

# 20230420\_Wirtschaftsinformatik\_MPW2

April 20, 2023

## 0.1 Case Study HR Analytics

```
[1]: # Daten - Wissen - Wertschöpfung
```

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

```
[3]: # download bei MAC 'option'+Raw
# download bei Microsoft 'Alt'+Raw

# \ sollte zu /

hr = pd.read_csv('/Users/h4/Desktop/20230420_Case_Study_Wirtschaftsinformatik.
↳csv', header = 0)
hr.head()
```

```
[3]:
```

	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		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]

```
[4]: import numpy as np
      np.shape(hr)
```

[4]: (1470, 35)

```
[6]: # Graphische Darstellung von unterschiedliche Verteilungen

import seaborn as sns
import matplotlib.pyplot as plt

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

plt.suptitle('Unterschiedliche Datenverteilungen', fontsize = 15)

sns.distplot(hr['Age'], ax = ax[0,0])
# plot ist oben links
ax[0][0].set_title('Altersverteilung')

sns.distplot(hr['TotalWorkingYears'], ax = ax[0,1])
# plot ist oben rechts
ax[0][1].set_title('Berufserfahrung')

sns.distplot(hr['YearsAtCompany'], ax = ax[1,0])
# plot ist unten links
ax[1][0].set_title('Jahre in der Firma')

sns.distplot(hr['YearsInCurrentRole'], ax = ax[1,1])
# plot ist unten rechts
ax[1][1].set_title('Jahre in der Rolle')

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)
```

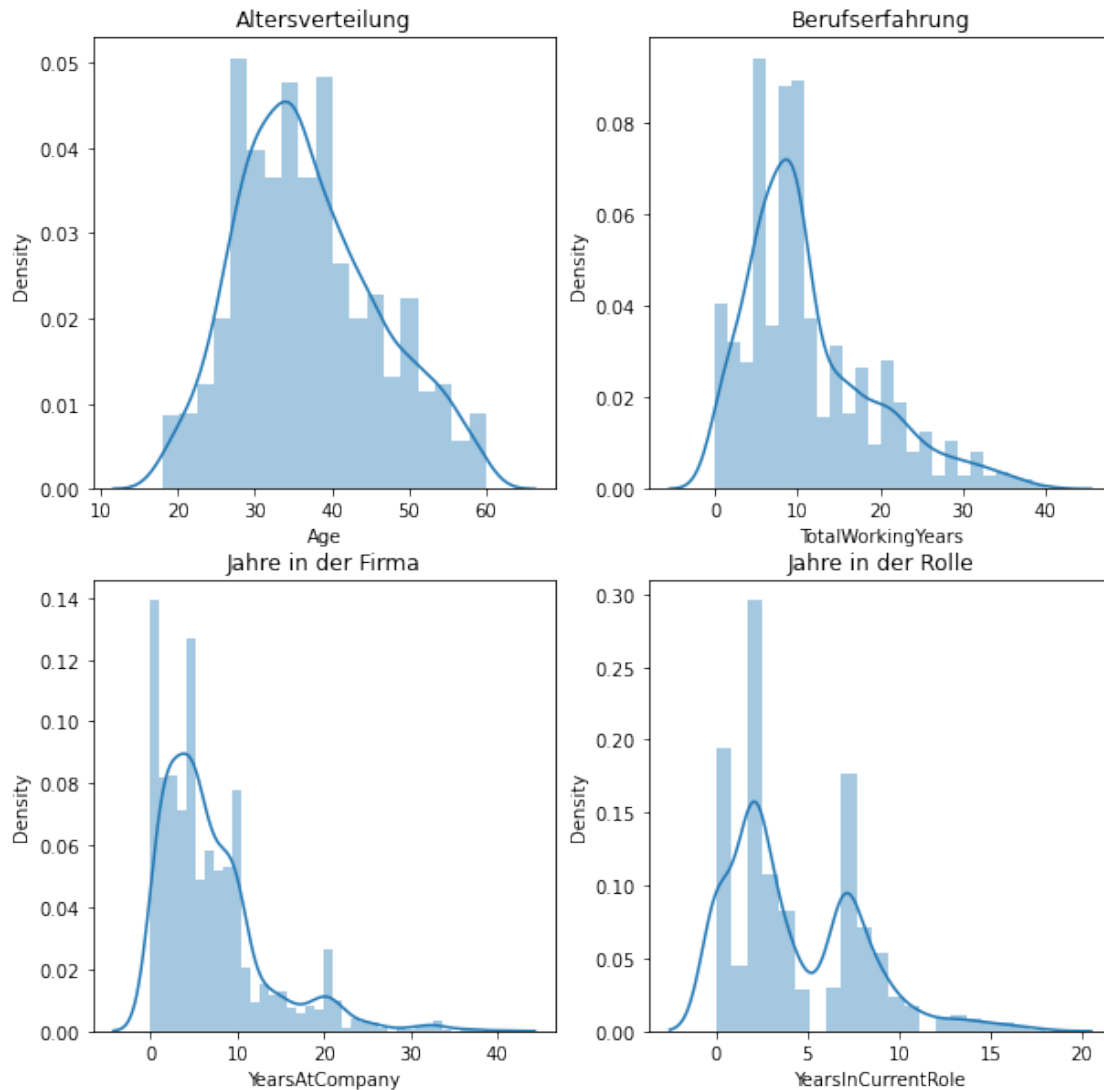
```
/Users/h4/anaconda3/lib/python3.9/site-packages/seaborn/distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level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

## Unterschiedliche Datenverteilungen



**0.1.1** die meisten Mitarbeiter sind zw 35 und 40 Jahre alt, die meisten haben ca. 10 Jahre Berufserfahrung, ...

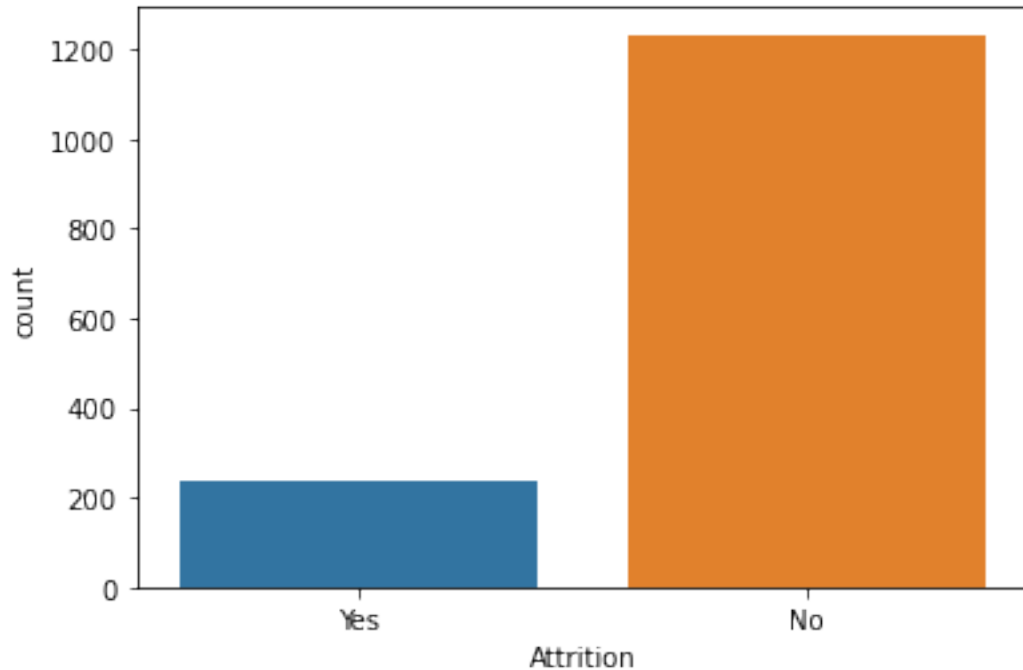
```
[7]: # wie viele Mitarbeiter haben die Fa. verlassen
```

```
[8]: sns.countplot(hr['Attrition'])  
plt.show()
```

/Users/h4/anaconda3/lib/python3.9/site-packages/seaborn/\_decorators.py:36:  
FutureWarning: Pass the following variable as a keyword arg: x. From version

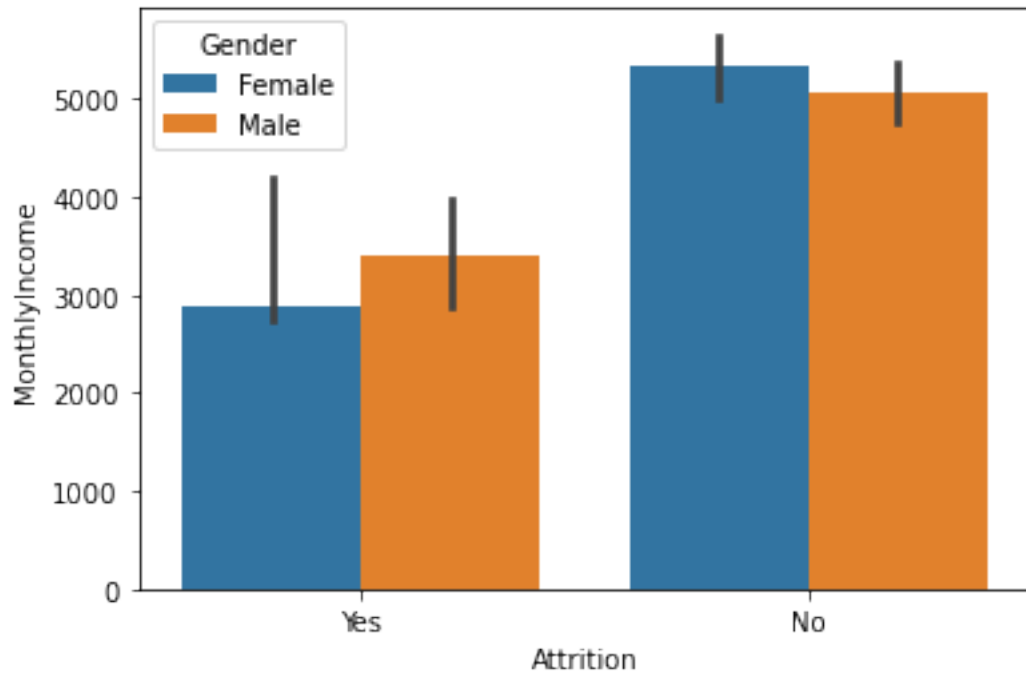
0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

```
warnings.warn(
```



0.1.2 ca. 250 Mitarbeiter von den 1470 haben die Fa. verlassen

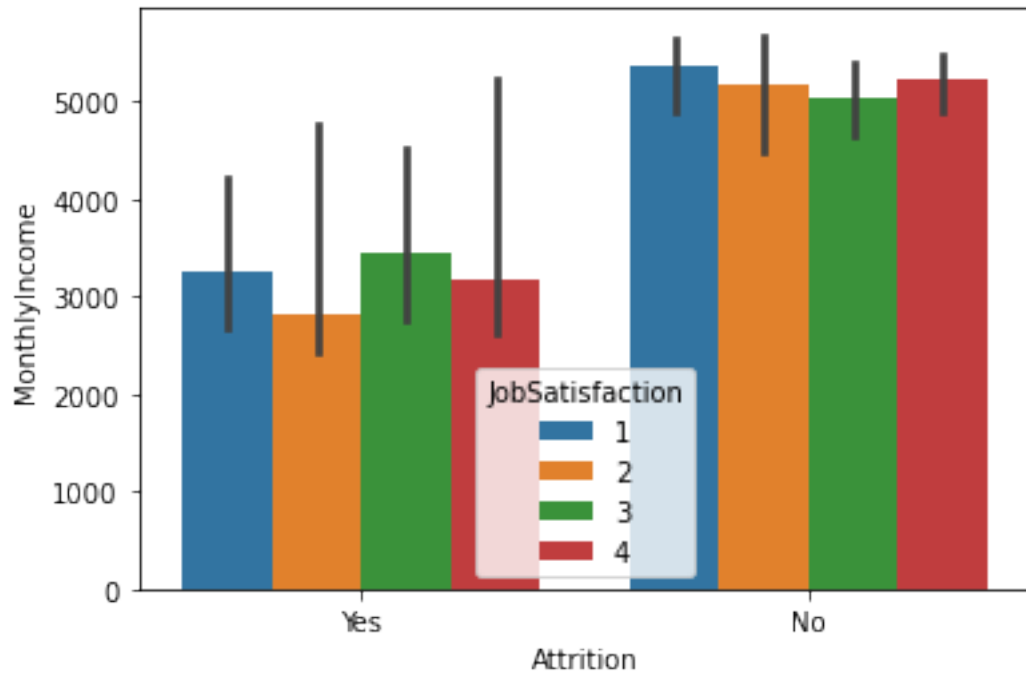
```
[12]: # barplot von den Monatlichen Einkommen und Attrition nach Geschlecht. der ↵  
      ↪ Estimator ist median  
  
from numpy import median  
  
sns.barplot(x = 'Attrition', y = 'MonthlyIncome', hue = 'Gender', data = hr, ↵  
            ↪ estimator = median)  
plt.show()
```



```
[14]: # barplot von Jobsatisfaction abhängig von den monatlichen Einkommen und
      ↳ Geschlecht
      # estimator median

      from numpy import median

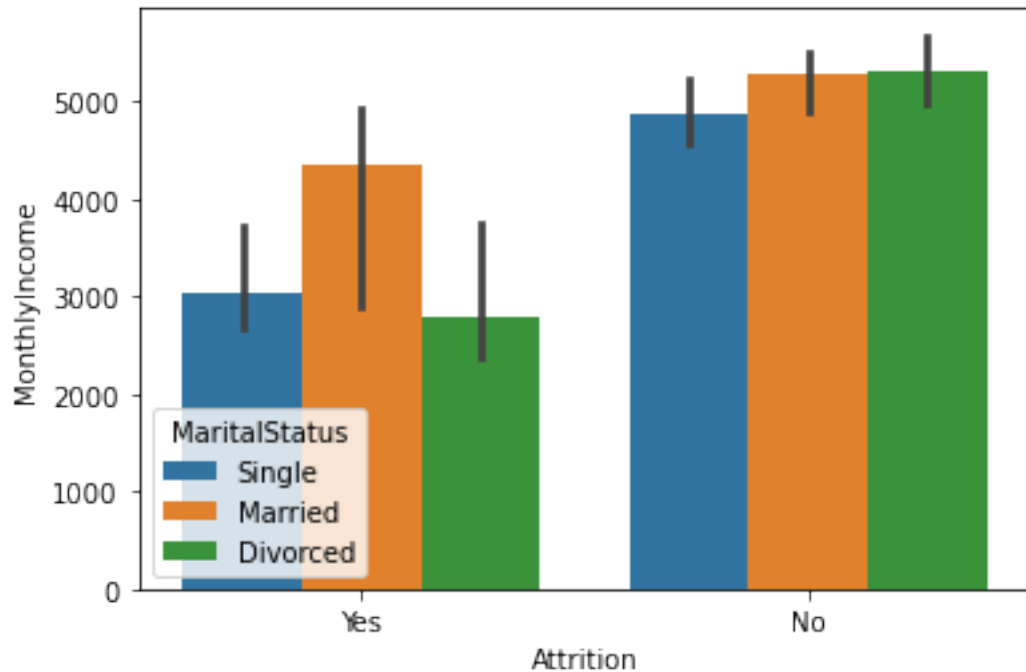
      sns.barplot(x = 'Attrition', y = 'MonthlyIncome', hue = 'JobSatisfaction', data_
      ↳ hr, estimator = median)
      plt.show()
```



```
[15]: # was hat für ein Effekt ob die Menschen Verheiratet, Geschieden oder Solo sind,
      ↪ für die Attrition?

from numpy import median

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



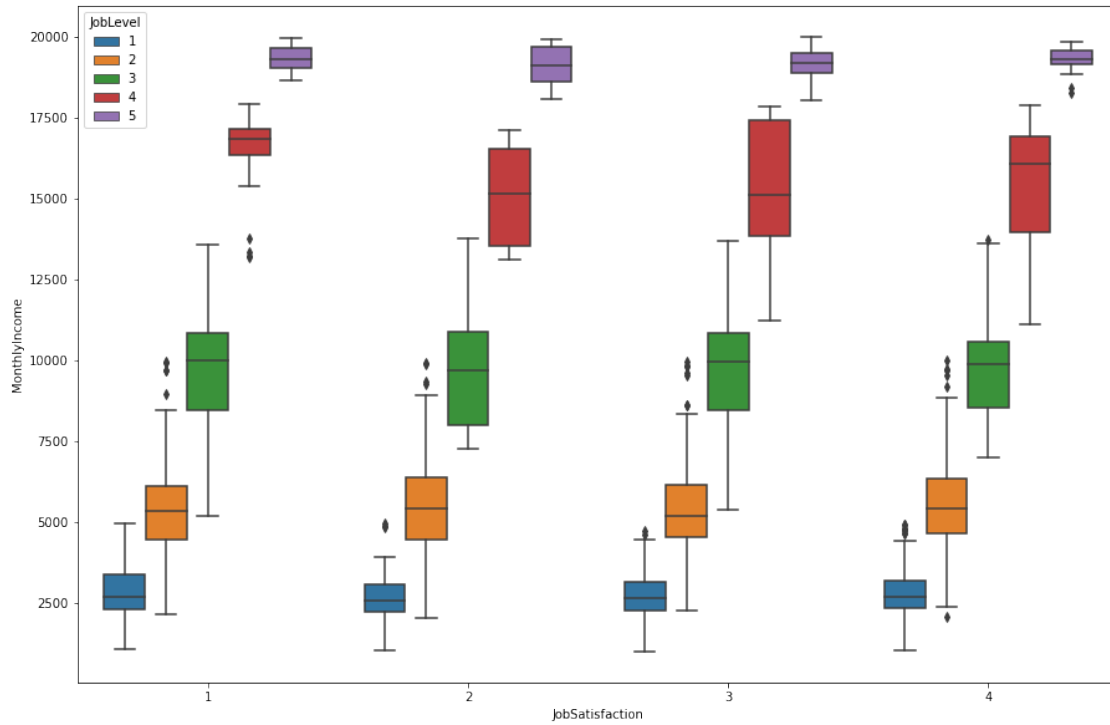
0.1.3 wir können feststellen, dass Menschen mit weniger als 3000 monthly Income,

0.1.4 werden die Firma höchstwahrscheinlich mehr verlassen als andere mit über 5000 income.

```
[16]: # wir können die Job Satisfaction und den monatlichen Einkommen gemeinsam
      ↪ darstellen
```

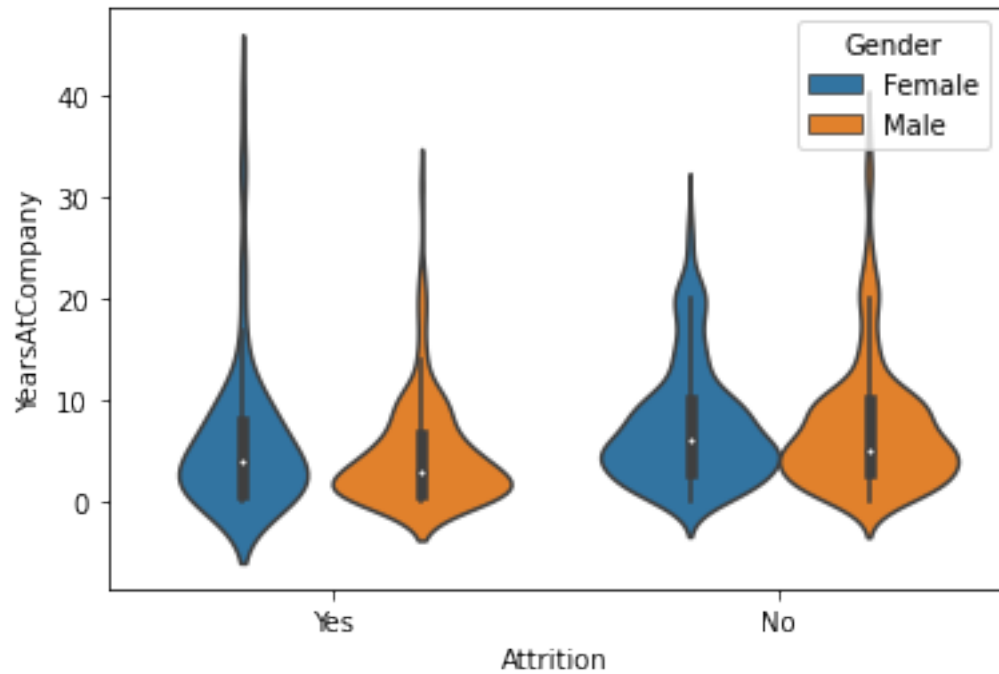
```
[17]: f, ax = plt.subplots(figsize=(15,10))
      sns.boxplot(x = 'JobSatisfaction', y = 'MonthlyIncome', data = hr, hue=
      ↪ 'JobLevel')
      plt.show()
```



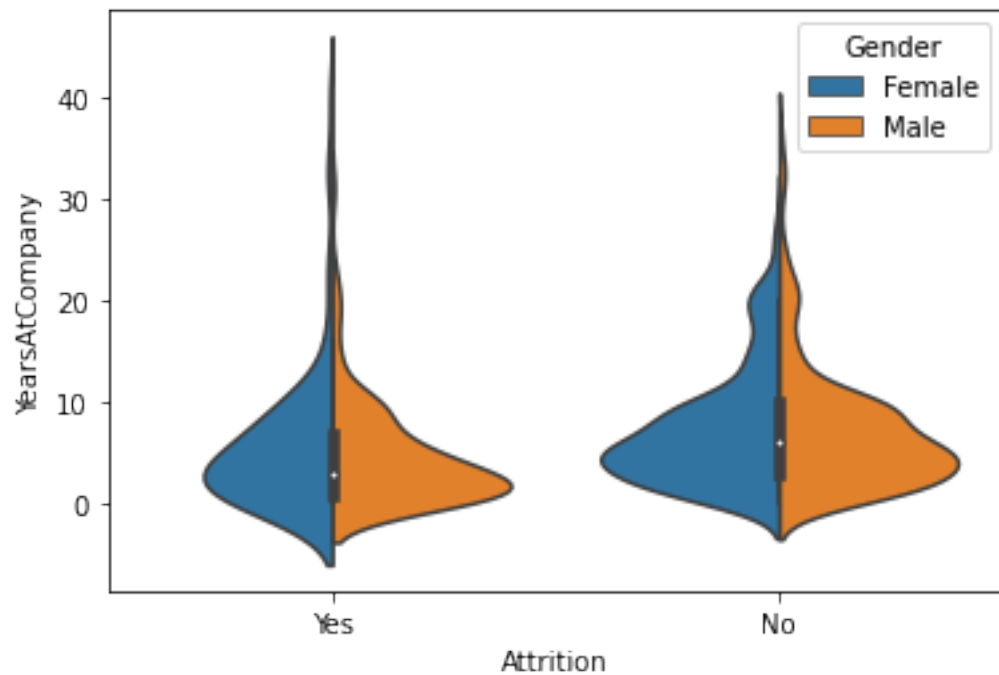


```
[18]: # wir können auch die "Attrition" als funktion der Zugehörigkeitsdauer und
      ↳ Geschlecht
```

```
[19]: sns.violinplot(x = 'Attrition', y = 'YearsAtCompany', hue = 'Gender', data = hr)
      plt.show()
```

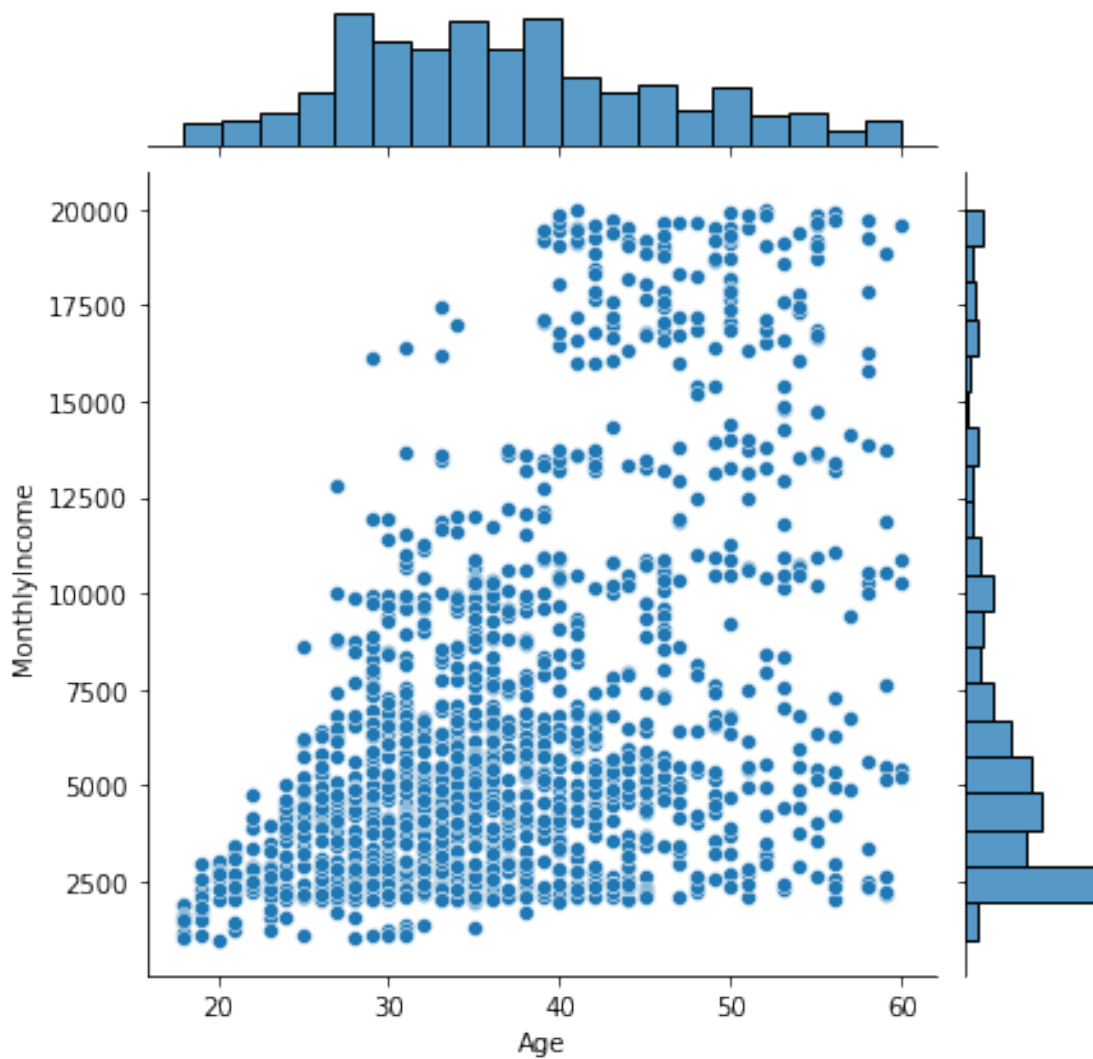


```
[20]: sns.violinplot(x = 'Attrition', y = 'YearsAtCompany', hue = 'Gender', data = hr,
    ↳hr, split = True)
plt.show()
```

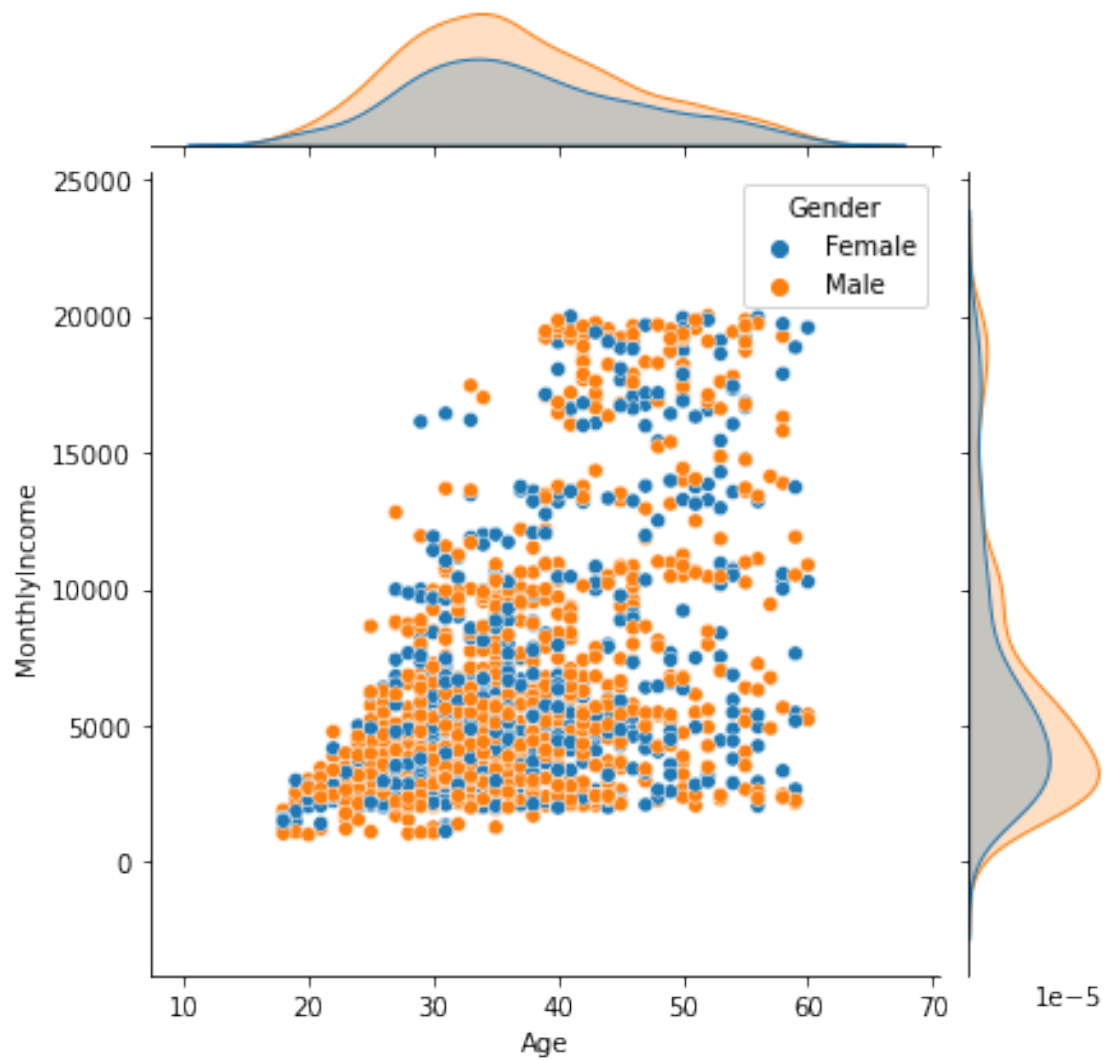


```
[22]: # Darstellung des monatlichen Einkommen nach Alter (Scatterplot)
```

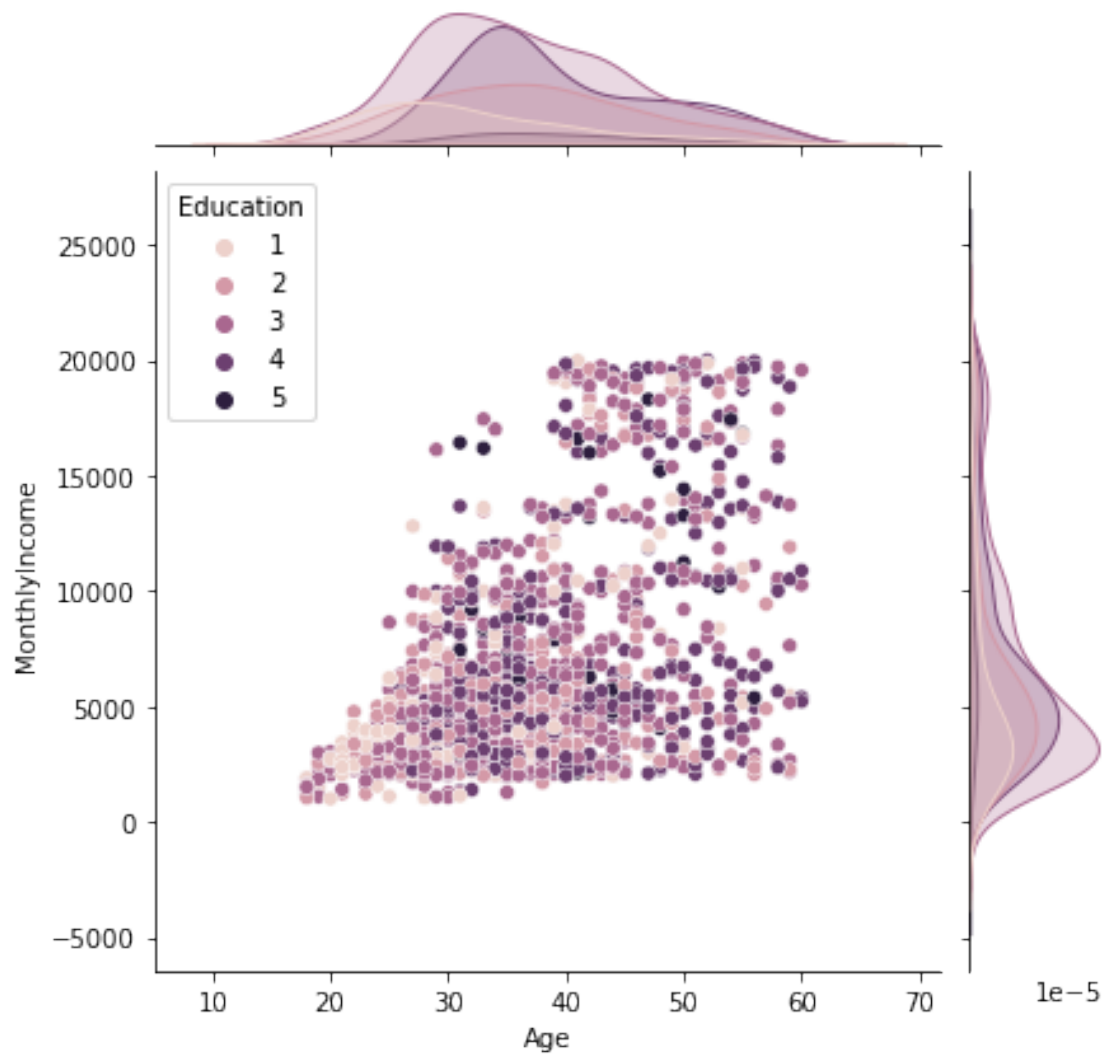
```
[25]: sns.jointplot(x = 'Age', y = 'MonthlyIncome', data = hr, kind = 'scatter')  
plt.show()
```



```
[28]: sns.jointplot(x = 'Age', y = 'MonthlyIncome', hue = 'Gender', data = hr, kind = 'scatter')  
      ↪ 'scatter')  
plt.show()
```

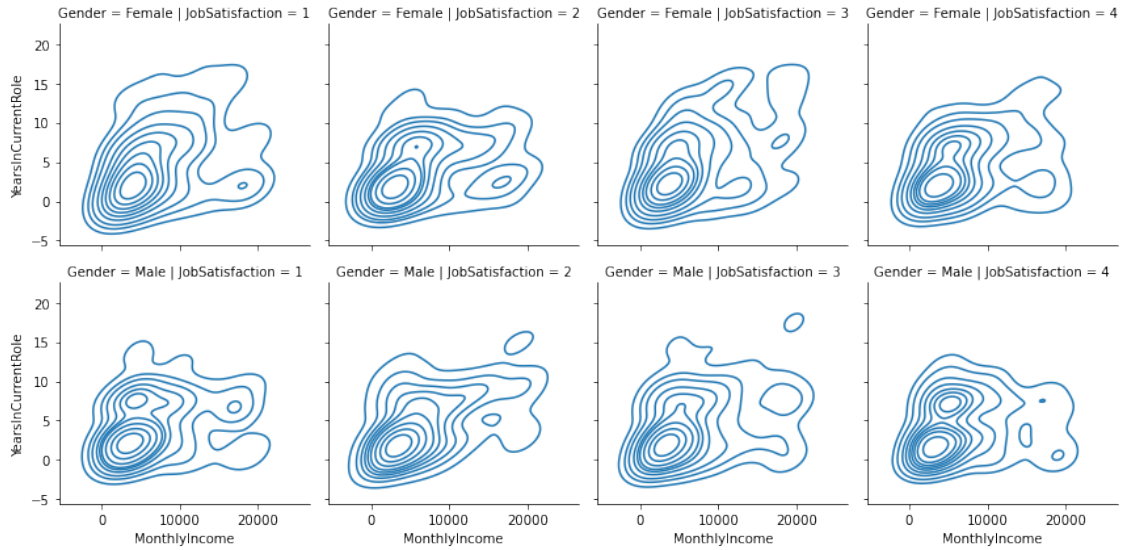


```
[29]: sns.jointplot(x = 'Age', y = 'MonthlyIncome', hue = 'Education', data = hr, kind = 'scatter')  
plt.show()
```

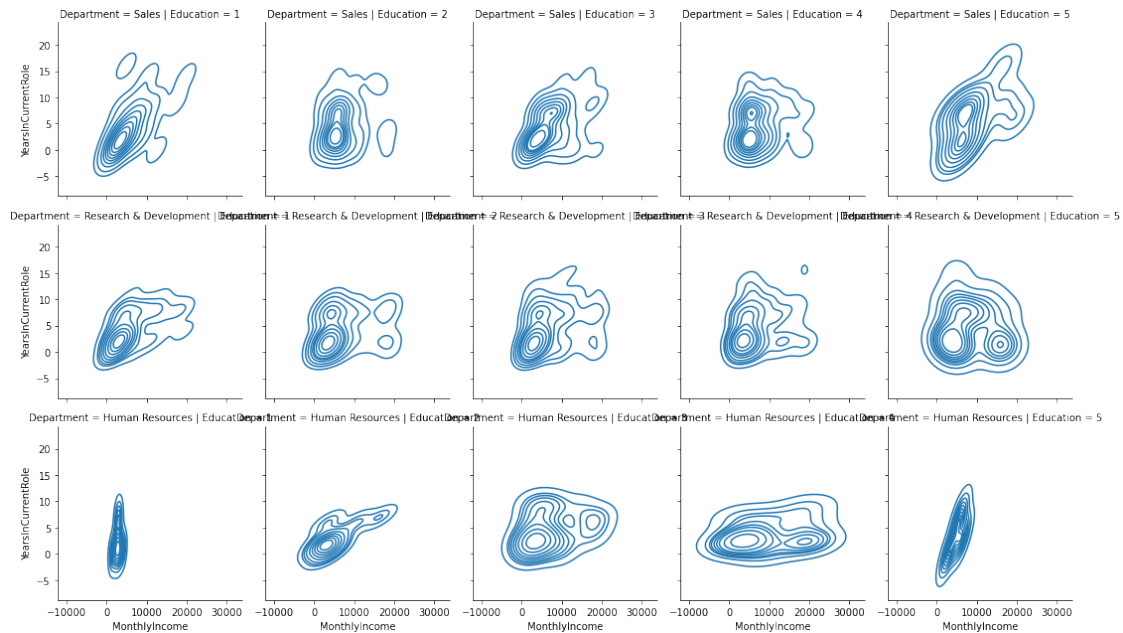


```
[30]: # Facetgrid um mehrere nicht numerische Variablen darzustellen
```

```
[32]: g = sns.FacetGrid(hr, col='JobSatisfaction', row = 'Gender')
g.map(sns.kdeplot, 'MonthlyIncome', 'YearsInCurrentRole')
plt.show()
```



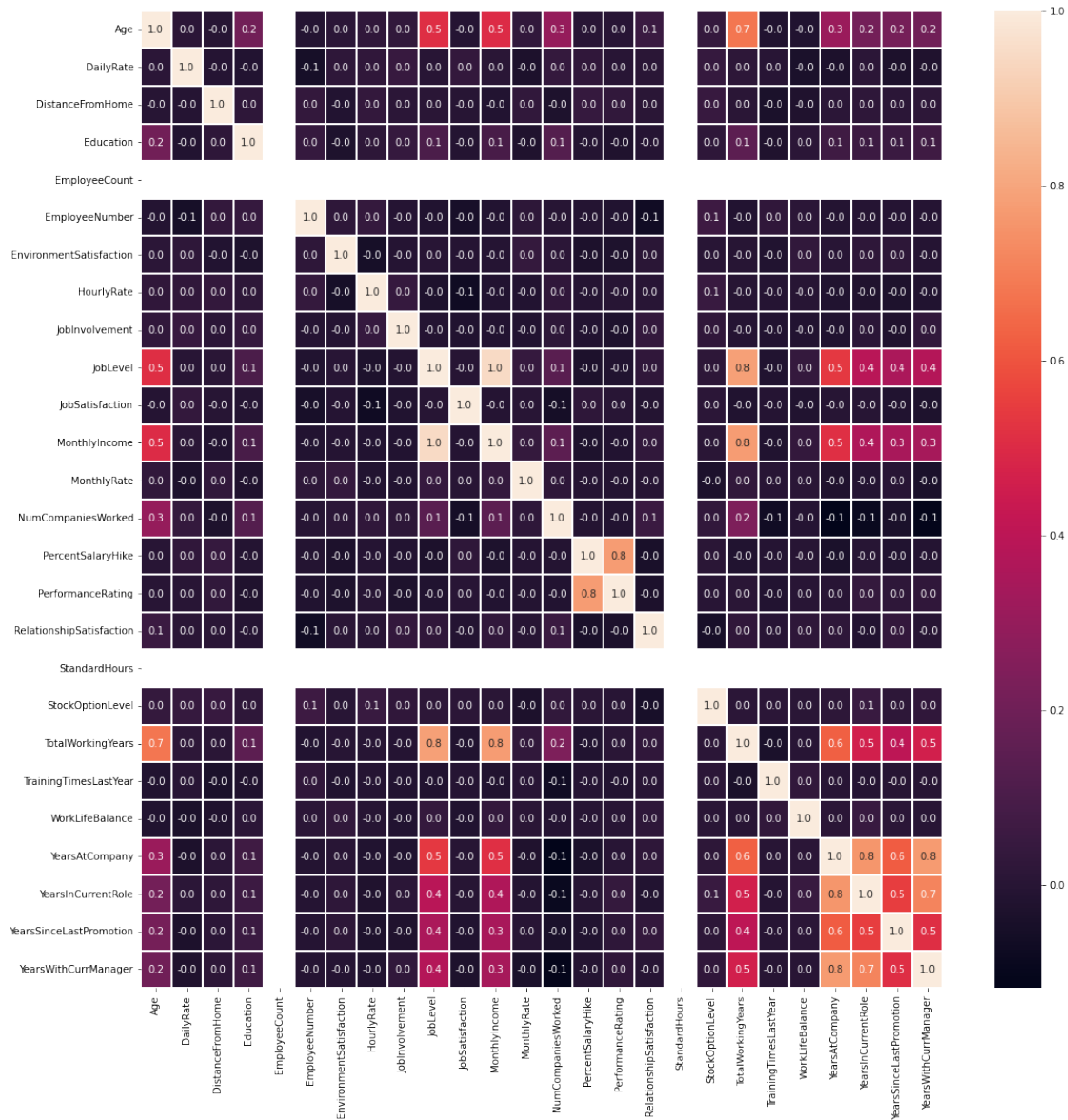
```
[34]: g = sns.FacetGrid(hr, col='Education', row = 'Department')
g.map(sns.kdeplot, 'MonthlyIncome', 'YearsInCurrentRole')
plt.show()
```



```
[35]: # plot Korrelationsmatrix für Alle numerische Variabeln
```

```
[36]: f, ax = plt.subplots(figsize=(18,18))
```

```
sns.heatmap(hr.corr(), annot = True, linewidths = 0.4, fmt='.1f', ax=ax)
# hr.corr bedeutet die Korrelationsmatrix
# annot bedeutet die Zahlen werden gezeigt
# 'linewidth' bedeutet eine kleine linie zw. den Elementen
# fmt = '.1f' bedeutet nur ein dezimal nach dem Komma
plt.show()
```



[ ]: