Human Resources analysis

The goal of this notebook is to investigate the factors that make an employee le
ave his/her company.

Main question to be answered

- Which variables contribute the most to an employee leaving the company?

Outline

- Reading data and importing libaries
- Basic exploration
- Features
- Cleaning the Data
- General overview by basic plots
- Probabilistic analysis

Reading the Data

In [1]:

```
# Importing libraries

%matplotlib inline

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:

```
sns.set_style('whitegrid')
sns.despine()
```

```
<matplotlib.figure.Figure at 0x269e4f18978>
```

In [3]:

```
hr = pd.read_csv("HR_comma_sep.csv")
```

In [4]:

```
HR_copy=hr.copy() # A copy of the data set is made for probability analysis
```

Basic Exploration - Is there any problem with our dataset?

In [5]:

```
hr.head(3)
```

Out[5]:

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_
0	0.38	0.53	2	157	3
1	0.80	0.86	5	262	6
2	0.11	0.88	7	272	4

In [6]:

```
hr.tail(3)
```

Out[6]:

	satisfaction_level	last_evaluation	number_project	average_montly_hours
14996	0.37	0.53	2	143
14997	0.11	0.96	6	280
14998	0.37	0.52	2	158

In [7]:

```
hr.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14999 entries, 0 to 14998
Data columns (total 10 columns):
satisfaction_level       14999 non-null float64
last_evaluation          14999 non-null float64
number_project           14999 non-null int64
average_montly_hours     14999 non-null int64
time_spend_company       14999 non-null int64
Work_accident            14999 non-null int64
left                     14999 non-null int64
promotion_last_5years    14999 non-null int64
sales                    14999 non-null object
salary                   14999 non-null object
dtypes: float64(2), int64(6), object(2)
memory usage: 1.1+ MB
```

In [8]:

```
print('Percent of employees who left: {}'.format(round(hr.left.sum()/hr.shape[0]*100)))
```

```
Percent of employees who left: 24
```

In [9]:

```
hr.describe()
```

Out[9]:

	satisfaction_level	last_evaluation	number_project	average_montly_hours
count	14999.000000	14999.000000	14999.000000	14999.000000
mean	0.612834	0.716102	3.803054	201.050337
std	0.248631	0.171169	1.232592	49.943099
min	0.090000	0.360000	2.000000	96.000000
25%	0.440000	0.560000	3.000000	156.000000
50%	0.640000	0.720000	4.000000	200.000000
75%	0.820000	0.870000	5.000000	245.000000
max	1.000000	1.000000	7.000000	310.000000

In [10]:

```
hr.isnull().values.any()
```

Out[10]:

```
False
```

Findings:
So far, the dataset seems to be clean and not much is needed to make sure all parts are easily analysed. There are no missing values and only sales and salary contain strings. Salary seems like a strange name for what seems to be departments, so that will be changed later. About 1/4th of the employees left, which at first seems like a substantial number. However, it is not known why those employees left the company which makes analyses somewhat more difficult. Next, we will take a more in depth look at the features.

What are the features?

According to website below, the following are the features:

- **satisfaction_level**: Level of satisfaction between 0 and 1
- **last_evaluation**: Their last evaluation between 0 and 1
- **number_project**: Number of projects (ranging from 2 to 7)
- **average_montly_hours**: The average work hours per month at workplace (ranging from 96 to 310)
- **time_spend_company**: The time spend in the company (in years ranging from 2 to 10)
- **Work_accident**: Whether they have had a work accident
- **left**: If they left the company (0 = no, 1 = yes)
- **promotion_last_5years**: Whether they have had a promotion in the last 5 years
- **sales**: Department (sales, accounting, hr, technical, support, management, IT, product_mng, marketing, RandD)
- **salary**: The salary of employees (low, medium, high)

Source: https://www.kaggle.com/ludobenistant/hr-analytics (https://www.kaggle.com/ludobenistant/hr-analytics)

Next, we will look at the values and whether some of them need to be changed.

In [11]:

```
print('Unique values of:\n')
print('number_project \n{}\n'.format(hr["number_project"].unique()))
print('salary \n{}\n'.format(hr["salary"].unique()))
print('Work_accident \n{}\n'.format(hr["Work_accident"].unique()))
print('time_spend_company \n{}\n'.format(hr["time_spend_company"].unique()))
print('sales \n{}\n'.format(hr["sales"].unique()))
print('promotion_last_5years \n{}\n'.format(hr["promotion_last_5years"].unique()))
print('average_montly_hours: \n{}\n'.format(hr["average_montly_hours"].unique()))
print('last_evaluation: \n{}\n'.format(hr["last_evaluation"].unique()))
print('satisfaction_level: \n{}\n'.format(hr["satisfaction_level"].unique()))
```

```
Unique values of:

number_project
[2 5 7 6 4 3]

salary
['low' 'medium' 'high']

Work_accident
[0 1]

time_spend_company
[ 3  6  4  5  2  8 10  7]

sales
['sales' 'accounting' 'hr' 'technical' 'support' 'management' 'IT'
 'product_mng' 'marketing' 'RandD']

promotion_last_5years
[0 1]

average_montly_hours:
[157 262 272 223 159 153 247 259 224 142 135 305 234 148 137 143 160 255
 282 147 304 139 158 242 239 128 132 294 134 145 140 246 126 306 152 269
 127 281 276 182 273 307 309 225 226 308 244 286 161 264 277 275 149 295
 151 249 291 232 130 129 155 265 279 284 221 154 150 267 257 177 144 289
 258 263 251 133 216 300 138 260 183 250 292 283 245 256 278 240 136 301
 243 296 274 164 146 261 285 141 297 156 287 219 254 228 131 252 236 270
 298 192 248 266 238 229 233 268 231 253 302 271 290 235 293 241 218 199
 180 195 237 227 172 206 181 217 310 214 198 211 222 213 202 184 204 288
 220 299 303 212 196 179 205 230 203 280 169 188 178 175 166 163 168 165
 189 162 215 193 176 191 174 201 208 171 111 104 106 100 194 209 185 200
 207 187 210 186 167 108 122 110 115 197 102 109 190  99 101  97 173 121
 170 105 118 119 117 114  96  98 107 123 116 125 113 120 112 124 103]

last_evaluation:
[ 0.53  0.86  0.88  0.87  0.52  0.5   0.77  0.85  1.    0.54  0.81  0.92
  0.55  0.56  0.47  0.99  0.51  0.89  0.83  0.95  0.57  0.49  0.46  0.62
  0.94  0.48  0.8   0.74  0.7   0.78  0.91  0.93  0.98  0.97  0.79  0.59
  0.84  0.45  0.96  0.68  0.82  0.9   0.71  0.6   0.65  0.58  0.72  0.67
  0.75  0.73  0.63  0.61  0.76  0.66  0.69  0.37  0.64  0.39  0.41  0.43
  0.44  0.36  0.38  0.4   0.42]

satisfaction_level:
[ 0.38  0.8   0.11  0.72  0.37  0.41  0.1   0.92  0.89  0.42  0.45  0.84
  0.36  0.78  0.76  0.09  0.46  0.4   0.82  0.87  0.57  0.43  0.13  0.44
  0.39  0.85  0.81  0.9   0.74  0.79  0.17  0.24  0.91  0.71  0.86  0.14
  0.75  0.7   0.31  0.73  0.83  0.32  0.54  0.27  0.77  0.88  0.48  0.19
  0.6   0.12  0.61  0.33  0.56  0.47  0.28  0.55  0.53  0.59  0.66  0.25
  0.34  0.58  0.51  0.35  0.64  0.5   0.23  0.15  0.49  0.3   0.63  0.21
  0.62  0.29  0.2   0.16  0.65  0.68  0.67  0.22  0.26  0.99  0.98  1.
  0.52  0.93  0.97  0.69  0.94  0.96  0.18  0.95]
```

Findings:

A quick inspection of the unique values in each column shows that there are no weird values that should be considered when analysing the data. This does not regard outliers, merely NaN (i.e., missing data), strings etc.

Cleaning Data

There are some features that have strings as values. We will change them so they can be more easily viewed and analysed later. Furthermore, the labels of departments and salary is kept so they can be used in plotting.

In [12]:

```
salary_labels = {'low':0,'medium':1,'high':2}
hr['salary'] = hr['salary'].map(salary_labels)
```

In [13]:

```
hr.rename(columns={'sales': 'department'}, inplace=True)

department_labels = {'sales':0,'accounting':1,'hr':2, 'technical':3, 'support':4,
                     'management':5, 'IT':6, 'product_mng':7, 'marketing':8, 'RandD':9}
hr['department'] = hr['department'].map(department_labels)

hr.loc[hr['satisfaction_level'] < .2, 'satisfaction_bins'] = 0
hr.loc[(hr['satisfaction_level'] >= .2) & (hr['satisfaction_level'] < .4), 'satisfactio
n_bins'] = 1
hr.loc[(hr['satisfaction_level'] >= .4) & (hr['satisfaction_level'] < .6), 'satisfactio
n_bins'] = 2
hr.loc[(hr['satisfaction_level'] >= .6) & (hr['satisfaction_level'] < .8), 'satisfactio
n_bins'] = 3
hr.loc[hr['satisfaction_level'] >= .8, 'satisfaction_bins'] = 4

hr.loc[hr['last_evaluation'] < .2, 'evaluation_bins'] = 0
hr.loc[(hr['last_evaluation'] >= .2) & (hr['last_evaluation'] < .4), 'evaluation_bins']
 = 1
hr.loc[(hr['last_evaluation'] >= .4) & (hr['last_evaluation'] < .6), 'evaluation_bins']
 = 2
hr.loc[(hr['last_evaluation'] >= .6) & (hr['last_evaluation'] < .8), 'evaluation_bins']
 = 3
hr.loc[hr['last_evaluation'] >= .8, 'evaluation_bins'] = 4
```

In [14]:

```
hr.head()
```

Out[14]:

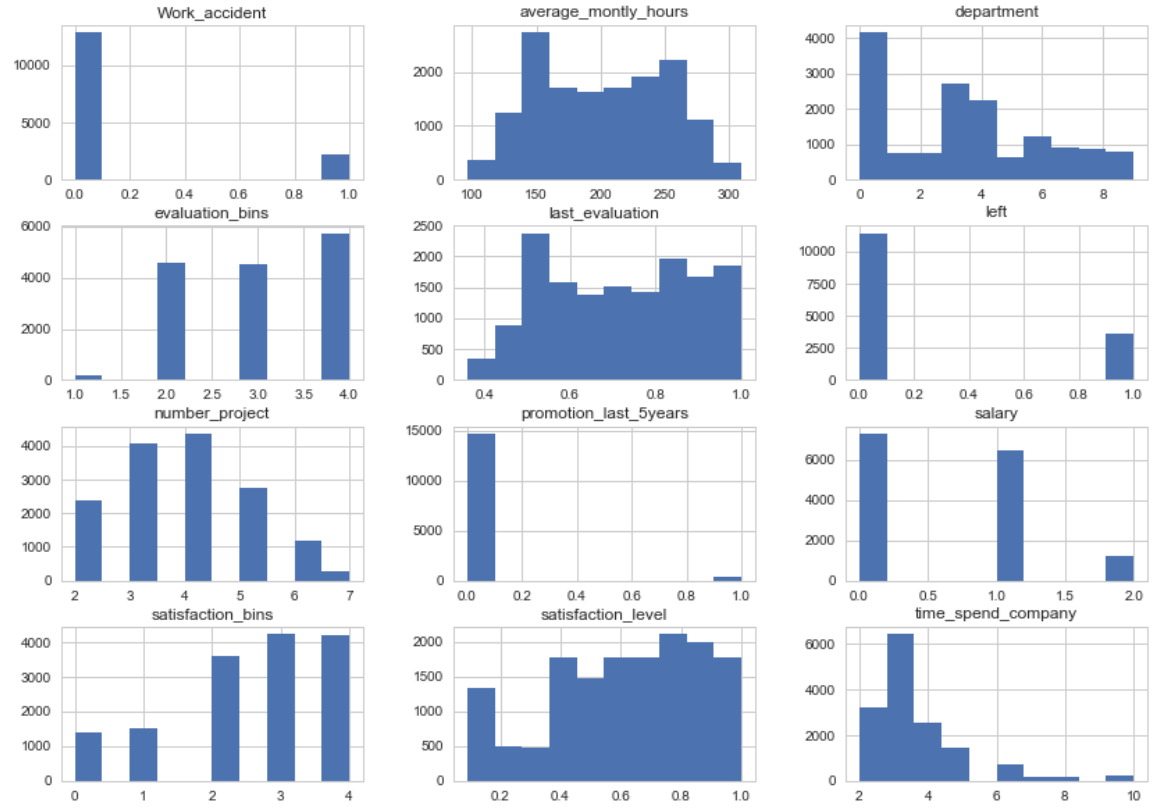
	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_
0	0.38	0.53	2	157	3
1	0.80	0.86	5	262	6
2	0.11	0.88	7	272	4
3	0.72	0.87	5	223	5
4	0.37	0.52	2	159	3

Basic Plots

We will start with some basic plots to simply see the distribution of the features. How does everything look? Are there any outliers we should be worried about? etc.

In [15]:

```
hr.hist(figsize =(14,10.0))
plt.show()
```



Findings:

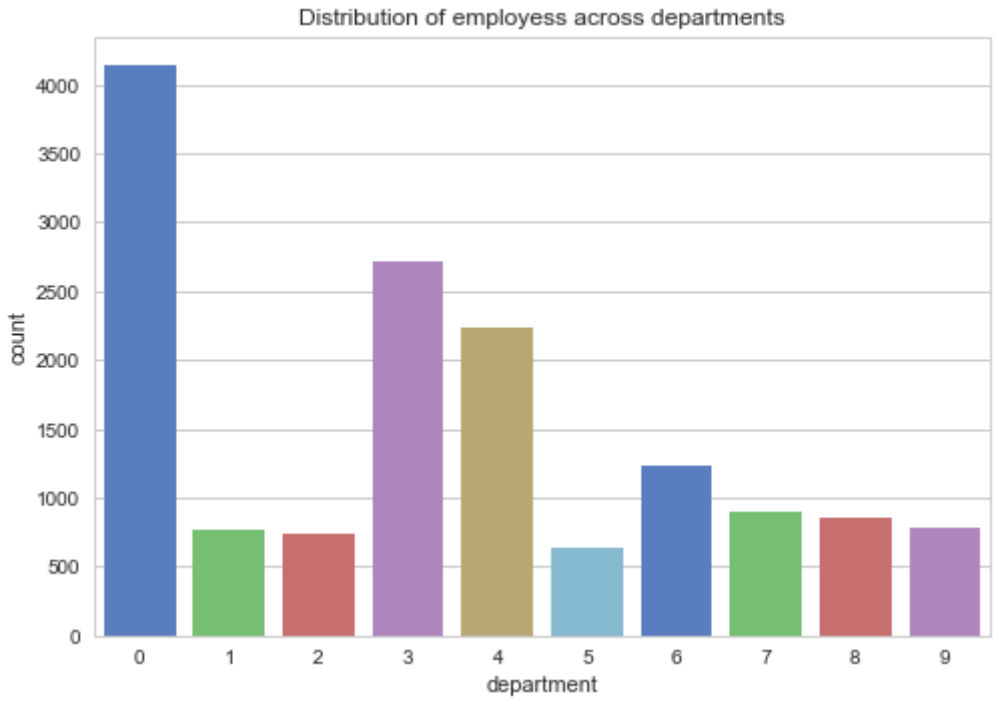
So far, the most surprising distribution is the very low percentage of employees that were promoted in the last 5 years. Thus, let's explore that first before we go into the feature 'left'.

In [16]:

```
sns.countplot(x="department", data=hr,palette="muted")
plt.title('Distribution of employess across departments')
```

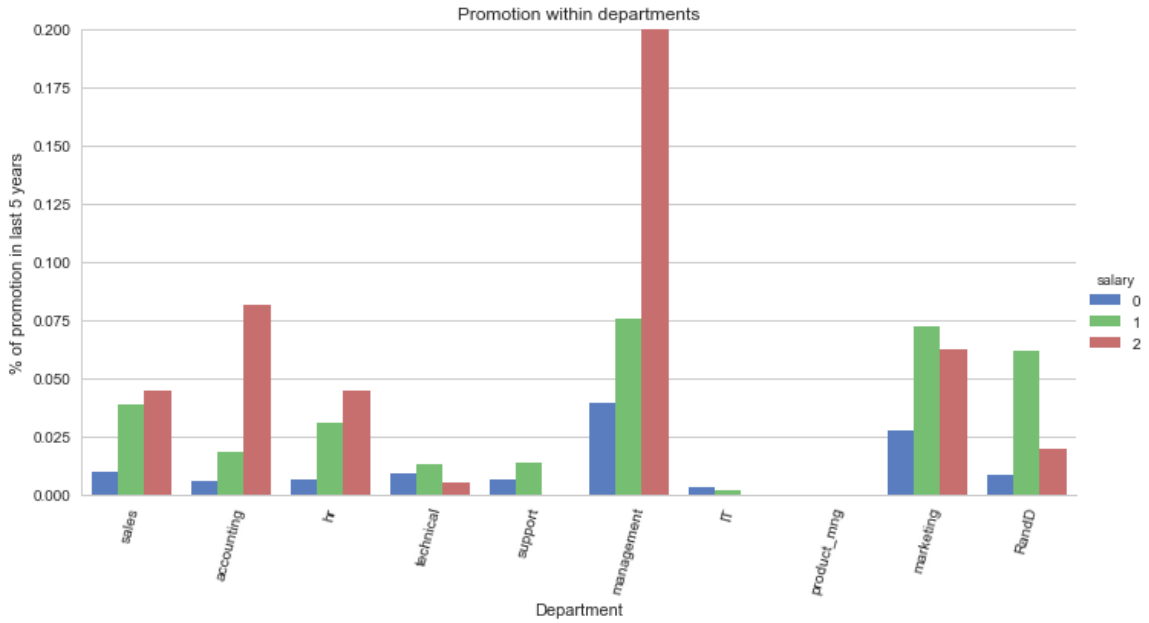
Out[16]:

```
<matplotlib.text.Text at 0x269e9c0c1d0>
```



In [17]:

```
sns.factorplot(y="promotion_last_5years", x="department", data=hr, kind = "bar",
               hue = "salary", size = 5, aspect = 2, palette='muted', ci=None)
plt.ylim(0,0.20)
plt.title('Promotion within departments')
plt.xlabel('Department')
plt.ylabel('% of promotion in last 5 years')
plt.xticks(list(department_labels.values()),department_labels.keys(),rotation=75)
plt.show()
```

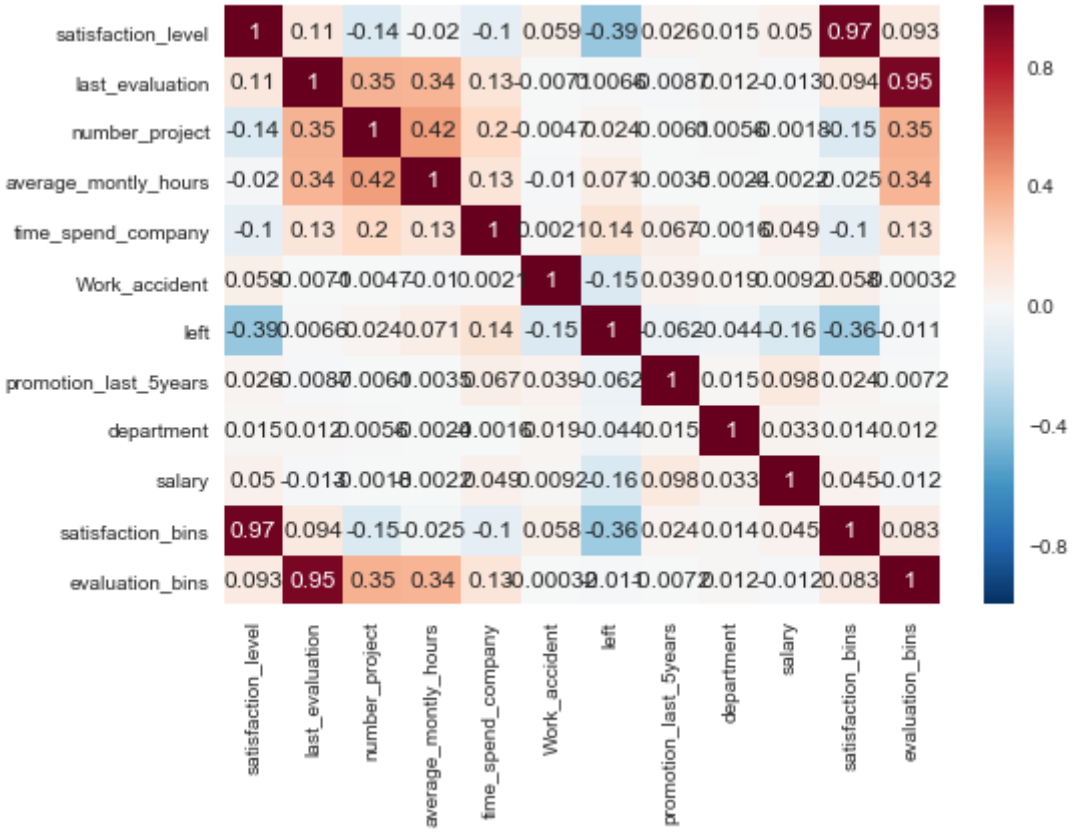


Findings:
Apparently, in most departments, employees with at least a medium salary are more likely to get promoted. This effect is even greater for employees with a high salary in the accounting and management departments. This suggests that the company is not a place where you'll want to be if you start off with a low salary. However, getting promoted is easier if you already have a high salary.

Correlations

In [18]:

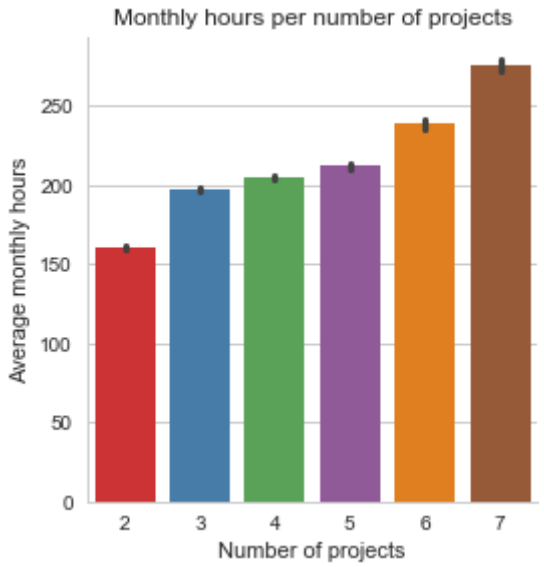
```
corr = hr.corr()
sns.heatmap(corr,
            xticklabels=corr.columns.values,
            yticklabels=corr.columns.values, annot=True, fmt='.2g')
plt.show()
```



Findings:
It seems that the number of projects and the average monthly hours is reasonably correlated. Furthermore, last evaluation and average monthly hours, and last evaluation and number of projects seems to be correlated. Thus, let's plot those correlations before going into the leaving of employees.

In [19]:

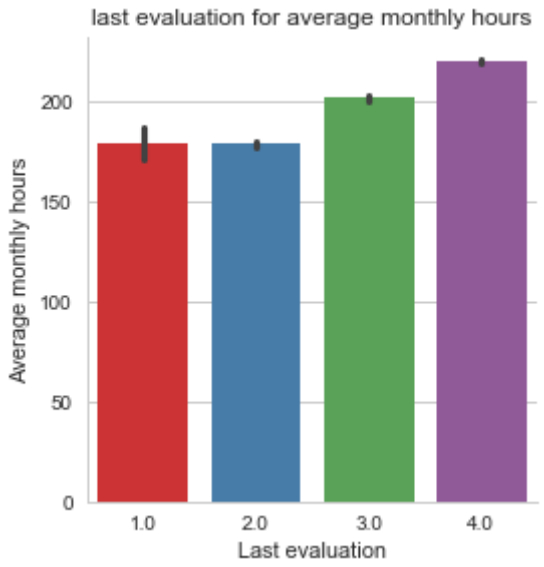
```
palette = ["#e41a1c", "#377eb8", '#4daf4a','#984ea3','#ff7f00','#a65628','#f781bf']
sns.factorplot(x='number_project',y='average_montly_hours',data=hr,kind='bar',
palette=palette)
plt.title('Monthly hours per number of projects')
plt.xlabel('Number of projects')
plt.ylabel('Average monthly hours')
plt.show()
```



The correlation that we have seen in the correlation matrix above suggested a correlation between number of projects and average monthly hours. We can clearly see that as the number of projects increases, the number of average monthly hours also increases. This makes a lot of sense, seeing that having more projects results in more work and therefore, more hours spent on that work.

In [22]:

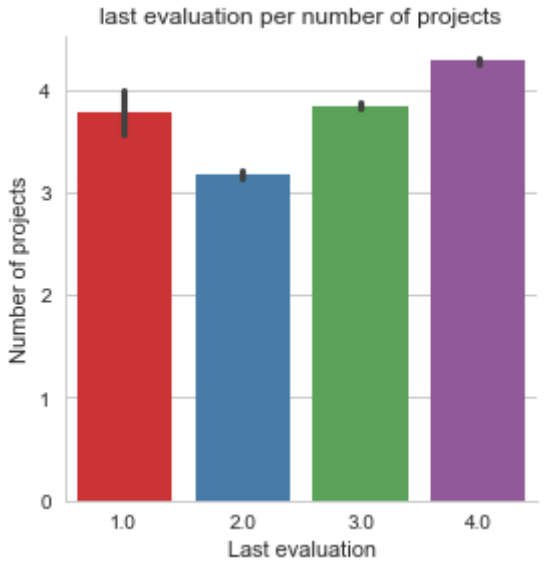
```
palette = ["#e41a1c", "#377eb8", '#4daf4a','#984ea3','#ff7f00','#a65628','#f781bf']
sns.factorplot(x='evaluation_bins',y='average_montly_hours',data=hr,kind='bar',
palette=palette)
plt.title('last evaluation for average monthly hours')
plt.xlabel('Last evaluation')
plt.ylabel('Average monthly hours')
plt.show()
```



Similarly, we noticed a correlation between evaluation and average monthly hours. As seen before, it seems that the better the last evaluation, the more hours employees have spent on their work. It may be that because they worked more hours, their evaluation would increase, but that is only a hypothesis.

In [20]:

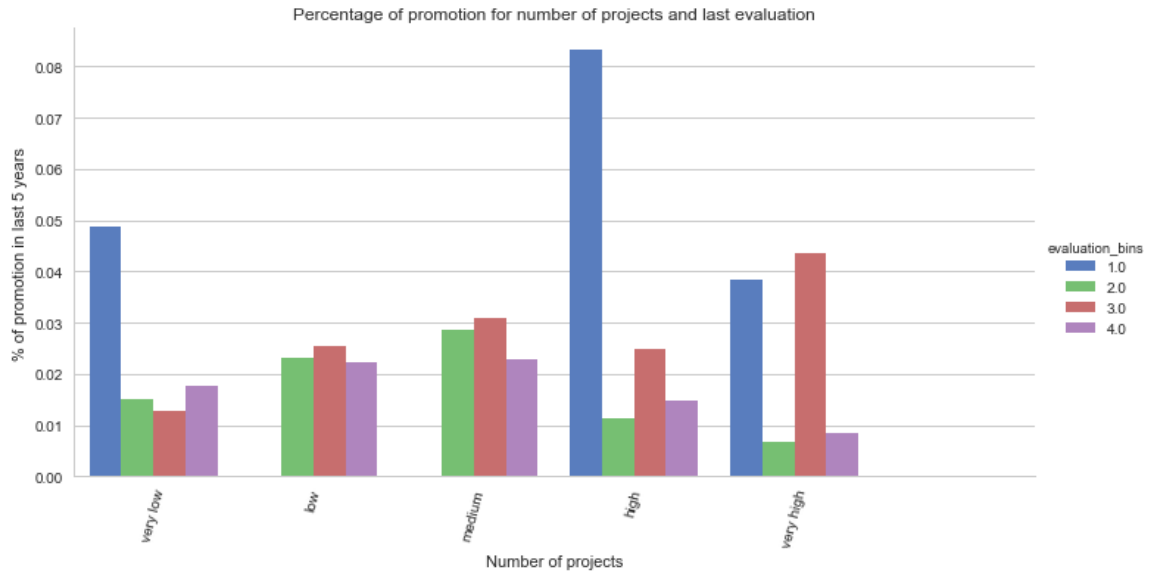
```
palette = ["#e41a1c", "#377eb8", '#4daf4a','#984ea3','#ff7f00','#a65628','#f781bf']
sns.factorplot(x='evaluation_bins',y='number_project',data=hr,kind='bar', palette=palet
te)
plt.title('last evaluation per number of projects')
plt.xlabel('Last evaluation')
plt.ylabel('Number of projects')
plt.show()
```



For an evaluation of .4 and higher we see that the higher the evaluation the more likely an employee will have more projects to work on. Seeing as how the number of projects and average monthly hours were correlated, this seems to be in line with the previous plots.

In [21]:

```
sns.factorplot(y="promotion_last_5years", x="number_project", data=hr, kind = "bar",
               hue = "evaluation_bins", size = 5, aspect = 2, palette='muted', ci=None,
 legend= True)
plt.xticks([0,1,2,3,4],['very low', 'low', 'medium', 'high', 'very high'],rotation=75)
plt.title('Percentage of promotion for number of projects and last evaluation')
plt.xlabel('Number of projects')
plt.ylabel('% of promotion in last 5 years')
plt.show()
```



Findings:
Interestingly, we see that the last evaluation and number of projects is not dependent on the chances of getting a promotion. Thus, although the number of projects are more likely to get employees a higher evaluation, it seems to not influence the chances of them getting a promotion.

Why do they leave?

Now, for the main question. Why are employees leaving? We have seen that evaluation seem to be based on the number of projects and hours worked, but that there is little to do with getting a promotion.

In [23]:

```
sns.pairplot(hr, hue="left", palette = "husl")
```

Out[23]:

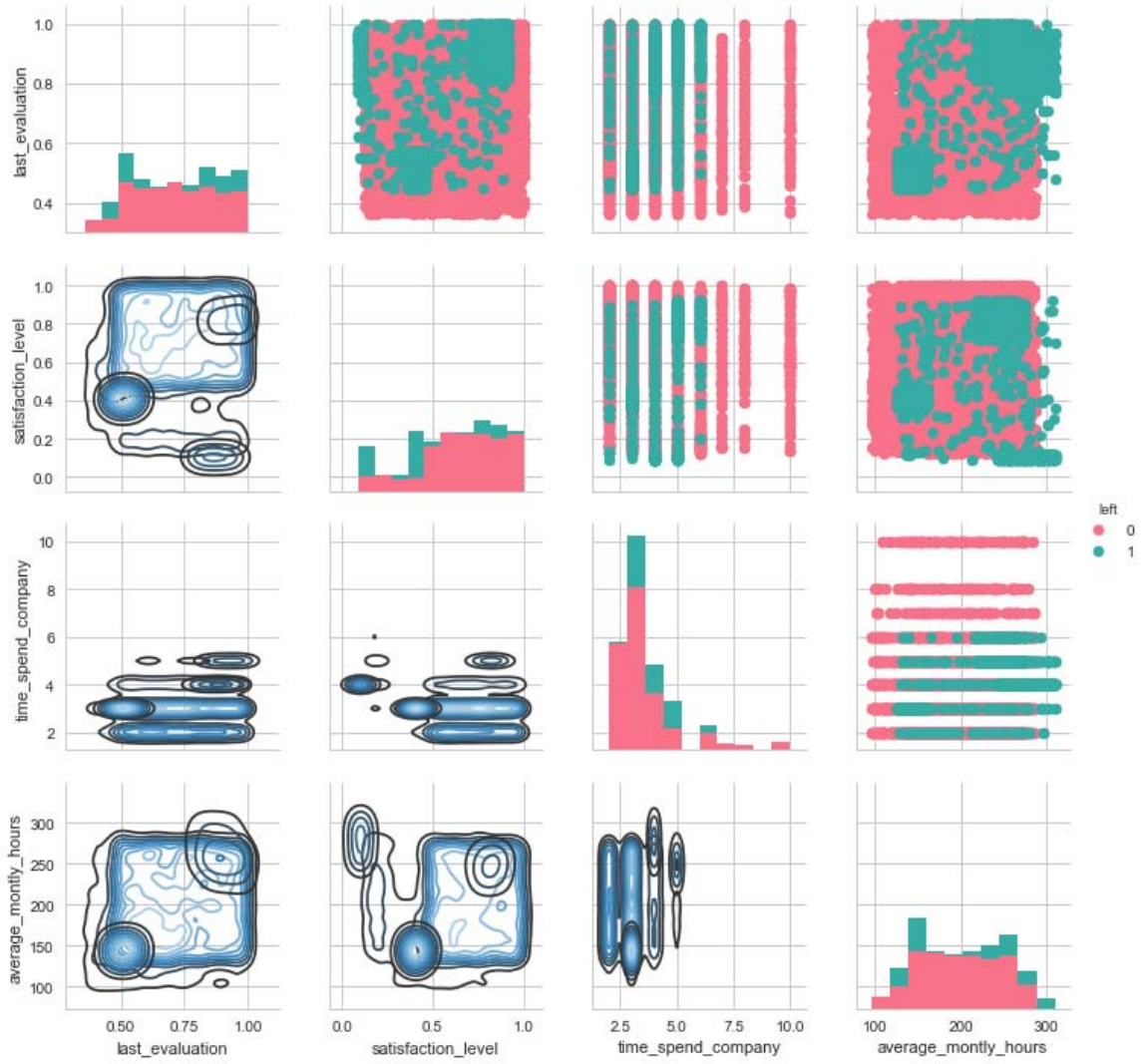
<seaborn.axisgrid.PairGrid at 0x269f2eb6048>

Although a pair plot creates a large pair of plots, we can immediately see some interesting clusters. If we mainly look at last evaluation, satisfaction level, average monthly hours and time spend at company, we can see that some values are clusters very closely together. However, that is difficult to see in the plot above, so we go into a narrower view using those four features.

In [24]:

```
g = sns.PairGrid(hr, vars=["last_evaluation",
"satisfaction_level","time_spend_company", "average_montly_hours"],
                 hue='left', palette = "husl")
g = g.map_diag(plt.hist)
g = g.map_upper(plt.scatter)
g = g.map_lower(sns.kdeplot, cmap="Blues_d")

g = g.add_legend()
```



In the pair grid above we can see the suggested relations more clearly. Especially for average monthly hours, satisfaction level and last evaluation we can see some clusters that indicate that employees are almost certain to leave. Thus, let's plot those features in more detail.

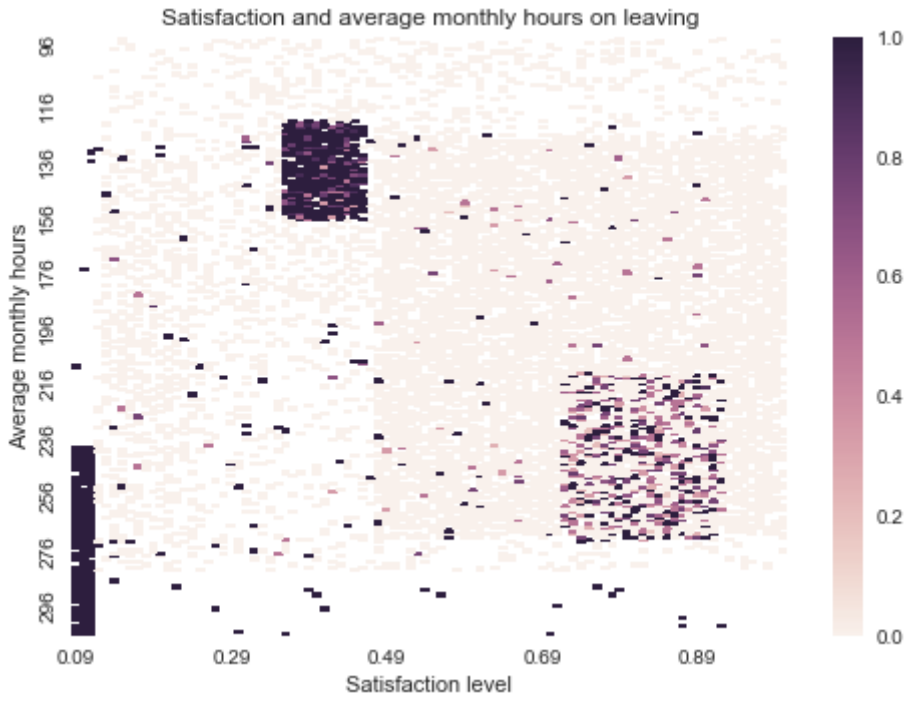
Heatmaps

In [25]:

```
hr_pivot = hr.pivot_table(index='last_evaluation', columns='satisfaction_level',
                          values='left')
sns.heatmap(hr_pivot, xticklabels=20, yticklabels=20, linecolor='white')
plt.title('Satisfaction and last evaluation on leaving')
plt.xlabel('Satisfaction level')
plt.ylabel('Last evalution')
plt.show()
```

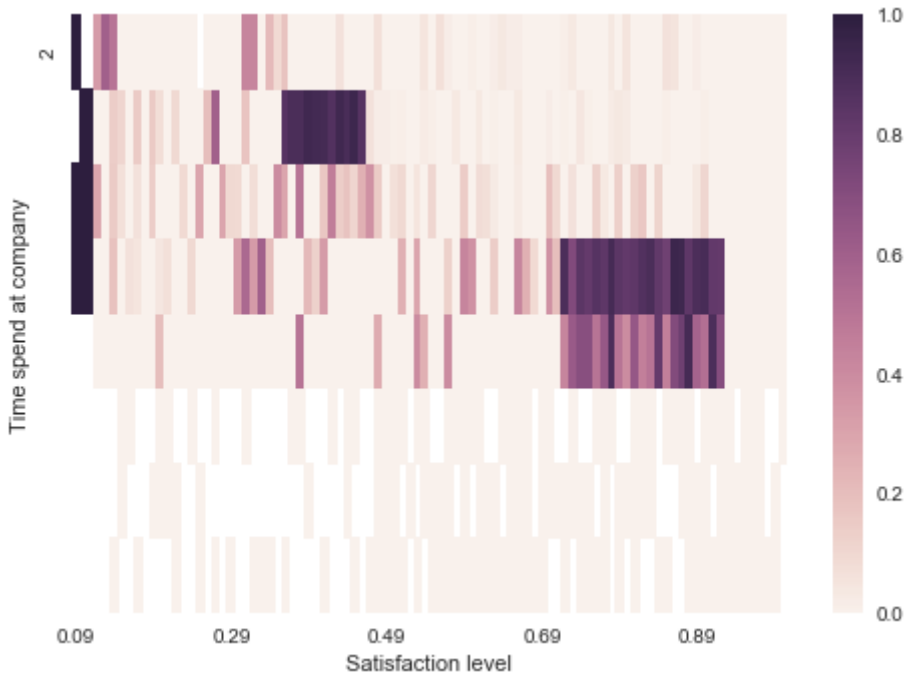
In [26]:

```
hr_pivot = hr.pivot_table(index='average_montly_hours', columns='satisfaction_level',
                          values='left')
sns.heatmap(hr_pivot, xticklabels=20, yticklabels=20, linecolor='white')
plt.title('Satisfaction and average monthly hours on leaving')
plt.xlabel('Satisfaction level')
plt.ylabel('Average monthly hours')
plt.show()
```



In [27]:

```
hr_pivot = hr.pivot_table(index='time_spend_company', columns='satisfaction_level',
                          values='left')
sns.heatmap(hr_pivot, xticklabels=20, yticklabels=20, linecolor='white')
plt.xlabel('Satisfaction level')
plt.ylabel('Time spend at company')
plt.show()
```



Findings:

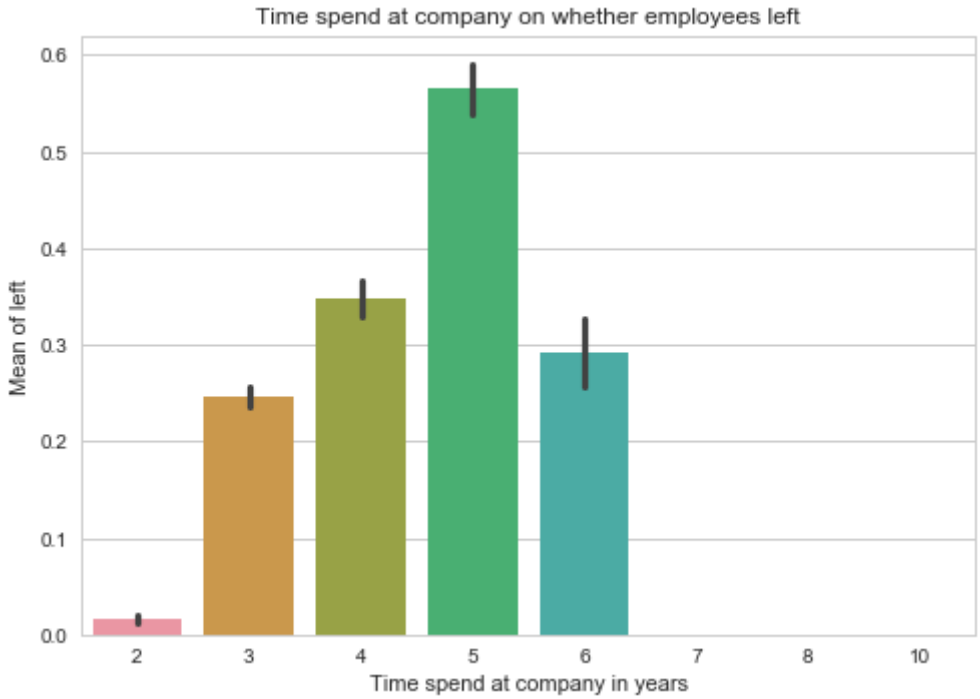
It seems that the features that we have selected for further exploration are sufficient to warrant some more in-depth plots. Next, we will look at each of the features.

Time spend at company

Time spend at the company showed some clusters in the pair plots, so we looked at it more in depth by plotting it versus left, number of projects and department.

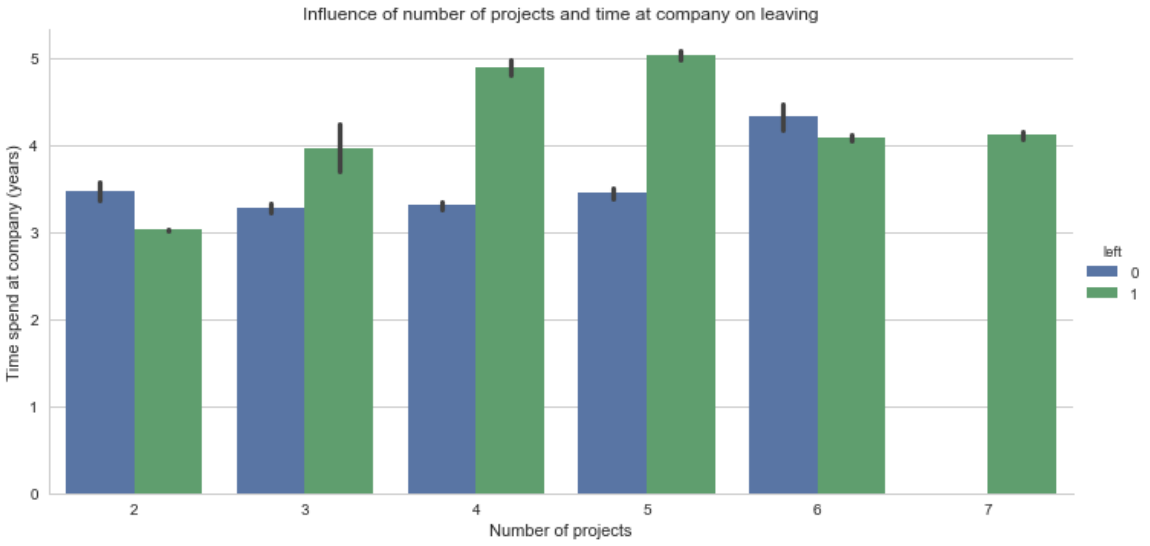
In [28]:

```
sns.barplot(x = 'time_spend_company', y = 'left', data = hr)
plt.xlabel('Time spend at company in years')
plt.ylabel('Mean of left')
plt.title('Time spend at company on whether employees left')
plt.show()
```



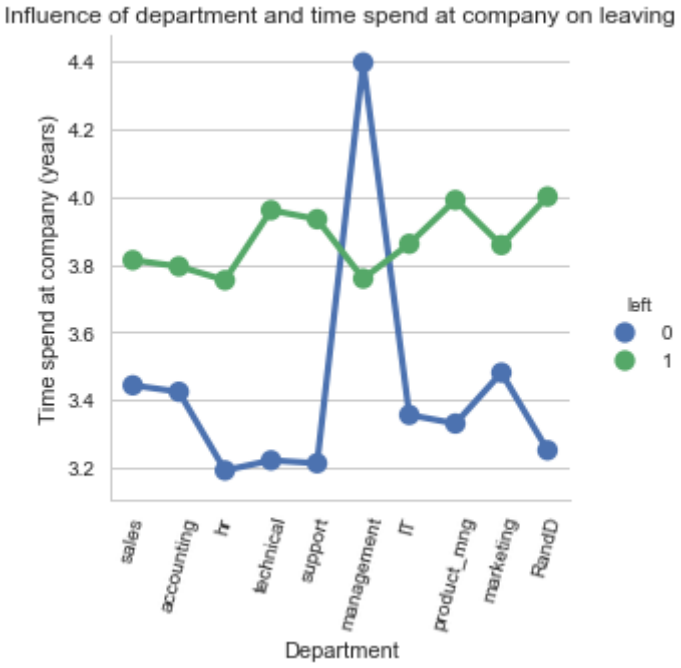
In [29]:

```
sns.factorplot(y="time_spend_company", x="number_project", data=hr, kind = "bar", hue =
 "left", size = 5, aspect = 2)
plt.xlabel('Number of projects')
plt.ylabel('Time spend at company (years)')
plt.title('Influence of number of projects and time at company on leaving')
plt.show()
```



In [30]:

```
sns.factorplot(x='department', y='time_spend_company', hue='left',data=hr, ci=None)
plt.xlabel('Department')
plt.ylabel('Time spend at company (years)')
labels = ['sales','accounting','hr', 'technical', 'support',
                    'management', 'IT', 'product_mng', 'marketing', 'RandD']
plt.xticks([0,1,2,3,4,5,6,7,8,9],labels,rotation=75)
plt.title('Influence of department and time spend at company on leaving')
plt.show()
```

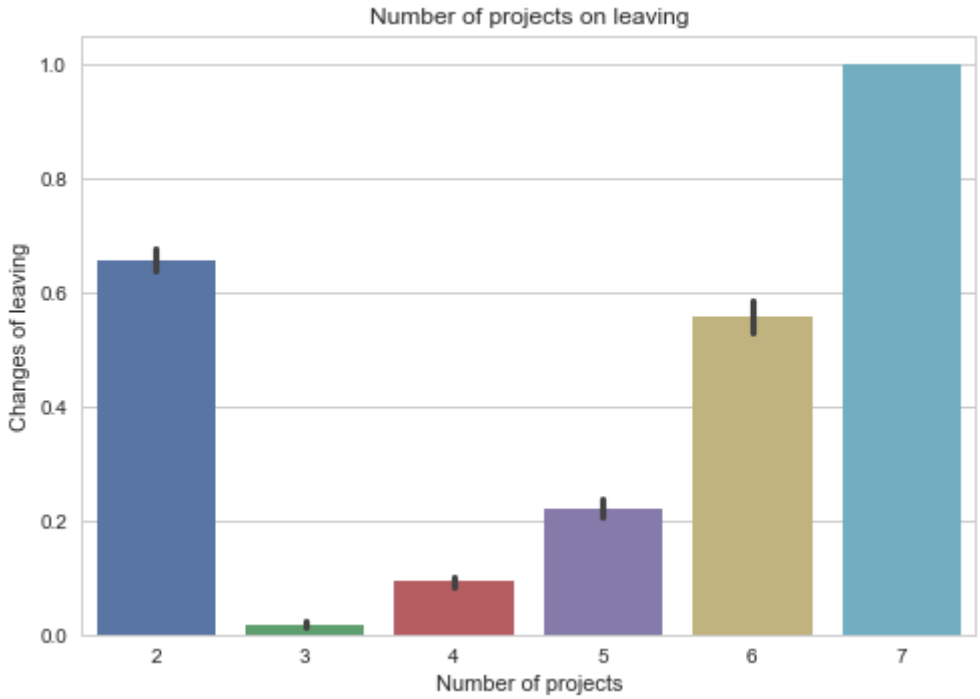


What we can clearly see is that management is more likely to stay if they spend more time at the company, which is in contrast with the other departments. It might be that employees in the management department make more money and have a higher change to receive a promotion.

Satisfaction level and number of projects

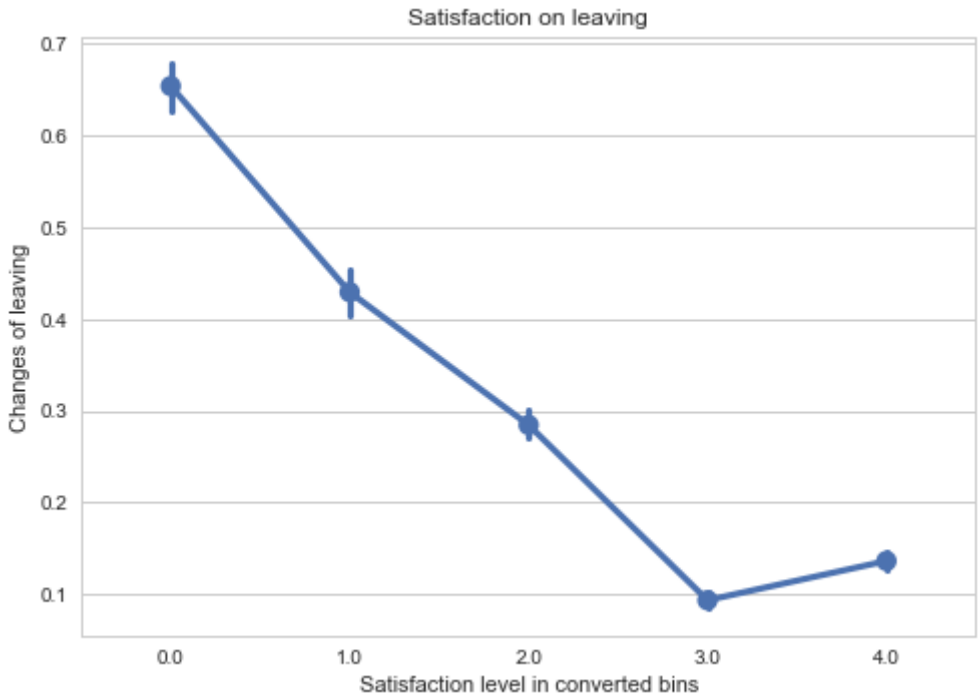
In [31]:

```
sns.barplot(x = 'number_project', y = 'left', data = hr)
plt.xlabel('Number of projects')
plt.title('Number of projects on leaving')
plt.ylabel('Changes of leaving')
plt.show()
```



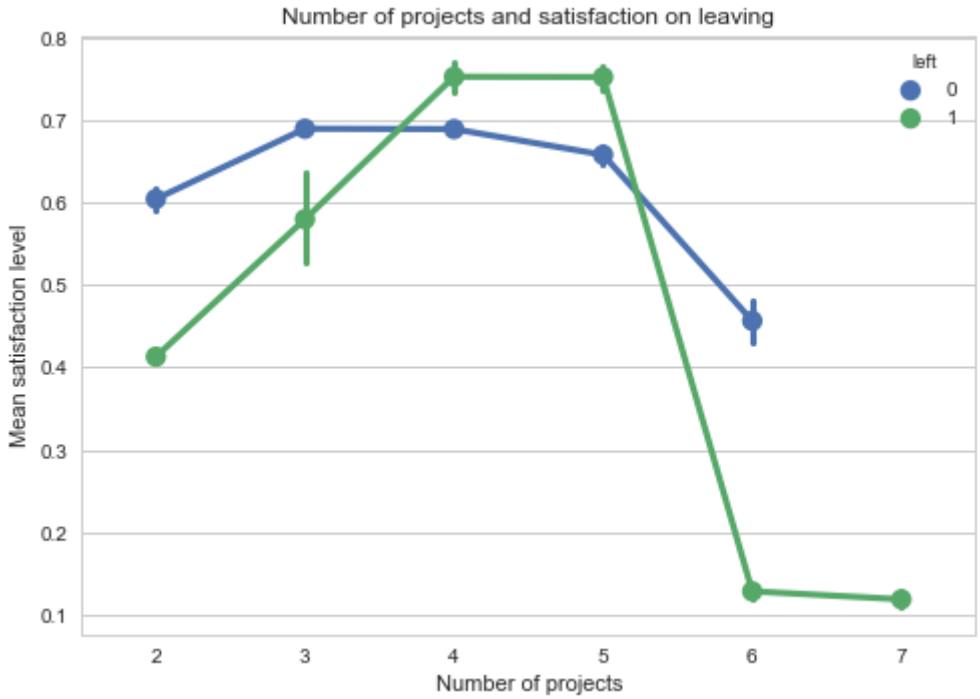
In [32]:

```
sns.pointplot(hr['satisfaction_bins'],hr['left'])
plt.title('Satisfaction on leaving')
plt.xlabel('Satisfaction level in converted bins')
plt.ylabel('Changes of leaving')
plt.show()
```



In [33]:

```
sns.pointplot(hr['number_project'],hr['satisfaction_level'], hue=hr['left'])
plt.title('Number of projects and satisfaction on leaving')
plt.xlabel('Number of projects')
plt.ylabel('Mean satisfaction level')
plt.show()
```



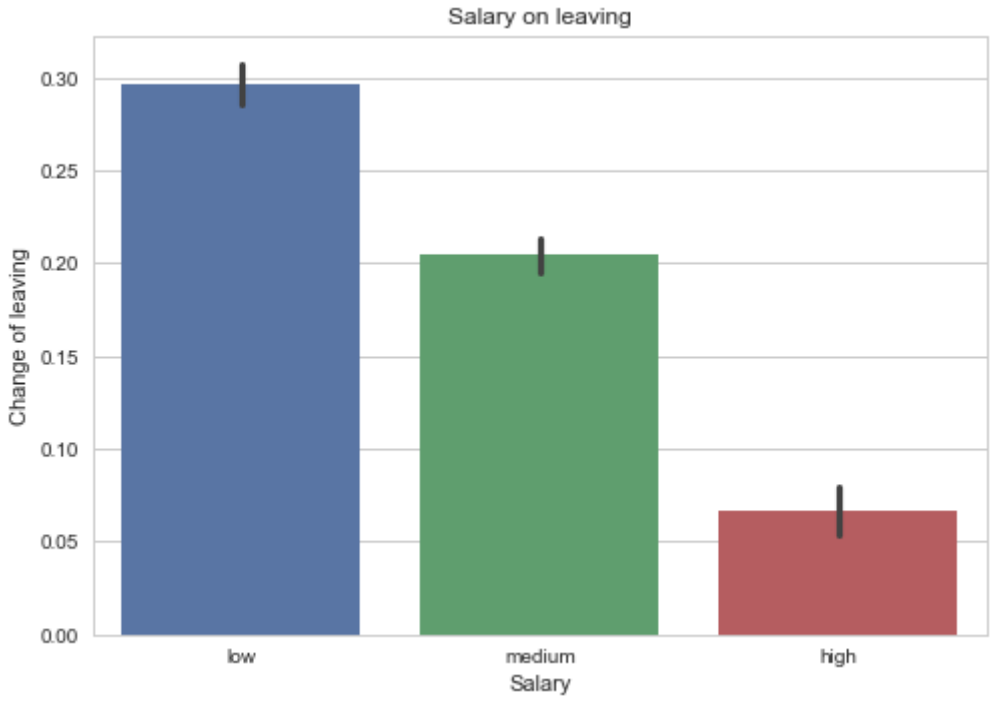
From these plots, we can see that when an employee must work on more projects, the chances of leaving are gets higher, except for when an employee is assigned for only two projects. In addition, counterintuitive, on average, someone that gave their job satisfaction level a 4 has a higher chance of leaving than those who ranged their job satisfaction level a 3.

From the last plot, we see how the number of projects is correlated with satisfaction rate for the employees how stayed or left.

General plots on leaving

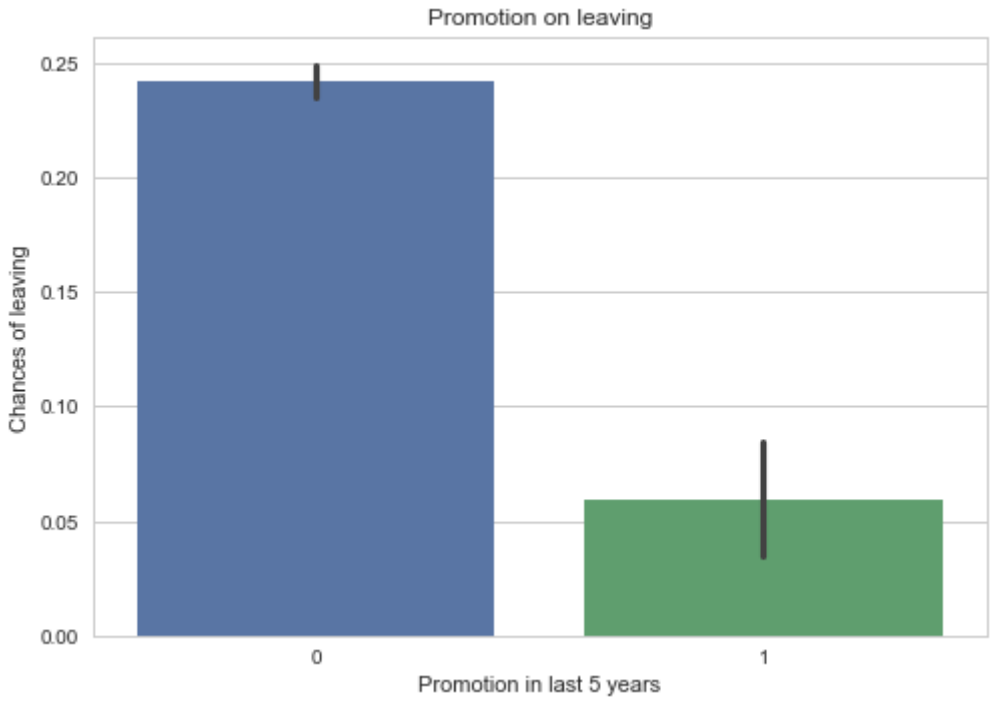
In [34]:

```
sns.barplot(x = 'salary', y = 'left', data = hr)
plt.xlabel('Salary')
plt.ylabel('Change of leaving')
plt.title('Salary on leaving')
plt.xticks([0,1,2],['low','medium','high'])
plt.show()
```



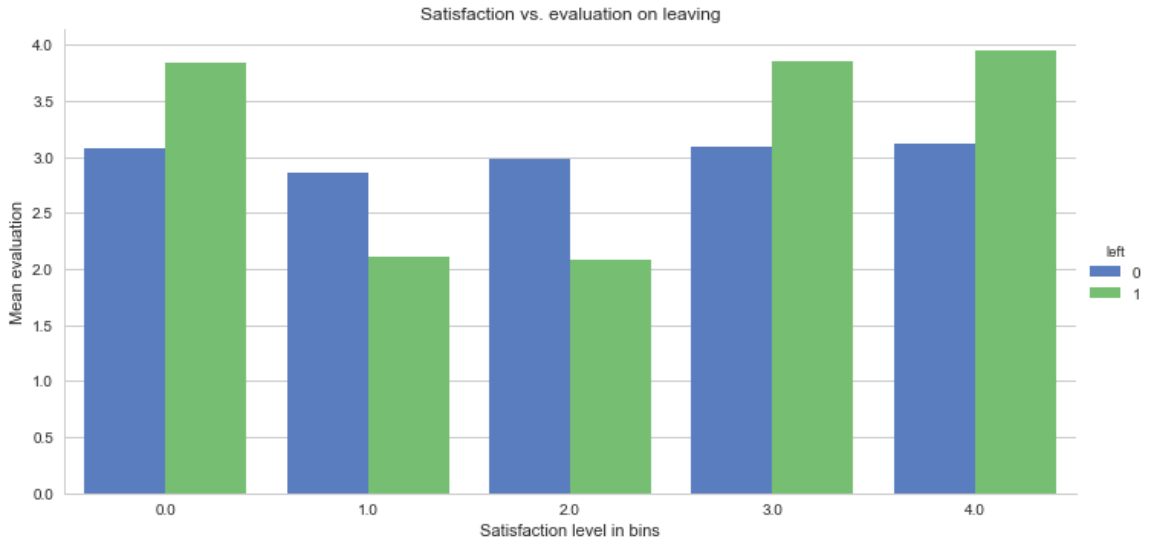
In [35]:

```
sns.barplot(x = 'promotion_last_5years', y = 'left', data = hr)
plt.xlabel('Promotion in last 5 years')
plt.ylabel('Chances of leaving')
plt.title('Promotion on leaving')
plt.show()
```



In [36]:

```
sns.factorplot(y="evaluation_bins", x="satisfaction_bins", data=hr, kind = "bar", hue =
 "left",
               size = 5, aspect = 2, palette='muted', ci=None)
plt.xlabel('Satisfaction level in bins')
plt.ylabel('Mean evaluation')
plt.title('Satisfaction vs. evaluation on leaving')
plt.show()
```



From these plots, we see that the change on leaving gets lower if the salary is higher and when there was a promotion in the past 5 years.

Probability analysis

Before starting the probability analysis some manipulation on the dataset is necessary, since we will be using, for example, the 'department' column.

In [37]:

```
HR_copy.rename(columns={"sales":"department", "Work_accident":"work_accident",
"left":"left_job"},inplace=True)
HR_copy.head(3)
```

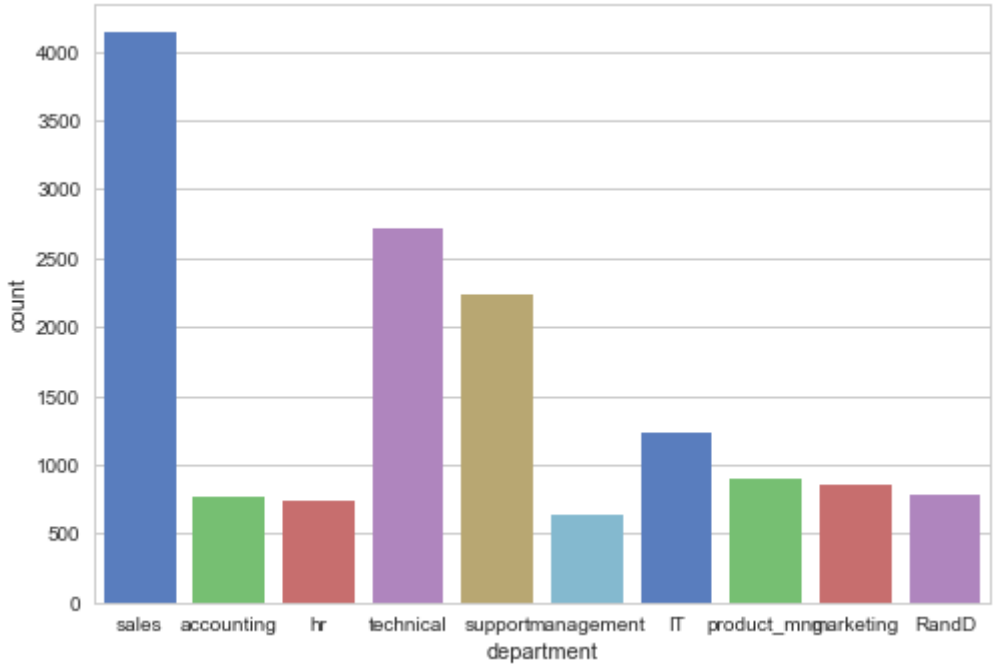
Out[37]:

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_
0	0.38	0.53	2	157	3
1	0.80	0.86	5	262	6
2	0.11	0.88	7	272	4

How many employees distributed by department?

In [38]:

```
ax = sns.countplot(x="department", data=HR_copy,palette="muted")
```



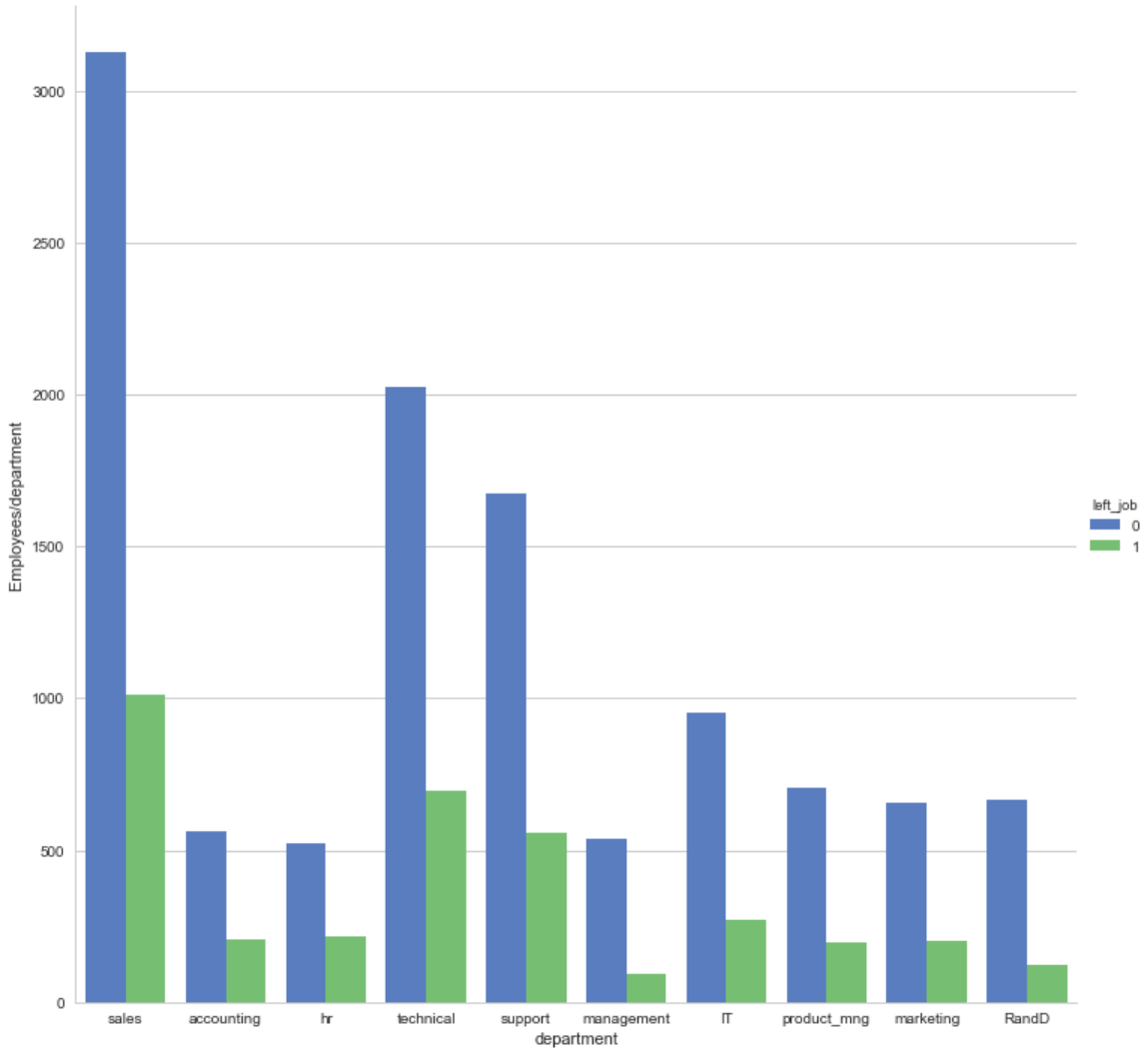
Percentage of employees that left per department

In [39]:

```
g = sns.factorplot(x="department", hue="left_job", kind="count",
                   data=HR_copy,size=10,palette="muted")
g.set_ylabels("Employees/department ")
```

Out[39]:

```
<seaborn.axisgrid.FacetGrid at 0x269862e04e0>
```



In [40]:

```
HR_copy['left_job'][HR_copy['department']=='sales'].value_counts(normalize = True)
```

Out[40]:

```
0    0.755072
1    0.244928
Name: left_job, dtype: float64
```

In [41]:

```
HR_copy.head()
```

Out[41]:

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_
0	0.38	0.53	2	157	3
1	0.80	0.86	5	262	6
2	0.11	0.88	7	272	4
3	0.72	0.87	5	223	5
4	0.37	0.52	2	159	3

In [42]:

```
# Creating a crosstab view to examine department relationship with the leaves
department_df = pd.crosstab(HR_copy.department, HR_copy.left_job, margins=True,
            normalize = 'index', rownames = ['Department Names'], colnames = ['Left Job
 Status'])
department_df = pd.DataFrame(department_df.loc[['accounting', 'hr', 'sales', 'technica
l', 'support', 'management', 'IT', 'product_mng', 'marketing','RandD'],:])
department_df.columns = ['Still Works', 'Left Job']
```

In [43]:

```
department_df.round(2)
```

Out[43]:

	Still Works	Left Job
Department Names		
accounting	0.73	0.27
hr	0.71	0.29
sales	0.76	0.24
technical	0.74	0.26
support	0.75	0.25
management	0.86	0.14
IT	0.78	0.22
product_mng	0.78	0.22
marketing	0.76	0.24
RandD	0.85	0.15

The HR department was the one with highest number of employees leaving, 29.09%. And management the one with least, 14.44%.

Satisfaction level X leaving job

In [44]:

```
HR_copy['satisfaction_level'].describe()
```

Out[44]:

```
count    14999.000000
mean         0.612834
std          0.248631
min          0.090000
25%          0.440000
50%          0.640000
75%          0.820000
max          1.000000
Name: satisfaction_level, dtype: float64
```

Taking in account the information obtained above (quartiles), we will calculate the probabilities of an employee leaving the job in relation to his/her level of satisfaction.

In [45]:

```
# Creating a new column to see satisfaction levels in categories
HR_copy['satisfaction_level_categorical'] = None
HR_copy.loc[HR_copy['satisfaction_level']<=0.44, 'satisfaction_level_categorical'] = 'L
ess Than 44%'
HR_copy.loc[(HR_copy['satisfaction_level']>0.44) & (HR_copy['satisfaction_level']
<=0.64), 'satisfaction_level_categorical'] = '44% - 64%'
HR_copy.loc[(HR_copy['satisfaction_level']>0.64) & (HR_copy['satisfaction_level']
<=0.82), 'satisfaction_level_categorical'] = '65% - 82%'
HR_copy.loc[(HR_copy['satisfaction_level']>0.82), 'satisfaction_level_categorical'] =
'More Than 82%'

# Satisfaction level crosstab view
satisfaction_level_df = pd.crosstab(HR_copy.satisfaction_level_categorical, HR_copy.lef
t_job,
            normalize = 'index', rownames = ['Satisfaction Levels'],
                                 colnames = ['Left Job Status'])
satisfaction_level_df.columns = ['Still Works', 'Left Job']

# Sorting the rows
satisfaction_level_df = satisfaction_level_df.loc[['Less Than 44%', '44% - 64%', '65% -
 82%', 'More Than 82%'], :]
```

In [46]:

```
satisfaction_level_df.round(2)
```

Out[46]:

	Still Works	Left Job
Satisfaction Levels		
Less Than 44%	0.41	0.59
44% - 64%	0.92	0.08
65% - 82%	0.87	0.13
More Than 82%	0.88	0.12

Confirming what was observed at other sections, the employee with lower satisfaction level has greater probability to leave the job. However, it is interesting to observe that it doesn't need to be the highest satisfaction level to have the highest probability to stay. For example, having an evaluation between 0.44 and 0.64 gives a probability of 91.52% while an evaluation above 0.82 gives a probability of staying of 86.83%.

To go further with the investigation let's consider other factors that should influence the employee in leaving his/her job, as pointed in previous sections.

Let's consider, for example the last evaluation.

How satisfaction level and last evaluation influences leaving the job.

Last evaluation influence

In [47]:

```
HR_copy['last_evaluation'].describe()
```

Out[47]:

```
count    14999.000000
mean         0.716102
std          0.171169
min          0.360000
25%          0.560000
50%          0.720000
75%          0.870000
max          1.000000
Name: last_evaluation, dtype: float64
```

In [48]:

```
# Creating a new column for last evaluation to be able to view that in categories
HR_copy['last_evaluation_categorical'] = None
HR_copy.loc[HR_copy['last_evaluation']<=0.56, 'last_evaluation_categorical'] = 'Less Th
an 57%'
HR_copy.loc[(HR_copy['last_evaluation']>0.56) & (HR_copy['last_evaluation']<=0.72), 'la
st_evaluation_categorical'] = '57% - 72%'
HR_copy.loc[(HR_copy['last_evaluation']>0.72) & (HR_copy['last_evaluation']<=0.87), 'la
st_evaluation_categorical'] = '73% - 87%'
HR_copy.loc[(HR_copy['last_evaluation']>0.87), 'last_evaluation_categorical'] = 'More T
han 87%'

# Creating a crosstab view
last_evaluation_df = pd.crosstab(HR_copy.last_evaluation_categorical, HR_copy.left_job,
 margins=True,
            normalize = 'index', rownames = ['Last Evaluation Grades'], colnames = ['Le
ft Job Status'])
last_evaluation_df = pd.DataFrame(last_evaluation_df.loc[['Less Than 57%', '57% - 72%',
 '73% - 87%', 'More Than 87%', 'All'],:])
last_evaluation_df.columns = ['Still Works', 'Left Job']
```

In [49]:

```
last_evaluation_df.round(2)
```

Out[49]:

	Still Works	Left Job
Last Evaluation Grades		
Less Than 57%	0.63	0.37
57% - 72%	0.94	0.06
73% - 87%	0.80	0.20
More Than 87%	0.69	0.31
All	0.76	0.24

It seems that last evaluation alone doesn't play a role in someone leaving the job. We can observe for example, that someone with low evaluation (under 0.56) has almost the same probability of leaving the job as someone with a high evaluation above 0.87, respectively, 37,44% and 32.01%.

Let's see what happens when satisfaction and last evaluation are put together.

In [50]:

```
# Creating a crosstab view
satisfaction_evaluation_df = pd.crosstab([HR_copy.last_evaluation_categorical,
HR_copy.satisfaction_level_categorical], HR_copy.left_job,
            normalize = 'index', rownames = ['Last Evaluation Grades', 'Satisfaction Le
vels'],
                                    colnames = ['Left Job Status'])

# Sorting the rows
satisfaction_evaluation_df = pd.DataFrame(satisfaction_evaluation_df.loc[[
            ('Less Than 57%','Less Than 44%'),
            ('Less Than 57%','44% - 64%'),
            ('Less Than 57%','65% - 82%'),
            ('Less Than 57%','More Than 82%'),
            ('57% - 72%','Less Than 44%'),
            ('57% - 72%','44% - 64%'),
            ('57% - 72%','65% - 82%'),
            ('57% - 72%','More Than 82%'),
            ('73% - 87%','Less Than 44%'),
            ('73% - 87%','44% - 64%'),
            ('73% - 87%','65% - 82%'),
            ('73% - 87%','More Than 82%'),
            ('More Than 87%','Less Than 44%'),
            ('More Than 87%','44% - 64%'),
            ('More Than 87%','65% - 82%'),
            ('More Than 87%','More Than 82%')],:])
satisfaction_evaluation_df.columns = ['Still Works', 'Left Job']
```

In [51]:

```
satisfaction_evaluation_df.round(2)
```

Out[51]:

		Still Works	Left Job
Last Evaluation Grades	**Satisfaction Levels**		
Less Than 57%	**Less Than 44%**	0.27	0.73
	44% - 64%	0.79	0.21
	65% - 82%	0.98	0.02
	More Than 82%	1.00	0.00
57% - 72%	**Less Than 44%**	0.79	0.21
	44% - 64%	0.95	0.05
	65% - 82%	0.98	0.02
	More Than 82%	0.99	0.01
73% - 87%	**Less Than 44%**	0.42	0.58
	44% - 64%	0.97	0.03
	65% - 82%	0.86	0.14
	More Than 82%	0.89	0.11
More Than 87%	**Less Than 44%**	0.39	0.61
	44% - 64%	0.97	0.03
	65% - 82%	0.70	0.30
	More Than 82%	0.69	0.31

It is interesting to observe that when putting together satisfaction level and last evaluation, the ones that are in the range 0.44 to 0.64 of satisfaction level have greater probability to stay as high is the evaluation. Above 0.64 in the satisfaction level it seems that high the evaluation, higher the probability of leaving the job.

What is the influence of the time spent in the company?

In [52]:

```
HR_copy['time_spend_company'].describe()
```

Out[52]:

```
count    14999.000000
mean         3.498233
std          1.460136
min          2.000000
25%          3.000000
50%          3.000000
75%          4.000000
max         10.000000
Name: time_spend_company, dtype: float64
```

In [53]:

```
# Creating categorical column for time spent in the company
HR_copy['time_spend_company_categorical'] = None
HR_copy.loc[HR_copy['time_spend_company'] < 3, 'time_spend_company_categorical'] = 'Les
s Than 3'
HR_copy.loc[(HR_copy['time_spend_company'] >= 3) & (HR_copy['time_spend_company'] < 4),
 'time_spend_company_categorical'] = '3 - 4'
HR_copy.loc[(HR_copy['time_spend_company'] >= 4) & (HR_copy['time_spend_company'] < 5),
 'time_spend_company_categorical'] = '4 - 5'
HR_copy.loc[(HR_copy['time_spend_company'] >= 5) & (HR_copy['time_spend_company'] < 6),
 'time_spend_company_categorical'] = '5 - 6'
HR_copy.loc[HR_copy['time_spend_company'] >= 6, 'time_spend_company_categorical'] = 'Mo
re Than 6'

# Creating crosstab and sorting the rows
time_spend_company_df = pd.crosstab(HR_copy.time_spend_company_categorical, HR_copy.lef
t_job, margins=True,
            normalize = 'index', rownames = ['Time Spent in Company'], colnames = ['Lef
t Job Status'])
time_spend_company_df = pd.DataFrame(time_spend_company_df.loc[['Less Than 3', '3 - 4',
 '4 - 5', '5 - 6', 'More Than 6'],:])
time_spend_company_df.columns = ['Still Works', 'Left Job']
```

In [54]:

```
time_spend_company_df.round(2)
```

Out[54]:

	Still Works	Left Job
Time Spent in Company		
Less Than 3	0.98	0.02
3 - 4	0.75	0.25
4 - 5	0.65	0.35
5 - 6	0.43	0.57
More Than 6	0.84	0.16