In [6]:
```
#Cuong Phan
#This program implements Data Exploration + Analysis
#Goal: PREDICTING WHY EMPLOYEES ARE LEAVING THE COMPANY, AND LEARN TO
PREDICT WHO LEAVE THE COMPANY USING HR_comma_sep dataset

#=====================================================================
=======================================================

#Step 1: Import libraries needed to summarize characteristics of data:
pattern, trend, outliner, hypo thesis testing


#=====================================================================
=======================================================
#import libraries
import numpy as np
import pandas as pd #for dataframes
import matplotlib.pyplot as plt #for plotting graphs
import seaborn as sns #for plotting graphs
#loading dataset
data = pd.read_csv("HR_comma_sep.csv")
#get the first 5 values
data.head()
#get the last 5 values
data.tail()
#Get information about attributes names and datatypes
data.info()
#Get a list of the columns names
col_names = data.columns.tolist()
#Rename the column from "sales" to "department"
data = data.rename(columns = {'sales' : 'department'})
#Describe data attributes in English
#=====================================================================
=======================================================

''' satisfaction_level : the employee satisfaction point, ranges from
0 - 1
    last_evaluation: evaluated performance by the employer, ranges fro
m 0 -1
    number_projects: How many projects are assigned to an employee
    average_monthly_hours: How many average numbers of hours worked by
an employee in a month
    time_spent_company: employee experience. The number of years spent
by an employee in the company
    work_accident: Whether an employee has had a work accident or not
    promotion_last_5years: Whether an employee has had a promotion in
the last 5 years or not
    Departments: Employee's working department/division
```

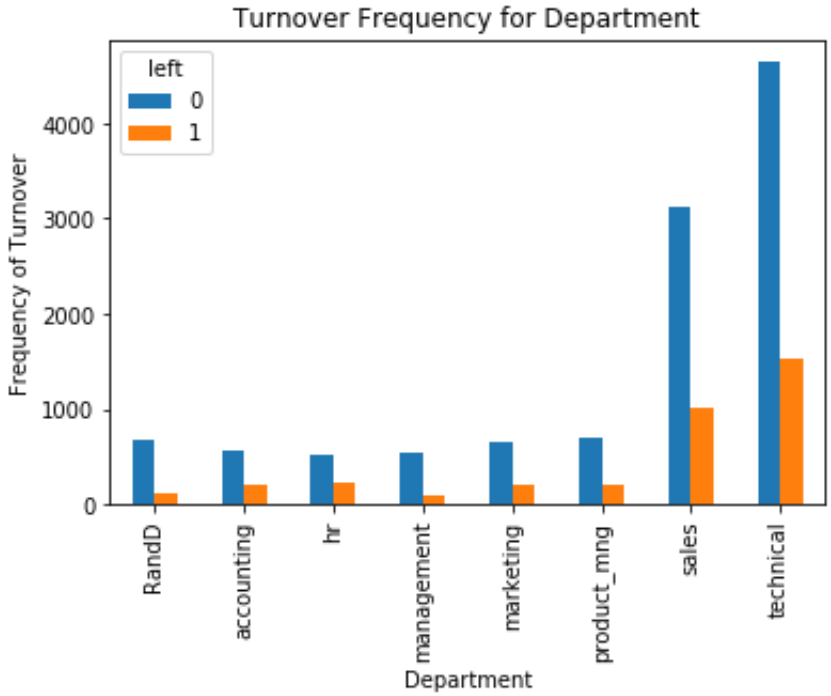
```
    Salary: Salary level of the employee such as low, medium, high
    left: whether the employee has left the company or not '''

#=================================================================
=============================================================
```

```
<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
```

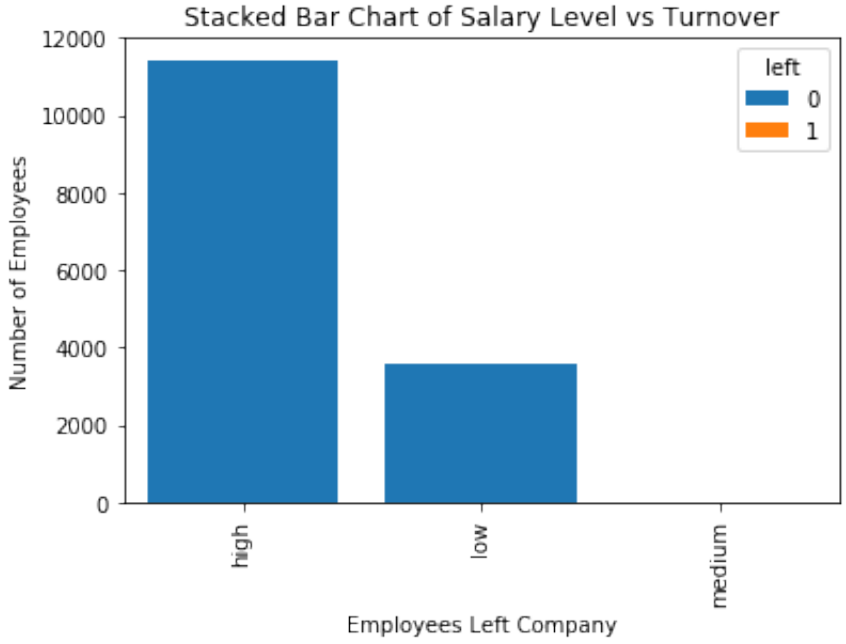
Out[6]:
```
" satisfaction_level : the employee satisfaction point, ranges from
0 - 1\n    last_evaluation: evaluated performance by the employer, r
anges from 0 -1 \n    number_projects: How many projects are assigne
d to an employee\n    average_monthly_hours: How many average number
s of hours worked by an employee in a month\n    time_spent_company:
employee experience. The number of years spent by an employee in the
company\n    work_accident: Whether an employee has had a work accid
ent or not\n    promotion_last_5years: Whether an employee has had a
promotion in the last 5 years or not\n    Departments: Employee's wo
rking department/division\n    Salary: Salary level of the employee
such as low, medium, high\n    left: whether the employee has left t
he company or not "
```

In [7]:
```
#Print the types
data.dtypes
#check if data is clean and no missing values
data.isnull().any()
#number of records and features
data.shape
data['department'].unique()

#combine "technical", "support" and "IT" these three together and call
them "technical"
data['department'] = np.where(data['department'] == 'support','technic
al', data['department'])
data['department'] = np.where(data['department'] == 'IT','technical',
data['department'])

''' Analyze data insights:  two types of employee one stayed and anoth
er left the company. So we can divide data into two groups
and compare their characteristics '''
left = data.groupby('left')
left.mean()
data['left'].value_counts()

#summary Satistics
data.describe()

#Data visualization
pd.crosstab(data.department, data.left).plot(kind='bar')
plt.title('Turnover Frequency for Department')
plt.xlabel('Department')
plt.ylabel('Frequency of Turnover')
plt.savefig('department_bart_chart')

''' This show that the frequency of employee turnover depends a great
deal on the department they work for. Thus, department
can be a good predictor of the outcome variable '''
```

Out[7]: ' This show that the frequency of employee turnover depends a great deal on the department they work for. Thus, department\ncan be a good predictor of the outcome variable '

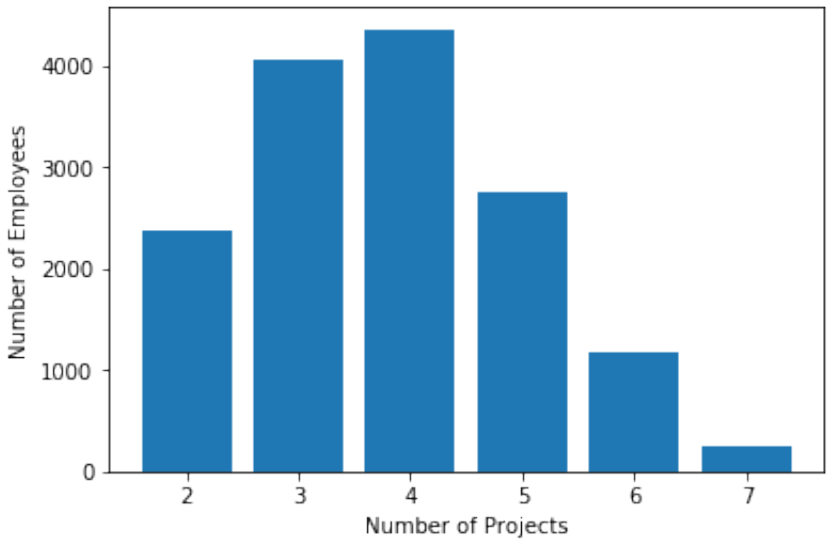


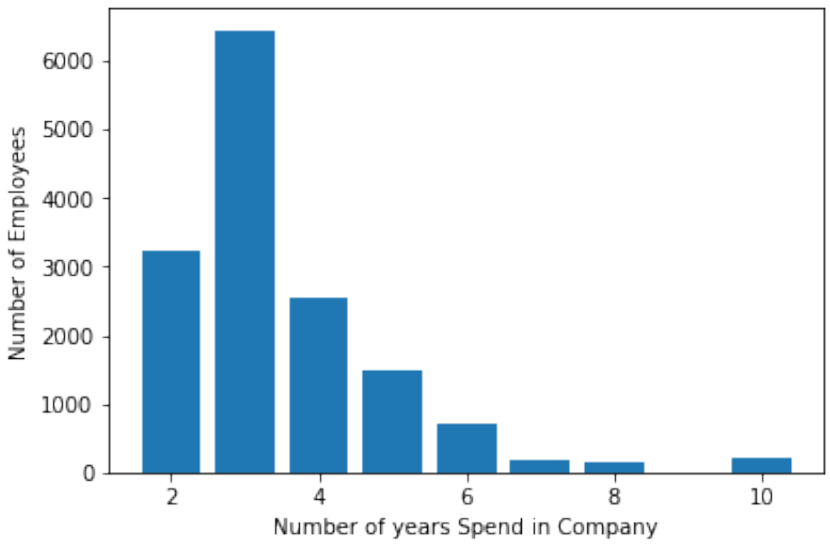
In [8]:
```
table = pd.crosstab(data.salary, data.left)
table.div(table.sum(1).astype(float), axis=0).plot(kind='bar', stacked
=True)
plt.title('Stacked Bar Chart of Salary Level vs Turnover')
plt.xlabel('Department')
plt.ylabel('Frequency of Turnover')
plt.savefig('department_bart_chart')

left_count = data.groupby('left').count()
plt.bar(left_count.index.values, left_count['satisfaction_level'])
plt.xlabel('Employees Left Company')
plt.ylabel('Number of Employees')
plt.show()
data.left.value_counts()

''' out of 15,0000 approx 3,571 were left and 11,428 stayled. The no o
f employee left is 23% of the total employement '''
```

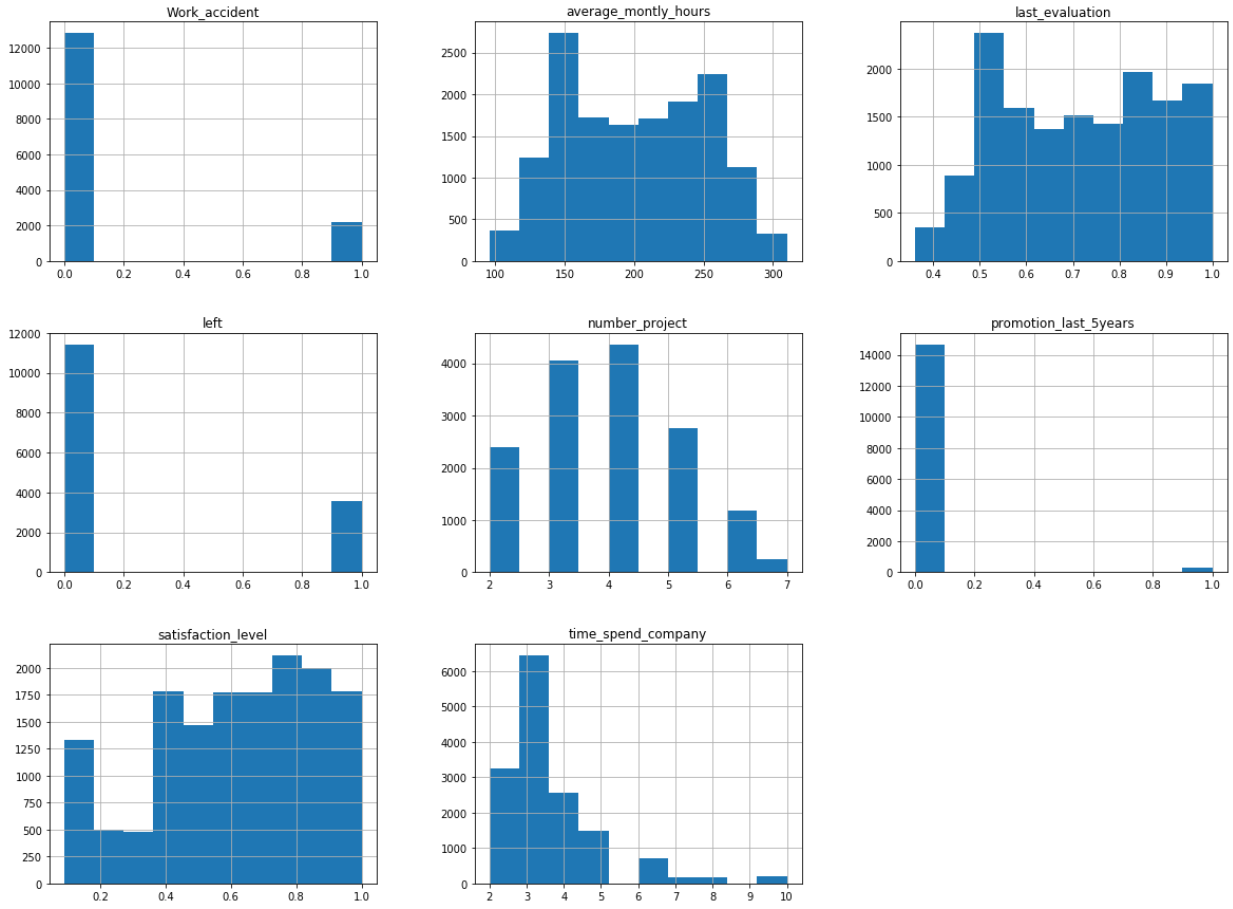


Out[8]:
```
' out of 15,0000 approx 3,571 were left and 11,428 stayled. The no o
f employee left is 23% of the total employement '
```

In [9]:
```
num_projects = data.groupby('number_project').count()
plt.bar(num_projects.index.values, num_projects['satisfaction_level'])
plt.xlabel('Number of Projects')
plt.ylabel('Number of Employees')
plt.show()

time_spent = data.groupby('time_spend_company').count()
plt.bar(time_spent.index.values, time_spent['satisfaction_level'])
plt.xlabel('Number of years Spend in Company')
plt.ylabel('Number of Employees')
plt.show()

num_bins = 10
data.hist(bins=num_bins, figsize =(20,15))
plt.savefig("hr_histogram_plots")
plt.show()
```

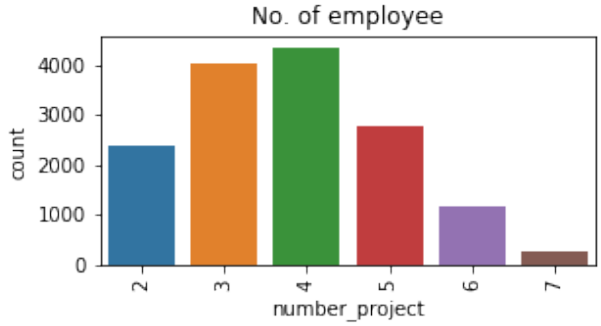


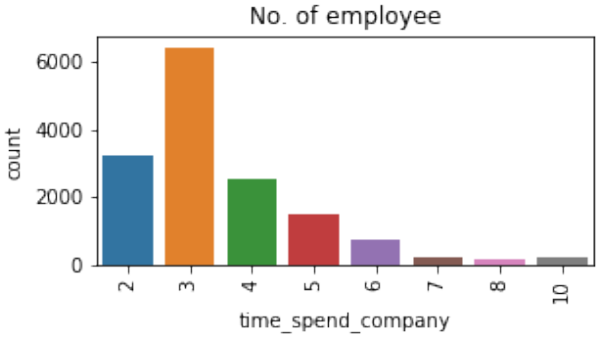




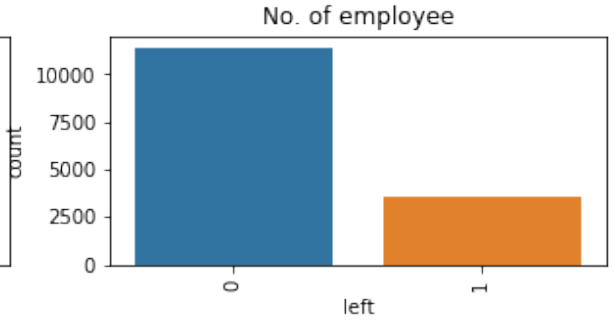
In [10]:
```
#Subplots using Seaborn:: plot all the graphs at once
features = ['number_project', 'time_spend_company', 'Work_accident', '
left', 'promotion_last_5years', 'Departments', 'salary']
fig = plt.subplots(figsize = (10,15))
for i, j in enumerate(features):
    plt.subplot(4,2,i+1)
    plt.subplots_adjust(hspace=1.0)
    sns.countplot(x=j, data = data)
    plt.xticks(rotation = 90)
    plt.title("No. of employee")

''' Most of the employee is doing the project from 3-5
    There is a huge drop between 3 years and 4 years experienced emplo
yee
    The no of employee left is 23% of the total employment
    A decidedly less number of employee get the promotion in the last
5 year
    The sales department is having maximum no.of employee followed by
technical and support
    Most of the employees are getting salary either medium or low'''
```

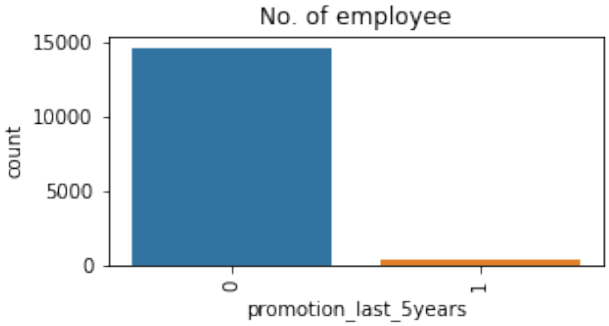
```
---------------------------------------------------------------------------
-------
ValueError                                Traceback (most recent cal
l last)
<ipython-input-10-e0e2e4c34ec1> in <module>
      5     plt.subplot(4,2,i+1)
      6     plt.subplots_adjust(hspace=1.0)
----> 7     sns.countplot(x=j, data = data)
      8     plt.xticks(rotation = 90)
      9     plt.title("No. of employee")

/anaconda3/lib/python3.7/site-packages/seaborn/categorical.py in cou
ntplot(x, y, hue, data, order, hue_order, orient, color, palette, sa
turation, dodge, ax, **kwargs)
   3551                           estimator, ci, n_boot, units,
   3552                           orient, color, palette, saturation
,
-> 3553                           errcolor, errwidth, capsize, dodge
)
   3554
   3555     plotter.value_label = "count"

/anaconda3/lib/python3.7/site-packages/seaborn/categorical.py in __i
nit__(self, x, y, hue, data, order, hue_order, estimator, ci, n_boot
, units, orient, color, palette, saturation, errcolor, errwidth, cap
size, dodge)
   1605         """Initialize the plotter."""
   1606         self.establish_variables(x, y, hue, data, orient,
-> 1607                                  order, hue_order, units)
   1608         self.establish_colors(color, palette, saturation)
   1609         self.estimate_statistic(estimator, ci, n_boot)

/anaconda3/lib/python3.7/site-packages/seaborn/categorical.py in est
ablish_variables(self, x, y, hue, data, orient, order, hue_order, un
its)
    153                 if isinstance(input, string_types):
    154                     err = "Could not interpret input '{}'".f
ormat(input)
--> 155                     raise ValueError(err)
    156
    157             # Figure out the plotting orientation

ValueError: Could not interpret input 'Departments'
```

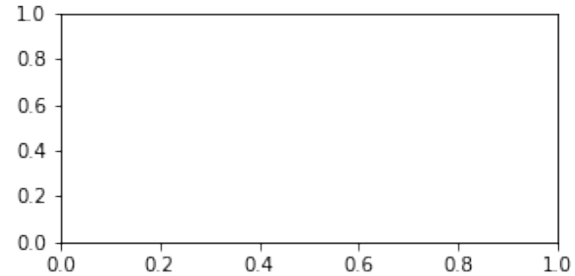

No. of employee
count
number_project
2
3
4
5
6
7


No. of employee
count
time_spend_company
2
3
4
5
6
7
8
10


No. of employee
count
Work_accident
0
1

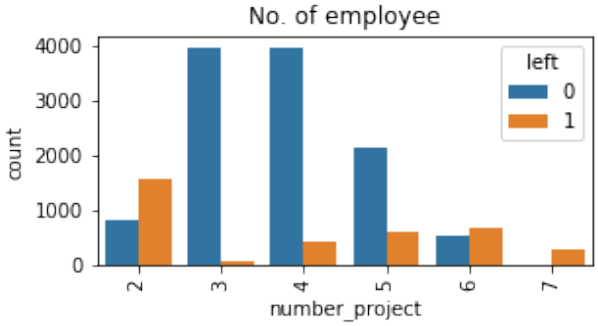

No. of employee
count
left
0
1

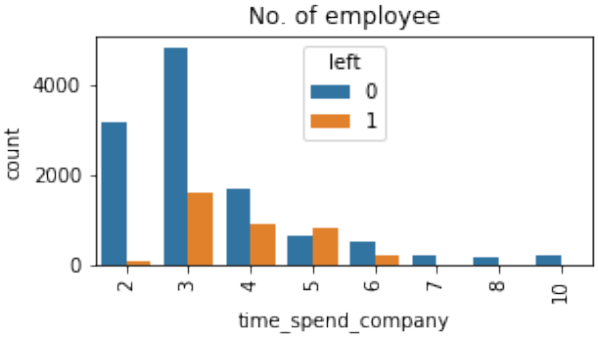

No. of employee
count
promotion_last_5years
0
1



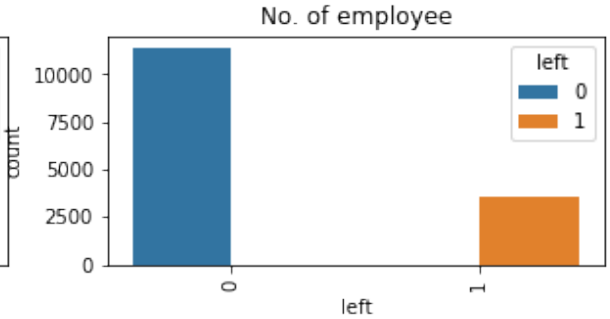
In [11]:
```
fig = plt.subplots(figsize = (10,15))
for i, j in enumerate(features):
    plt.subplot(4,2,i+1)
    plt.subplots_adjust(hspace=1.0)
    sns.countplot(x=j, data = data, hue = 'left')
    plt.xticks(rotation = 90)
    plt.title("No. of employee")
''' Those employees who have the number of projects more than 5 were l
eft the company
    The employee who had done 6 and 7 projects left the company it see
ms to like that they were loaded with work
    The employee with five-year experience is leaving more because of
no. promotions in last 5 years and more than 6 years experience
    are not leaving because of affection with the company
    Those who promotion in last 5 years they didn't leave, all those l
eft they didn't get the promotion in the previous 5 years '''
```

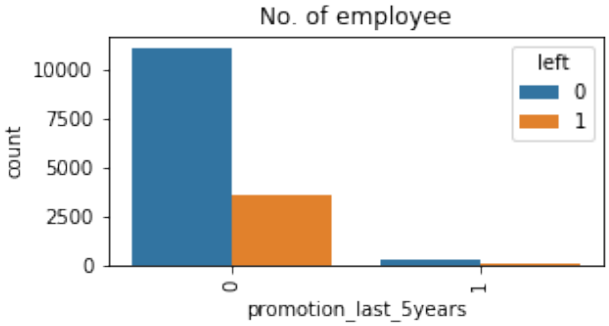
```
---------------------------------------------------------------------------
-------
ValueError                                Traceback (most recent cal
l last)
<ipython-input-11-1238aaa6e479> in <module>
      3     plt.subplot(4,2,i+1)
      4     plt.subplots_adjust(hspace=1.0)
----> 5     sns.countplot(x=j, data = data, hue = 'left')
      6     plt.xticks(rotation = 90)
      7     plt.title("No. of employee")

/anaconda3/lib/python3.7/site-packages/seaborn/categorical.py in cou
ntplot(x, y, hue, data, order, hue_order, orient, color, palette, sa
turation, dodge, ax, **kwargs)
   3551                           estimator, ci, n_boot, units,
   3552                           orient, color, palette, saturation
,
-> 3553                           errcolor, errwidth, capsize, dodge
)
   3554
   3555     plotter.value_label = "count"

/anaconda3/lib/python3.7/site-packages/seaborn/categorical.py in __i
nit__(self, x, y, hue, data, order, hue_order, estimator, ci, n_boot
, units, orient, color, palette, saturation, errcolor, errwidth, cap
size, dodge)
   1605         """Initialize the plotter."""
   1606         self.establish_variables(x, y, hue, data, orient,
-> 1607                                  order, hue_order, units)
   1608         self.establish_colors(color, palette, saturation)
   1609         self.estimate_statistic(estimator, ci, n_boot)

/anaconda3/lib/python3.7/site-packages/seaborn/categorical.py in est
ablish_variables(self, x, y, hue, data, orient, order, hue_order, un
its)
    153                 if isinstance(input, string_types):
    154                     err = "Could not interpret input '{}'".f
ormat(input)
--> 155                     raise ValueError(err)
    156
    157             # Figure out the plotting orientation

ValueError: Could not interpret input 'Departments'
```

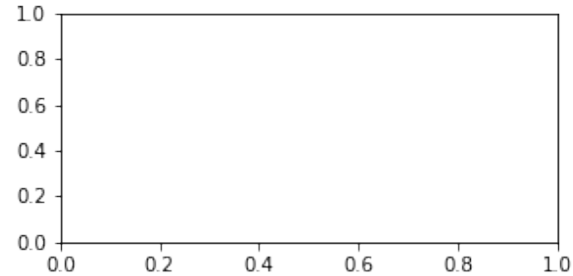

No. of employee
left
0
1
count
number_project


No. of employee
left
0
1
count
time_spend_company


No. of employee
left
0
1
count
Work_accident


No. of employee
left
0
1
count
left


No. of employee
left
0
1
count
promotion_last_5years



In [12]:
```
#Cluster Analysis based on satisfaction and performance
#import module
from sklearn.cluster import KMeans
#Filter data
left_emp = data[['satisfaction_level','last_evaluation']][data.left == 1]
#Create groups using K-means clustering.
kmeans = KMeans(n_clusters = 3, random_state = 0).fit(left_emp)
#Add new column "label" and assign cluster labels.
left_emp['label'] = kmeans.labels_
#Draw scatter plot
plt.scatter(left_emp['satisfaction_level'],left_emp['last_evaluation'], c=left_emp['label'], cmap = 'Accent')
plt.xlabel('Satisfaction Level')
plt.ylabel('Last Evaluation')
plt.title('3 Clusters of employees who left')
plt.show()

'''High Satisfaction and High Evaluation (Shared by the green color in the graph) =>Winners
    Low Satisfaction and High Evaluation (Shared by blue color) =>Frustrated
    Moderate Satisfaction and moderate Evaluation (Shared by grey color in the graph), => Bad Match '''
```

Out[12]:
```
'High Satisfaction and High Evaluation (Shared by the green color in the graph) =>Winners\n    Low Satisfaction and High Evaluation (Shared by blue color) =>Frustrated \n    Moderate Satisfaction and moderate Evaluation (Shared by grey color in the graph), => Bad Match '
```

In [ ]: