DSC530 Final Project_Michelle Rice

July 5, 2021

[2]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import thinkstats2
# if using a Jupyter notebook, includue:
%matplotlib inline
import thinkplot
```

[3]:
```
def plot_hist(var):
    hist = thinkstats2.Hist(var)
    thinkplot.Hist(hist,align='left')

    return hist
```

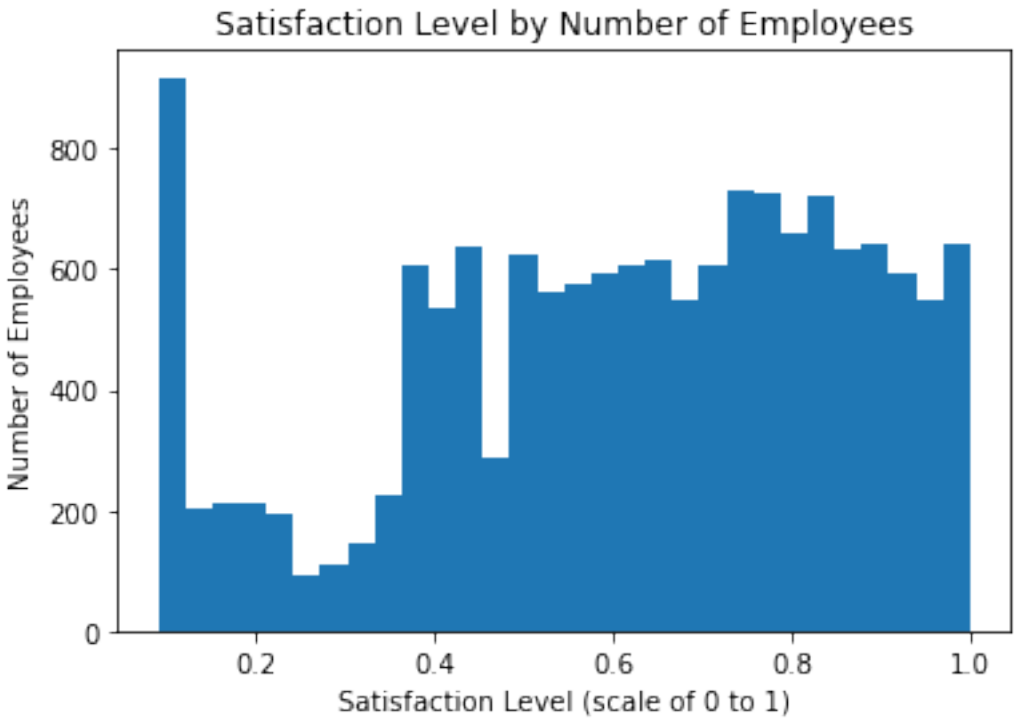
[4]:
```
employeeTurnoverData = pd.read_csv('HR_Data_Predict Employee Turnover.csv')
employeeTurnoverData.head()
```

[4]:

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company	Work_accident	left	promotion_last_5years	Department	salary
0	0.38	0.53	4	157	3	0	1	0	sales	medium
1	0.80	0.86	5	262	6	0	1	0	sales	medium
2	0.11	0.88	7	272	4	0	1	0	sales	medium
3	0.72	0.87	5	223	5	0	1	0	sales	low
4	0.37	0.52	2	159	3	0	1	0	sales	low

[4]:
```
plt.hist(employeeTurnoverData.satisfaction_level, bins = 30)
plt.title('Satisfaction Level by Number of Employees')
plt.xlabel('Satisfaction Level (scale of 0 to 1)')
plt.ylabel('Number of Employees')
plt.show
```

[4]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```

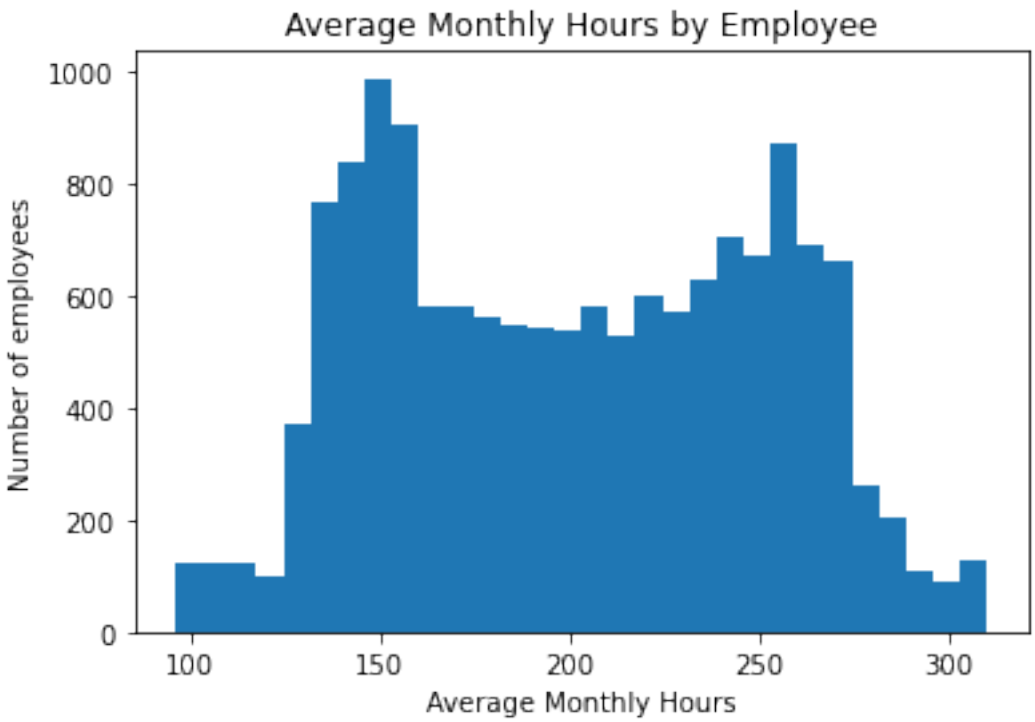


[5]:
```
employeeTurnoverData.satisfaction_level.mean(), employeeTurnoverData.
↪satisfaction_level.std()
```

[5]:
```
(0.6128335222348166, 0.2486306510611418)
```

[6]:
```
plt.hist(employeeTurnoverData.average_montly_hours, bins = 30)
plt.title ('Average Monthly Hours by Employee')
plt.xlabel ('Average Monthly Hours')
plt.ylabel('Number of employees')
plt.show
```

[6]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```

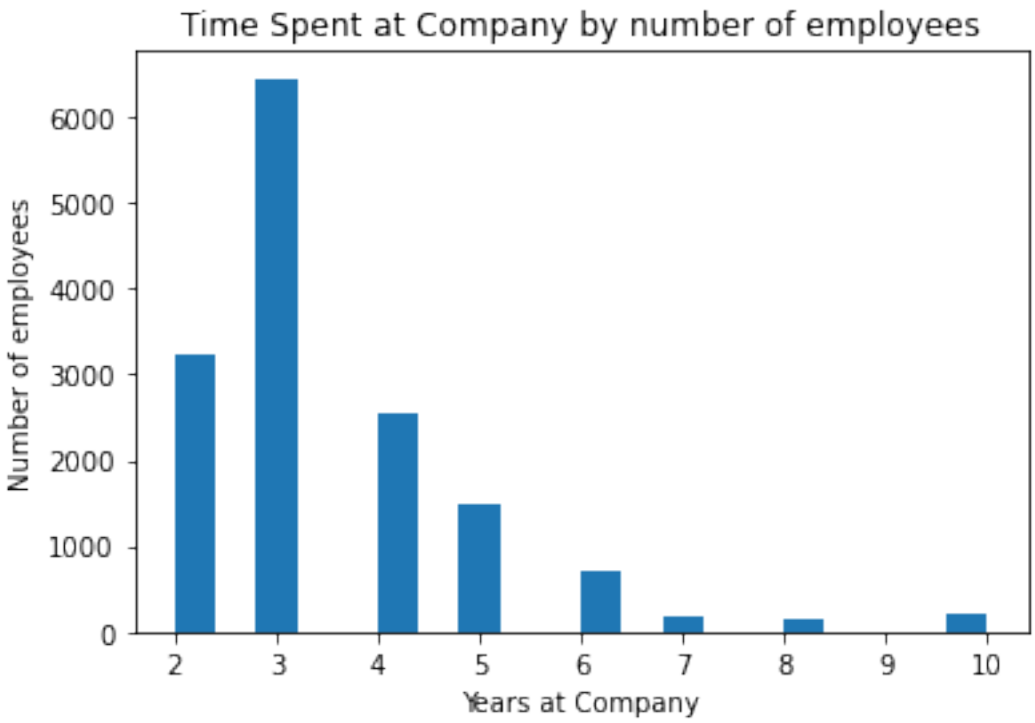


[7]:
```
employeeTurnoverData.average_montly_hours.mean(), employeeTurnoverData.
↪average_montly_hours.std()
```

[7]:
```
(201.0503366891126, 49.943099371284305)
```

[8]:
```
plt.hist(employeeTurnoverData.time_spend_company, bins = 20)
plt.title ('Time Spent at Company by number of employees')
plt.xlabel ('Years at Company')
plt.ylabel('Number of employees')
plt.show
```

[8]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```

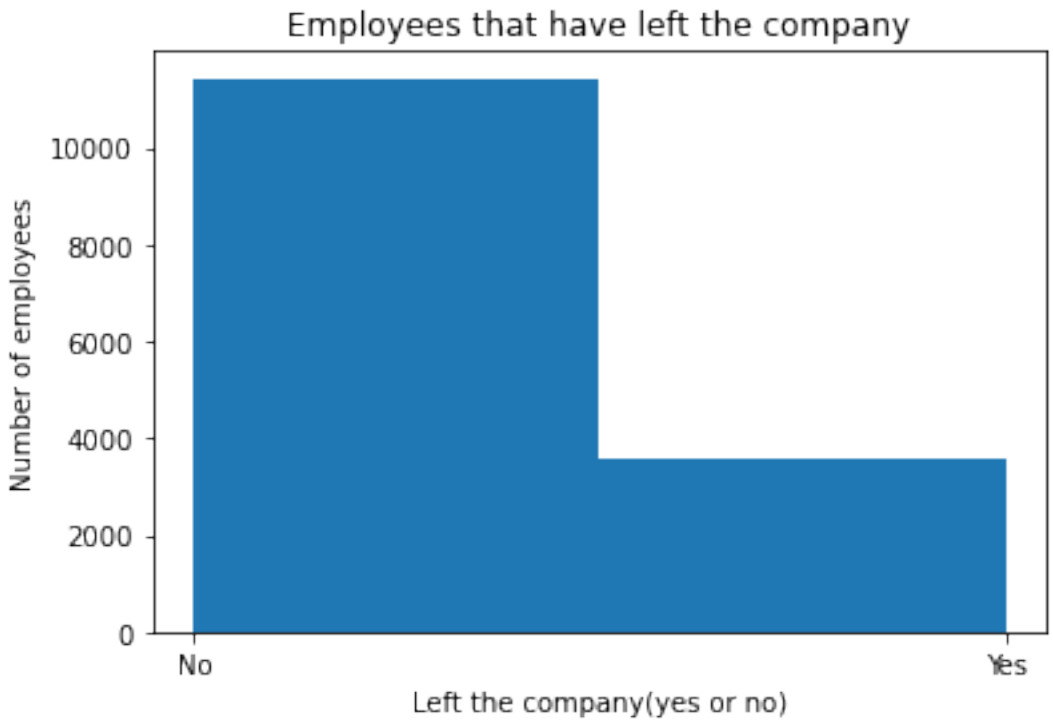


[9]:
```
employeeTurnoverData.time_spend_company.mean(), employeeTurnoverData.
↪time_spend_company.std()
```

[9]:
```
(3.498233215547703, 1.4601362305354546)
```

[10]:
```
plt.hist(employeeTurnoverData.left, bins=2)
plt.xticks([0,1],['No', 'Yes'],)
plt.title('Employees that have left the company')
plt.xlabel('Left the company(yes or no)')
plt.ylabel('Number of employees')
plt.show
```

[10]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



[11]:
```
employeeTurnoverData.left.mean(), employeeTurnoverData.left.std()
```

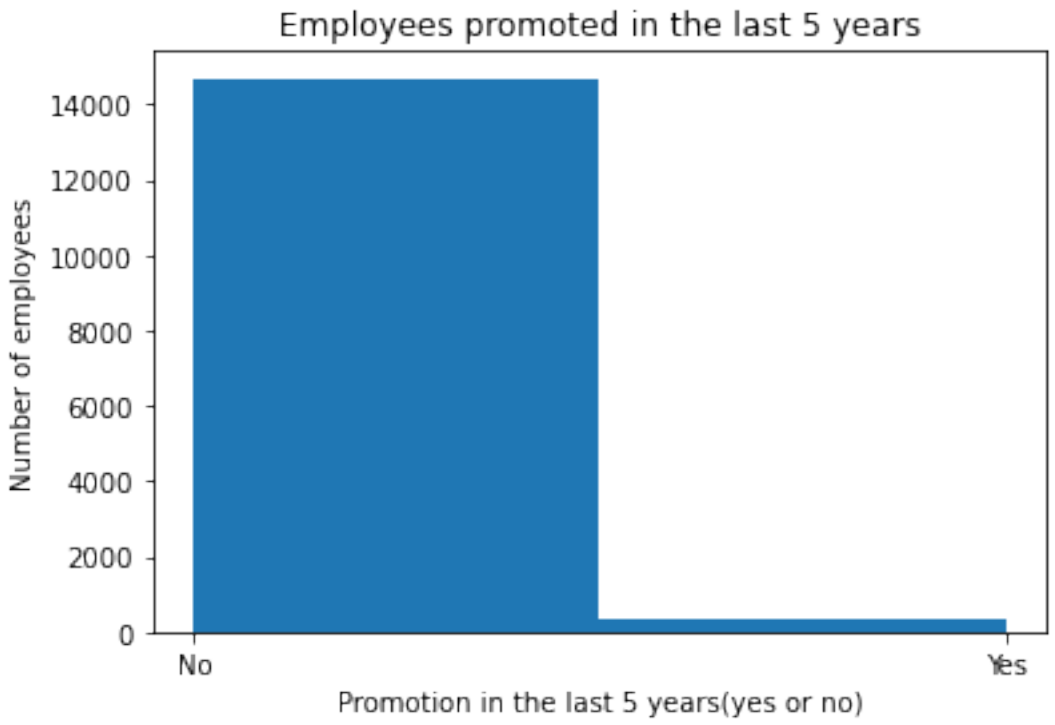
[11]:
```
(0.2380825388359224, 0.425924099380363)
```

[12]:
```
employeeTurnoverData['left'].value_counts(normalize=True)
```

[12]:
```
0    0.761917
1    0.238083
Name: left, dtype: float64
```

[13]:
```
plt.hist(employeeTurnoverData.promotion_last_5years, bins=2)
plt.xticks([0, 1], ['No', 'Yes'])
plt.title('Employees promoted in the last 5 years')
plt.xlabel('Promotion in the last 5 years(yes or no)')
plt.ylabel('Number of employees')
plt.show
```

[13]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



[14]:
```
employeeTurnoverData.promotion_last_5years.mean(), employeeTurnoverData.
↪promotion_last_5years.std()
```

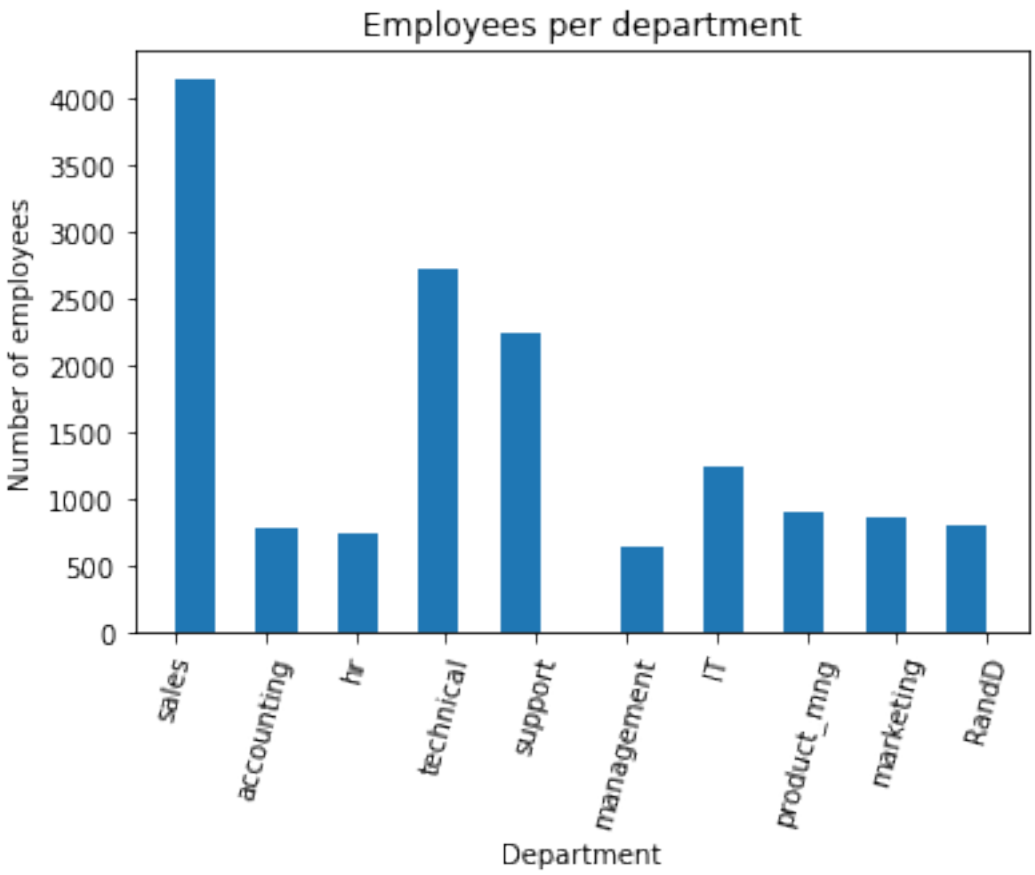
[14]:
```
(0.021268084538969265, 0.1442814645785774)
```

[15]:
```
employeeTurnoverData['promotion_last_5years'].value_counts(normalize=False)
```

[15]:
```
0    14680
1      319
Name: promotion_last_5years, dtype: int64
```

[16]:
```
plt.hist(employeeTurnoverData.Department, bins=20)
plt.xticks(['sales', 'accounting', 'hr', 'technical', 'support', ␣
↪'product_mng', 'marketing', 'IT', 'management', 'RandD'],rotation=75)
plt.title('Employees per department')
plt.xlabel('Department')
plt.ylabel('Number of employees')
plt.show
```

[16]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```

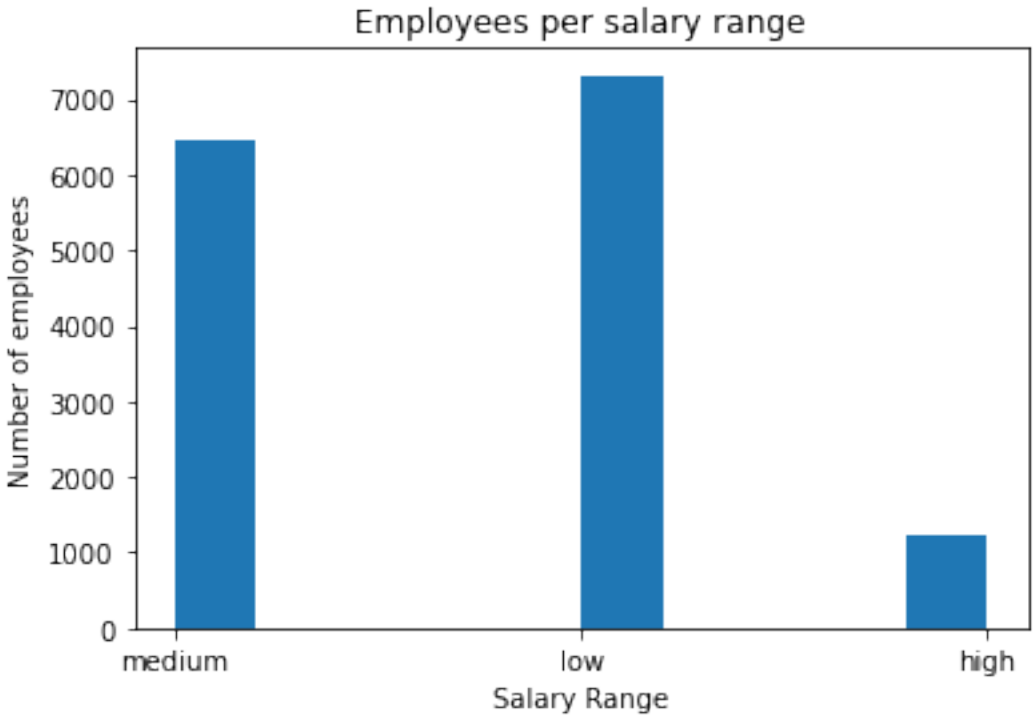


[17]:
```
employeeTurnoverData['Department'].value_counts(normalize=False)
```

[17]:
```
sales          4140
technical      2720
support        2229
IT             1227
product_mng     902
marketing       858
RandD           787
accounting      767
hr              739
management      630
Name: Department, dtype: int64
```

[18]:
```
plt.hist(employeeTurnoverData.salary)
plt.title('Employees per salary range')
plt.xlabel('Salary Range')
plt.ylabel('Number of employees')
plt.show
```

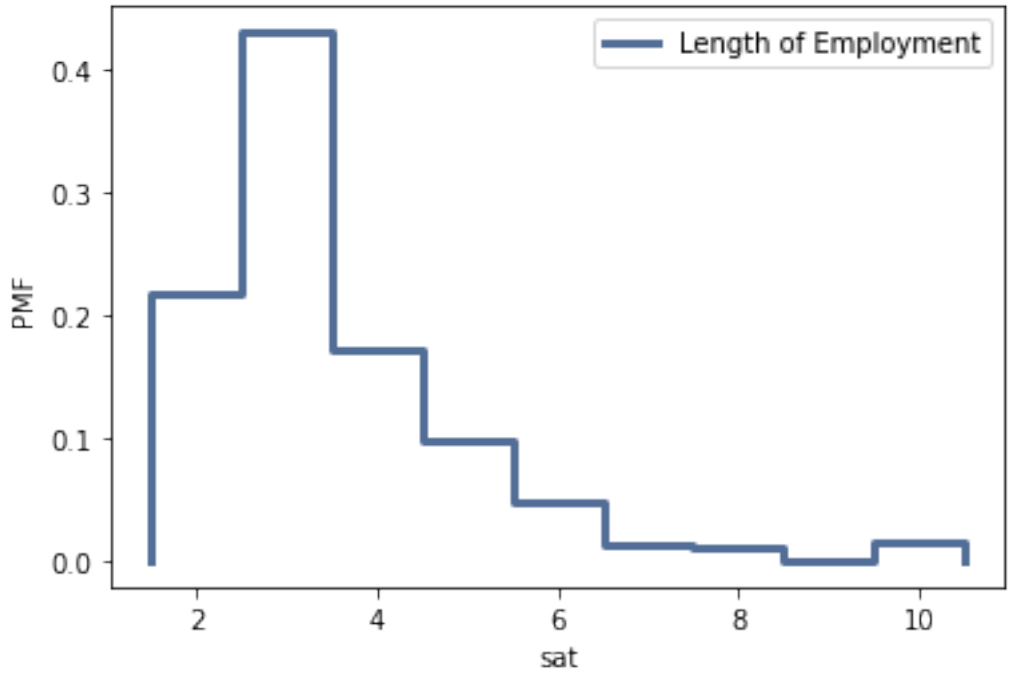
[18]: <function matplotlib.pyplot.show(close=None, block=None)>



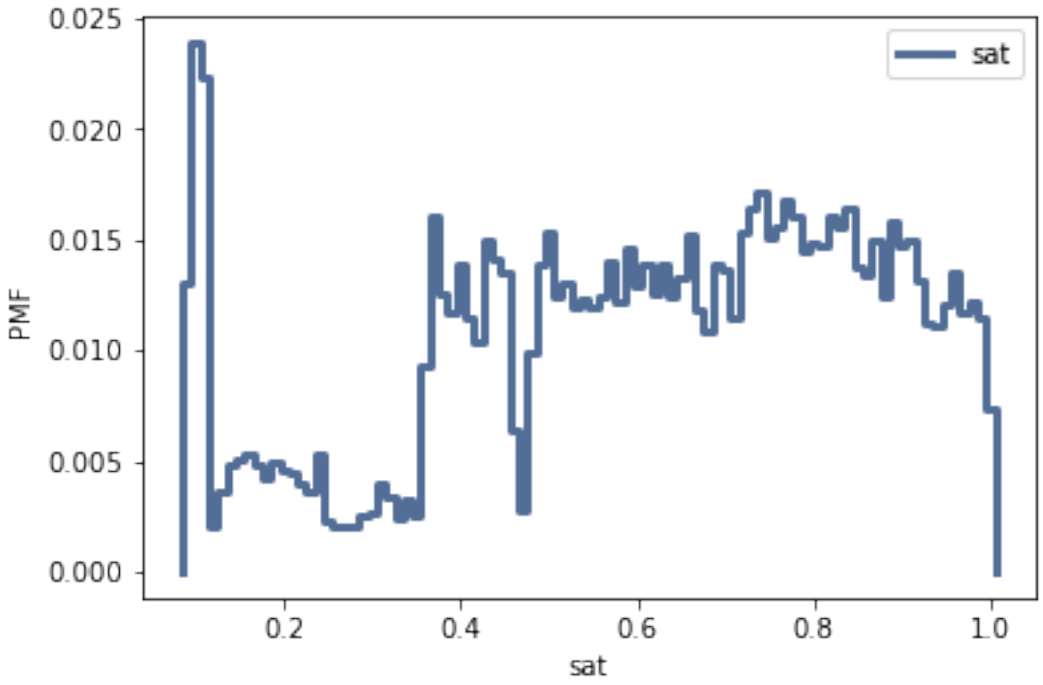
[19]:
```
employeeTurnoverData['salary'].value_counts(normalize=False)
```

[19]:
```
low       7315
medium    6447
high      1237
Name: salary, dtype: int64
```

[20]:
```
pmf = thinkstats2.Pmf(employeeTurnoverData.time_spend_company, 'Length of␣
↪Employment')
thinkplot.Pmf(pmf)
thinkplot.Config(xlabel='sat', ylabel='PMF')
```

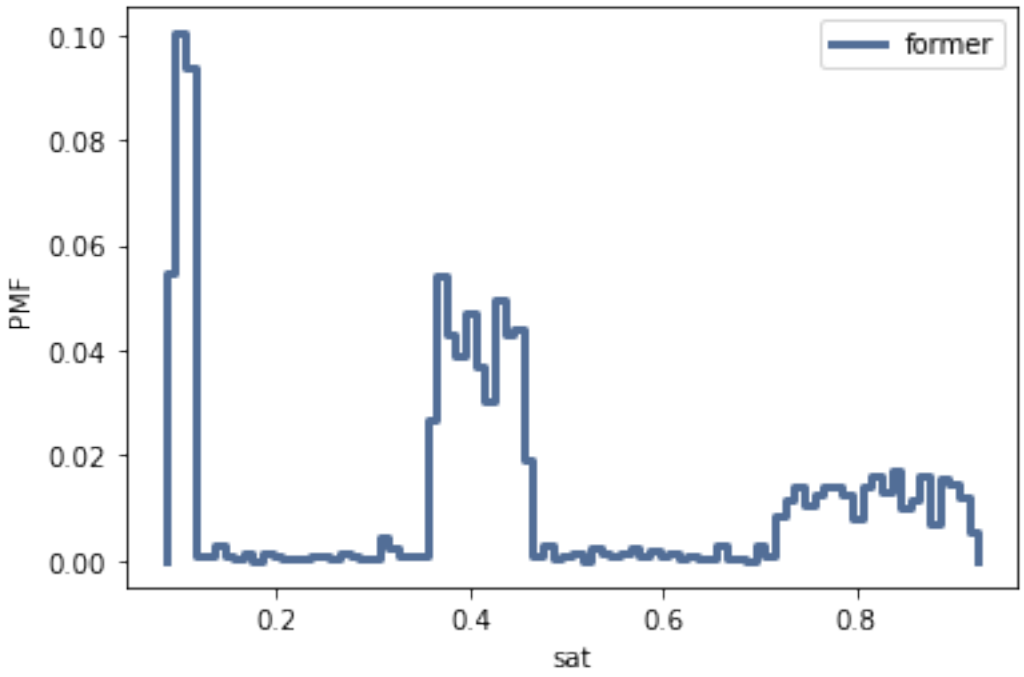


[21]:
```
pmf = thinkstats2.Pmf(employeeTurnoverData.satisfaction_level, 'sat')
thinkplot.Pmf(pmf)
thinkplot.Config(xlabel='sat', ylabel='PMF')
```

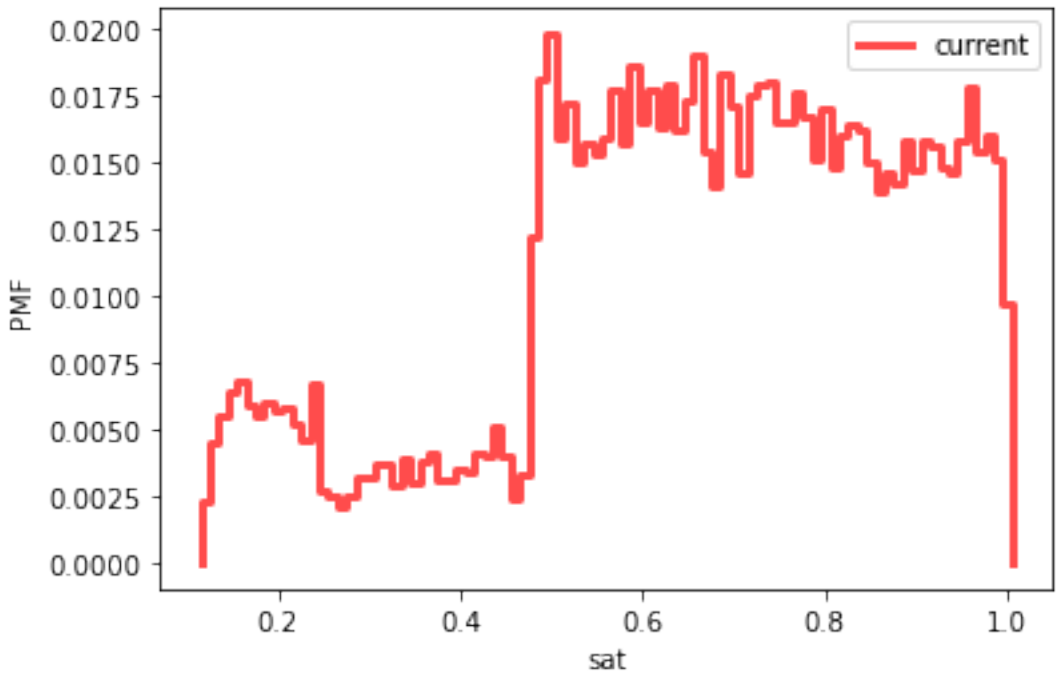


[22]:
```
former = employeeTurnoverData[employeeTurnoverData.left == 1]
current = employeeTurnoverData[employeeTurnoverData.left == 0]
```

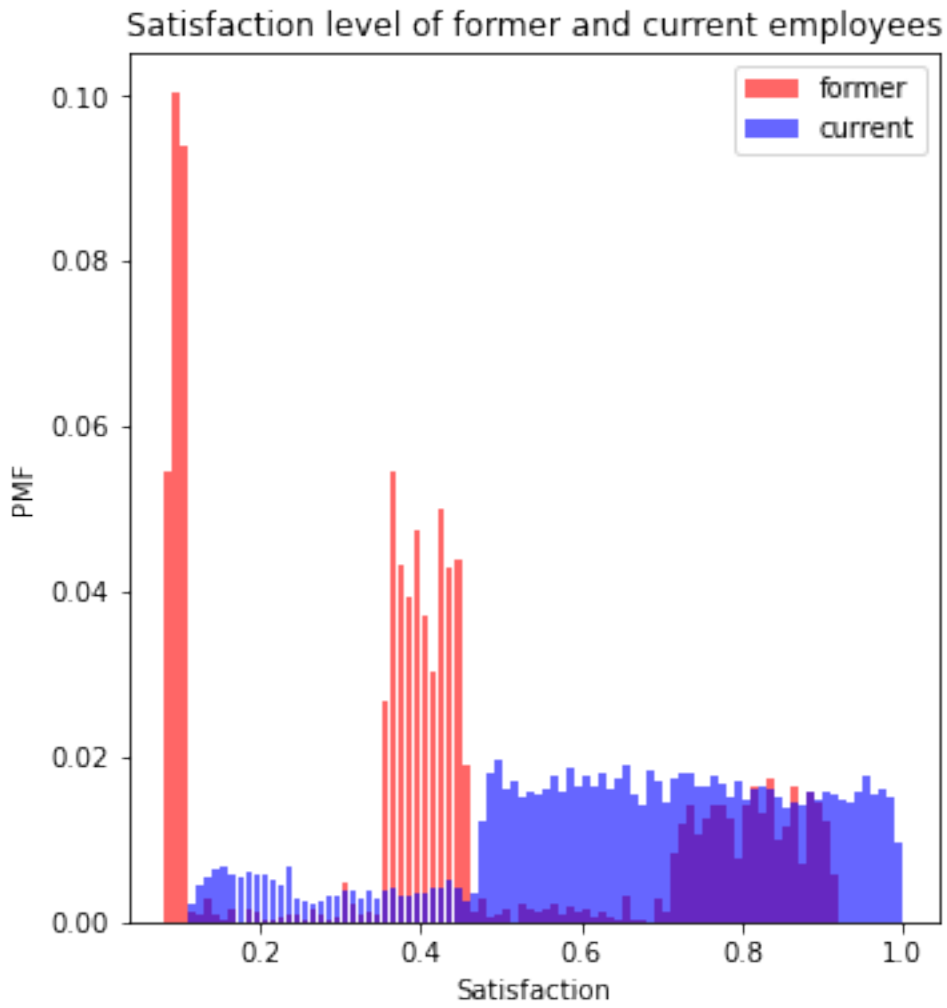
[23]:
```
pmf = thinkstats2.Pmf(former.satisfaction_level, 'former')
thinkplot.Pmf(pmf)
thinkplot.Config(xlabel='sat', ylabel='PMF')
```



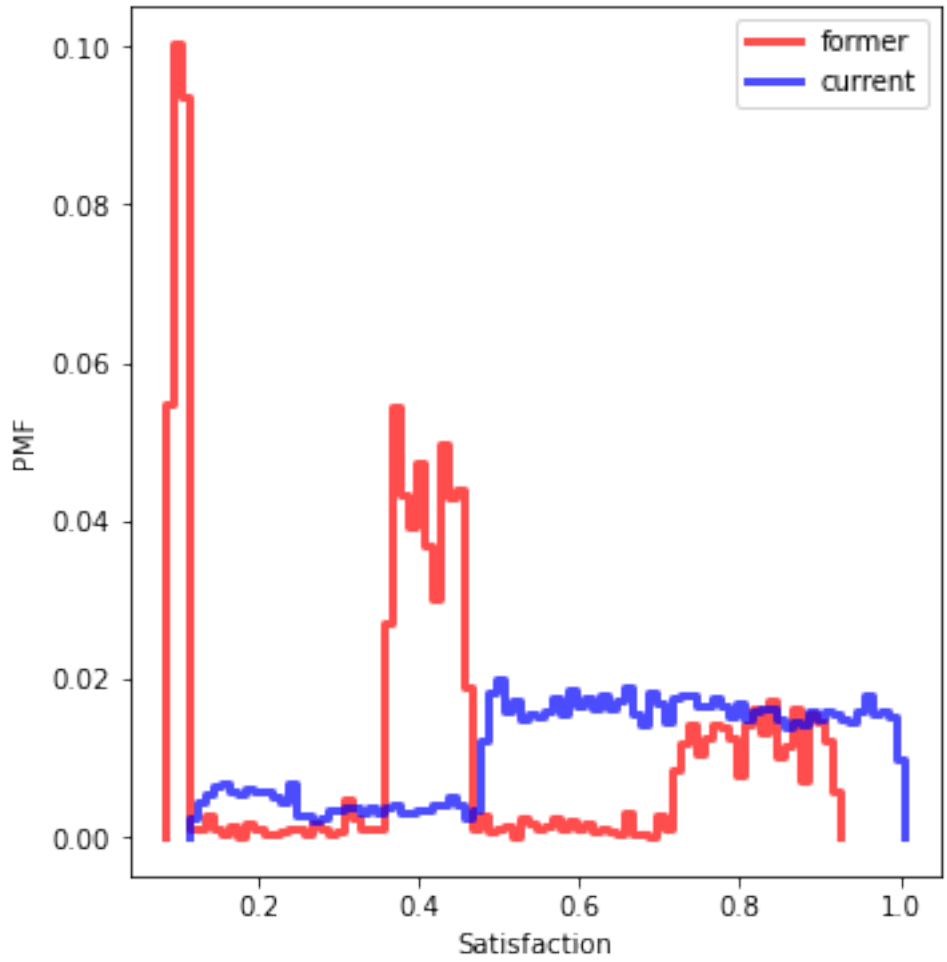
[24]:
```
pmf2 = thinkstats2.Pmf(current.satisfaction_level, 'current')
thinkplot.Pmf(pmf2, color = 'red')
thinkplot.Config(xlabel='sat', ylabel='PMF')
```



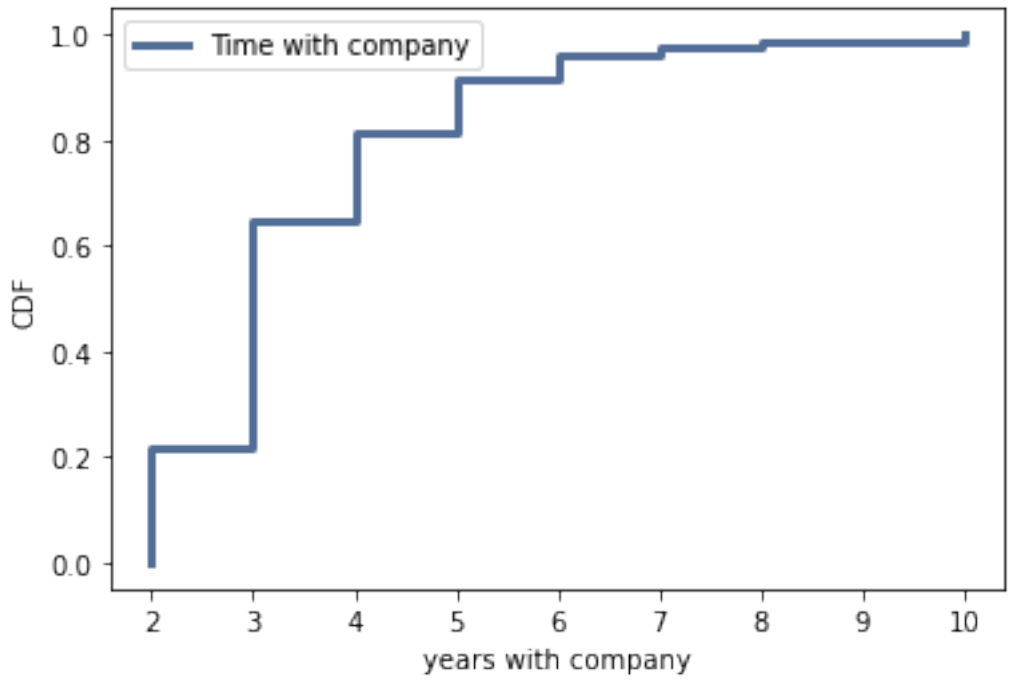
[25]:
```
thinkplot.PrePlot(2, cols=2)
thinkplot.Hist(pmf, align = 'right', color = 'red')
thinkplot.Hist(pmf2, align = 'right', color = 'blue')
thinkplot.Config(xlabel='Satisfaction', ylabel='PMF', title = 'Satisfaction␣
↪level of former and current employees')
```



[26]:
```
thinkplot.PrePlot(2, cols=2)
thinkplot.Pmf(pmf, color = 'red')
thinkplot.Pmf(pmf2, color = 'blue')
thinkplot.Config(xlabel='Satisfaction', ylabel='PMF')
```



[27]:
```
cdf = thinkstats2.Cdf(employeeTurnoverData.time_spend_company, label = 'Time␣
↪with company')
thinkplot.Cdf(cdf)
thinkplot.Show(xlabel = 'years with company', ylabel = 'CDF')
```




```
<Figure size 576x432 with 0 Axes>
```

[28]:
```
import scipy.stats
```

[29]:
```
scipy.stats.norm.cdf(.613)
```

[29]:
```
0.7300618301391996
```

[51]:
```
low = dist.cdf(.3)
high = dist.cdf(.5)
low, high, high-low
```

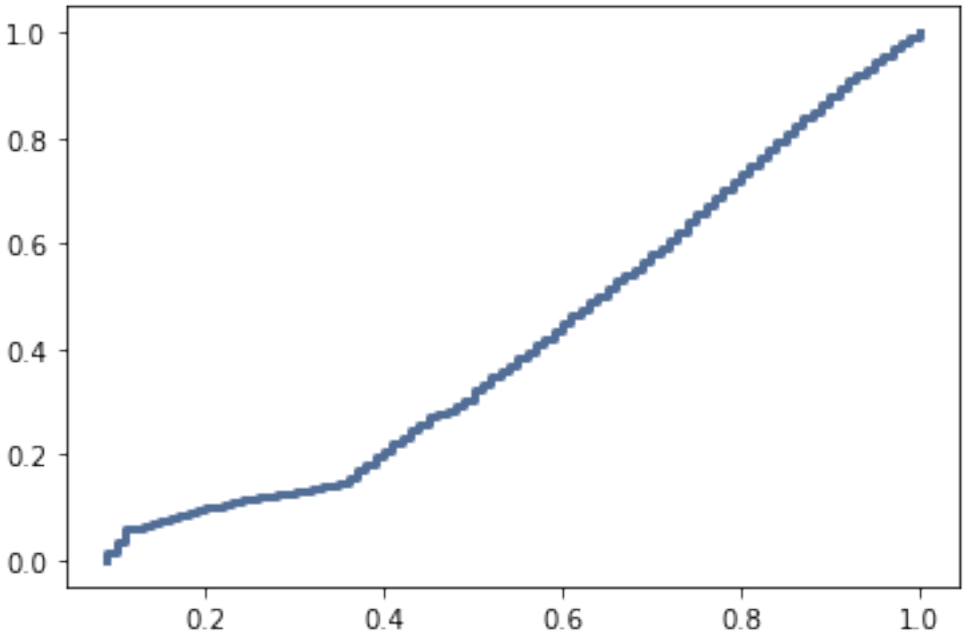
```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-51-e71852e68a74> in <module>
----> 1 low = dist.cdf(.3)
      2 high = dist.cdf(.5)
      3 low, high, high-low

NameError: name 'dist' is not defined
```

[31]:
```
cdf=thinkstats2.Cdf(employeeTurnoverData.satisfaction_level, label =␣
↪'Satisfaction')
thinkplot.Cdf(cdf)
```

```
thinkplot.Show
```

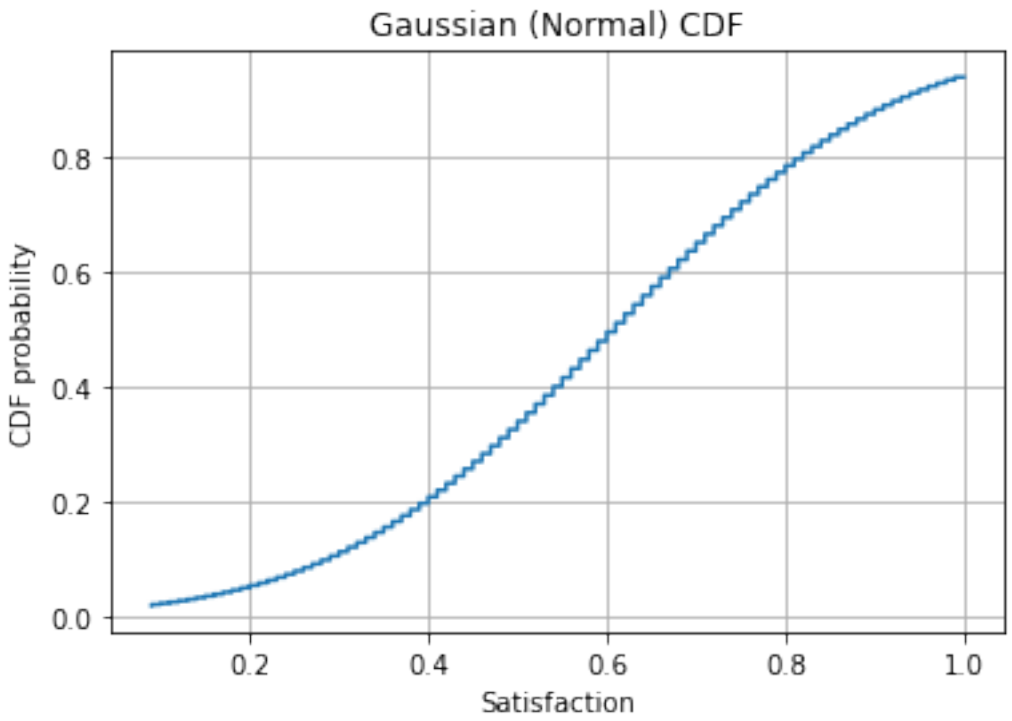
[31]: <function thinkplot.Show(**options)>



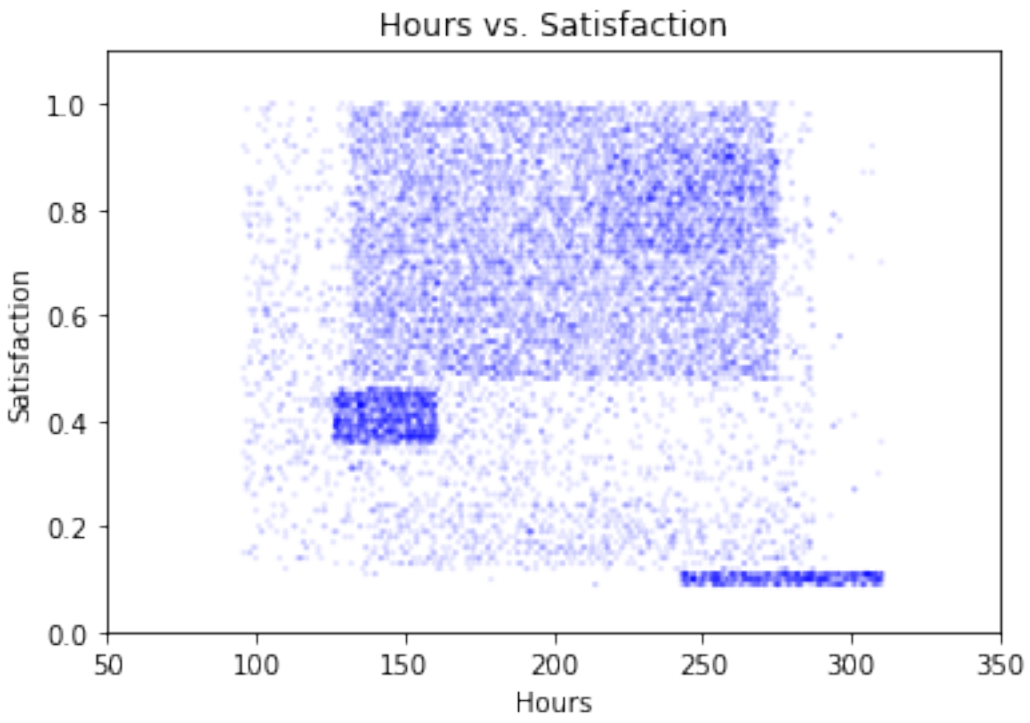
[33]:
```
def EvalNormalCdf (x, mu=.613, sigma =.249):
    return scipy.stats.norm.cdf(x, loc=mu, scale=sigma)
```

[34]:
```
def step_plot(values, probabilities, title, xlabel, ylabel = "CDF probability"):
    plt.step(values, probabilities)
    plt.grid()
    plt.title(title)
    plt.xlabel(xlabel)
    plt.ylabel(ylabel)
```

[35]:
```
step_plot(sorted(employeeTurnoverData["satisfaction_level"]),EvalNormalCdf(sorted(employeeTurn
 ↪mu=.613, sigma=.249), "Gaussian (Normal) CDF", "Satisfaction")
```



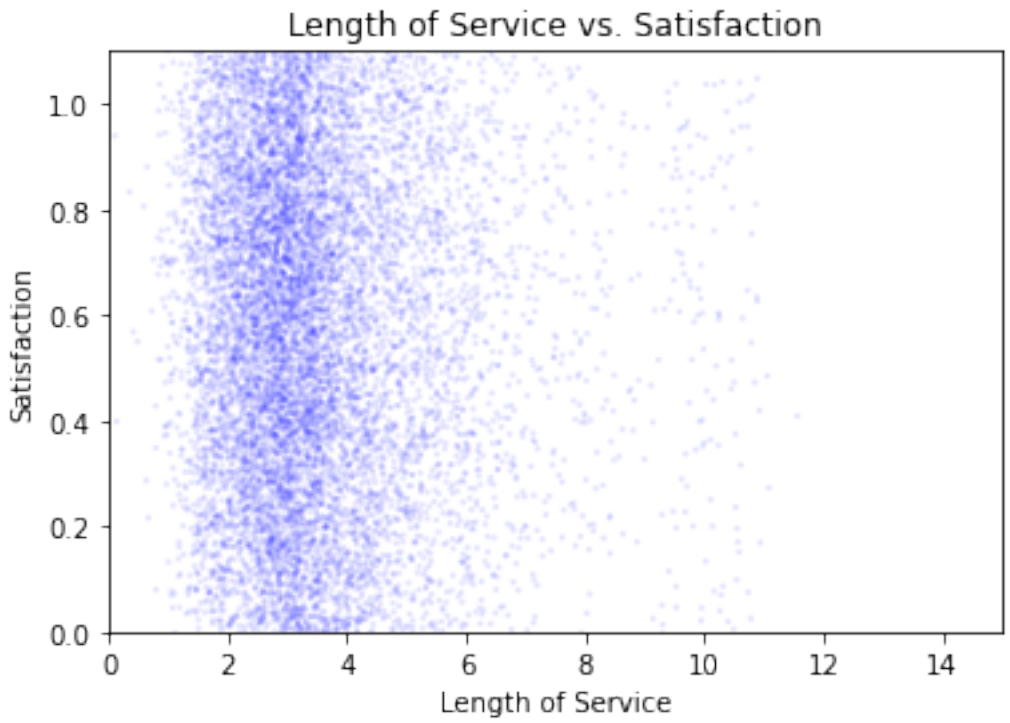
[36]:
```
thinkplot.Scatter(employeeTurnoverData.average_montly_hours,␣
↪employeeTurnoverData.satisfaction_level, alpha = .1, s=5)
thinkplot.Config(xlabel='Hours',
                 ylabel='Satisfaction',
                 axis=[50,350,0,1.1],
                 title = "Hours vs. Satisfaction",
                 legend=False)
```



[37]:
```
def Jitter(values, jitter=0.5):
    n = len(values)
    return np.random.normal(0, jitter, n) + values
```

[38]:
```
empDuration = Jitter(employeeTurnoverData.time_spend_company)
empSatisfaction = Jitter(employeeTurnoverData.satisfaction_level)
```

[39]:
```
thinkplot.Scatter(empDuration, empSatisfaction, alpha = .1, s=5)
thinkplot.Config(xlabel='Length of Service',
                 ylabel='Satisfaction',
                 axis=[0,15,0,1.1],
                 title = "Length of Service vs. Satisfaction",
                 legend=False)
```

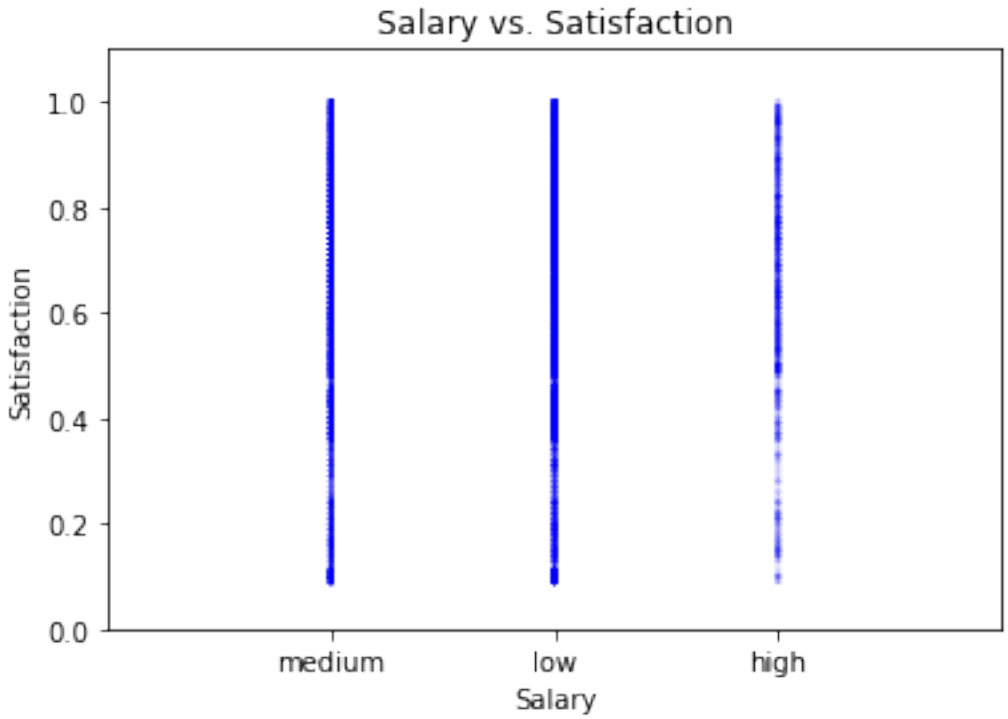


[40]:
```
employeeTurnoverData.salary[employeeTurnoverData.salary == 'low'] == '1'
employeeTurnoverData.salary[employeeTurnoverData.salary == 'medium'] == '2'
employeeTurnoverData.salary[employeeTurnoverData.salary == 'high'] == '3'
```

[40]:
```
72       False
111      False
189      False
267      False
306      False
         …
14829    False
14868    False
14902    False
14941    False
14980    False
Name: salary, Length: 1237, dtype: bool
```

[41]:
```
thinkplot.Scatter(employeeTurnoverData.salary, employeeTurnoverData.
 ↪satisfaction_level, alpha = .1, s=5)
thinkplot.Config(xlabel='Salary',
                 ylabel='Satisfaction',
                 axis=[-1,3,0,1.1],
                 title = "Salary vs. Satisfaction",
```

```
                    legend=False)
```



[42]:
```
class CorrelationPermute(thinkstats2.HypothesisTest):

    def TestStatistic(self, data):
        xs, ys = data
        test_stat = abs(thinkstats2.Corr(xs, ys))
        return test_stat

    def RunModel(self):
        xs, ys = self.data
        xs = np.random.permutation(xs)
        return xs, ys
```

[43]:
```
data = employeeTurnoverData.average_montly_hours, employeeTurnoverData.
 ↪satisfaction_level
ht = CorrelationPermute(data)
pvalue = ht.PValue()
pvalue
```

[43]: 0.009

[44]:
```
ht.actual, ht.MaxTestStat()
```

[44]:
```
(0.020048113219472988, 0.02454129192235911)
```

[45]:
```
import statsmodels.formula.api as smf
```

[46]:
```
formula = 'left ~ satisfaction_level + salary + time_spend_company'
model = smf.logit(formula, data=employeeTurnoverData)
results = model.fit()
results.summary()
```

```
Optimization terminated successfully.
         Current function value: 0.453298
         Iterations 7
```

[46]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                           Logit Regression Results
==============================================================================
Dep. Variable:                   left   No. Observations:                14999
Model:                          Logit   Df Residuals:                    14994
Method:                           MLE   Df Model:                            4
Date:                Sat, 06 Mar 2021   Pseudo R-squ.:                  0.1741
Time:                        23:34:03   Log-Likelihood:                -6799.0
converged:                       True   LL-Null:                       -8232.3
Covariance Type:            nonrobust   LLR p-value:                     0.000
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -1.5050      0.144    -10.470      0.000      -1.787      -1.223
salary[T.low]  1.9854      0.125     15.933      0.000       1.741       2.230
salary[T.medium]  1.4524      0.125     11.575      0.000       1.206       1.698
satisfaction_level    -3.7239      0.089    -42.070      0.000      -3.897      -3.550
time_spend_company     0.2115      0.014     14.919      0.000       0.184       0.239
==============================================================================
"""
```

[47]:
```
formula = 'satisfaction_level ~ salary + time_spend_company +␣
 ↪promotion_last_5years + average_montly_hours'
model = smf.logit(formula, data=employeeTurnoverData)
```

```
results = model.fit()
results.summary()
```

```
Optimization terminated successfully.
         Current function value: 0.645558
         Iterations 4
```

[47]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                           Logit Regression Results
==============================================================================
Dep. Variable:     satisfaction_level   No. Observations:                14999
Model:                          Logit   Df Residuals:                    14993
Method:                           MLE   Df Model:                            5
Date:                Sat, 06 Mar 2021   Pseudo R-squ.:                -0.07135
Time:                        23:34:03   Log-Likelihood:                -9682.7
converged:                       True   LL-Null:                       -9037.9
Covariance Type:            nonrobust   LLR p-value:                     1.000
==============================================================================
=========
                             coef    std err          z      P>|z|      [0.025
0.975]
------------------------------------------------------------------------------
---------
Intercept                  0.8560      0.096      8.910      0.000       0.668
1.044
salary[T.low]             -0.1659      0.064     -2.581      0.010      -0.292
-0.040
salary[T.medium]          -0.0740      0.065     -1.143      0.253      -0.201
0.053
time_spend_company        -0.0741      0.011     -6.441      0.000      -0.097
-0.052
promotion_last_5years      0.2057      0.120      1.710      0.087      -0.030
0.441
average_montly_hours      -0.0001      0.000     -0.408      0.683      -0.001
0.001
==============================================================================
=========
"""
```

[48]:
```
employeeTurnoverData.groupby(employeeTurnoverData.Department).mean()
```

[48]:

Department	satisfaction_level	last_evaluation	number_project	\
IT	0.618142	0.716830	3.816626	
RandD	0.619822	0.712122	3.853875	
accounting	0.582151	0.717718	3.825293	

```
hr                    0.598809         0.708850        3.654939
management            0.621349         0.724000        3.860317
marketing             0.618601         0.715886        3.687646
product_mng           0.619634         0.714756        3.807095
sales                 0.614447         0.709717        3.776812
support               0.618300         0.723109        3.803948
technical             0.607897         0.721099        3.877941

             average_montly_hours  time_spend_company  Work_accident  \
Department
IT                     202.215974            3.468623       0.133659
RandD                  200.800508            3.367217       0.170267
accounting             201.162973            3.522816       0.125163
hr                     198.684709            3.355886       0.120433
management             201.249206            4.303175       0.163492
marketing              199.385781            3.569930       0.160839
product_mng            199.965632            3.475610       0.146341
sales                  200.911353            3.534058       0.141787
support                200.758188            3.393001       0.154778
technical              202.497426            3.411397       0.140074

                left  promotion_last_5years
Department
IT          0.222494               0.002445
RandD       0.153748               0.034307
accounting  0.265971               0.018253
hr          0.290934               0.020298
management  0.144444               0.109524
marketing   0.236597               0.050117
product_mng 0.219512               0.000000
sales       0.244928               0.024155
support     0.248991               0.008973
technical   0.256250               0.010294
```

[49]:
```
former.describe()
```

[49]:
```
       satisfaction_level  last_evaluation  number_project  \
count         3571.000000      3571.000000     3571.000000
mean             0.440098         0.718113        3.856063
std              0.263933         0.197673        1.817902
min              0.090000         0.450000        2.000000
25%              0.130000         0.520000        2.000000
50%              0.410000         0.790000        4.000000
75%              0.730000         0.900000        6.000000
max              0.920000         1.000000        7.000000

       average_montly_hours  time_spend_company  Work_accident    left  \
```

```
count           3571.000000         3571.000000    3571.000000  3571.0
mean             207.419210            3.876505       0.047326     1.0
std               61.202825            0.977698       0.212364     0.0
min              126.000000            2.000000       0.000000     1.0
25%              146.000000            3.000000       0.000000     1.0
50%              224.000000            4.000000       0.000000     1.0
75%              262.000000            5.000000       0.000000     1.0
max              310.000000            6.000000       1.000000     1.0

       promotion_last_5years
count            3571.000000
mean                0.005321
std                 0.072759
min                 0.000000
25%                 0.000000
50%                 0.000000
75%                 0.000000
max                 1.000000
```

[50]:
```
current.describe()
```

[50]:
```
       satisfaction_level  last_evaluation  number_project  \
count        11428.000000     11428.000000    11428.000000
mean             0.666810         0.715473        3.786664
std              0.217104         0.162005        0.979884
min              0.120000         0.360000        2.000000
25%              0.540000         0.580000        3.000000
50%              0.690000         0.710000        4.000000
75%              0.840000         0.850000        4.000000
max              1.000000         1.000000        6.000000

       average_montly_hours  time_spend_company  Work_accident     left  \
count          11428.000000        11428.000000   11428.000000  11428.0
mean             199.060203            3.380032       0.175009      0.0
std               45.682731            1.562348       0.379991      0.0
min               96.000000            2.000000       0.000000      0.0
25%              162.000000            2.000000       0.000000      0.0
50%              198.000000            3.000000       0.000000      0.0
75%              238.000000            4.000000       0.000000      0.0
max              287.000000           10.000000       1.000000      0.0

       promotion_last_5years
count           11428.000000
mean                0.026251
std                 0.159889
min                 0.000000
25%                 0.000000
```

```
50%                 0.000000
75%                 0.000000
max                 1.000000
```

[ ]:

[ ]: