Survival Analysis Lab

Complete the following exercises to solidify your knowledge of survival analysis.

In [1]:
```
import pandas as pd
import plotly.plotly as py
import cufflinks as cf
from lifelines import KaplanMeierFitter

cf.go_offline()
```

In [2]:
```
data = pd.read_csv('../data/attrition.csv')
```

In [3]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1470 entries, 0 to 1469
Data columns (total 35 columns):
Age                         1470 non-null int64
Attrition                   1470 non-null int64
BusinessTravel              1470 non-null object
DailyRate                   1470 non-null int64
Department                  1470 non-null object
DistanceFromHome            1470 non-null int64
Education                   1470 non-null int64
EducationField              1470 non-null object
EmployeeCount               1470 non-null int64
EmployeeNumber              1470 non-null int64
EnvironmentSatisfaction     1470 non-null int64
Gender                      1470 non-null object
HourlyRate                  1470 non-null int64
JobInvolvement              1470 non-null int64
JobLevel                    1470 non-null int64
JobRole                     1470 non-null object
JobSatisfaction             1470 non-null int64
MaritalStatus               1470 non-null object
MonthlyIncome               1470 non-null int64
MonthlyRate                 1470 non-null int64
NumCompaniesWorked          1470 non-null int64
Over18                      1470 non-null object
OverTime                    1470 non-null object
PercentSalaryHike           1470 non-null int64
PerformanceRating           1470 non-null int64
RelationshipSatisfaction    1470 non-null int64
StandardHours               1470 non-null int64
StockOptionLevel            1470 non-null int64
TotalWorkingYears           1470 non-null int64
TrainingTimesLastYear       1470 non-null int64
WorkLifeBalance             1470 non-null int64
YearsAtCompany              1470 non-null int64
YearsInCurrentRole          1470 non-null int64
YearsSinceLastPromotion     1470 non-null int64
YearsWithCurrManager        1470 non-null int64
dtypes: int64(27), object(8)
memory usage: 402.1+ KB
```

In [4]:
```
data.head()
```

Out[4]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	Educa
0	41	1	Travel_Rarely	1102	Sales	1	2	Life
1	49	0	Travel_Frequently	279	Research & Development	8	1	Life
2	37	1	Travel_Rarely	1373	Research & Development	2	2	
3	33	0	Travel_Frequently	1392	Research & Development	3	4	Life
4	27	0	Travel_Rarely	591	Research & Development	2	1	

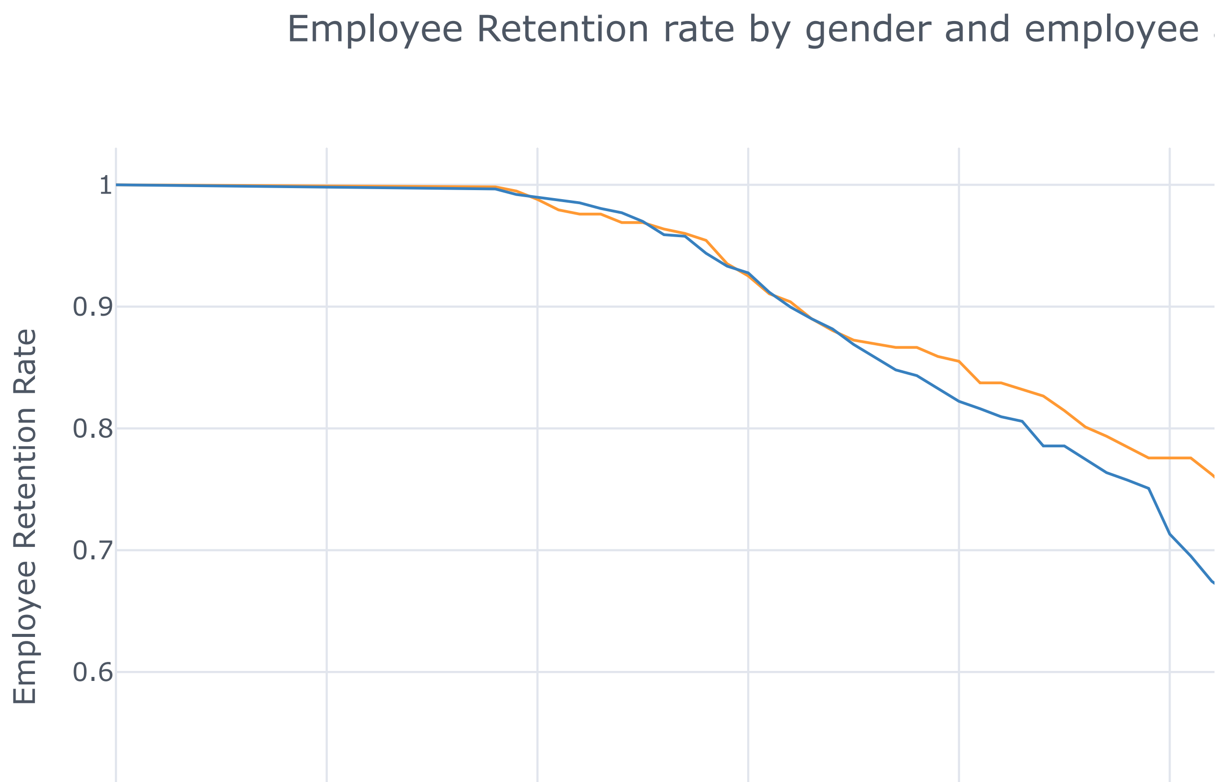
5 rows × 35 columns

1. Generate and plot a survival function that shows how employee retention rates vary by gender and employee age.

Tip: If your lines have gaps in them, you can fill them in by using the `fillna(method=ffill)` *and the* `fillna(method=bfill)` *methods and then taking the average. We have provided you with a revised survival function below that you can use for the exercises in this lab*

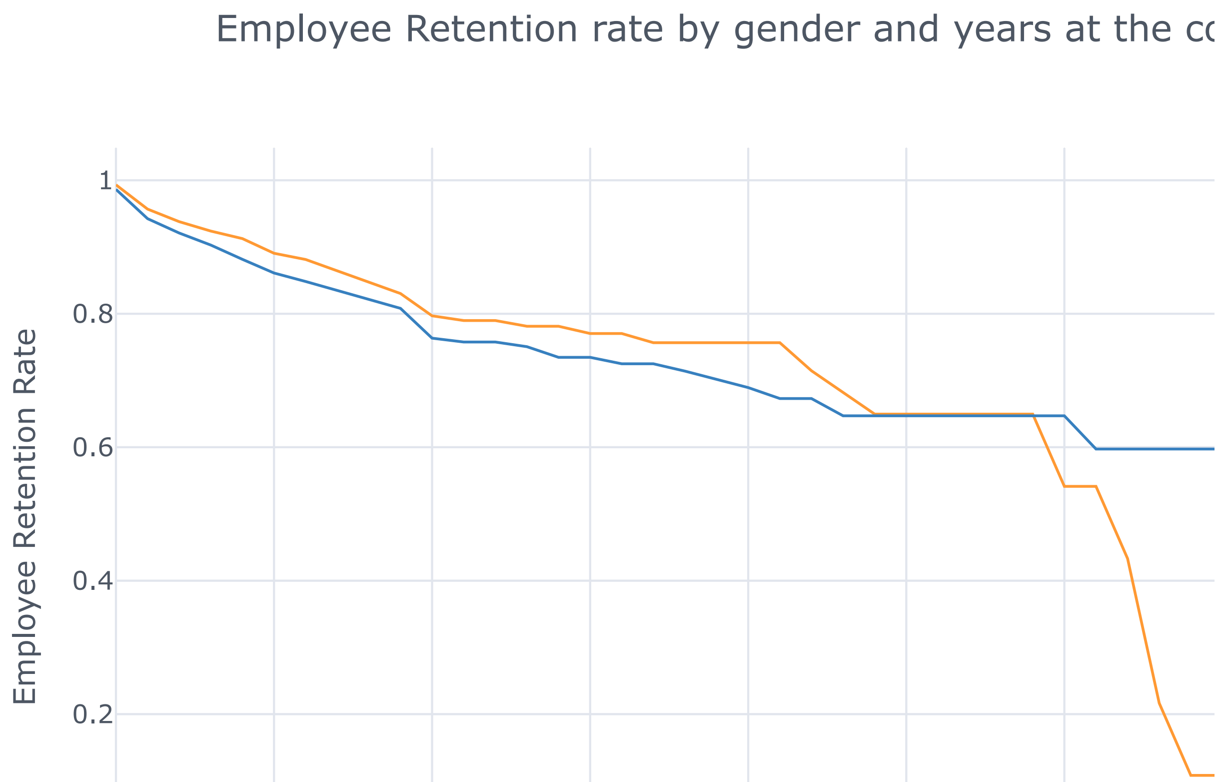
In [5]:
```
def survival(data, group_field, time_field, event_field):
    kmf = KaplanMeierFitter()
    results = []

    for i in data[group_field].unique():
        group = data[data[group_field]==i]
        T = group[time_field]
        E = group[event_field]
        kmf.fit(T, E, label=str(i))
        results.append(kmf.survival_function_)

    survival = pd.concat(results, axis=1)
    front_fill = survival.fillna(method='ffill')
    back_fill = survival.fillna(method='bfill')
    smoothed = (front_fill + back_fill) / 2
    return smoothed
```

In [6]:
```
rates = survival(data, 'Gender', 'Age', 'Attrition')

rates.iplot(kind='line', xTitle='Age', yTitle='Employee Retention Rate',\
            title= 'Employee Retention rate by gender and employee age')
```



2. Compare the plot above with one that plots employee retention rates by gender over the number of years the employee has been working for the company.

In [7]:
```
rates = survival(data, 'Gender', 'YearsAtCompany', 'Attrition')

rates.iplot(kind='line', xTitle='Years At Company', yTitle='Employee Rete
            title= 'Employee Retention rate by gender and years at the co
```

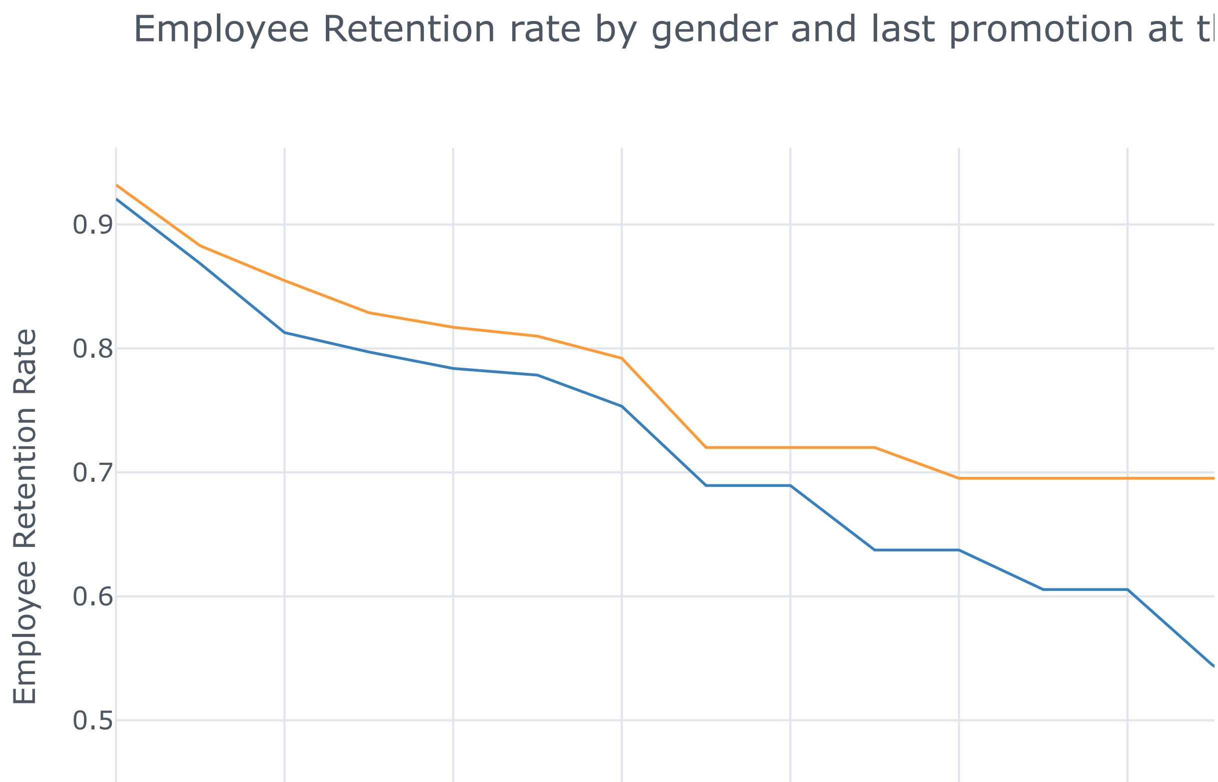


3. Let's look at retention rate by gender from a third perspective - the number of years since the employee's last promotion. Generate and plot a survival curve showing this.

In [8]:
```
data.columns
```

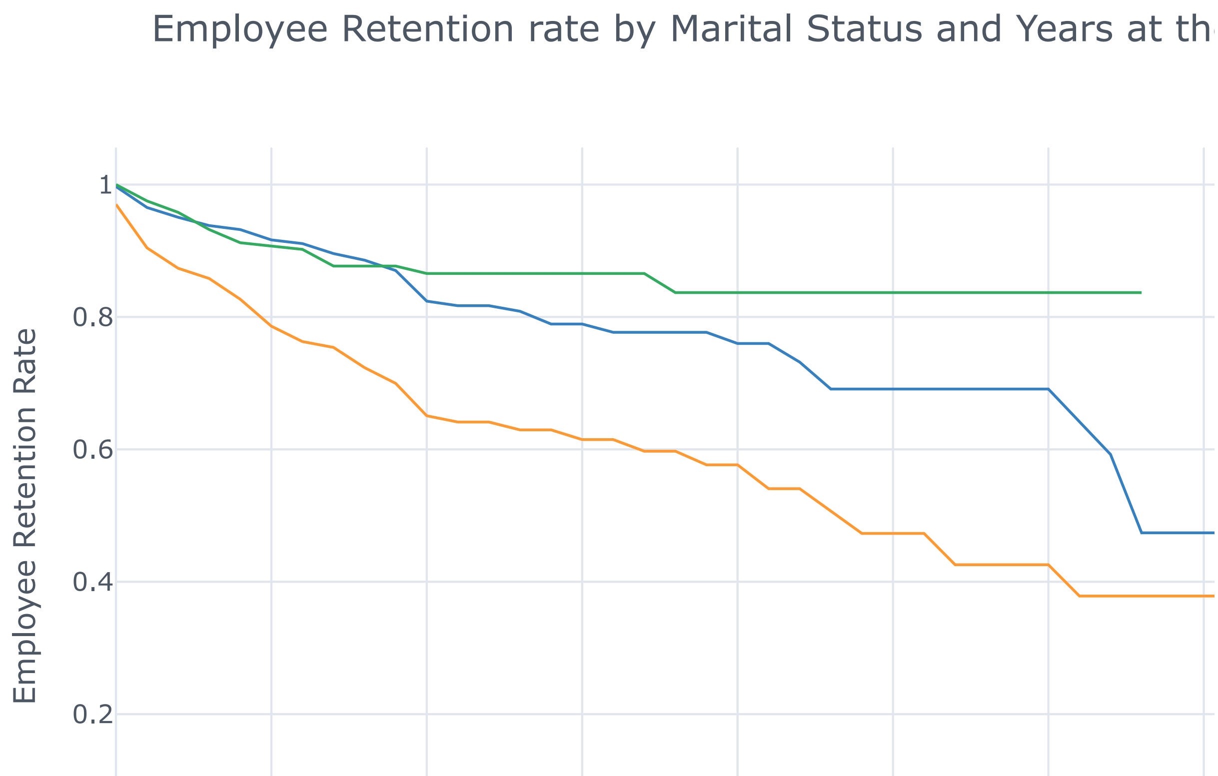
Out[8]:
```
Index(['Age', 'Attrition', 'BusinessTravel', 'DailyRate', 'Department'
,
       'DistanceFromHome', 'Education', 'EducationField', 'EmployeeCou
nt',
       'EmployeeNumber', 'EnvironmentSatisfaction', 'Gender', 'HourlyR
ate',
       'JobInvolvement', 'JobLevel', 'JobRole', 'JobSatisfaction',
       'MaritalStatus', 'MonthlyIncome', 'MonthlyRate', 'NumCompaniesW
orked',
       'Over18', 'OverTime', 'PercentSalaryHike', 'PerformanceRating',
       'RelationshipSatisfaction', 'StandardHours', 'StockOptionLevel'
,
       'TotalWorkingYears', 'TrainingTimesLastYear', 'WorkLifeBalance'
,
       'YearsAtCompany', 'YearsInCurrentRole', 'YearsSinceLastPromotio
n',
       'YearsWithCurrManager'],
      dtype='object')
```

In [9]:
```
rates = survival(data, 'Gender', 'YearsSinceLastPromotion', 'Attrition')

rates.iplot(kind='line', xTitle='Years Since Last Promotion', yTitle='Emp
            title= 'Employee Retention rate by gender and last promotion
```



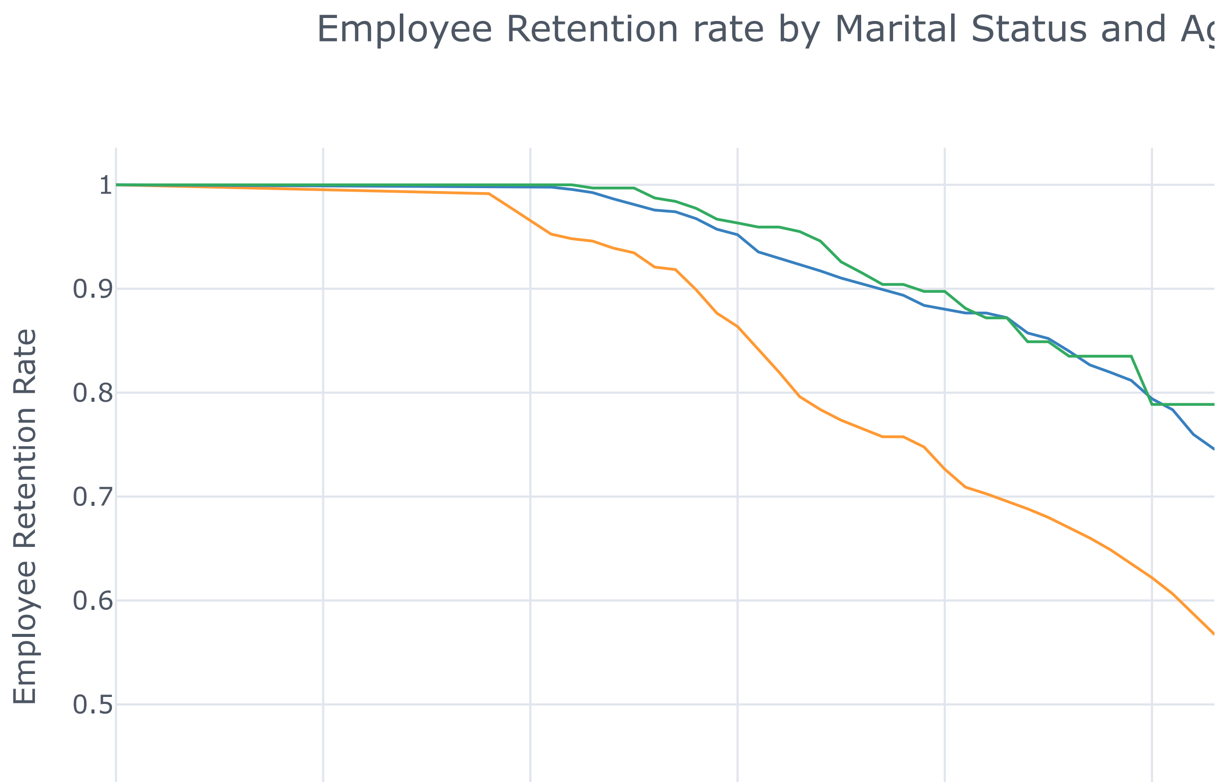
4. Let's switch to looking at retention rates from another demographic perspective: marital status. Generate and plot survival curves for the different marital statuses by number of years at the company.

```
In [10]: rates = survival(data, 'MaritalStatus', 'YearsAtCompany', 'Attrition')

         rates.iplot(kind='line', xTitle='Years At Company', yTitle='Employee Rete
                     title= 'Employee Retention rate by Marital Status and Years a
```



5. Let's also look at the marital status curves by employee age. Generate and plot the survival curves showing retention rates by marital status and age.

In [11]:
```
rates = survival(data, 'MaritalStatus', 'Age', 'Attrition')

rates.iplot(kind='line', xTitle='Age', yTitle='Employee Retention Rate',\
            title= 'Employee Retention rate by Marital Status and Age')
```

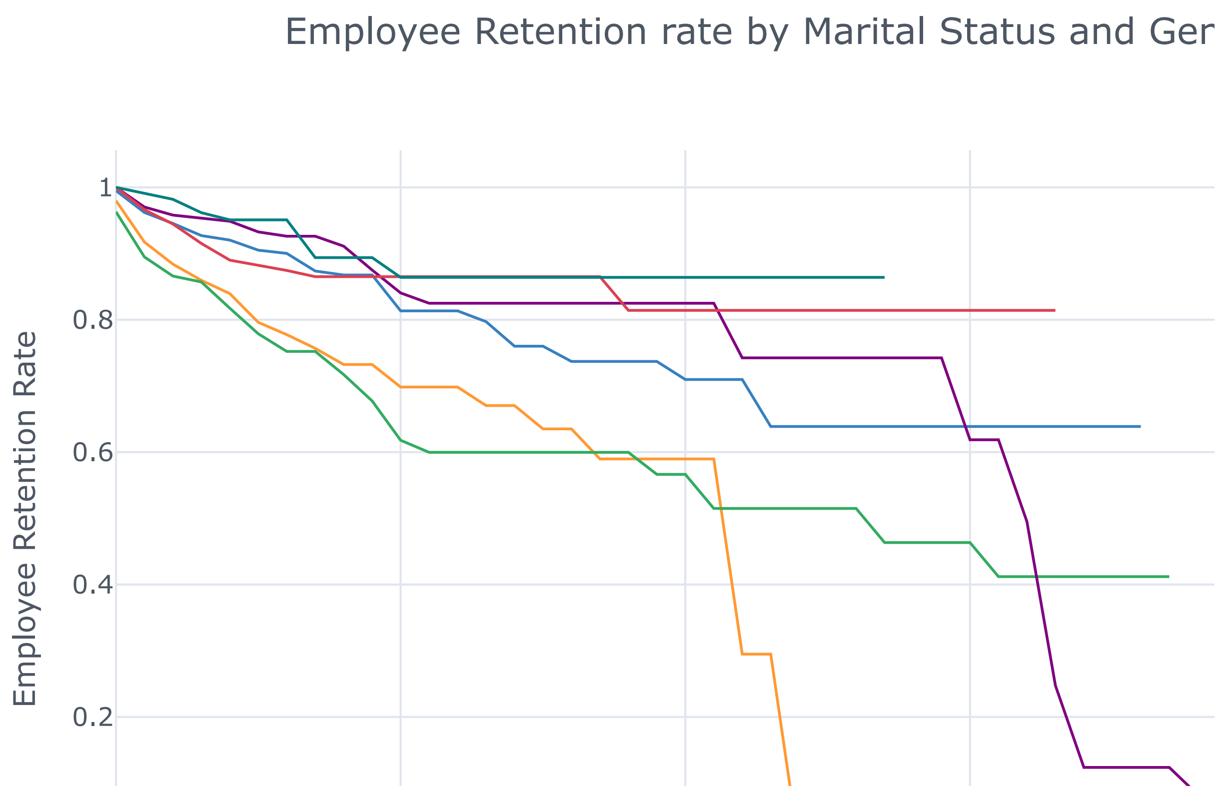


6. Now that we have looked at the retention rates by gender and marital status individually, let's look at them together.

Create a new field in the data set that concatenates marital status and gender, and then generate and plot a survival curve that shows the retention by this new field over the age of the employee.

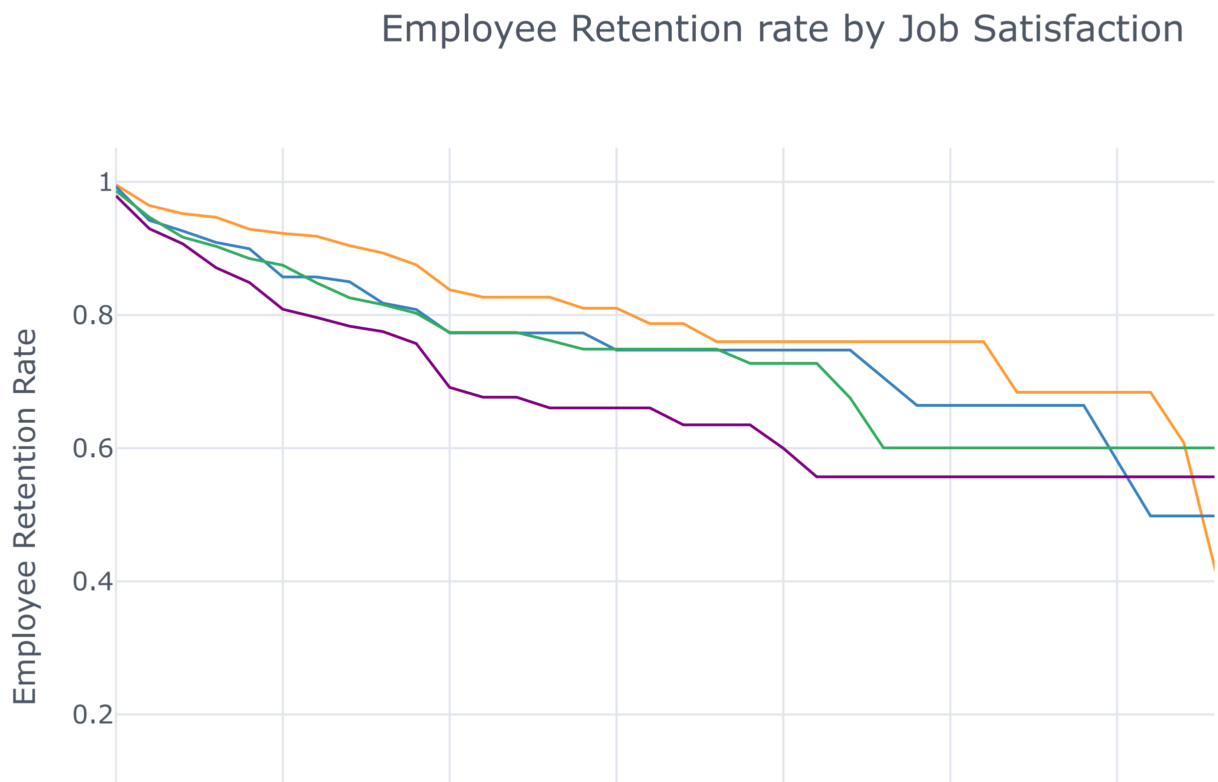
In [12]:
```
data['GenderandMaritalStatus']= data['Gender']+'-'+data['MaritalStatus']
```

In [13]:
```
rates = survival(data, 'GenderandMaritalStatus', 'YearsAtCompany', 'Attri

rates.iplot(kind='line', xTitle='Years at the Company', yTitle='Employee
            title= 'Employee Retention rate by Marital Status and Gender'
```



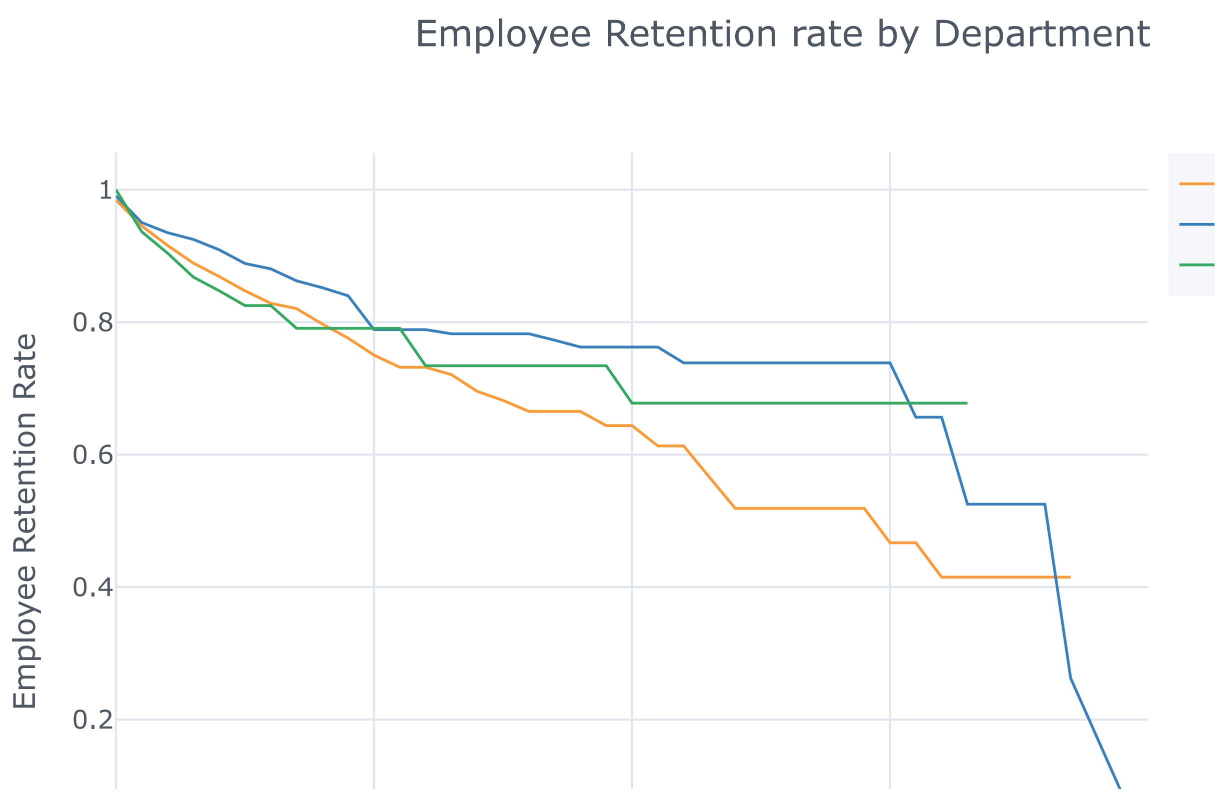
6. Let's find out how job satisfaction affects retention rates. Generate and plot survival curves for each level of job satisfaction by number of years at the company.

In [14]:
```
rates = survival(data, 'JobSatisfaction', 'YearsAtCompany', 'Attrition')

rates.iplot(kind='line', xTitle='Years at the Company', yTitle='Employee
          title= 'Employee Retention rate by Job Satisfaction')
```



7. Let's investigate whether the department the employee works in has an impact on how long they stay with the company. Generate and plot survival curves showing retention by department and years the employee has worked at the company.

In [15]:
```
rates = survival(data, 'Department', 'YearsAtCompany', 'Attrition')

rates.iplot(kind='line', xTitle='Years at the Company', yTitle='Employee
          title= 'Employee Retention rate by Department')
```

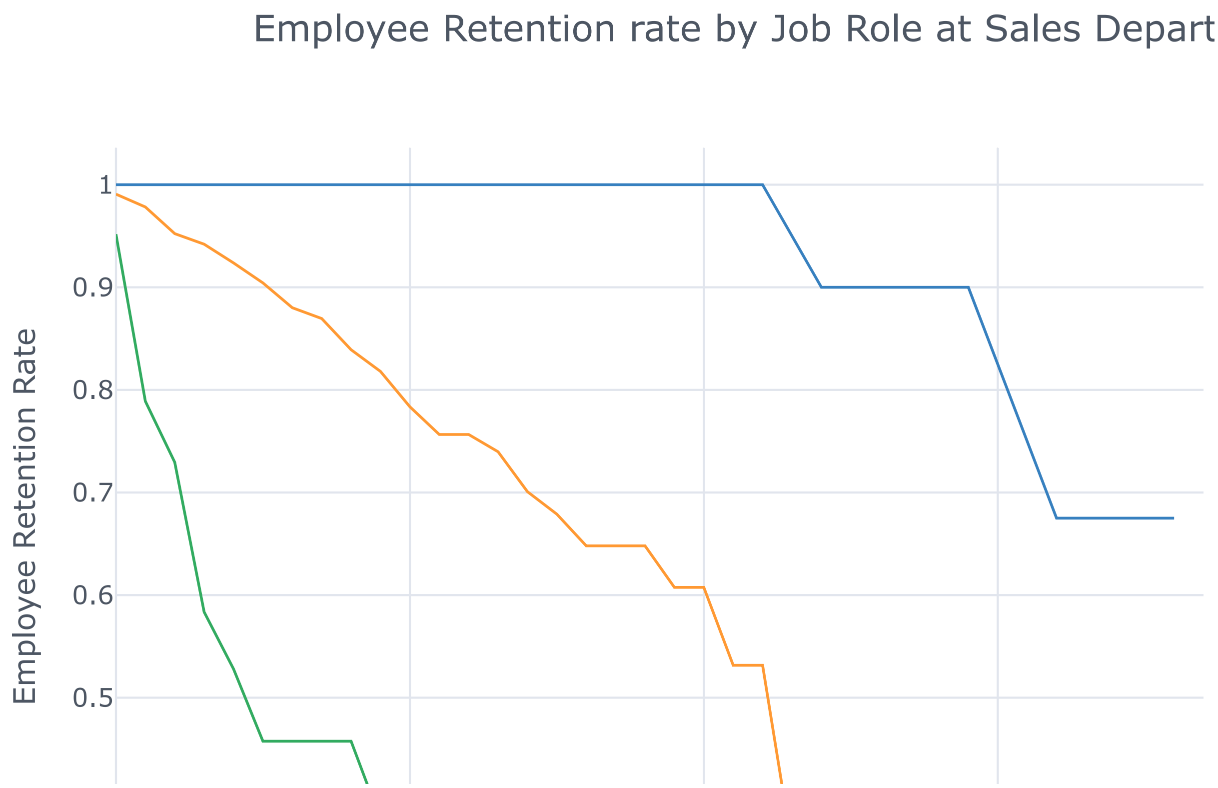


8. From the previous example, it looks like the sales department has the highest attrition. Let's drill down on this and look at what the survival curves for specific job roles within that department look like.

Filter the data set for just the sales department and then generate and plot survival curves by job role and the number of years at the company.

In [16]:
```
dep_filt= data[data['Department']=='Sales']
```

In [17]:
```
rates = survival(dep_filt, 'JobRole', 'YearsAtCompany', 'Attrition')

rates.iplot(kind='line', xTitle='Years at the Company', yTitle='Employee
            title= 'Employee Retention rate by Job Role at Sales Departme
```



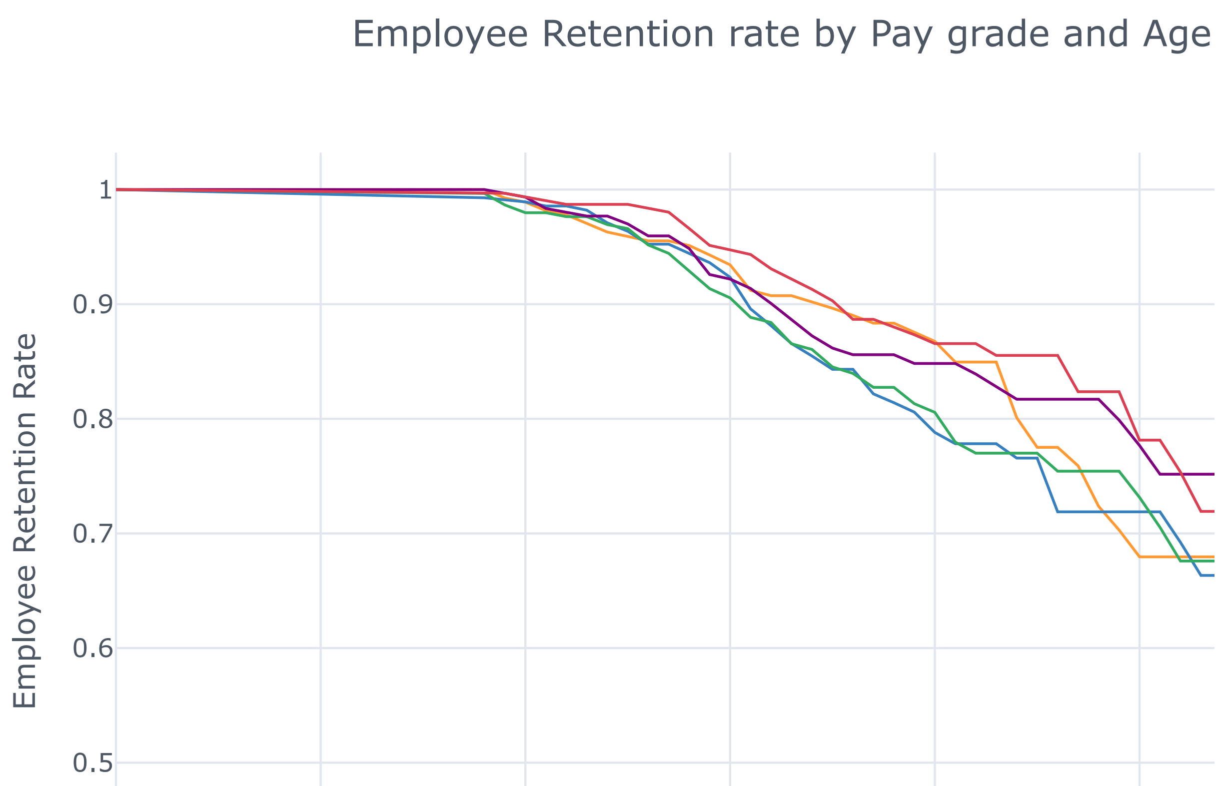
Use the pd.qcut method to bin the HourlyRate field into 5 different pay grade categories (Very Low, Low, Moderate, High, and Very High). Generate and plot survival curves showing employee retention by pay grade and age.

In [18]:
```
data.columns
```

Out[18]:
```
Index(['Age', 'Attrition', 'BusinessTravel', 'DailyRate', 'Department'
,
       'DistanceFromHome', 'Education', 'EducationField', 'EmployeeCou
nt',
       'EmployeeNumber', 'EnvironmentSatisfaction', 'Gender', 'HourlyR
ate',
       'JobInvolvement', 'JobLevel', 'JobRole', 'JobSatisfaction',
       'MaritalStatus', 'MonthlyIncome', 'MonthlyRate', 'NumCompaniesW
orked',
       'Over18', 'OverTime', 'PercentSalaryHike', 'PerformanceRating',
       'RelationshipSatisfaction', 'StandardHours', 'StockOptionLevel'
,
       'TotalWorkingYears', 'TrainingTimesLastYear', 'WorkLifeBalance'
,
       'YearsAtCompany', 'YearsInCurrentRole', 'YearsSinceLastPromotio
n',
       'YearsWithCurrManager', 'GenderandMaritalStatus'],
      dtype='object')
```

In [19]:
```
pay_grade = pd.qcut(data['HourlyRate'], 5, labels=['Very Low', 'Low', 'Mo
                                                    'Very High'])
data['Pay_Grade'] = pay_grade
```

In [20]:
```
rates = survival(data, 'Pay_Grade', 'Age', 'Attrition')

rates.iplot(kind='line', xTitle='Years', yTitle='Employee Retention Rate'
            title= 'Employee Retention rate by Pay grade and Age')
```



10. Finally, let's take a look at how the demands of the job impact employee attrition.

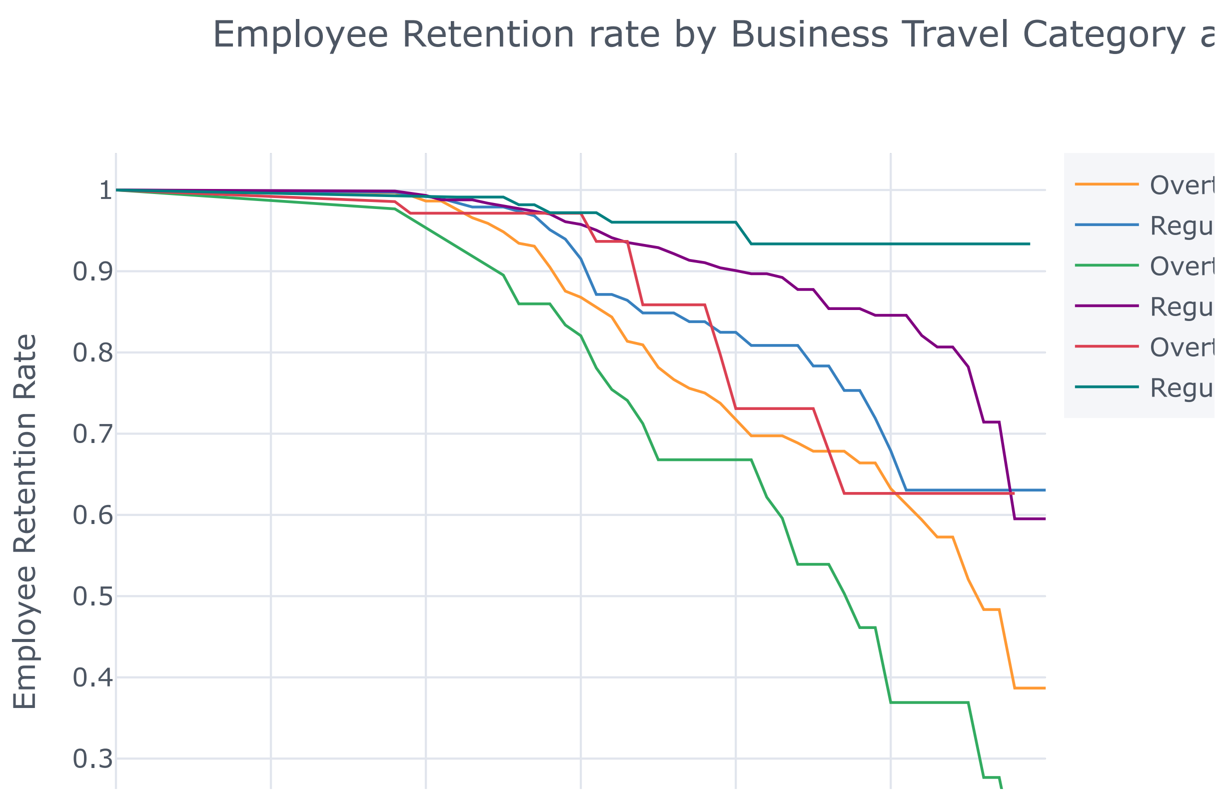
- Create a new field whose values are 'Overtime' or 'Regular Hours' depending on whether there is a Yes or a No in the OverTime field.
- Create a new field that concatenates that field with the BusinessTravel field.
- Generate and plot survival curves showing employee retention based on these conditions and employee age.

In [21]:
```
import numpy as np
```

In [22]:
```
data['Category']= np.where(data['OverTime']=='Yes', 'Overtime', 'Regular
```

In [23]:
```
data['Categ_Travel']= data['Category'] + '-' + data['BusinessTravel']
```

In [24]:
```
rates = survival(data, 'Categ_Travel', 'Age', 'Attrition')

rates.iplot(kind='line', xTitle='Years', yTitle='Employee Retention Rate'
            title= 'Employee Retention rate by Business Travel Category a
```



In [ ]: