In [167]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [168]:
```
import os
os.chdir("C:\\DataSet Company\\absenteeism")
print(os.getcwd())
```

```
C:\DataSet Company\absenteeism
```

In [169]:
```
dataset = pd.read_csv('MFGEmployees4.csv')
dataset
```

Out[169]:

	EmployeeNumber	Surname	GivenName	Gender	City	JobTitle	DepartmentName	StoreLocation	Division
0	1	Gutierrez	Molly	F	Burnaby	Baker	Bakery	Burnaby	Stores
1	2	Hardwick	Stephen	M	Courtenay	Baker	Bakery	Nanaimo	Stores
2	3	Delgado	Chester	M	Richmond	Baker	Bakery	Richmond	Stores
3	4	Simon	Irene	F	Victoria	Baker	Bakery	Victoria	Stores
4	5	Delvalle	Edward	M	New Westminster	Baker	Bakery	New Westminster	Stores
5	6	Jones	Ernie	M	Richmond	Baker	Bakery	Richmond	Stores
6	7	Buford	Ralph	M	Vancouver	Accounting Clerk	Accounting	Vancouver	FinanceAndAccounting
7	8	Lee	Gregory	M	Sechelt	Baker	Bakery	West Vancouver	Stores
8	9	Smith	Jerry	M	New	Baker	Bakery	New	Stores

In [170]:
```
dataset.head()
```

Out[170]:

	EmployeeNumber	Surname	GivenName	Gender	City	JobTitle	DepartmentName	StoreLocation	Division	Age	LengthSe
0	1	Gutierrez	Molly	F	Burnaby	Baker	Bakery	Burnaby	Stores	32.028816	6.0
1	2	Hardwick	Stephen	M	Courtenay	Baker	Bakery	Nanaimo	Stores	40.320902	5.5
2	3	Delgado	Chester	M	Richmond	Baker	Bakery	Richmond	Stores	48.822047	4.3
3	4	Simon	Irene	F	Victoria	Baker	Bakery	Victoria	Stores	44.599357	3.0
4	5	Delvalle	Edward	M	New Westminster	Baker	Bakery	New Westminster	Stores	35.697876	3.6

In [171]:
```
print('Shape of dataset is:{}'.format(dataset.shape))
```

```
Shape of dataset is:(8336, 13)
```

In [172]:
```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8336 entries, 0 to 8335
Data columns (total 13 columns):
EmployeeNumber    8336 non-null int64
Surname           8336 non-null object
GivenName         8336 non-null object
Gender            8336 non-null object
City              8336 non-null object
JobTitle          8336 non-null object
DepartmentName    8336 non-null object
StoreLocation     8336 non-null object
Division          8336 non-null object
Age               8336 non-null float64
LengthService     8336 non-null float64
AbsentHours       8336 non-null float64
BusinessUnit      8336 non-null object
dtypes: float64(3), int64(1), object(9)
memory usage: 846.7+ KB
```

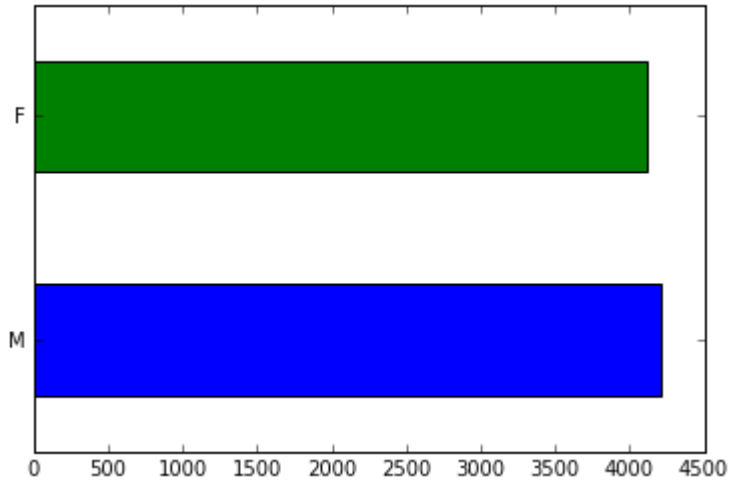
In [173]:
```
dataset.describe()
```

Out[173]:

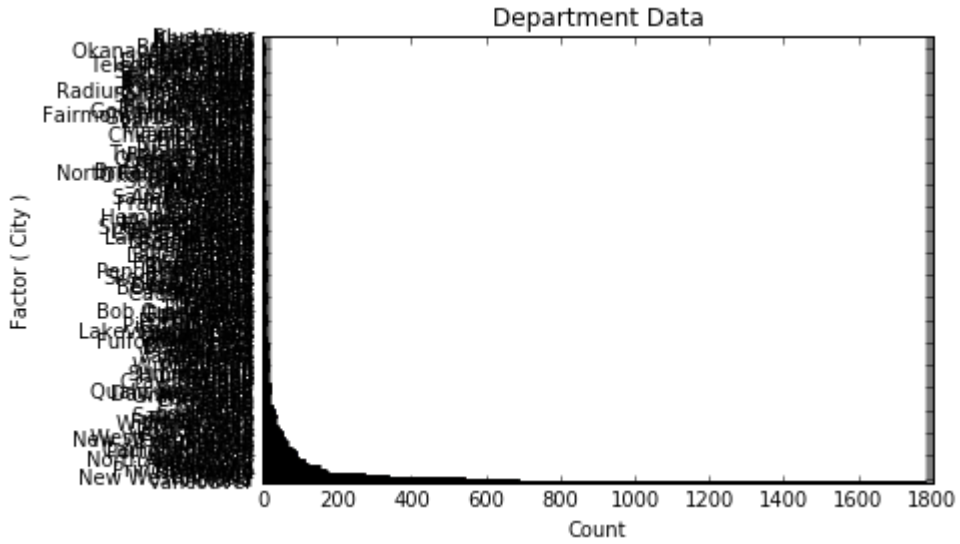
	EmployeeNumber	Age	LengthService	AbsentHours
count	8336.000000	8336.000000	8336.000000	8336.000000
mean	4168.500000	42.007086	4.782910	61.283978
std	2406.540255	9.939798	2.462990	49.038365
min	1.000000	3.504743	0.012098	0.000000
25%	2084.750000	35.298748	3.575892	19.127590
50%	4168.500000	42.114924	4.600248	56.005808
75%	6252.250000	48.666943	5.623922	94.284692
max	8336.000000	77.938003	43.735239	272.530123

In [174]:
```
# Gender Barplot
dataset['Gender'].value_counts().plot.barh()
```

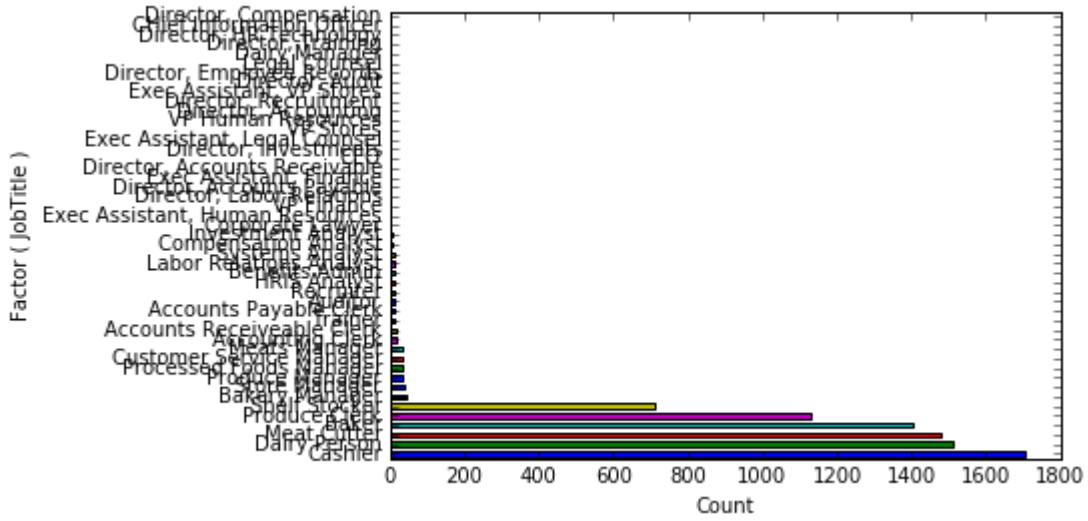
Out[174]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x25d1bd19a58>
```



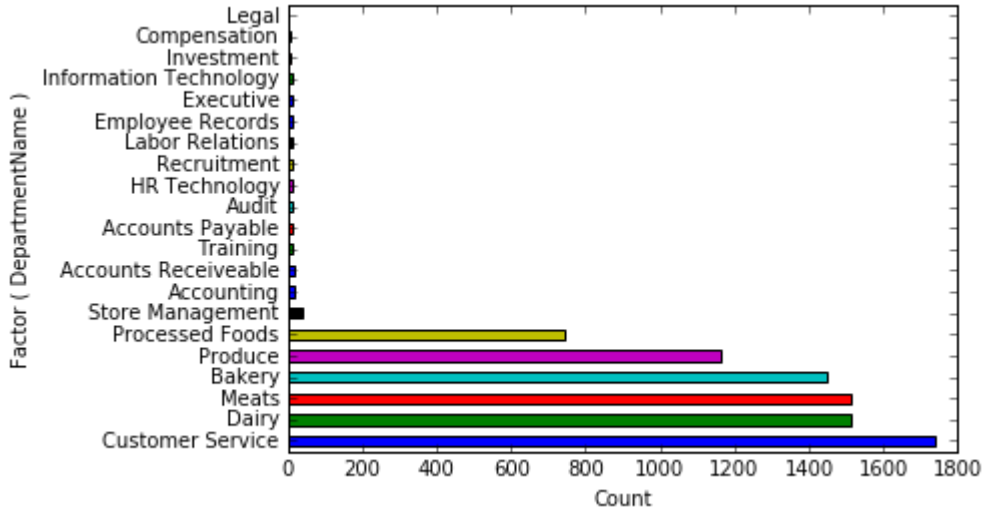
In [175]:
```
# Department Barplot
dataset['City'].value_counts().plot.barh()
plt.title('Department Data')
plt.xlabel('Count')
plt.ylabel('Factor ( City )')
plt.show()
```



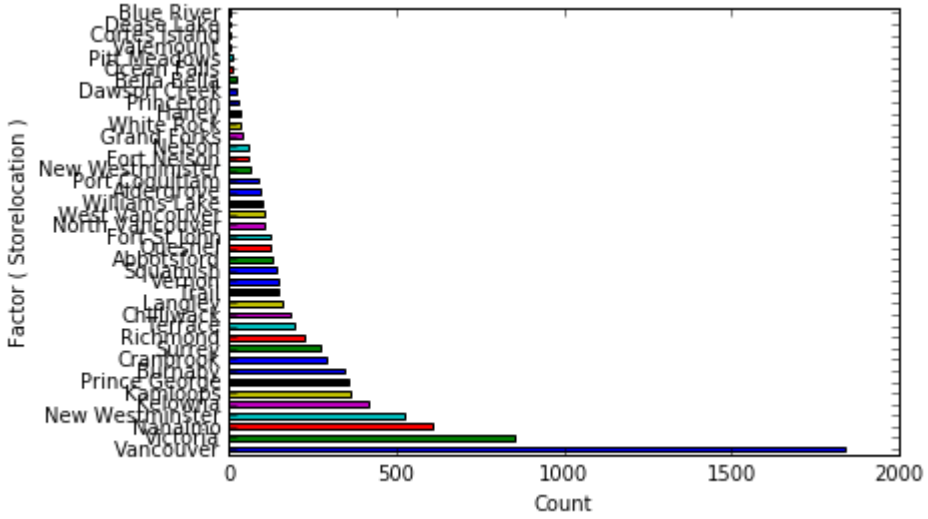
In [176]:
```
# JobTitle Barplot
dataset['JobTitle'].value_counts().plot.barh()
plt.xlabel('Count')
plt.ylabel('Factor ( JobTitle )')
plt.show()
```



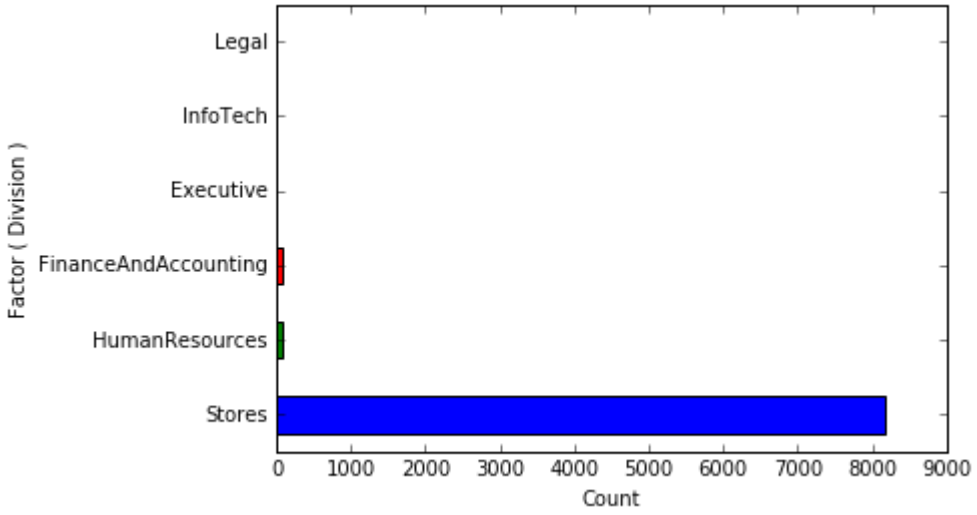
In [177]:
```
# Department Barplot
dataset['DepartmentName'].value_counts().plot.barh()
plt.xlabel('Count')
plt.ylabel('Factor ( DepartmentName )')
plt.show()
```



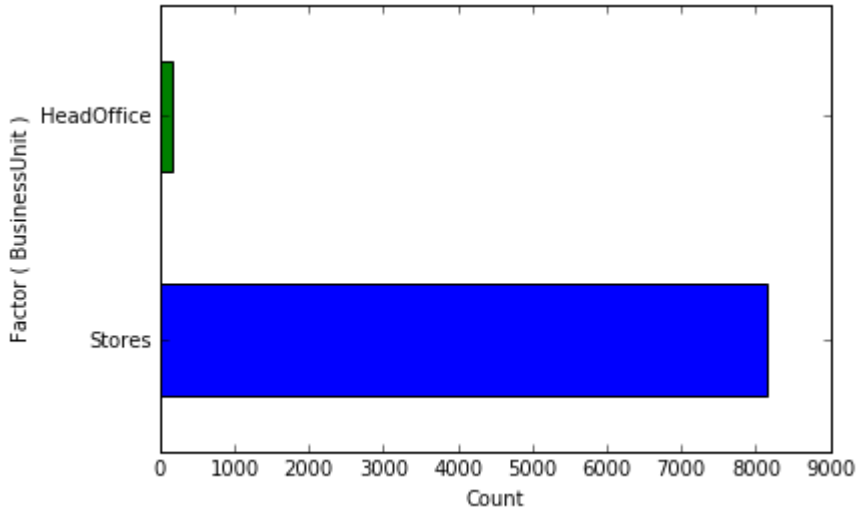
In [178]:
```
# Store Location Barplot
dataset['StoreLocation'].value_counts().plot.barh()
plt.xlabel('Count')
plt.ylabel('Factor ( Storelocation )')
plt.show()
```



In [179]:
```
# Division Barplot
dataset['Division'].value_counts().plot.barh()
plt.xlabel('Count')
plt.ylabel('Factor ( Division )')
plt.show()
```



In [180]:
```
# Business Unit Barplot
dataset['BusinessUnit'].value_counts().plot.barh()
plt.xlabel('Count')
plt.ylabel('Factor ( BusinessUnit )')
plt.show()
```

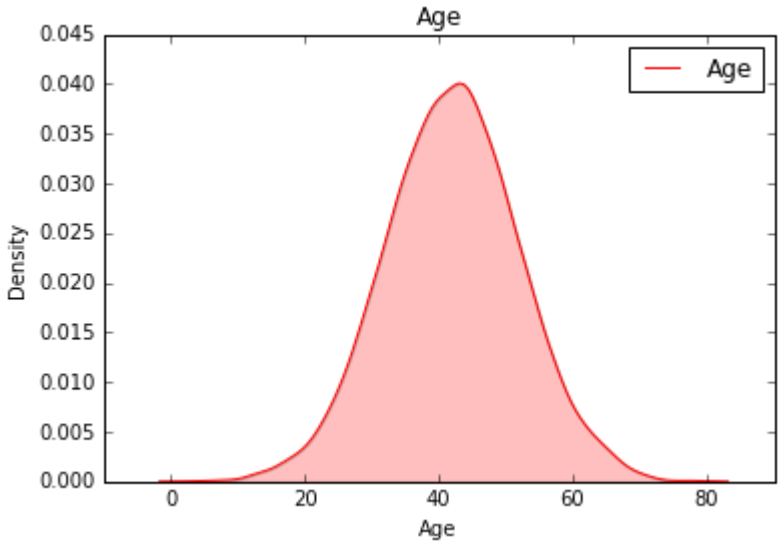


In [181]:
```
dataset['City'].value_counts()
```

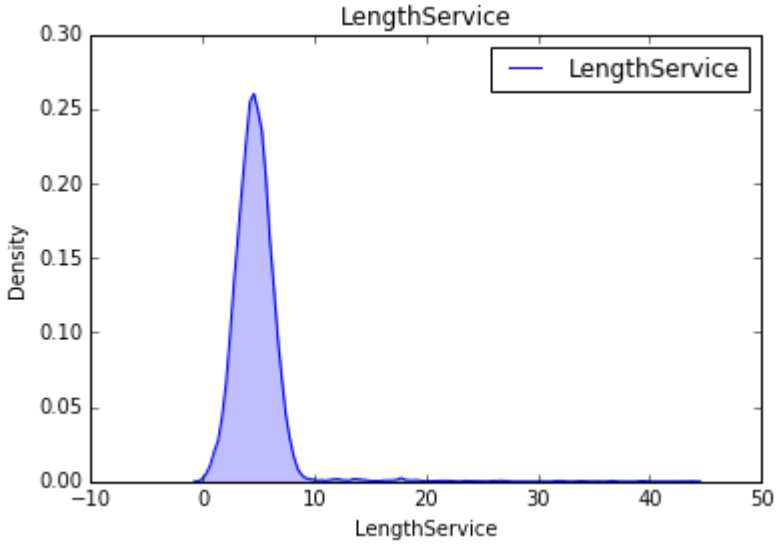
Out[181]:
```
Vancouver           1780
Victoria             690
New Westminster      540
Burnaby              339
Surrey               275
Richmond             228
Nanaimo              176
Prince George        174
Kelowna              158
Kamloops             156
Langley              122
Abbotsford           114
North Vancouver      111
Aldergrove            94
Chilliwack            90
Vernon                90
Port Coquitlam        89
Campbell River        87
Penticton             82
Whistler              82
Cranbrook             72
Duncan                72
Fort St John          69
New Westminister      62
West Vancouver        60
Fort Nelson           55
Terrace               55
Port Alberni          54
Courtenay             49
Nelson                45
                     ...
Cluculz Lake           5
Nakusp                 5
Toad River             5
Port Renfrew           5
Bowen Island           5
Bouchie Lake           5
Pouce Coupe            5
Sayward                5
Rosedale               5
Field                  4
```

```
Seton Portage          4
Bougie Creek           4
Port Mellon            4
Telegraph Creek        4
Dease Lake             4
Pemberton              4
Douglas Lake           4
Lillooet               4
Little Fort            4
Sicamous               4
Alkali Lake            3
Okanagan Mission       3
Bear Lake              3
Bridge Lake            3
Salmo                  3
Wells                  3
Black Pool             3
Keremeos               2
Lytton                 2
Blue River             2
Name: City, Length: 243, dtype: int64
```

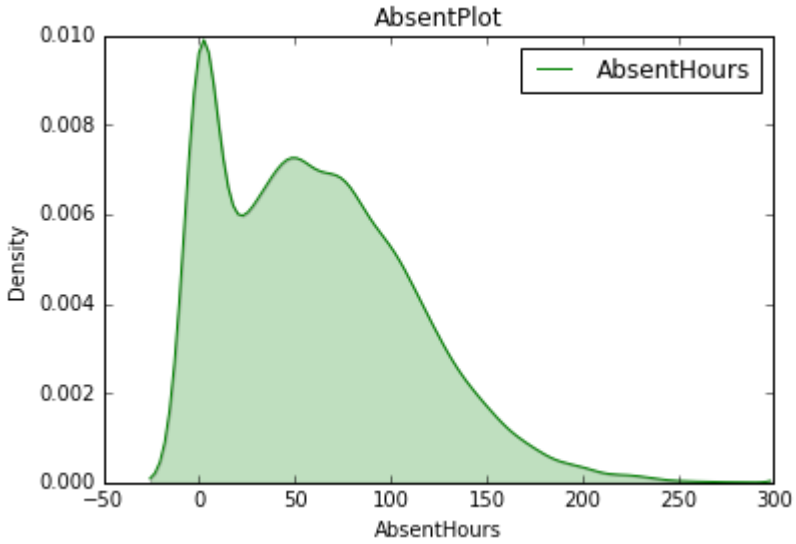
In [182]:
```
sns.kdeplot(dataset['Age'], color = 'red', shade = True)
plt.title('Age')
plt.xlabel('Age')
plt.ylabel('Density')
plt.show()
```



In [183]:
```
sns.kdeplot(dataset['LengthService'], color = 'blue', shade = True)
plt.title('LengthService')
plt.xlabel('LengthService')
plt.ylabel('Density')
plt.show()
```



In [184]:
```
sns.kdeplot(dataset['AbsentHours'], color = 'green', shade = True)
plt.title('AbsentPlot')
plt.xlabel('AbsentHours')
plt.ylabel('Density')
plt.show()
```



In [185]:
```
#----------- Eliminate Outliers -----------------
dataset=dataset[(dataset['Age'] >= 18) & (dataset['Age'] <= 65)]
dataset['Age'].describe()
```

Out[185]:
```
count    8165.000000
mean       41.985633
std         9.276915
min        18.204720
25%        35.456296
50%        42.097897
75%        48.513876
max        65.000000
Name: Age, dtype: float64
```

In [186]:
```
dataset.describe()
```

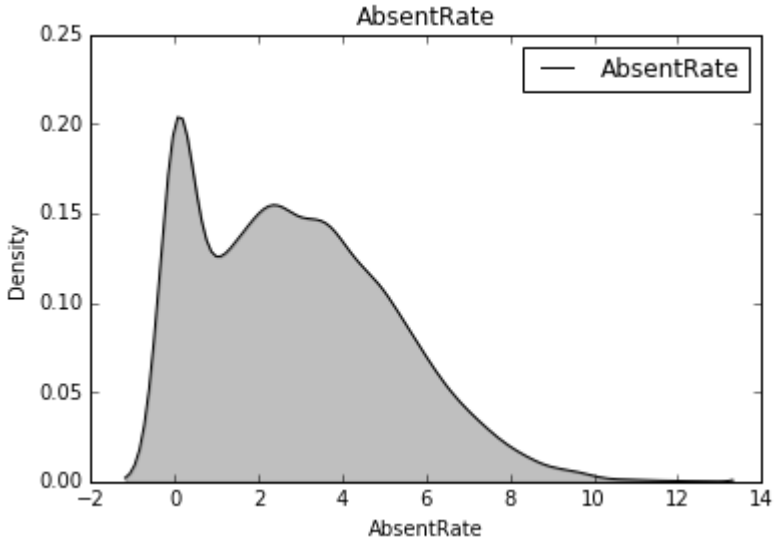
Out[186]:

	EmployeeNumber	Age	LengthService	AbsentHours
count	8165.000000	8165.000000	8165.000000	8165.000000
mean	4164.661237	41.985633	4.788871	60.471110
std	2403.600726	9.276915	2.478484	47.107030
min	1.000000	18.204720	0.053279	0.000000
25%	2081.000000	35.456296	3.582605	20.067078
50%	4166.000000	42.097897	4.597999	55.862962
75%	6245.000000	48.513876	5.623582	93.381290
max	8336.000000	65.000000	43.735239	252.193535

In [187]:
```
#-------- Create a column AbsentRate form AbsentHours ----------------------
dataset['AbsentRate']=dataset['AbsentHours']/2080*100
sns.kdeplot(dataset['AbsentRate'], color = 'black', shade = True)
plt.title('AbsentRate')
plt.xlabel('AbsentRate')
plt.ylabel('Density')
plt.show()
```

```
C:\Users\Hajimalang\Anaconda3\lib\site-packages\ipykernel_launcher.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy (http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy)
```



In [188]:
```
np.mean(dataset['AbsentRate'])
```

Out[188]:
```
2.9072649276669615
```

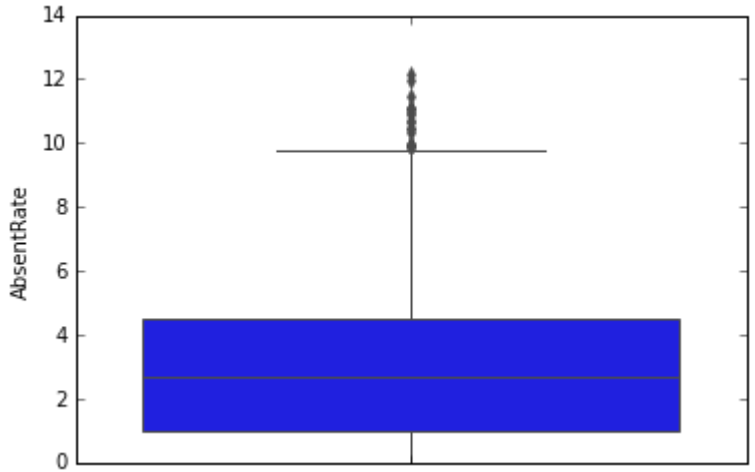
In [189]:
```
dataset.describe()
```

Out[189]:

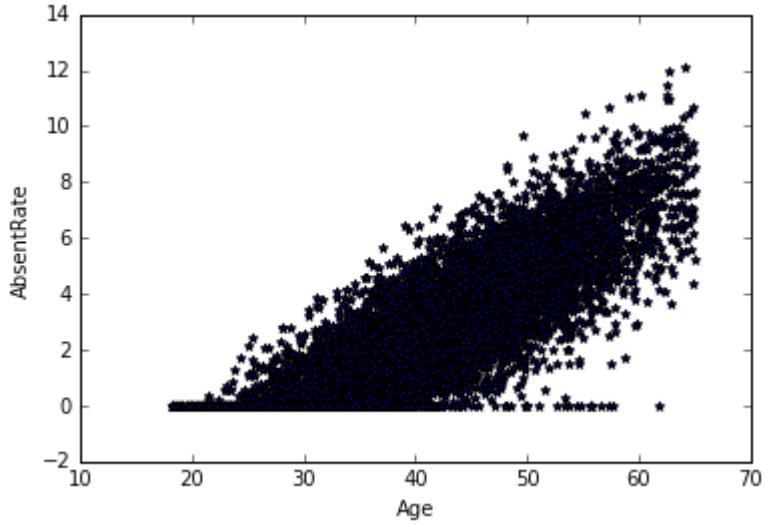
	EmployeeNumber	Age	LengthService	AbsentHours	AbsentRate
count	8165.000000	8165.000000	8165.000000	8165.000000	8165.000000
mean	4164.661237	41.985633	4.788871	60.471110	2.907265
std	2403.600726	9.276915	2.478484	47.107030	2.264761
min	1.000000	18.204720	0.053279	0.000000	0.000000
25%	2081.000000	35.456296	3.582605	20.067078	0.964763
50%	4166.000000	42.097897	4.597999	55.862962	2.685719
75%	6245.000000	48.513876	5.623582	93.381290	4.489485
max	8336.000000	65.000000	43.735239	252.193535	12.124689

In [190]:
```
#------------ import seaborn as sns ----------------------------
sns.boxplot(y=dataset['AbsentRate'])
```

Out[190]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x25d1c05b518>
```



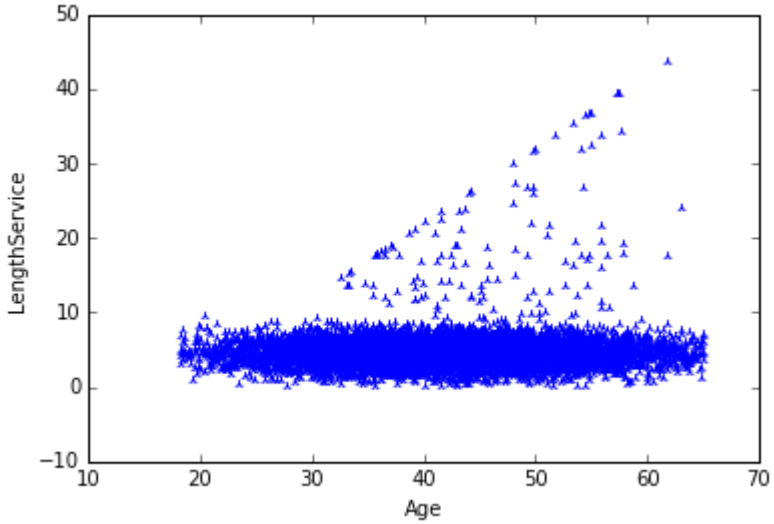
In [191]:
```
plt.scatter(x=dataset['Age'], y = dataset['AbsentRate'],marker = '*')
plt.xlabel('Age')
plt.ylabel('AbsentRate')
plt.show()
```



In [192]:
```
np.corrcoef(dataset['Age'], dataset['AbsentRate'])
```

Out[192]:
```
array([[ 1.        ,  0.82461291],
       [ 0.82461291,  1.        ]])
```

In [193]:
```
plt.scatter(x=dataset['Age'],y=dataset['LengthService'], marker = '2')
plt.xlabel('Age')
plt.ylabel('LengthService')
plt.show()
```



In [194]:
```
np.corrcoef(dataset['Age'], dataset['LengthService'])
```

Out[194]:
```
array([[ 1.        ,  0.05623405],
       [ 0.05623405,  1.        ]])
```

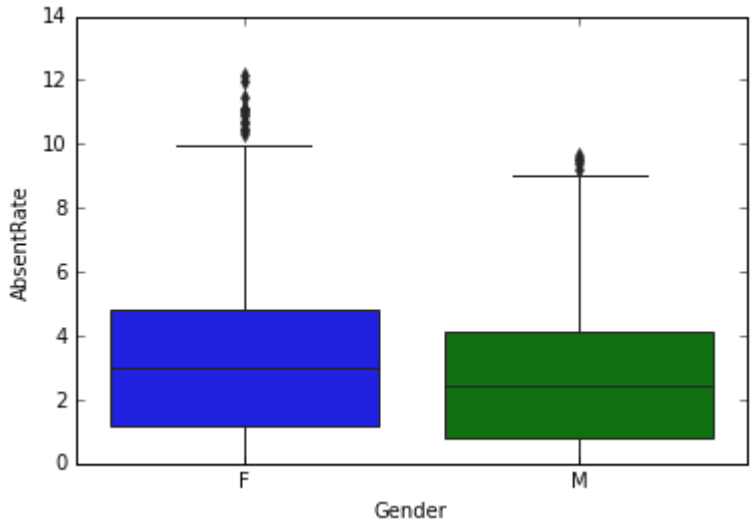
In [195]:
```
dataset.corr()
```

Out[195]:

	EmployeeNumber	Age	LengthService	AbsentHours	AbsentRate
EmployeeNumber	1.000000	-0.024625	-0.123113	0.001608	0.001608
Age	-0.024625	1.000000	0.056234	0.824613	0.824613
LengthService	-0.123113	0.056234	1.000000	-0.046692	-0.046692
AbsentHours	0.001608	0.824613	-0.046692	1.000000	1.000000
AbsentRate	0.001608	0.824613	-0.046692	1.000000	1.000000

In [196]:
```
sns.boxplot(y=dataset['AbsentRate'], x = dataset['Gender'])
```

Out[196]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x25d1c078278>
```



In [197]:
```
dataset['Gender'].value_counts()   #-------- Getting a count of Male and Female in the column Gender -----------
dataset.groupby(['Gender'])['AbsentRate'].mean()    #----- Getting mean of AbsentRate with each Gender ----------
```

Out[197]:
```
Gender
F    3.157624
M    2.664813
Name: AbsentRate, dtype: float64
```

In [198]:
```
#-------------- Linear Regression ------------------------------------
from sklearn.linear_model import LinearRegression
lm = LinearRegression()

#--------- Converting Categorical Variables to numeric using get_dummies ------------------
Gen_Male = pd.get_dummies(dataset.Gender).iloc[:,1:]
datasetF = pd.concat([dataset,Gen_Male],axis=1)  #------------ axis rows ke liye 0 columns ke liye 1 ----------
```

In [199]:
```
X = Gen_Male.values.reshape((8165,1))
X.shape
y=dataset['AbsentRate'].values.reshape((8165,1))
y.shape
```

Out[199]:
```
(8165, 1)
```

In [200]:
```
dataset.head()
```

Out[200]:

	EmployeeNumber	Surname	GivenName	Gender	City	JobTitle	DepartmentName	StoreLocation	Division	Age	LengthSe
0	1	Gutierrez	Molly	F	Burnaby	Baker	Bakery	Burnaby	Stores	32.028816	6.0
1	2	Hardwick	Stephen	M	Courtenay	Baker	Bakery	Nanaimo	Stores	40.320902	5.5
2	3	Delgado	Chester	M	Richmond	Baker	Bakery	Richmond	Stores	48.822047	4.3
3	4	Simon	Irene	F	Victoria	Baker	Bakery	Victoria	Stores	44.599357	3.0
4	5	Delvalle	Edward	M	New Westminster	Baker	Bakery	New Westminster	Stores	35.697876	3.6

In [201]:
```
#--------- Linear Regression ------------
data = lm.fit(X,y)
data
lm.intercept_
lm.coef_
```

Out[201]:
```
array([[-0.49281131]])
```

In [202]:
```
import scipy
from scipy import stats
import statsmodels
import statsmodels.api as sm
from statsmodels.formula.api import ols

mod = ols('AbsentRate~Gender',data=dataset).fit()
anoval = sm.stats.anova_lm(mod,type=1)
print(anoval)
```

```
              df        sum_sq     mean_sq          F        PR(>F)
Gender       1.0    495.616457  495.616457  97.772928  6.310896e-23
Residual  8163.0  41378.704733    5.069056        NaN           NaN
```

In [203]:
```
dataset['Gender'].value_counts()       #------ Getting a count of Male and Female in the column Gender --------
dataset.groupby(['Gender'])['AbsentRate'].mean()    #-------- Getting mean of AbsentRate with each Gender -------
```

Out[203]:
```
Gender
F    3.157624
M    2.664813
Name: AbsentRate, dtype: float64
```

In [204]:
```
#------------ One Hot Encoding of City ------------------
city1 = pd.get_dummies(dataset.City).iloc[:,1:]
l2 = lm.fit(city1,y)
l2
l2.intercept_
l2.coef_
```

Out[204]:
```
array([[ -8.45336936e-02,   5.64178113e-01,   2.32103279e-02,
          5.98033165e-01,  -5.77787225e-02,   1.62170969e-01,
          8.05505828e-01,  -7.87094686e-02,  -2.77479108e-01,
          1.88612978e+00,  -2.72640432e-01,  -1.04681003e-01,
         -5.86711663e-01,  -9.39786733e-01,  -1.19062643e+00,
         -5.73577396e-01,  -6.89777986e-01,   5.40658387e-02,
          8.86629218e-01,   5.40972021e-01,   3.15899157e-01,
         -1.01982794e+00,   1.95149143e-01,  -8.76022879e-01,
          9.37355207e-01,  -5.45709713e-01,  -4.00760487e-01,
          7.17227369e-02,  -1.09740577e+00,   1.78791836e-01,
          5.34488641e-01,  -1.75008443e-01,  -1.05337636e+00,
          1.34440634e-01,  -9.69169096e-01,  -2.94804939e-01,
          3.69926206e-01,   2.76166765e-01,   2.33324731e-01,
         -1.23483019e+00,   1.98951771e-01,   1.99520634e-01,
          1.12104161e-01,  -5.60028430e-01,   1.50471071e-01,
          1.90706821e+00,   2.48015947e-02,  -4.13596599e-01,
         -1.45106644e-01,  -3.72594337e-01,   5.25533673e-01,
          3.63877128e-02,  -2.19696875e-01,  -1.66168240e-04,
         -1.46389399e-01,   2.34806434e-01,   8.94156226e-01,
         -1.40094240e+00,   4.56051078e-01,   1.85391379e+00,
          2.53512130e-01,   2.79374737e-01,   6.75886422e-01,
         -3.21248258e-01,  -9.23786841e-01,   7.51396786e-01,
          1.01442937e+00,  -1.38626235e+00,   1.48098922e+00,
          1.23325594e+00,   4.21057219e-01,   5.29727298e-01,
         -4.40901645e-01,  -5.89093496e-02,   6.70422728e-01,
         -4.67556503e-01,  -9.68757624e-01,   2.67175145e-01,
          2.24670116e-01,   6.01218534e-01,   3.16589725e-01,
          7.40257680e-01,  -6.36607801e-01,  -8.33793775e-01,
          5.55180770e-02,   5.63239426e-01,   3.07163839e-01,
          7.27960142e-02,   6.48860784e-01,  -5.74930685e-01,
          1.53316703e-01,  -1.03056393e-01,   1.11714989e-01,
         -3.08024242e-01,   2.21359583e+00,   2.03061234e+00,
         -9.41803214e-01,   1.15197967e+00,   8.06268921e-01,
          1.34533111e-01,  -3.87588951e-01,  -4.79509507e-02,
          6.36592865e-01,  -5.12620487e-01,   4.59247863e-01,
          8.20618264e-01,  -1.17504827e-01,  -4.60626604e-02,
```

```
-1.58409036e-01,  -5.09092554e-01,  -1.77855311e-01,
 1.04524050e+00,   5.60441546e-03,  -8.61879650e-01,
 7.03819573e-01,   1.05167367e+00,  -1.56773169e+00,
-3.65385948e-01,   3.29645795e-01,  -2.42852061e-01,
 6.32896792e-01,   9.43034277e-01,   3.47471942e-01,
-6.16025372e-01,  -1.24506257e-02,  -2.78571608e-02,
 1.10994478e+00,  -2.42739868e-01,  -1.61841323e+00,
-1.28909656e+00,   3.73997680e-01,  -4.62063227e-01,
-2.09011944e-01,   4.10603140e-01,   5.81398369e-01,
 7.07176428e-01,  -6.27509715e-01,  -8.11137710e-01,
 7.11346329e-01,  -1.33803980e-01,   3.45968335e-01,
 6.58774765e-02,   7.15773491e-02,  -2.37245954e-01,
-2.71863509e-01,   1.19746294e-01,   8.53535378e-01,
-4.26173875e-01,   2.27376510e+00,  -8.26879073e-02,
 5.36214301e-01,   1.08588987e+00,   7.87104857e-02,
-2.22893805e-01,  -8.26193636e-01,   2.47100870e-01,
-1.04566045e+00,  -3.62196039e-01,   3.51556830e-02,
-5.09944250e-01,  -8.68560459e-02,   7.57643600e-01,
 2.87889332e-01,   4.69240189e-01,   3.52104001e-01,
 5.65223097e-01,  -8.97648084e-01,  -5.42207005e-01,
-8.84907646e-01,   2.31194428e-01,   6.80322730e-02,
 3.46060735e-01,   3.61216130e-01,  -1.06392323e-01,
 1.47189316e-01,   3.28636025e-01,   4.15310184e-01,
-2.00213388e-01,   2.36126638e-01,   9.84370368e-01,
 4.74427009e-01,  -2.21783691e-01,   1.17938974e+00,
-3.29126655e-01,  -4.28406109e-01,  -2.73757958e-02,
 9.81436391e-01,  -2.23118719e+00,  -8.11464228e-02,
-8.05316378e-01,  -2.95491914e-01,   1.02462781e+00,
-9.46307016e-01,   3.19090417e-01,  -3.43048291e-01,
 1.17126246e-01,   5.98189845e-01,  -5.25469386e-01,
-1.39261442e-01,   1.20499821e+00,   7.76798907e-01,
 2.80123213e-01,   1.28037269e+00,  -1.23567565e-02,
 1.17557109e-01,   1.75564597e-01,  -1.64229522e+00,
 5.38450966e-01,  -1.95415248e+00,   1.48452061e+00,
 1.16495371e-01,  -7.29414637e-01,   3.60703277e-01,
-1.26328144e-01,   3.88106849e-01,  -1.00068801e-01,
 4.04622237e-01,   1.27391125e+00,   5.24109253e-01,
-1.84074153e-01,   4.99938158e-02,   4.55668244e-03,
-6.87831507e-01,   6.19456117e-01,   3.05282362e-01,
-5.79042744e-02,  -1.06425120e+00,   2.85323427e-01,
 1.55390423e-01,   5.56428083e-01,   6.57890359e-02,
 4.12560970e-01,  -5.86515999e-01,  -1.11468944e-01,
-6.79337281e-01,   5.27283748e-03,  -1.26944672e+00,
```

```
        2.32240721e+00,  -7.10158747e-01,   1.31393492e+00,
        1.65158288e+00,  -4.15966110e-01]])
```

In [205]:
```
#------- OLS for each variable against AbsentRate ----------------

m2 = ols('AbsentRate~City',data=dataset).fit()      #------ AbsentRate ~ City ---------
anova2 = sm.stats.anova_lm(m2,type=2)
print(anova2)
```

```
              df        sum_sq   mean_sq         F    PR(>F)
City       242.0   1084.592061  4.481785  0.870432  0.925378
Residual  7922.0  40789.729129  5.148918       NaN       NaN
```

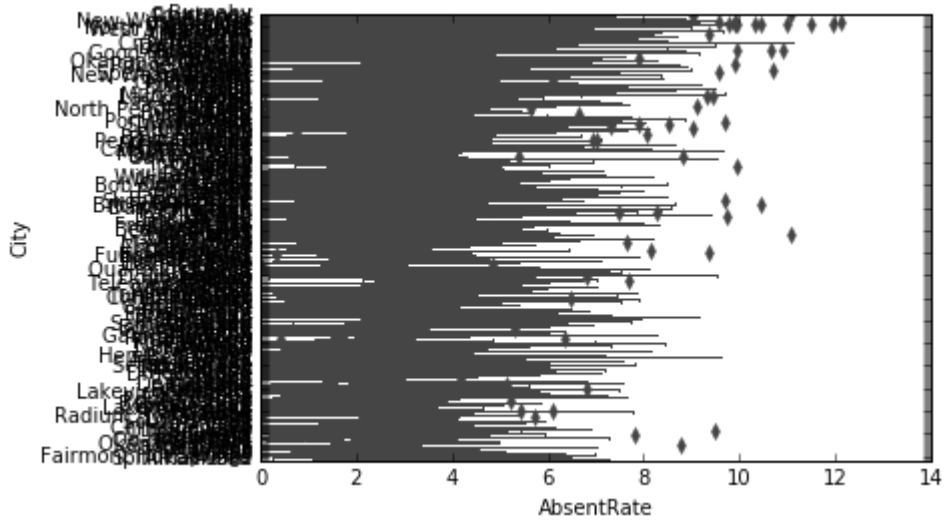
In [206]:
```
dataset.groupby(['City'])['AbsentRate'].mean()
```

Out[206]:
```
City
Abbotsford         2.857907
Agassiz            2.773373
Aiyansh            3.422085
Aldergrove         2.881117
Alexis Creek       3.455940
Alkali Lake        2.800128
Armstrong          3.020078
Ashcroft           3.663413
Atlin              2.779197
Avola              2.580428
Balfour            4.744037
Bamfield           2.585266
Barriere           2.753226
Bear Lake          2.271195
Beaver Valley      1.918120
Bella Bella        1.667280
Black Point        2.284329
Black Pool         2.168129
Blue River         2.911973
Blueberry          3.744536
Bob Quinn Lake     3.398879
Boston Bar         3.173806
Bouchie Lake       1.838079
Bougie Creek       3.053056
Bowen Island       1.981884
Brackendale        3.795262
Bridge Lake        2.312197
Britannia Beach    2.457146
Burnaby            2.929630
Burns Lake         1.760501
                      ...
Tofino             3.218610
Topley             2.731579
Trail              3.246014
Tumbler Ridge      2.757838
Ucluelet           3.262529
Union Bay          4.131818
Valemount          3.382016
Vallican           2.673833
Vananda            2.907901
```

```
Vancouver          2.862464
Vanderhoof         2.170075
Vavenby            3.477363
Vernon             3.163189
Victoria           2.800003
Wells              1.793656
West Vancouver     3.143230
Westbank           3.013297
Westwold           3.414335
Whistler           2.923696
White Rock         3.270468
Wildwood           2.271391
Williams Lake      2.746438
Willow Point       2.178570
Winfield           2.863180
Woss               1.588460
Wynndel            5.180314
Yahk               2.147748
Yale               4.171842
Yarrow             4.509490
Youbou             2.441941
Name: AbsentRate, Length: 243, dtype: float64
```

In [207]:
```
sns.boxplot(x='AbsentRate',y='City',data=dataset)
```

Out[207]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x25d1116e358>
```



In [208]:
```
m3 = ols('AbsentRate~JobTitle',data=dataset).fit()      #-------- AbsentRate ~ JobTitle ------------
anova3 = sm.stats.anova_lm(m3,type=2)
print(anova3)
```

```
               df         sum_sq   mean_sq         F    PR(>F)
JobTitle     46.0    281.136153  6.111655  1.19285  0.174517
Residual   8118.0  41593.185037  5.123575      NaN       NaN
```

In [209]:
```
dataset.groupby(['JobTitle'])['AbsentRate'].mean()
```

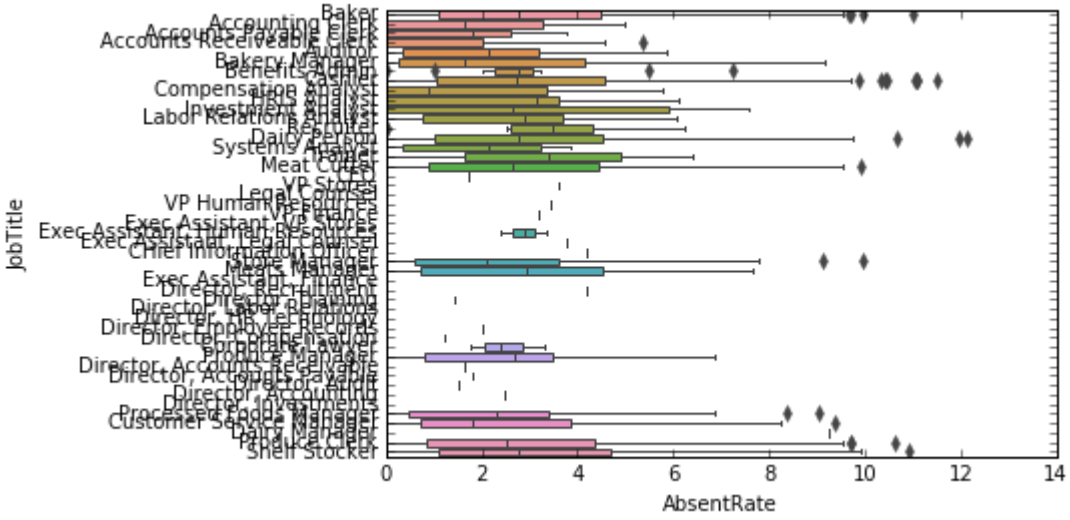
Out[209]:

JobTitle	
Accounting Clerk	1.946229
Accounts Payable Clerk	1.623203
Accounts Receiveable Clerk	1.419389
Auditor	2.160575
Baker	2.923429
Bakery Manager	2.579606
Benefits Admin	2.972834
CEO	1.709012
CHief Information Officer	4.170790
Cashier	2.996985
Compensation Analyst	1.790482
Corporate Lawyer	2.471724
Customer Service Manager	2.546558
Dairy Manager	9.250115
Dairy Person	2.991750
Director, Accounting	2.462047
Director, Accounts Payable	1.818838
Director, Accounts Receivable	1.647439
Director, Audit	1.499403
Director, Compensation	1.218407
Director, Employee Records	2.006652
Director, HR Technology	0.000000
Director, Investments	0.000000
Director, Labor Relations	0.000000
Director, Recruitment	4.177417
Director, Training	1.412981
Exec Assistant, Finance	0.000000
Exec Assistant, Human Resources	2.872417
Exec Assistant, Legal Counsel	3.758468
Exec Assistant, VP Stores	0.000000
HRIS Analyst	2.493649
Investment Analyst	3.190082
Labor Relations Analyst	2.655482
Legal Counsel	0.000000
Meat Cutter	2.846953
Meats Manager	3.006050
Processed Foods Manager	2.617214
Produce Clerk	2.800071
Produce Manager	2.684084
Recruiter	3.186081

```
Shelf Stocker                   3.002924
Store Manager                   2.517101
Systems Analyst                 1.925995
Trainer                         3.084251
VP Finance                      3.173603
VP Human Resources              3.416529
VP Stores                       3.586138
Name: AbsentRate, dtype: float64
```

In [210]:
```
sns.boxplot(x='AbsentRate',y='JobTitle',data=dataset)
```

Out[210]: <matplotlib.axes._subplots.AxesSubplot at 0x25d167ee7f0>



In [211]:
```
m4 = ols('AbsentRate~DepartmentName',data=dataset).fit()       #----- AbsentRate ~ DepartmentName ---------
anova4 = sm.stats.anova_lm(m4,type=2)
print(anova4)
```

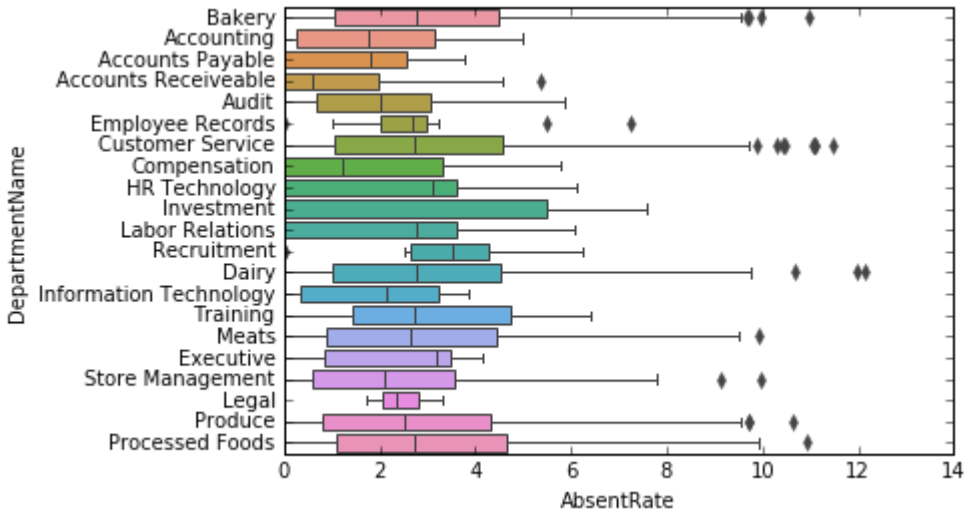
```
                    df         sum_sq   mean_sq         F    PR(>F)
DepartmentName    20.0     171.543244  8.577162  1.675006  0.029954
Residual        8144.0   41702.777946  5.120675       NaN       NaN
```

In [212]:
```
dataset.groupby(['DepartmentName'])['AbsentRate'].mean()
```

Out[212]:
```
DepartmentName
Accounting                1.974886
Accounts Payable          1.636245
Accounts Receiveable      1.433642
Audit                     2.116497
Bakery                    2.912533
Compensation              1.726918
Customer Service          2.988482
Dairy                     2.995988
Employee Records          2.892319
Executive                 2.323580
HR Technology             2.315531
Information Technology    1.925995
Investment                2.835628
Labor Relations           2.434192
Legal                     2.471724
Meats                     2.850572
Processed Foods           2.985082
Produce                   2.796795
Recruitment               3.262338
Store Management          2.517101
Training                  2.972833
Name: AbsentRate, dtype: float64
```

In [213]:
```
sns.boxplot(x='AbsentRate',y='DepartmentName',data=dataset)
```

Out[213]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x25d18905d68>
```



In [214]:
```
m5 = ols('AbsentRate~StoreLocation',data=dataset).fit()      #-------- AbsentRate ~ StoreLocation ----------
anova5 = sm.stats.anova_lm(m5,type=2)
print(anova5)
```

```
                   df        sum_sq   mean_sq         F    PR(>F)
StoreLocation    39.0    191.282363  4.904676  0.956036  0.548266
Residual       8125.0  41683.038827  5.130220       NaN       NaN
```

In [215]:
```
dataset.groupby(['StoreLocation'])['AbsentRate'].mean()
```

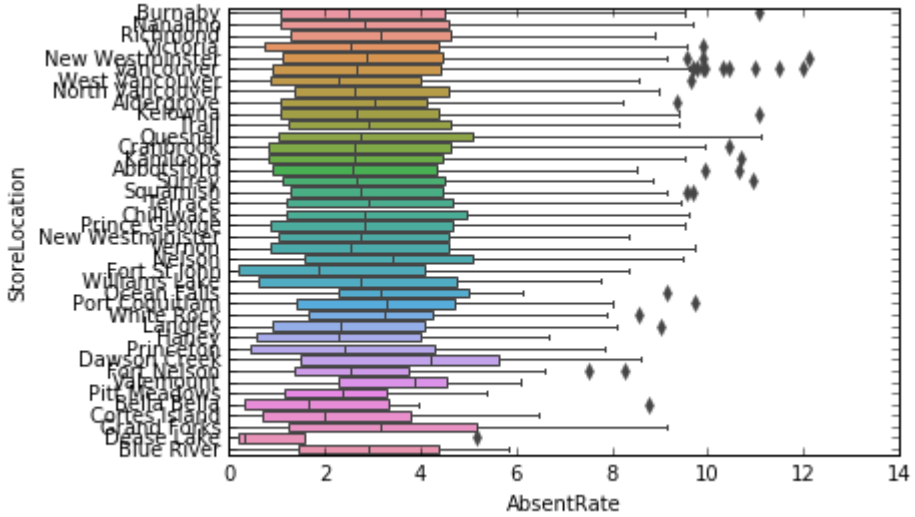
Out[215]:

StoreLocation	
Abbotsford	2.937386
Aldergrove	2.901258
Bella Bella	2.031680
Blue River	2.911973
Burnaby	2.944466
Chilliwack	3.082989
Cortes Island	2.485313
Cranbrook	2.952225
Dawson Creek	3.545463
Dease Lake	1.456964
Fort Nelson	2.741195
Fort St John	2.422886
Grand Forks	3.251701
Haney	2.549883
Kamloops	2.869252
Kelowna	2.898045
Langley	2.579410
Nanaimo	2.975264
Nelson	3.454472
New Westminister	2.930529
New Westminster	2.969692
North Vancouver	3.016872
Ocean Falls	3.711442
Pitt Meadows	2.347963
Port Coquitlam	3.160728
Prince George	2.919220
Princeton	2.679282
Quesnel	3.109154
Richmond	3.090128
Squamish	2.907684
Surrey	2.984772
Terrace	3.082298
Trail	3.115680
Valemount	3.382016
Vancouver	2.855725
Vernon	2.898143
Victoria	2.792531
West Vancouver	2.688103
White Rock	3.237530

```
Williams Lake       2.854828
Name: AbsentRate, dtype: float64
```

In [216]:
```
sns.boxplot(x='AbsentRate',y='StoreLocation',data=dataset)
```

Out[216]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x25d1be72400>
```



In [217]:
```
m6 = ols('AbsentRate~Division',data=dataset).fit()      #-------- AbsentRate ~ Division -----------
anova6 = sm.stats.anova_lm(m6,type=2)
print(anova6)
```

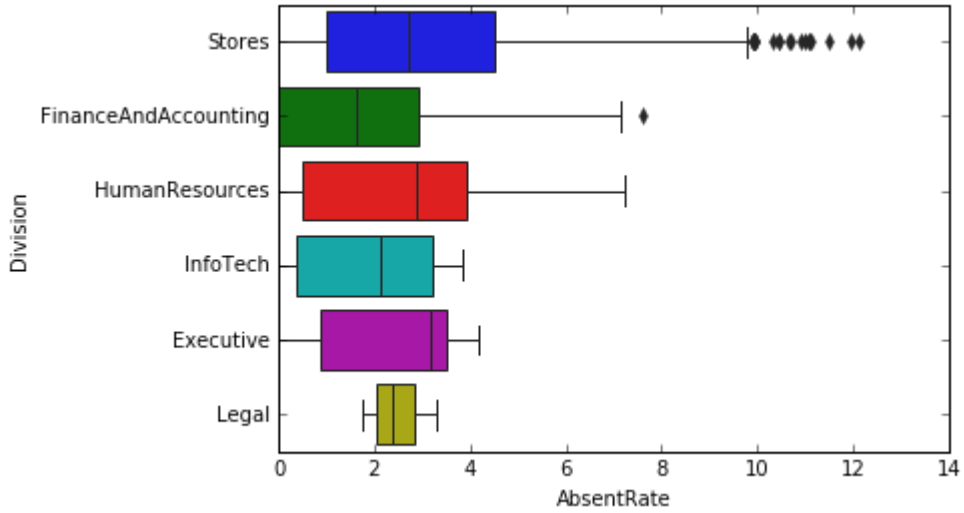
```
                df       sum_sq    mean_sq         F    PR(>F)
Division       5.0     91.19924  18.239848  3.561699  0.003218
Residual    8159.0  41783.12195   5.121108       NaN       NaN
```

In [218]:
```
dataset.groupby(['Division'])['AbsentRate'].mean()
```

Out[218]:
```
Division
Executive               2.323580
FinanceAndAccounting    1.921890
HumanResources          2.651743
InfoTech                1.925995
Legal                   2.471724
Stores                  2.920856
Name: AbsentRate, dtype: float64
```

In [219]:
```
sns.boxplot(x='AbsentRate',y='Division',data=dataset)
```

Out[219]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x25d1c1b2d68>
```



In [220]:
```
m7 = ols('AbsentRate~BusinessUnit',data=dataset).fit()     #------ AbsentRate ~ BusinessUnit --------
anova7 = sm.stats.anova_lm(m7,type=2)
print(anova7)
```

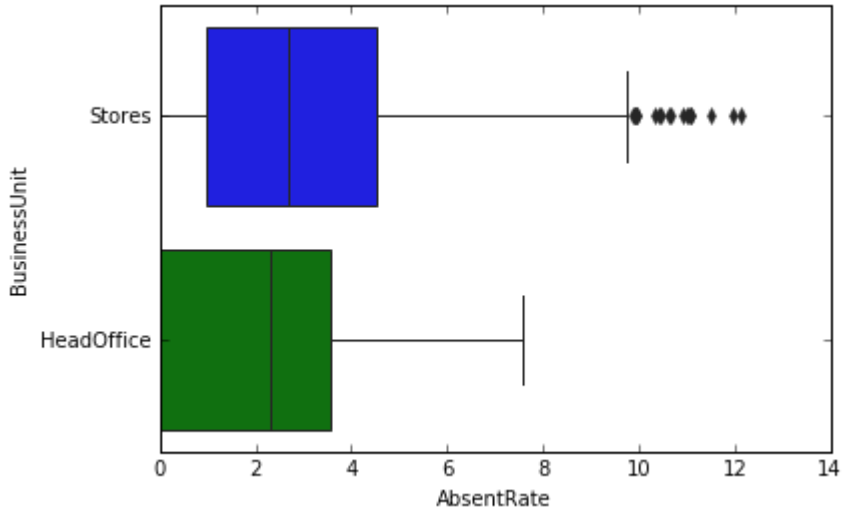
```
                  df        sum_sq    mean_sq          F    PR(>F)
BusinessUnit     1.0     70.091974  70.091974  13.686672  0.000217
Residual      8163.0  41804.229216   5.121185        NaN       NaN
```

In [221]:
```
dataset.groupby(['BusinessUnit'])['AbsentRate'].mean()
```

Out[221]:
```
BusinessUnit
HeadOffice    2.275658
Stores        2.920856
Name: AbsentRate, dtype: float64
```

In [222]:
```
sns.boxplot(x='AbsentRate',y='BusinessUnit',data=dataset)
```

Out[222]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x25d1e084ba8>
```



In [223]:
```
m8 = ols('AbsentRate~(Division*Gender)',data=dataset).fit()  #------ AbsentRate ~ Division * Gender --------
anova8 = sm.stats.anova_lm(m8,type=2)
print(anova8)
```

```
                    df        sum_sq     mean_sq          F        PR(>F)
Division           5.0     91.199240   18.239848   3.602225  2.953221e-03
Gender             1.0    495.927721  495.927721  97.941776  5.803030e-23
Division:Gender    5.0      4.517101    0.903420   0.178418  9.707833e-01
Residual        8153.0  41282.677128    5.063495        NaN           NaN
```

In [224]:
```
dataset.groupby(['Division','Gender'])['AbsentRate'].mean()
```

Out[224]:
```
Division              Gender
Executive             F         2.976419
                      M         1.779546
FinanceAndAccounting  F         2.172804
                      M         1.634077
HumanResources        F         3.014491
                      M         2.214311
InfoTech              F         3.298112
                      M         1.773538
Legal                 F         3.298112
                      M         2.058530
Stores                F         3.169049
                      M         2.680788
Name: AbsentRate, dtype: float64
```

In [225]:
```
#------------ Predictive Analysis ------------------
Mydataset = dataset
input = ("Gender", "DeptName", "StoreLocation","Division","Age","LengthService","BussUnit")
numeric = ("Age","LengthService")
categoric = ("Gender", "DeptName", "StoreLocation","Division","BussUnit")
target = ("AbsentRate")
```

In [226]:
```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 8165 entries, 0 to 8335
Data columns (total 14 columns):
EmployeeNumber    8165 non-null int64
Surname           8165 non-null object
GivenName         8165 non-null object
Gender            8165 non-null object
City              8165 non-null object
JobTitle          8165 non-null object
DepartmentName    8165 non-null object
StoreLocation     8165 non-null object
Division          8165 non-null object
Age               8165 non-null float64
LengthService     8165 non-null float64
AbsentHours       8165 non-null float64
BusinessUnit      8165 non-null object
AbsentRate        8165 non-null float64
dtypes: float64(4), int64(1), object(9)
memory usage: 1.2+ MB
```

In [227]:
```
#------ Convert Categoric variables to numeric -----------------
sha = pd.get_dummies(dataset.DepartmentName,drop_first=True).iloc[:,1:]
hm = pd.get_dummies(dataset.StoreLocation,drop_first=True).iloc[:,1:]
ss = pd.get_dummies(dataset.Division,drop_first=True).iloc[:,1:]
mmm = pd.get_dummies(dataset.BusinessUnit,drop_first=True).iloc[:,1:]
```

In [228]:
```
dataset = pd.concat([dataset,sha],axis=1)
dataset = pd.concat([dataset,hm],axis=1)
dataset = pd.concat([dataset,ss],axis=1)
dataset = pd.concat([dataset,mmm],axis=1)
```

In [229]:
```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 8165 entries, 0 to 8335
Data columns (total 75 columns):
EmployeeNumber            8165 non-null int64
Surname                   8165 non-null object
GivenName                 8165 non-null object
Gender                    8165 non-null object
City                      8165 non-null object
JobTitle                  8165 non-null object
DepartmentName            8165 non-null object
StoreLocation             8165 non-null object
Division                  8165 non-null object
Age                       8165 non-null float64
LengthService             8165 non-null float64
AbsentHours               8165 non-null float64
BusinessUnit              8165 non-null object
AbsentRate                8165 non-null float64
Accounts Receiveable      8165 non-null uint8
Audit                     8165 non-null uint8
Bakery                    8165 non-null uint8
Compensation              8165 non-null uint8
Customer Service          8165 non-null uint8
Dairy                     8165 non-null uint8
Employee Records          8165 non-null uint8
Executive                 8165 non-null uint8
HR Technology             8165 non-null uint8
Information Technology    8165 non-null uint8
Investment                8165 non-null uint8
Labor Relations           8165 non-null uint8
Legal                     8165 non-null uint8
Meats                     8165 non-null uint8
Processed Foods           8165 non-null uint8
Produce                   8165 non-null uint8
Recruitment               8165 non-null uint8
Store Management          8165 non-null uint8
Training                  8165 non-null uint8
Bella Bella               8165 non-null uint8
Blue River                8165 non-null uint8
Burnaby                   8165 non-null uint8
Chilliwack                8165 non-null uint8
Cortes Island             8165 non-null uint8
```

```
Cranbrook                8165 non-null uint8
Dawson Creek             8165 non-null uint8
Dease Lake               8165 non-null uint8
Fort Nelson              8165 non-null uint8
Fort St John             8165 non-null uint8
Grand Forks              8165 non-null uint8
Haney                    8165 non-null uint8
Kamloops                 8165 non-null uint8
Kelowna                  8165 non-null uint8
Langley                  8165 non-null uint8
Nanaimo                  8165 non-null uint8
Nelson                   8165 non-null uint8
New Westminister         8165 non-null uint8
New Westminster          8165 non-null uint8
North Vancouver          8165 non-null uint8
Ocean Falls              8165 non-null uint8
Pitt Meadows             8165 non-null uint8
Port Coquitlam           8165 non-null uint8
Prince George            8165 non-null uint8
Princeton                8165 non-null uint8
Quesnel                  8165 non-null uint8
Richmond                 8165 non-null uint8
Squamish                 8165 non-null uint8
Surrey                   8165 non-null uint8
Terrace                  8165 non-null uint8
Trail                    8165 non-null uint8
Valemount                8165 non-null uint8
Vancouver                8165 non-null uint8
Vernon                   8165 non-null uint8
Victoria                 8165 non-null uint8
West Vancouver           8165 non-null uint8
White Rock               8165 non-null uint8
Williams Lake            8165 non-null uint8
HumanResources           8165 non-null uint8
InfoTech                 8165 non-null uint8
Legal                    8165 non-null uint8
Stores                   8165 non-null uint8
dtypes: float64(4), int64(1), object(9), uint8(61)
memory usage: 1.7+ MB
```

In [230]:
```
sss = dataset
sss.drop(["GivenName","JobTitle","Gender", "DepartmentName", "StoreLocation","Division","BusinessUnit","City","Su
```

In [231]:
```
sss.info() #describe()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 8165 entries, 0 to 8335
Data columns (total 66 columns):
EmployeeNumber            8165 non-null int64
Age                       8165 non-null float64
LengthService             8165 non-null float64
AbsentHours               8165 non-null float64
AbsentRate                8165 non-null float64
Accounts Receiveable      8165 non-null uint8
Audit                     8165 non-null uint8
Bakery                    8165 non-null uint8
Compensation              8165 non-null uint8
Customer Service          8165 non-null uint8
Dairy                     8165 non-null uint8
Employee Records          8165 non-null uint8
Executive                 8165 non-null uint8
HR Technology             8165 non-null uint8
Information Technology    8165 non-null uint8
Investment                8165 non-null uint8
Labor Relations           8165 non-null uint8
Legal                     8165 non-null uint8
Meats                     8165 non-null uint8
Processed Foods           8165 non-null uint8
Produce                   8165 non-null uint8
Recruitment               8165 non-null uint8
Store Management          8165 non-null uint8
Training                  8165 non-null uint8
Bella Bella               8165 non-null uint8
Blue River                8165 non-null uint8
Burnaby                   8165 non-null uint8
Chilliwack                8165 non-null uint8
Cortes Island             8165 non-null uint8
Cranbrook                 8165 non-null uint8
Dawson Creek              8165 non-null uint8
Dease Lake                8165 non-null uint8
Fort Nelson               8165 non-null uint8
Fort St John              8165 non-null uint8
Grand Forks               8165 non-null uint8
Haney                     8165 non-null uint8
Kamloops                  8165 non-null uint8
Kelowna                   8165 non-null uint8
```

```
Langley                  8165 non-null uint8
Nanaimo                  8165 non-null uint8
Nelson                   8165 non-null uint8
New Westminister         8165 non-null uint8
New Westminster          8165 non-null uint8
North Vancouver          8165 non-null uint8
Ocean Falls              8165 non-null uint8
Pitt Meadows             8165 non-null uint8
Port Coquitlam           8165 non-null uint8
Prince George            8165 non-null uint8
Princeton                8165 non-null uint8
Quesnel                  8165 non-null uint8
Richmond                 8165 non-null uint8
Squamish                 8165 non-null uint8
Surrey                   8165 non-null uint8
Terrace                  8165 non-null uint8
Trail                    8165 non-null uint8
Valemount                8165 non-null uint8
Vancouver                8165 non-null uint8
Vernon                   8165 non-null uint8
Victoria                 8165 non-null uint8
West Vancouver           8165 non-null uint8
White Rock               8165 non-null uint8
Williams Lake            8165 non-null uint8
HumanResources           8165 non-null uint8
InfoTech                 8165 non-null uint8
Legal                    8165 non-null uint8
Stores                   8165 non-null uint8
dtypes: float64(4), int64(1), uint8(61)
memory usage: 1.2 MB
```

In [232]:
```
sss.head(3)
```

Out[232]:

	EmployeeNumber	Age	LengthService	AbsentHours	AbsentRate	Accounts Receiveable	Audit	Bakery	Compensation	Customer Service	...	Van
0	1	32.028816	6.018478	36.577306	1.758524	0	0	1	0	0	...	
1	2	40.320902	5.532445	30.165072	1.450244	0	0	1	0	0	...	
2	3	48.822047	4.389973	83.807798	4.029221	0	0	1	0	0	...	

3 rows × 66 columns

In [233]:
```
pd.get_dummies(dataset)  # ------- get all dummies row and columns ---------
```

Out[233]:

	EmployeeNumber	Age	LengthService	AbsentHours	AbsentRate	Accounts Receiveable	Audit	Bakery	Compensation	Customer Service	...	\
0	1	32.028816	6.018478	36.577306	1.758524	0	0	1	0	0	...	
1	2	40.320902	5.532445	30.165072	1.450244	0	0	1	0	0	...	
2	3	48.822047	4.389973	83.807798	4.029221	0	0	1	0	0	...	
3	4	44.599357	3.081736	70.020165	3.366354	0	0	1	0	0	...	
4	5	35.697876	3.619091	0.000000	0.000000	0	0	1	0	0	...	
5	6	48.440311	2.717692	81.830079	3.934138	0	0	1	0	0	...	
6	7	50.752730	10.157918	60.495072	2.908417	0	0	0	0	0	...	
7	8	36.216031	4.432123	30.072902	1.445813	0	0	1	0	0	...	
8	9	58.427380	6.940121	181.630819	8.732251	0	0	1	0	0	...	
9	10	39.853980	13.848321	30.664408	1.474250	0	0	0	0	0	...	
10	11	46.547581	4.872038	28.018353	1.347036	0	0	1	0	0	...	
12	13	37.728011	3.621142	0.000000	0.000000	0	0	1	0	0	...	
13	14	30.785191	4.583328	34.334443	1.650694	0	0	1	0	0	...	
14	15	49.923380	4.883225	0.000000	0.000000	0	0	0	0	0	...	
15	16	42.797890	19.107198	21.659823	1.041338	0	0	0	0	0	...	
16	17	48.621300	9.940272	0.000000	0.000000	0	0	0	0	0	...	
17	18	41.855812	2.559054	55.099831	2.649030	0	0	1	0	0	...	
18	19	51.008737	5.302773	81.595540	3.922863	0	0	1	0	0	...	
19	20	36.910410	11.226280	94.668561	4.551373	0	0	0	0	0	...	
20	21	57.903243	3.300304	108.380176	5.210585	0	0	1	0	0	...	
21	22	24.470303	3.147510	0.000000	0.000000	0	0	1	0	0	...	
22	23	49.516720	6.533500	67.740789	3.256769	0	0	0	0	0	...	
23	24	60.595509	4.465037	158.704509	7.630024	0	0	1	0	0	...	
24	25	35.804664	4.626920	0.000000	0.000000	0	0	1	0	0	...	

	EmployeeNumber	Age	LengthService	AbsentHours	AbsentRate	Accounts Receiveable	Audit	Bakery	Compensation	Customer Service	...	V
25	26	35.873388	3.101926	48.017051	2.308512	0	0	1	0	0	...	
26	27	41.935606	4.557230	42.788027	2.057117	0	0	1	0	0	...	
27	28	35.497906	5.174674	34.524272	1.659821	0	0	1	0	0	...	
28	29	40.028870	12.185467	59.883165	2.878998	0	0	0	0	0	...	
29	30	49.756585	4.313618	100.674099	4.840101	0	0	1	0	0	...	
30	31	38.111732	3.493534	9.522524	0.457814	0	0	1	0	0	...	
...	...	...	...	...	...	...	...	...	...	...	...	
8305	8306	57.153423	3.362375	69.583037	3.345338	0	0	0	0	1	...	
8306	8307	29.639450	4.685995	0.000000	0.000000	0	0	0	0	0	...	
8307	8308	45.790541	0.429435	89.085632	4.282963	0	0	0	0	1	...	
8308	8309	37.764318	3.572291	77.136720	3.708496	0	0	0	0	1	...	
8309	8310	35.027290	3.479817	0.000000	0.000000	0	0	0	0	1	...	
8310	8311	41.005533	6.827897	49.113985	2.361249	0	0	0	0	1	...	
8311	8312	47.484636	2.801183	117.252611	5.637145	0	0	0	0	0	...	
8312	8313	44.469159	4.153047	88.278983	4.244182	0	0	0	0	1	...	
8313	8314	40.136869	4.511522	21.271534	1.022670	0	0	0	0	1	...	
8314	8315	37.525723	2.111874	52.114955	2.505527	0	0	0	0	0	...	
8315	8316	43.625842	3.269938	109.118573	5.246085	0	0	0	0	1	...	
8316	8317	38.509250	3.110783	34.623534	1.664593	0	0	0	0	1	...	
8317	8318	30.040191	3.611187	0.000000	0.000000	0	0	0	0	1	...	
8318	8319	35.355472	2.062953	48.172195	2.315971	0	0	0	0	1	...	
8319	8320	45.213492	2.684577	128.078831	6.157636	0	0	0	0	1	...	
8320	8321	47.925425	4.396204	73.332561	3.525604	0	0	0	0	1	...	
8321	8322	56.641156	2.656240	114.634808	5.511289	0	0	0	0	1	...	
8322	8323	34.497413	3.172903	2.866901	0.137832	0	0	0	0	1	...	
8324	8325	31.729961	5.876275	0.000000	0.000000	0	0	0	0	1	...	

	EmployeeNumber	Age	LengthService	AbsentHours	AbsentRate	Accounts Receiveable	Audit	Bakery	Compensation	Customer Service	...	V
8325	8326	34.921911	3.329741	0.000000	0.000000	0	0	0	0	1	...	
8326	8327	50.978770	3.791685	110.906292	5.332033	0	0	0	0	1	...	
8327	8328	22.155280	6.621585	0.000000	0.000000	0	0	0	0	1	...	
8328	8329	37.615123	4.025016	40.367062	1.940724	0	0	0	0	1	...	
8329	8330	46.465588	4.838590	49.226319	2.366650	0	0	0	0	1	...	
8330	8331	46.548428	7.819204	151.285105	7.273322	0	0	0	0	0	...	
8331	8332	46.057544	4.838288	93.665111	4.503130	0	0	0	0	1	...	
8332	8333	34.455490	2.427274	0.000000	0.000000	0	0	0	0	1	...	
8333	8334	58.347160	4.009393	176.356940	8.478699	0	0	0	0	1	...	
8334	8335	43.340616	6.154837	60.321917	2.900092	0	0	0	0	0	...	
8335	8336	46.192782	5.174722	112.023389	5.385740	0	0	0	0	0	...	

8165 rows × 66 columns

In [234]:
```
y = dataset['AbsentRate']
```

In [235]:
```
def num(s):
    try:
        return int(s)
    except ValueError:
        return float(s)
```

In [236]:
```
dataset.dropna().head(15)
```

Out[236]:

	EmployeeNumber	Age	LengthService	AbsentHours	AbsentRate	Accounts Receiveable	Audit	Bakery	Compensation	Customer Service	...	Van
0	1	32.028816	6.018478	36.577306	1.758524	0	0	1	0	0	...	
1	2	40.320902	5.532445	30.165072	1.450244	0	0	1	0	0	...	
2	3	48.822047	4.389973	83.807798	4.029221	0	0	1	0	0	...	
3	4	44.599357	3.081736	70.020165	3.366354	0	0	1	0	0	...	
4	5	35.697876	3.619091	0.000000	0.000000	0	0	1	0	0	...	
5	6	48.440311	2.717692	81.830079	3.934138	0	0	1	0	0	...	
6	7	50.752730	10.157918	60.495072	2.908417	0	0	0	0	0	...	
7	8	36.216031	4.432123	30.072902	1.445813	0	0	1	0	0	...	
8	9	58.427380	6.940121	181.630819	8.732251	0	0	1	0	0	...	
9	10	39.853980	13.848321	30.664408	1.474250	0	0	0	0	0	...	
10	11	46.547581	4.872038	28.018353	1.347036	0	0	1	0	0	...	
12	13	37.728011	3.621142	0.000000	0.000000	0	0	1	0	0	...	
13	14	30.785191	4.583328	34.334443	1.650694	0	0	1	0	0	...	
14	15	49.923380	4.883225	0.000000	0.000000	0	0	0	0	0	...	
15	16	42.797890	19.107198	21.659823	1.041338	0	0	0	0	0	...	

15 rows × 66 columns

In [237]:
```
dataset.dropna(how='any')
dataset.dropna(how='all')
```

Out[237]:

	EmployeeNumber	Age	LengthService	AbsentHours	AbsentRate	Accounts Receiveable	Audit	Bakery	Compensation	Customer Service	...	\
0	1	32.028816	6.018478	36.577306	1.758524	0	0	1	0	0	...	
1	2	40.320902	5.532445	30.165072	1.450244	0	0	1	0	0	...	
2	3	48.822047	4.389973	83.807798	4.029221	0	0	1	0	0	...	
3	4	44.599357	3.081736	70.020165	3.366354	0	0	1	0	0	...	
4	5	35.697876	3.619091	0.000000	0.000000	0	0	1	0	0	...	
5	6	48.440311	2.717692	81.830079	3.934138	0	0	1	0	0	...	
6	7	50.752730	10.157918	60.495072	2.908417	0	0	0	0	0	...	
7	8	36.216031	4.432123	30.072902	1.445813	0	0	1	0	0	...	
8	9	58.427380	6.940121	181.630819	8.732251	0	0	1	0	0	...	
9	10	39.853980	13.848321	30.664408	1.474250	0	0	0	0	0	...	
10	11	46.547581	4.872038	28.018353	1.347036	0	0	1	0	0	...	
12	13	37.728011	3.621142	0.000000	0.000000	0	0	1	0	0	...	
13	14	30.785191	4.583328	34.334443	1.650694	0	0	1	0	0	...	
14	15	49.923380	4.883225	0.000000	0.000000	0	0	0	0	0	...	
15	16	42.797890	19.107198	21.659823	1.041338	0	0	0	0	0	...	
16	17	48.621300	9.940272	0.000000	0.000000	0	0	0	0	0	...	
17	18	41.855812	2.559054	55.099831	2.649030	0	0	1	0	0	...	
18	19	51.008737	5.302773	81.595540	3.922863	0	0	1	0	0	...	
19	20	36.910410	11.226280	94.668561	4.551373	0	0	0	0	0	...	
20	21	57.903243	3.300304	108.380176	5.210585	0	0	1	0	0	...	
21	22	24.470303	3.147510	0.000000	0.000000	0	0	1	0	0	...	
22	23	49.516720	6.533500	67.740789	3.256769	0	0	0	0	0	...	
23	24	60.595509	4.465037	158.704509	7.630024	0	0	1	0	0	...	
24	25	35.804664	4.626920	0.000000	0.000000	0	0	1	0	0	...	

	EmployeeNumber	Age	LengthService	AbsentHours	AbsentRate	Accounts Receiveable	Audit	Bakery	Compensation	Customer Service	...	V
25	26	35.873388	3.101926	48.017051	2.308512	0	0	1	0	0	...	
26	27	41.935606	4.557230	42.788027	2.057117	0	0	1	0	0	...	
27	28	35.497906	5.174674	34.524272	1.659821	0	0	1	0	0	...	
28	29	40.028870	12.185467	59.883165	2.878998	0	0	0	0	0	...	
29	30	49.756585	4.313618	100.674099	4.840101	0	0	1	0	0	...	
30	31	38.111732	3.493534	9.522524	0.457814	0	0	1	0	0	...	
...	...	...	...	...	...	...	...	...	...	...	...	
8305	8306	57.153423	3.362375	69.583037	3.345338	0	0	0	0	1	...	
8306	8307	29.639450	4.685995	0.000000	0.000000	0	0	0	0	0	...	
8307	8308	45.790541	0.429435	89.085632	4.282963	0	0	0	0	1	...	
8308	8309	37.764318	3.572291	77.136720	3.708496	0	0	0	0	1	...	
8309	8310	35.027290	3.479817	0.000000	0.000000	0	0	0	0	1	...	
8310	8311	41.005533	6.827897	49.113985	2.361249	0	0	0	0	1	...	
8311	8312	47.484636	2.801183	117.252611	5.637145	0	0	0	0	0	...	
8312	8313	44.469159	4.153047	88.278983	4.244182	0	0	0	0	1	...	
8313	8314	40.136869	4.511522	21.271534	1.022670	0	0	0	0	1	...	
8314	8315	37.525723	2.111874	52.114955	2.505527	0	0	0	0	0	...	
8315	8316	43.625842	3.269938	109.118573	5.246085	0	0	0	0	1	...	
8316	8317	38.509250	3.110783	34.623534	1.664593	0	0	0	0	1	...	
8317	8318	30.040191	3.611187	0.000000	0.000000	0	0	0	0	1	...	
8318	8319	35.355472	2.062953	48.172195	2.315971	0	0	0	0	1	...	
8319	8320	45.213492	2.684577	128.078831	6.157636	0	0	0	0	1	...	
8320	8321	47.925425	4.396204	73.332561	3.525604	0	0	0	0	1	...	
8321	8322	56.641156	2.656240	114.634808	5.511289	0	0	0	0	1	...	
8322	8323	34.497413	3.172903	2.866901	0.137832	0	0	0	0	1	...	
8324	8325	31.729961	5.876275	0.000000	0.000000	0	0	0	0	1	...	

	EmployeeNumber	Age	LengthService	AbsentHours	AbsentRate	Accounts Receiveable	Audit	Bakery	Compensation	Customer Service	...	V
8325	8326	34.921911	3.329741	0.000000	0.000000	0	0	0	0	1	...	
8326	8327	50.978770	3.791685	110.906292	5.332033	0	0	0	0	1	...	
8327	8328	22.155280	6.621585	0.000000	0.000000	0	0	0	0	1	...	
8328	8329	37.615123	4.025016	40.367062	1.940724	0	0	0	0	1	...	
8329	8330	46.465588	4.838590	49.226319	2.366650	0	0	0	0	1	...	
8330	8331	46.548428	7.819204	151.285105	7.273322	0	0	0	0	0	...	
8331	8332	46.057544	4.838288	93.665111	4.503130	0	0	0	0	1	...	
8332	8333	34.455490	2.427274	0.000000	0.000000	0	0	0	0	1	...	
8333	8334	58.347160	4.009393	176.356940	8.478699	0	0	0	0	1	...	
8334	8335	43.340616	6.154837	60.321917	2.900092	0	0	0	0	0	...	
8335	8336	46.192782	5.174722	112.023389	5.385740	0	0	0	0	0	...	

8165 rows × 66 columns

In [284]:
```
from sklearn.cross_validation import train_test_split

X =dataset[['Age','LengthService']]
X.values
y = dataset['AbsentRate']
y.values
```

Out[284]:
```
array([ 1.75852433,  1.45024386,  4.02922104, ...,  8.47869902,
        2.90009217,  5.38573985])
```

In [272]:
```
from sklearn.model_selection import train_test_split
X_train
```

Out[272]:
```
array([[1],
       [0],
       [0],
       ...,
       [0],
       [0],
       [0]], dtype=uint8)
```

In [296]:
```
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.3,random_state=100)
```

In [297]:
```
#--------- Linear Regression ------------------
from sklearn.linear_model import LinearRegression
model = LinearRegression()
```

In [298]:
```
model.fit(X_train,y_train)
```

Out[298]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1, normalize=False)
```

In [299]:
```
model.score(X_test,y_test)
```

Out[299]:
```
0.66934324464701422
```

In [270]:
```
from sklearn import metrics
```

In [300]:
```
#----------- Decision Tree -------------------------
from sklearn.tree import DecisionTreeRegressor


```

In [301]:
```
from sklearn import metrics
```

In [302]:
```
model1 = DecisionTreeRegressor()
model1.fit(X_train,y_train)
```

Out[302]:
```
DecisionTreeRegressor(criterion='mse', max_depth=None, max_features=None,
           max_leaf_nodes=None, min_impurity_decrease=0.0,
           min_impurity_split=None, min_samples_leaf=1,
           min_samples_split=2, min_weight_fraction_leaf=0.0,
           presort=False, random_state=None, splitter='best')
```

In [303]:
```
model1.score(X_test,y_test)
```

Out[303]:
```
0.40069827187233226
```