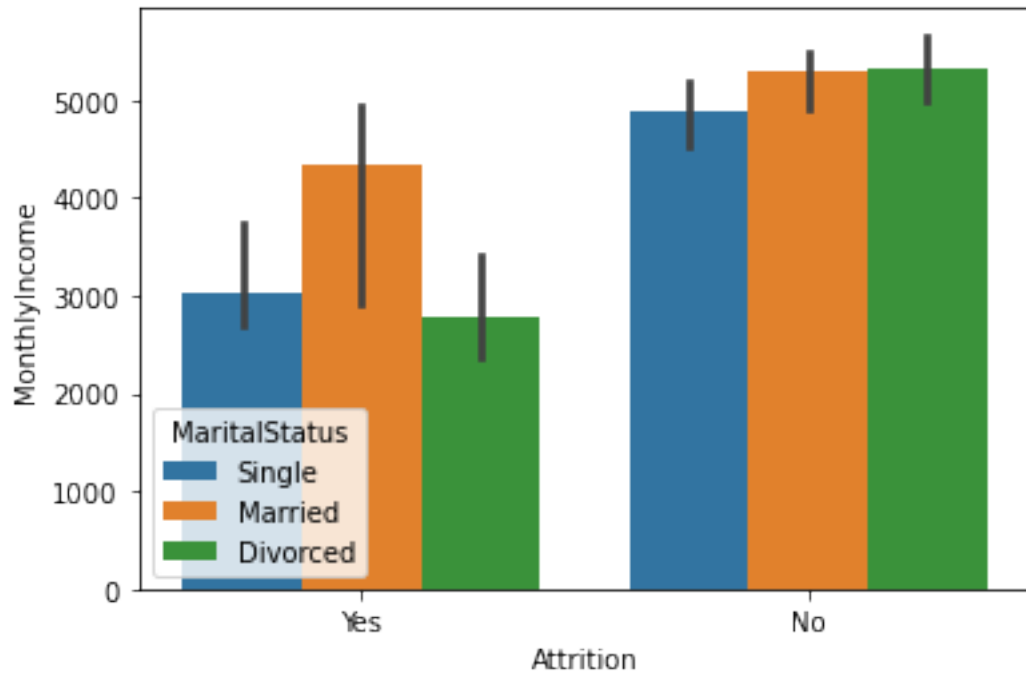
[15]: *# Miterbaiter Attrition abhängig von Monatseinkommen und Job Satisfaction*

```
sns.barplot(x = 'Attrition', y = 'MonthlyIncome', hue = 'JobSatisfaction', data = mv,
            estimator = median)
plt.show()
```



```
[16]: # Mitarbeiter Attrition abhängig von Monatseinkommen und "Marital Status"

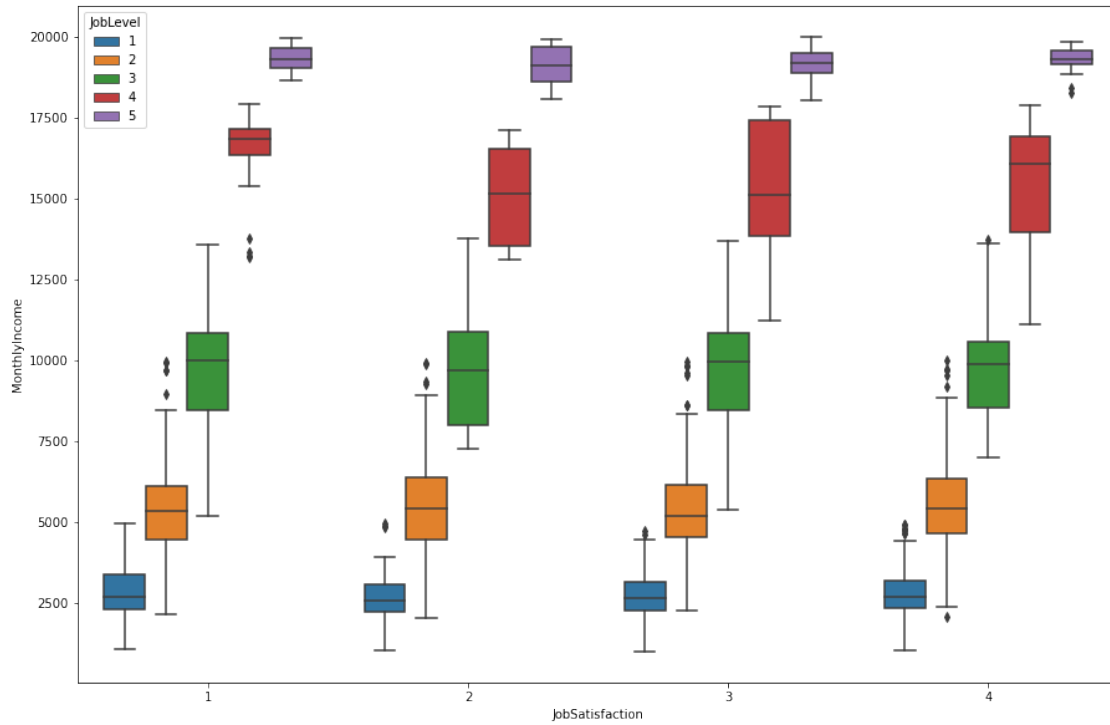
sns.barplot( x = 'Attrition', y = 'MonthlyIncome', hue = 'MaritalStatus', data_
    ↪= mv, estimator = median)
plt.show()
```



[18]: *# Wechselwirkung zw. JobSatisfaction und MonthlyIncome*

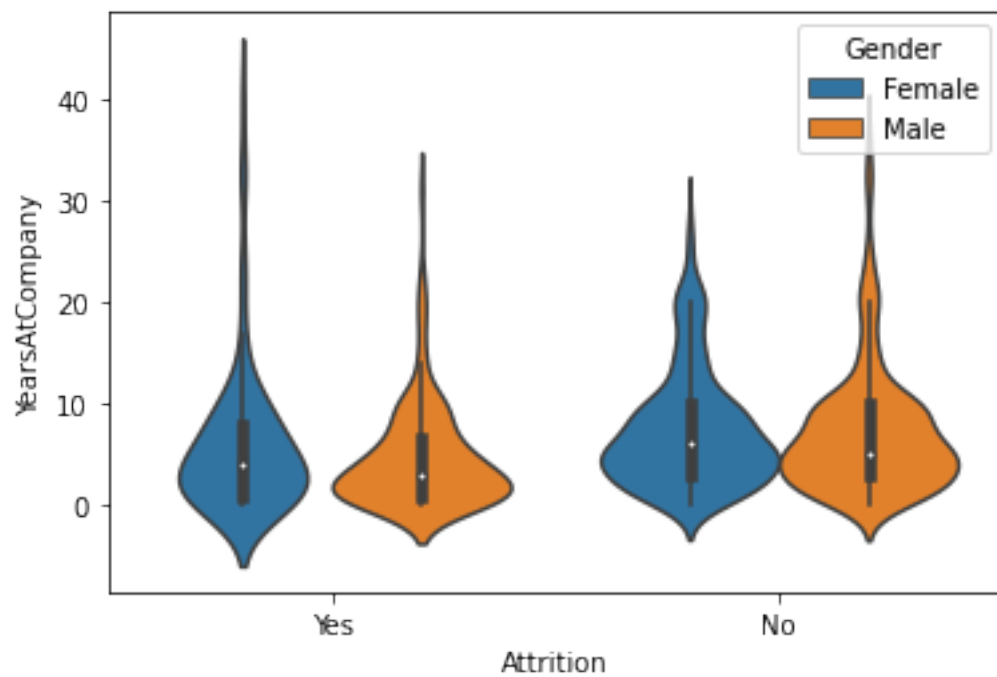
```
f, ax = plt.subplots(figsize=(15,10))
sns.boxplot(x = 'JobSatisfaction', y = 'MonthlyIncome', data = mv, hue = 'JobLevel')
plt.show()
```



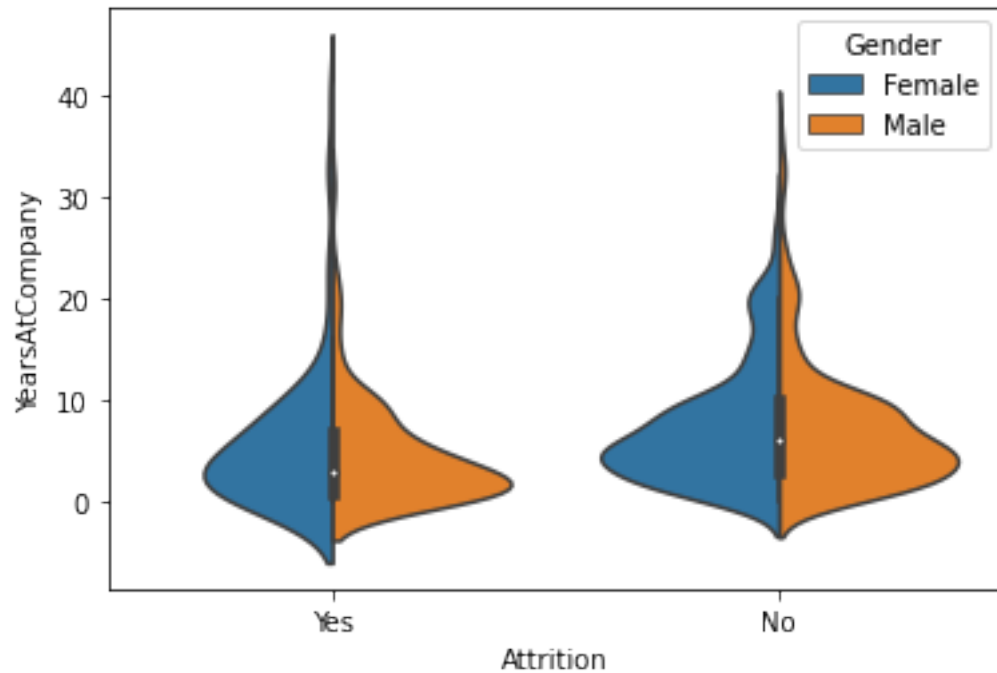


[20]: *# Wechselwirkung zw. Attrition und Jahre in der Firma und nach Geschlecht*

```
sns.violinplot(x = 'Attrition', y = 'YearsAtCompany', hue = 'Gender', data = mv)
plt.show()
```

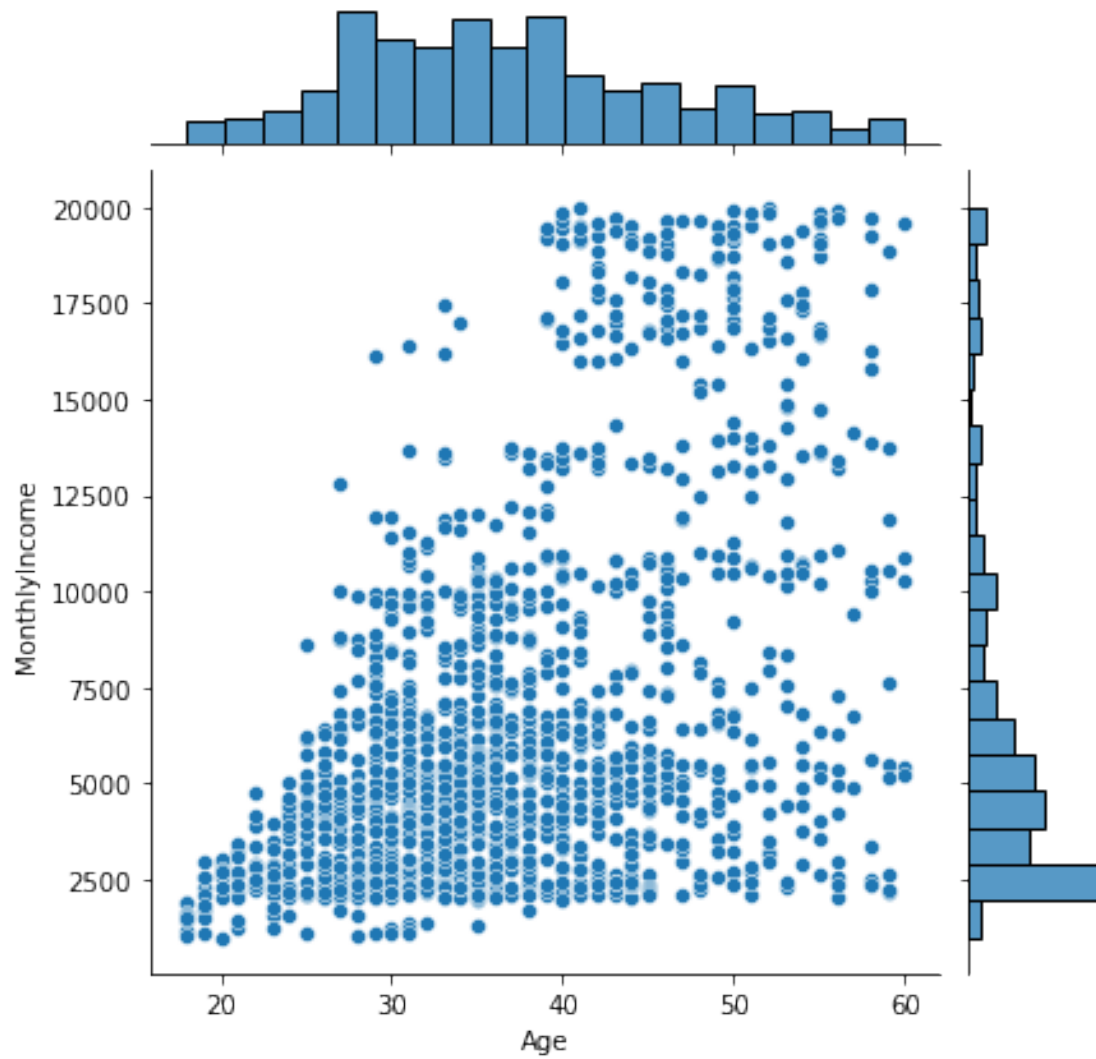


```
[21]: sns.violinplot(x = 'Attrition', y = 'YearsAtCompany', hue = 'Gender', data = mv,
    ↳mv, split = True)
plt.show()
```



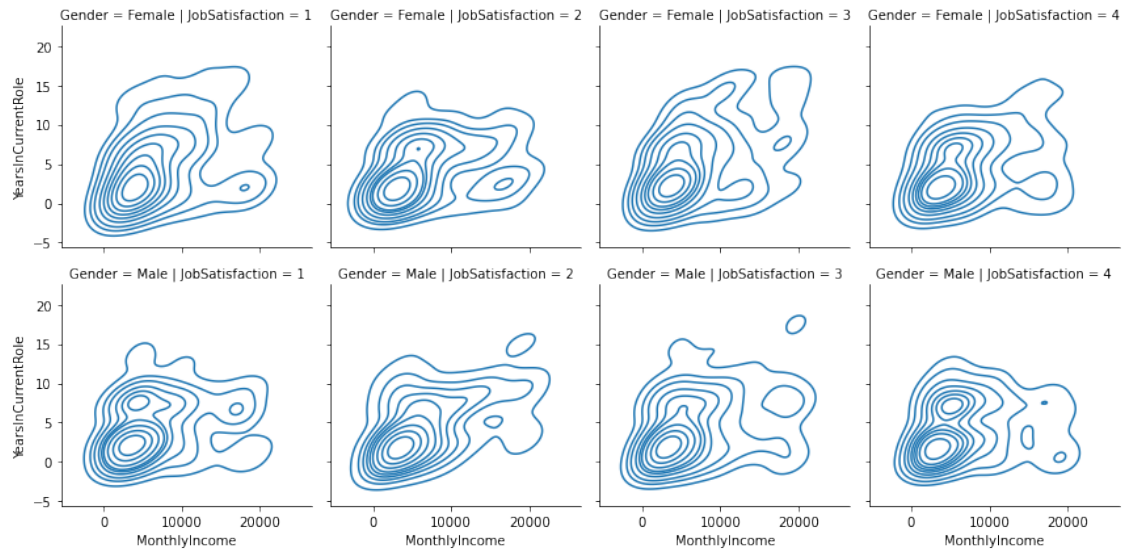
```
[23]: # wie viel Geld verdient man, abhängig vom Alter?

sns.jointplot(mv.Age, mv.MonthlyIncome, kind = 'scatter') # für Jointplot
    ↳brauche ich numerische Variablen
plt.show()
```



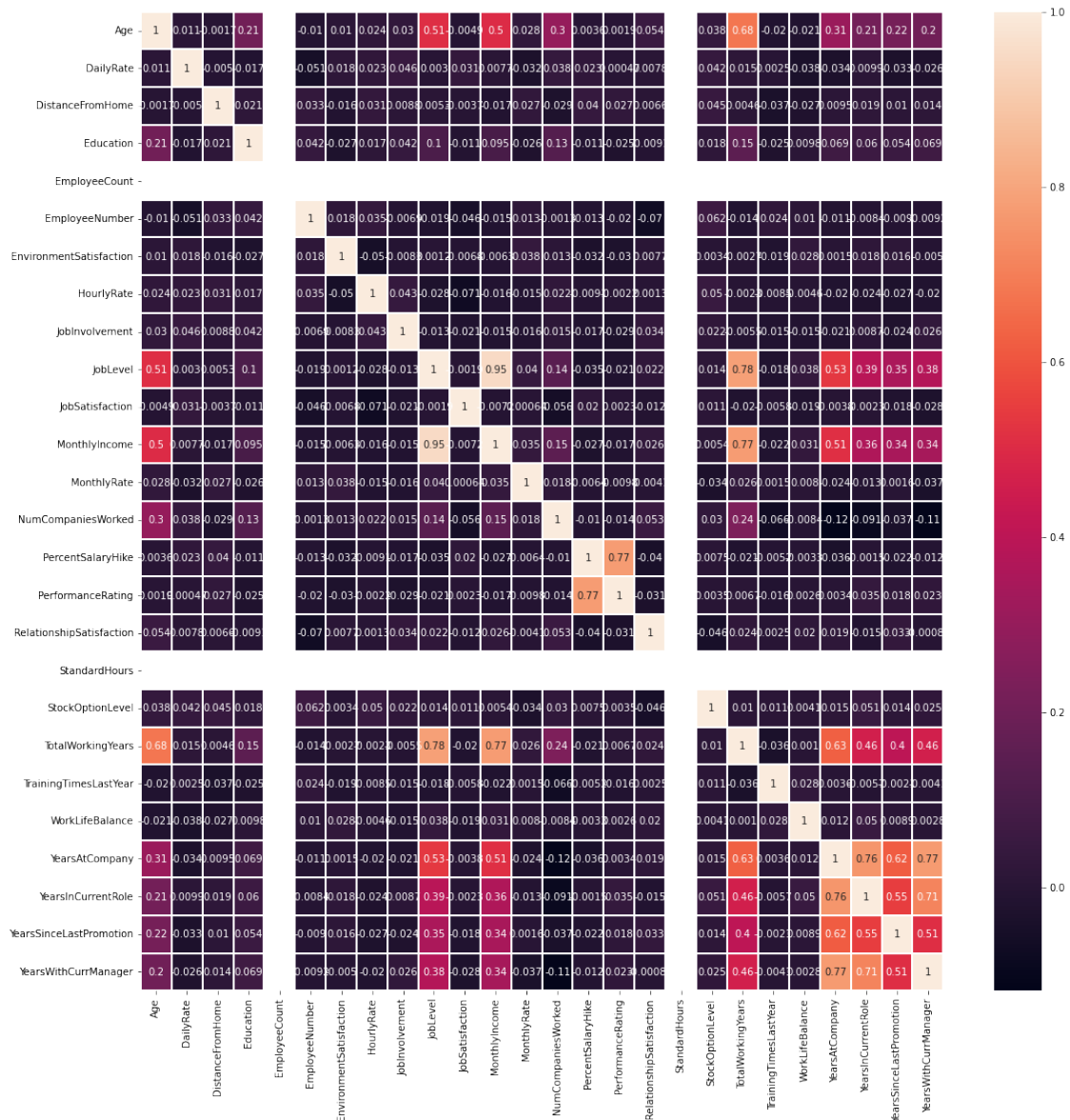
[24]: *# Jobsatisfaction und Gender in mehreren Graphen*

```
g = sns.FacetGrid(mv, col='JobSatisfaction', row='Gender')
g.map(sns.kdeplot, 'MonthlyIncome', 'YearsInCurrentRole')
plt.show()
```



[25]: *# Korrelationsmap für ALLE numerische Variablen*

```
f, ax = plt.subplots(figsize=(18,18))
sns.heatmap(mv.corr(), annot=True, linewidths=.4, ax=ax)
plt.show()
```



[ ]: