Survival Analysis Lab

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

In [33]:

```python
import pandas as pd
import chart_studio.plotly as py
#import plotly.plotly as py
import cufflinks as cf
from lifelines import KaplanMeierFitter
import numpy as np

cf.go_offline()
```

In [6]:

```python
data = pd.read_csv('C:/Users/rsher/lab-survival-analysis/data/attrition.csv')
data.head()
```

Out[6]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	...	RelationshipSatisfaction	StandardHours	StockOptionLevel
0	41	1	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	1	...	1	80	0
1	49	0	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	2	...	4	80	1
2	37	1	Travel_Rarely	1373	Research & Development	2	2	Other	1	4	...	2	80	0
3	33	0	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	1	5	...	3	80	0
4	27	0	Travel_Rarely	591	Research & Development	2	1	Medical	1	7	...	4	80	1

5 rows × 35 columns

In [10]:

```python
data.columns
```

Out[10]:

```
Index(['Age', 'Attrition', 'BusinessTravel', 'DailyRate', 'Department',
       'DistanceFromHome', 'Education', 'EducationField', 'EmployeeCount',
       'EmployeeNumber', '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')
```

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 [7]:

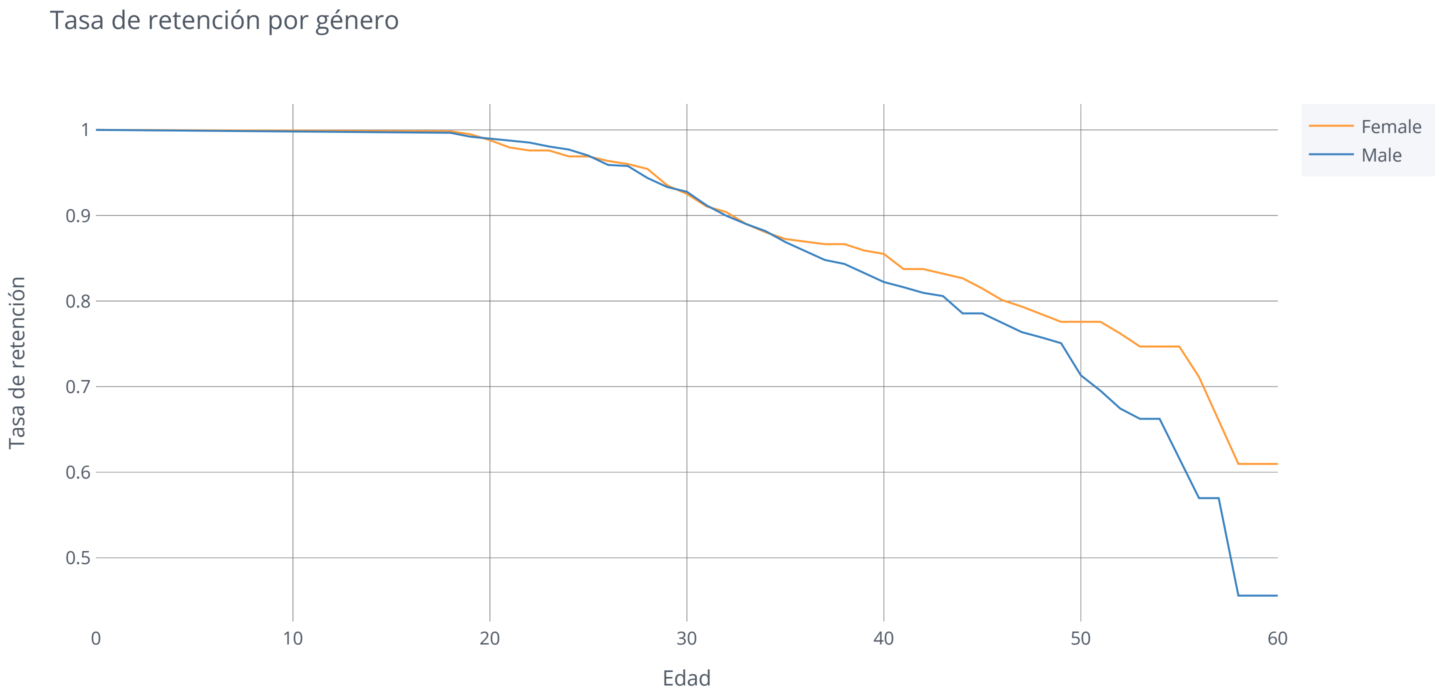
```
def survival(data, group_field, time_field, event_field):
    kmf = KaplanMeierFitter()
    results = []

    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 [14]:

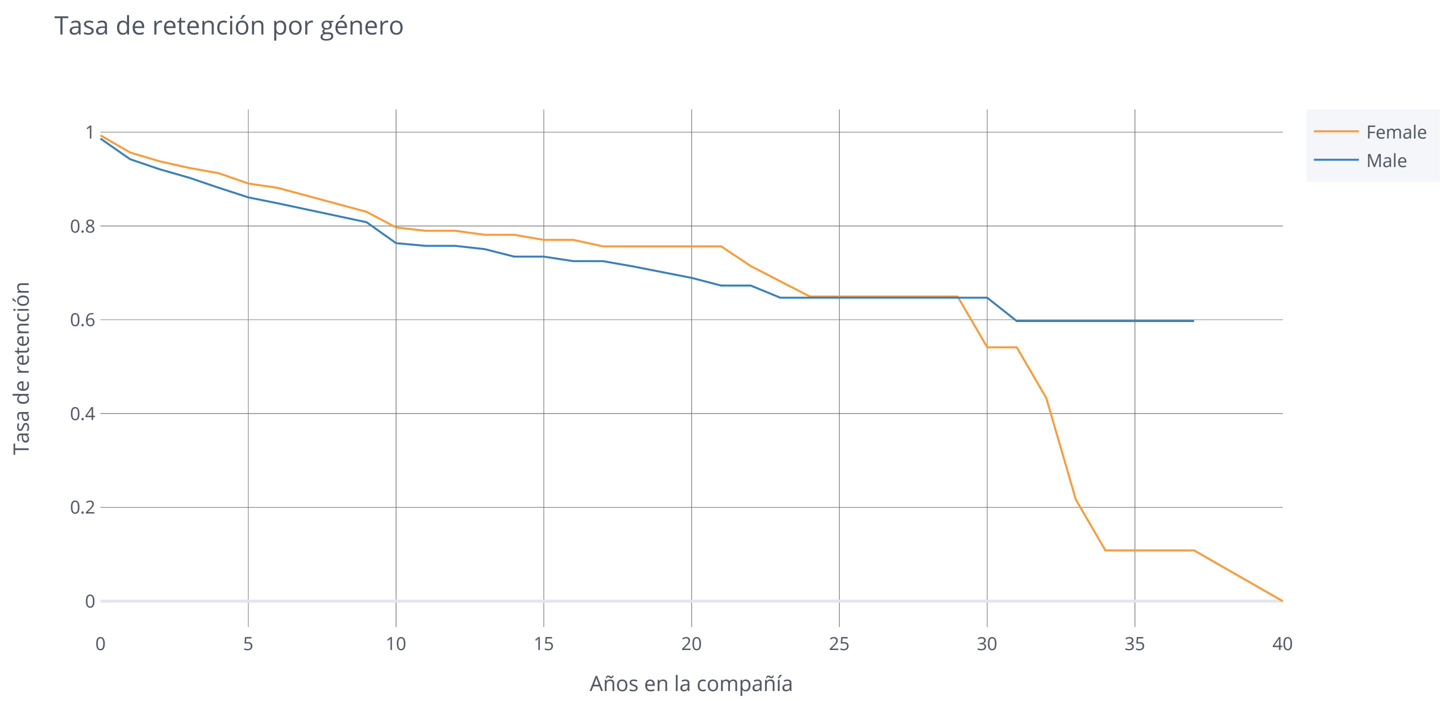
```
tasa_retencion = survival(data,'Gender','Age','Attrition')
tasa_retencion.iplot(kind='line',xTitle='Edad',yTitle='Tasa de retención',
                     title='Tasa de retención por género')
```



Export to plot.ly »

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 [15]: tasa_retencion = survival(data,'Gender','YearsAtCompany','Attrition')
         tasa_retencion.iplot(kind='line',xTitle='Años en la compañía',yTitle='Tasa de retención',
                              title='Tasa de retención por género')
```

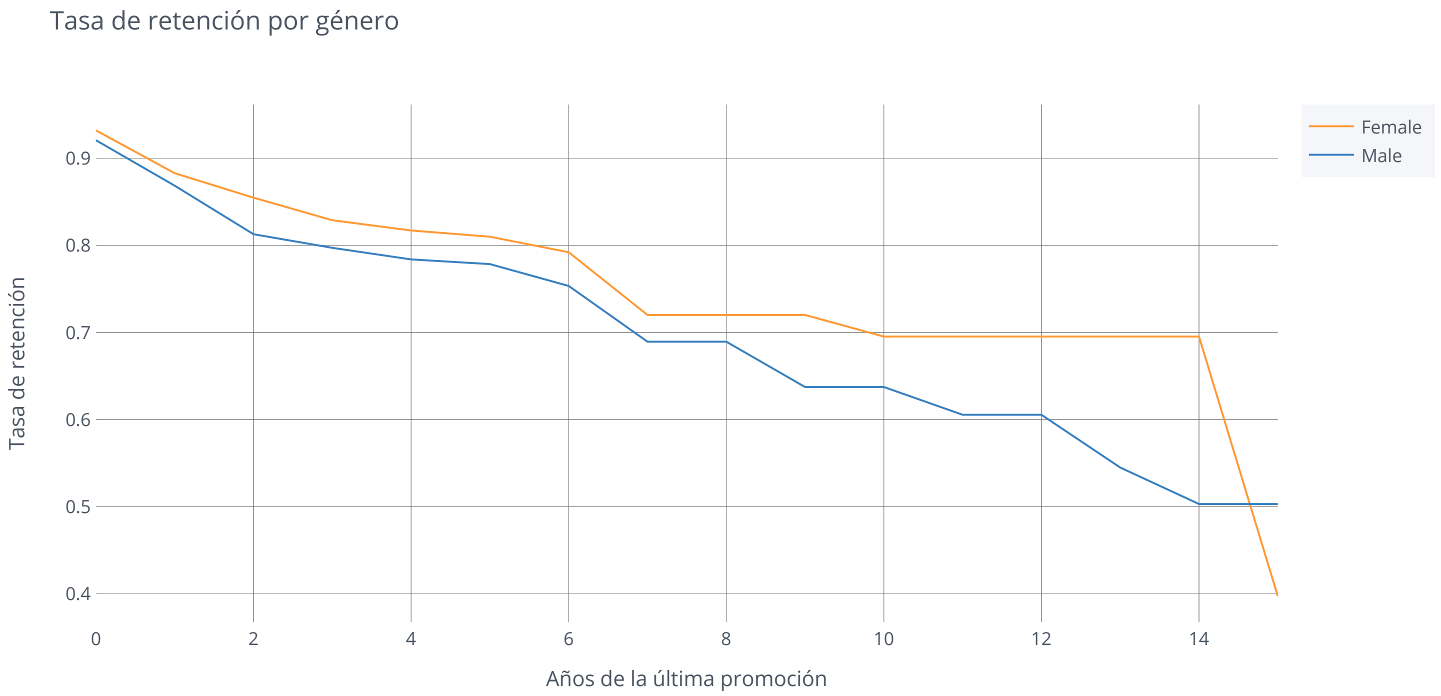


Export to plot.ly »

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 [17]:

```python
tasa_retencion = survival(data,'Gender','YearsSinceLastPromotion','Attrition')
tasa_retencion.iplot(kind='line',xTitle='Años de la última promoción',yTitle='Tasa de retención',
                     title='Tasa de retención por género')
```

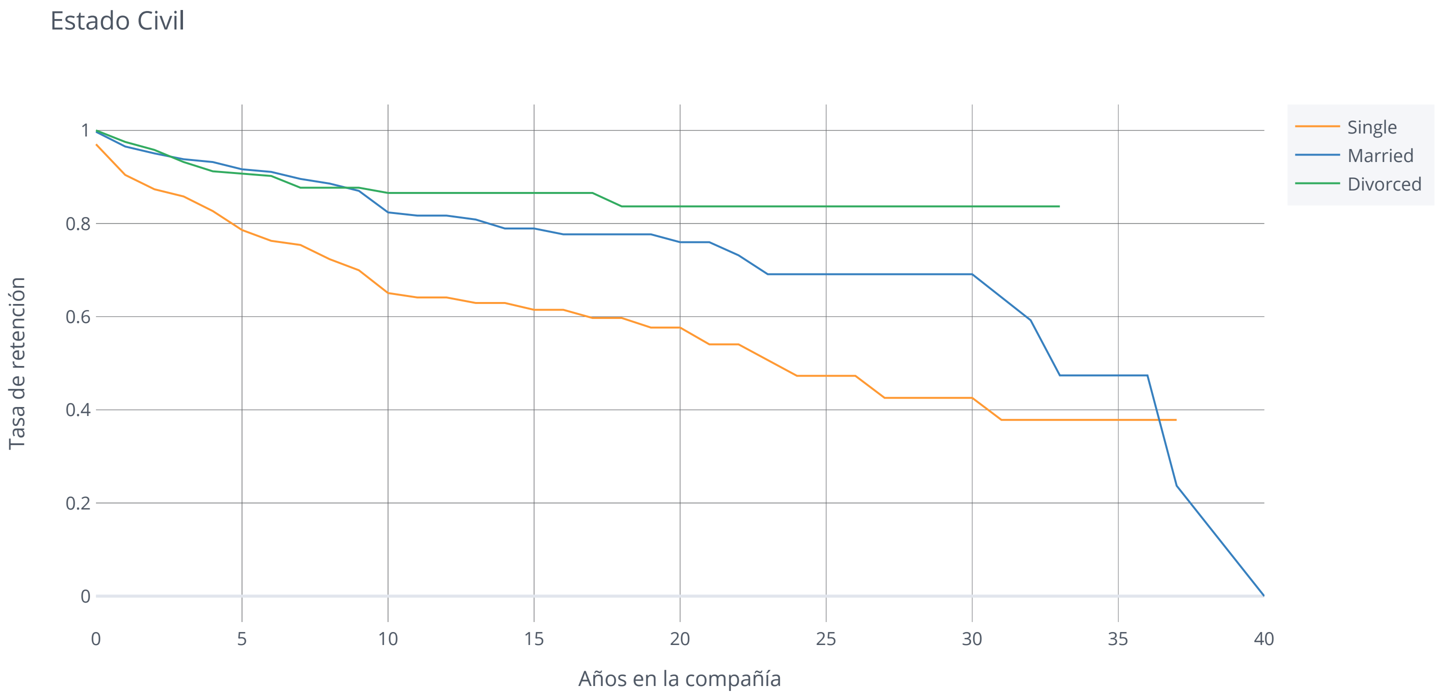


Export to plot.ly »

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 [18]:

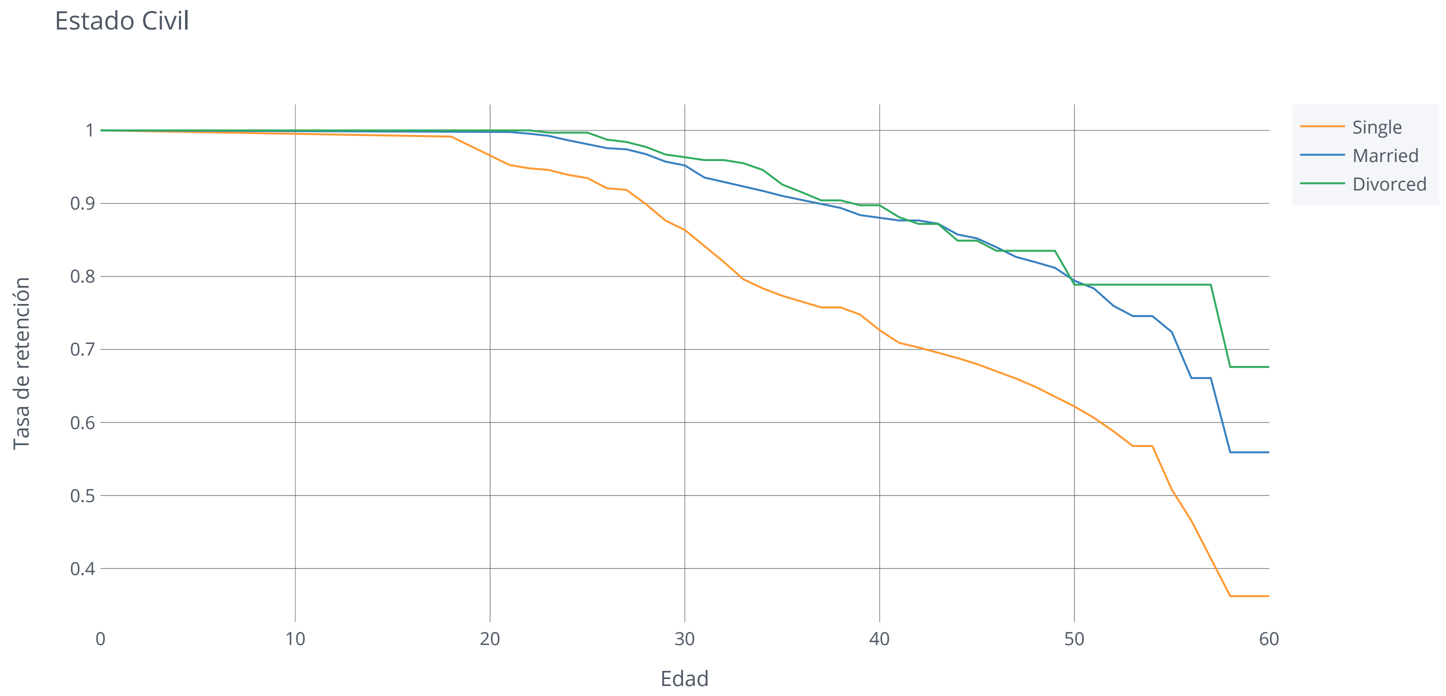
```
tasa_retencion = survival(data,'MaritalStatus','YearsAtCompany','Attrition')
tasa_retencion.iplot(kind='line',xTitle='Años en la compañía',yTitle='Tasa de retención',
                     title='Estado Civil')
```



Export to plot.ly »

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 [19]: tasa_retencion = survival(data,'MaritalStatus','Age','Attrition')
         tasa_retencion.iplot(kind='line',xTitle='Edad',yTitle='Tasa de retención',
                              title='Estado Civil')
```



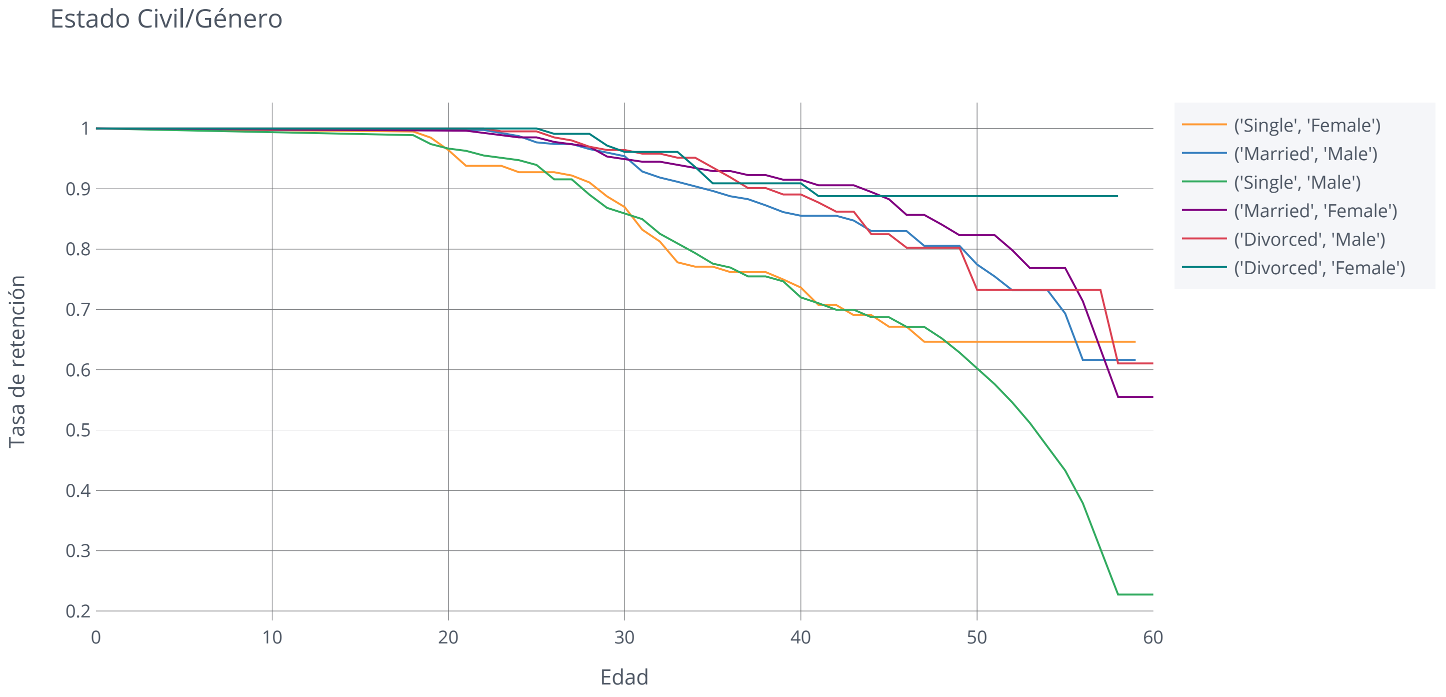
Export to plot.ly »

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 [23]: data['Gen_Mar'] = list(zip(data['MaritalStatus'],data['Gender']))

         tasa_retencion = survival(data,'Gen_Mar','Age','Attrition')
         tasa_retencion.iplot(kind='line',xTitle='Edad',yTitle='Tasa de retención',
                              title='Estado Civil/Género')
```

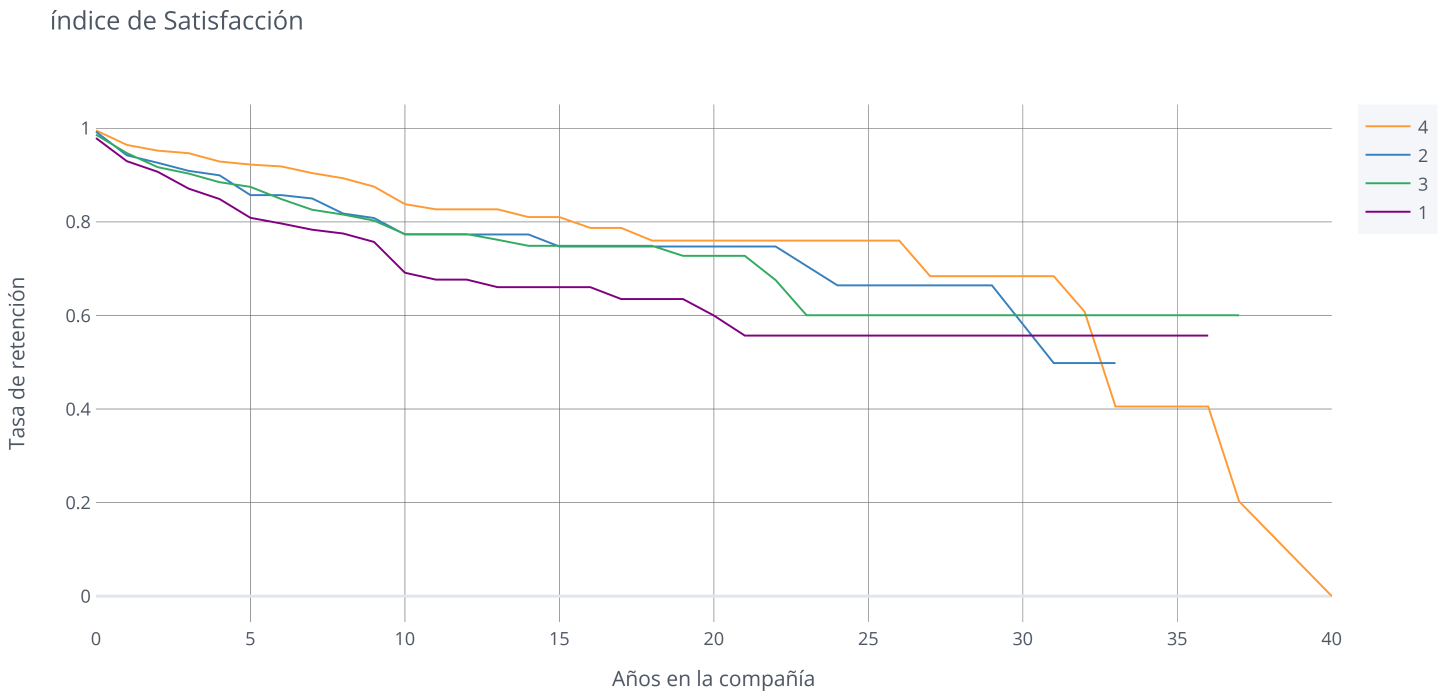


Export to plot.ly »

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 [24]:

```
tasa_retencion = survival(data,'JobSatisfaction','YearsAtCompany','Attrition')
tasa_retencion.iplot(kind='line',xTitle='Años en la compañía',yTitle='Tasa de retención',
                     title='índice de Satisfacción')
```

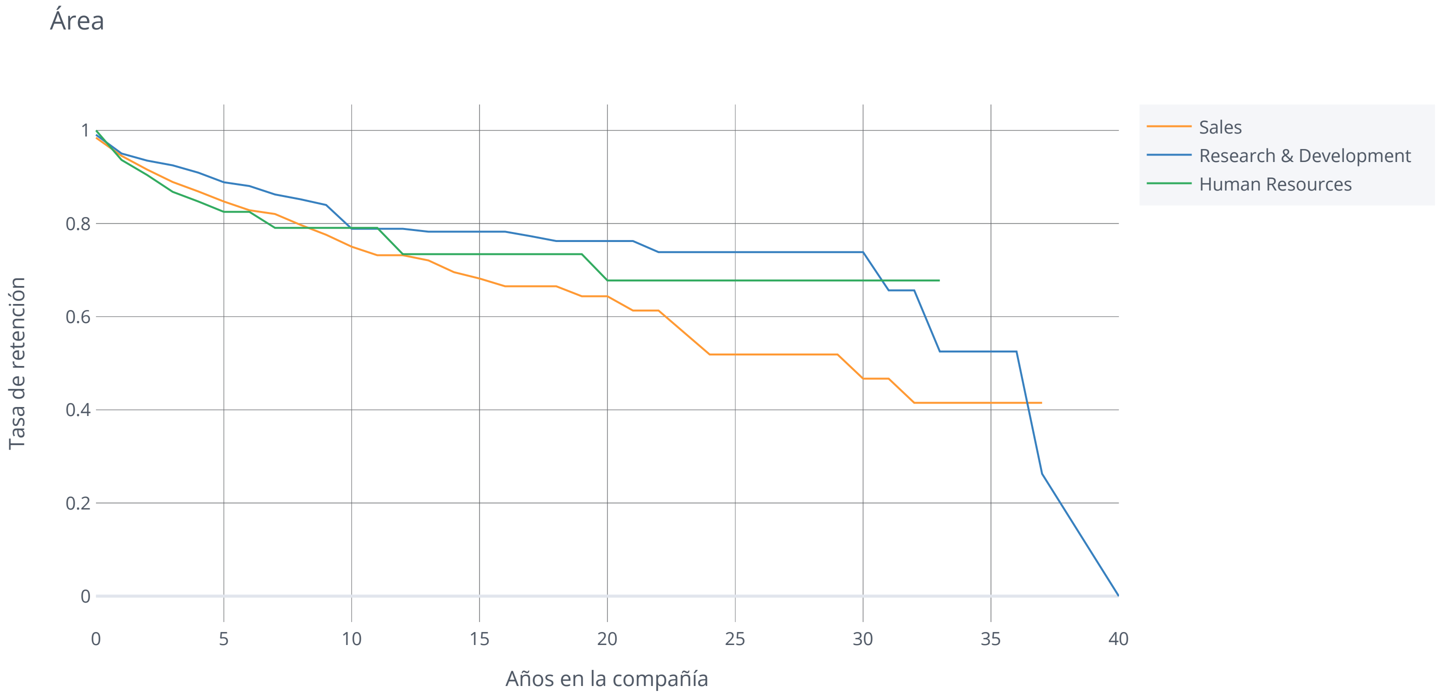


Export to plot.ly »

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 [26]:

```
tasa_retencion = survival(data,'Department','YearsAtCompany','Attrition')
tasa_retencion.iplot(kind='line',xTitle='Años en la compañía',yTitle='Tasa de retención',
                    title='Área')
```

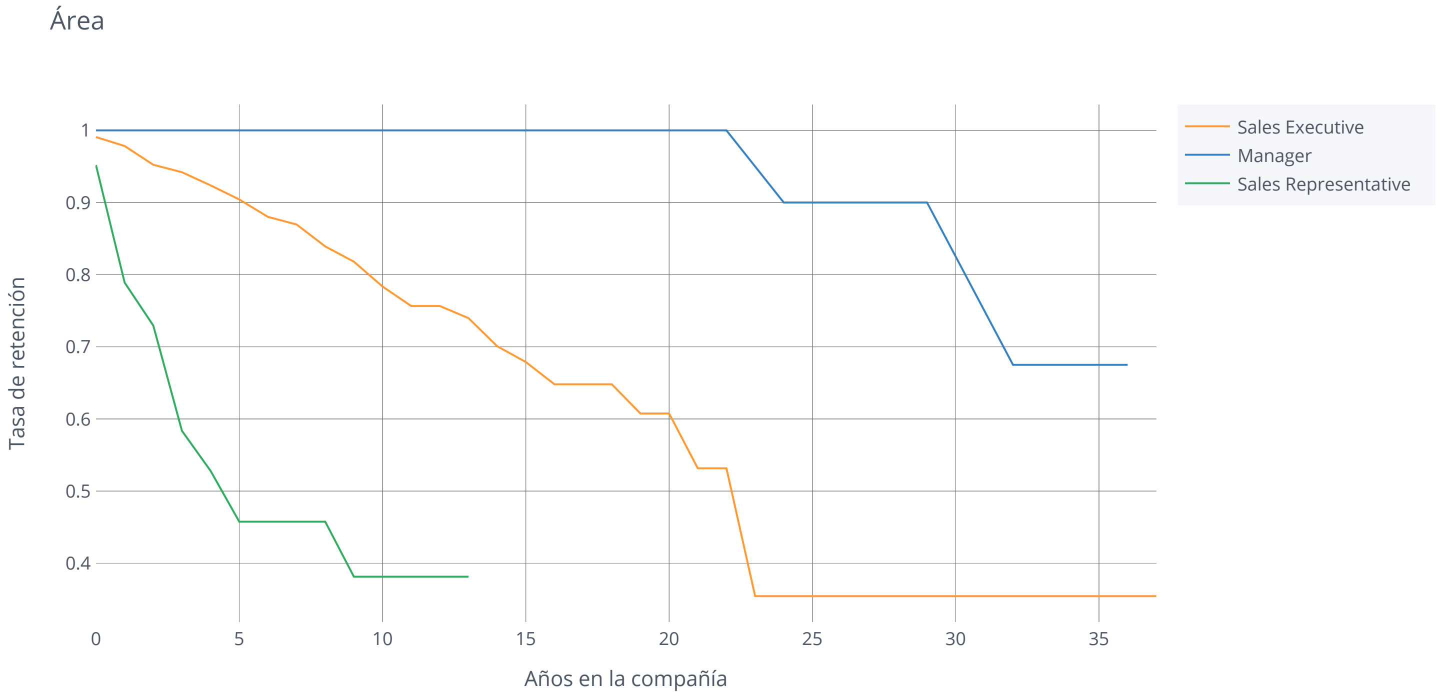


Export to plot.ly »

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 [28]:
```
tasa_retencion = survival(data[data['Department']=='Sales'],'JobRole','YearsAtCompany','Attrition')
tasa_retencion.iplot(kind='line',xTitle='Años en la compañía',yTitle='Tasa de retención',
                     title='Área')
```



Export to plot.ly »

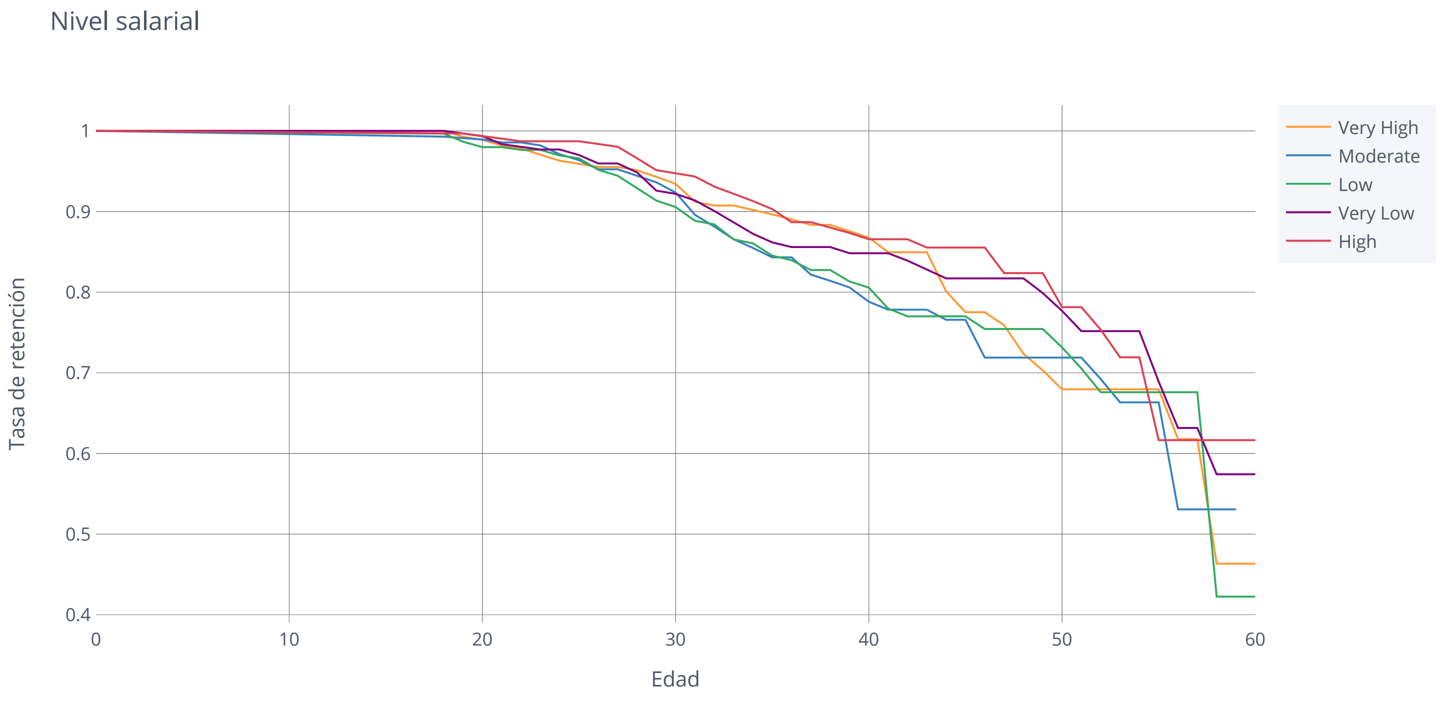
9. Let examine how compensation affects attrition.

- 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 [32]:
```
pay_grade = ['Very Low', 'Low', 'Moderate', 'High','Very High']

data['Pay_Grade'] = pd.qcut(data['HourlyRate'],5,labels=pay_grade)

tasa_retencion = survival(data,'Pay_Grade','Age','Attrition')
tasa_retencion.iplot(kind='line',xTitle='Edad',yTitle='Tasa de retención',
                     title='Nivel salarial')
```



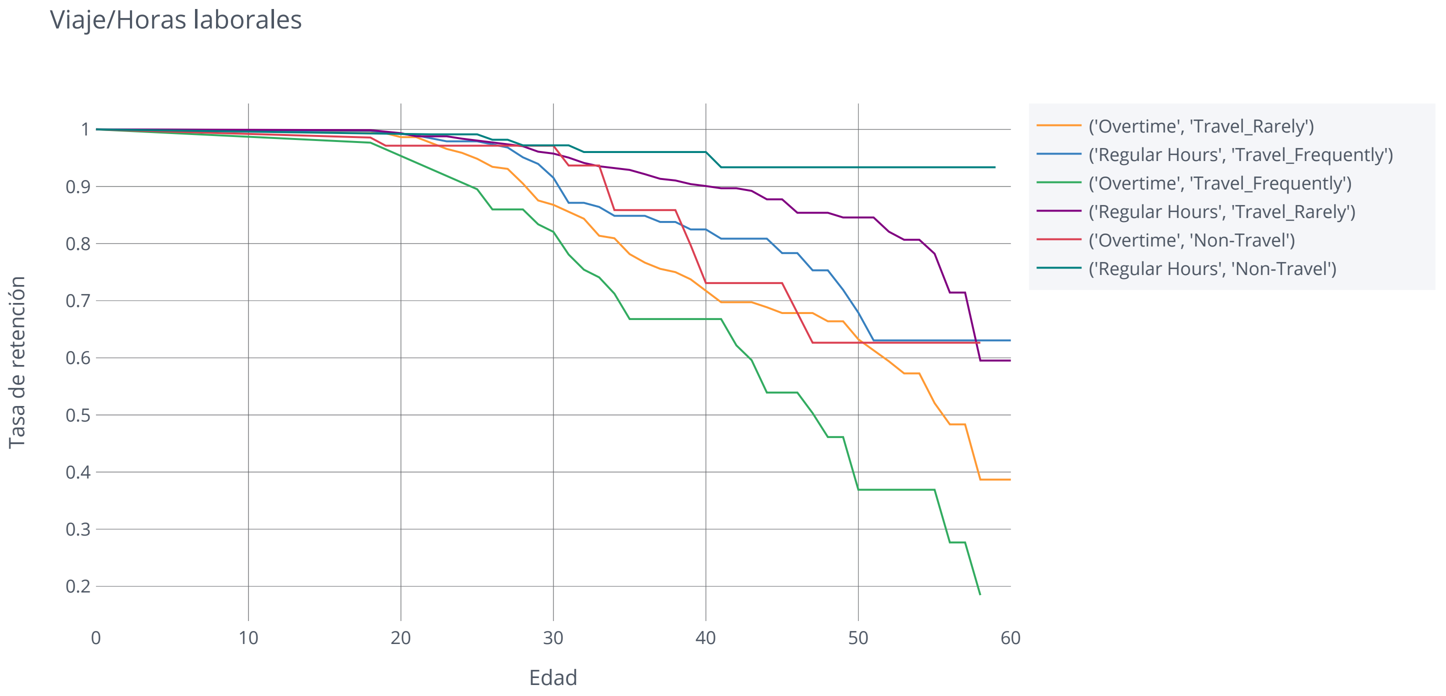
Export to plot.ly »

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 [44]:
```
data['Overtime/Regular'] = np.where(data['OverTime'] == 'Yes', 'Overtime', 'Regular Hours')
data['Travel+Over'] = list(zip(data['Overtime/Regular'],data['BusinessTravel']))

tasa_retencion = survival(data,'Travel+Over','Age','Attrition')
tasa_retencion.iplot(kind='line',xTitle='Edad',yTitle='Tasa de retención',
                     title='Viaje/Horas laborales')
```



Export to plot.ly »