Pandas Library

Dhafer Malouche

Contents

1 Description from the Pandas documentation:

- `Pandas` is a data analysis library providing fast, flexible, and expressive data structures designed to work with relational or table-like data (SQL table or Excel spreadsheet). It is a fundamental high-level building block for doing practical, real world data analysis in Python.
- `Pandas` is well suited for:
  - Tabular data with heterogeneously-typed columns, as in an SQL table or Excel spreadsheetOrdered and unordered (not necessarily fixed-frequency) time series data.
  - Arbitrary matrix data (homogeneously typed or heterogeneous) with row and column labels
  - Any other form of observational / statistical data sets.
- The data used with `Pandas` actually doesn't need be labeled at all to be placed into a `Pandas` data structure.
- The two primary data structures of `Pandas`, Series (1-dimensional) and DataFrame (2-dimensional), handle the vast majority of typical use cases in finance, statistics, social science, and many areas of engineering.
- `Pandas` is built **on top of** `NumPy` and is intended to integrate well within a scientific computing environment with many other 3rd party libraries.

Here are just a few of the things that `Pandas` does well:

- Easy handling of missing data (represented as NaN) in floating point as well as non-floating point data

- Size mutability: columns can be inserted and deleted from DataFrame and higher dimensional objects
- Automatic and explicit data alignment: objects can be explicitly aligned to a set of labels, or the user can simply ignore the labels and let Series, DataFrame, etc. automatically align the data for you in computations
- Powerful, flexible group by functionality to perform split-apply-combine operations on data sets, for both aggregating and transforming data
- Make it easy to convert ragged, differently-indexed data in other Python and NumPy data structures into DataFrame objects
- Intelligent label-based slicing, fancy indexing, and subsetting of large data sets
- Intuitive merging and joining data sets
- Flexible reshaping and pivoting of data sets
- Hierarchical labeling of axes (possible to have multiple labels per tick)
- Robust IO tools for loading data from flat files (CSV and delimited), Excel files, databases, and saving / loading data from the ultrafast HDF5 format
- Time series-specific functionality: date range generation and frequency conversion, moving window statistics, moving window linear regressions, date shifting and lagging, etc.

2 Series and DataFrames

We should first import `Pandas` into `Python` after installing it from the CMD promt: pip install pandas

[1]:
```
import pandas as pd
```

3 The Panda Series

The Series data structure in `Pandas` is a one-dimensional labeled array. + Data in the array can be of any type (integers, strings, floating point numbers, Python objects, etc.). + Data within the array is **homogeneous** + `Pandas` Series objects always have an index: this gives them both ndarray-like and dict-like properties.

Creating a `Panda` Series:

- Creation from a list
- Creation from a dictionary
- Creation from a ndarray
- From an external source file (.csv,.xls…)

From a list

[2]:
```
temperature = [34, 56, 15, -9, -121, -5, 39]
days = ['Mon','Tue','Wed','Thu','Fri','Sat','Sun']

# create series
series_from_list = pd.Series(temperature, index=days)
series_from_list
```

[2]:
```
Mon     34
Tue     56
Wed     15
Thu     -9
Fri   -121
Sat     -5
Sun     39
dtype: int64
```

The series should contains homogeneous types

[3]:
```
temperature = [34, 56, 'a', -9, -121, -5, 39]
days = ['Mon','Tue','Wed','Thu','Fri','Sat','Sun']
```

We create series

[4]:
```
series_from_list = pd.Series(temperature, index=days)
series_from_list
```

[4]:
```
Mon      34
Tue      56
Wed       a
Thu      -9
Fri    -121
Sat      -5
Sun      39
dtype: object
```

from a dictionary

[5]:
```
my_dict = {'Mon': 33, 'Tue': 19, 'Wed': 15, 'Thu': 89, 'Fri': 11, 'Sat': -5,␣
↪'Sun': 9}
my_dict
```

[5]:
```
{'Mon': 33, 'Tue': 19, 'Wed': 15, 'Thu': 89, 'Fri': 11, 'Sat': -5, 'Sun': 9}
```

[6]:
```
series_from_dict = pd.Series(my_dict)
series_from_dict
```

[6]:
```
Mon    33
Tue    19
Wed    15
Thu    89
Fri    11
Sat    -5
Sun     9
dtype: int64
```

From a numpy array

```
[7]: import numpy as np
```

I'm using `linspace` to create an array with spaced numbers over a specified interval: 15 numbers between 0 and 10

```
[8]: my_array = np.linspace(0,10,15)
     my_array
```

```
[8]: array([ 0.        ,  0.71428571,  1.42857143,  2.14285714,  2.85714286,
            3.57142857,  4.28571429,  5.        ,  5.71428571,  6.42857143,
            7.14285714,  7.85714286,  8.57142857,  9.28571429, 10.        ])
```

```
[9]: len(my_array)
```

```
[9]: 15
```

The `array` **must** be with dimension 1

```
[10]: series_from_ndarray = pd.Series(my_array)
      series_from_ndarray
```

```
[10]: 0      0.000000
      1      0.714286
      2      1.428571
      3      2.142857
      4      2.857143
      5      3.571429
      6      4.285714
      7      5.000000
      8      5.714286
      9      6.428571
      10     7.142857
      11     7.857143
      12     8.571429
      13     9.285714
      14    10.000000
      dtype: float64
```

4 Pandas DataFrames

DataFrame is a 2-dimensional labeled data structure with **columns** of potentially different types. You can think of it like a spreadsheet or SQL table, or a dict of Series objects. You can create a DataFrame from: + Dict of 1D ndarrays, lists, dicts, or Series + 2-D numpy.ndarray + From text, CSV, Excel files or databases + Many other ways

Reading the data.

Sample data: HR Employee Attrition and Performance You can get it from here and add it to your working directory:

https://www.ibm.com/communities/analytics/watson-analytics-blog/hr-employee-attrition/

Importing the xlsx file by considering the variable EmployeeNumber as an Index variable.

If Kaggle use this after uploading the xlsx into Kaggle

[11]:
```
## data = pd.read_excel(io="../input/WA_Fn-UseC_-HR-Employee-Attrition.xlsx",␣
↪sheetname=0, index_col='EmployeeNumber')
```

[11]:
```
data = pd.read_excel(io="data1.xlsx",  index_col='EmployeeNumber')
```

Types of the variables

[12]:
```
data.dtypes
```

[12]:
```
Age                          int64
Attrition                   object
BusinessTravel              object
DailyRate                    int64
Department                  object
DistanceFromHome             int64
Education                    int64
EducationField              object
EmployeeCount                int64
EnvironmentSatisfaction      int64
Gender                      object
HourlyRate                   int64
JobInvolvement               int64
JobLevel                     int64
JobRole                     object
JobSatisfaction              int64
MaritalStatus               object
MonthlyIncome                int64
MonthlyRate                  int64
NumCompaniesWorked           int64
Over18                      object
OverTime                    object
PercentSalaryHike            int64
PerformanceRating            int64
RelationshipSatisfaction     int64
StandardHours                int64
StockOptionLevel             int64
TotalWorkingYears            int64
TrainingTimesLastYear        int64
WorkLifeBalance              int64
YearsAtCompany               int64
YearsInCurrentRole           int64
YearsSinceLastPromotion      int64
```

```
YearsWithCurrManager        int64
dtype: object
```

A preview of the data (the first 3 rows)

[39]:
```
data.head(3)
```

[39]:
```
                Age Attrition     BusinessTravel  DailyRate  \
EmployeeNumber
1                41       Yes      Travel_Rarely       1102
2                49        No  Travel_Frequently        279
4                37       Yes      Travel_Rarely       1373

                            Department  DistanceFromHome  Education  \
EmployeeNumber
1                                Sales                 1          2
2               Research & Development                 8          1
4               Research & Development                 2          2

               EducationField  EmployeeCount  EnvironmentSatisfaction  ...  \
EmployeeNumber                                                         ...
1               Life Sciences              1                        2  ...
2               Life Sciences              1                        3  ...
4                       Other              1                        4  ...

               RelationshipSatisfaction  StandardHours  StockOptionLevel  \
EmployeeNumber
1                                     1             80                 0
2                                     4             80                 1
4                                     2             80                 0

                TotalWorkingYears TrainingTimesLastYear  WorkLifeBalance  \
EmployeeNumber
1                               8                     0                1
2                              10                     3                3
4                               7                     3                3

               YearsAtCompany  YearsInCurrentRole  YearsSinceLastPromotion  \
EmployeeNumber
1                           6                   4                        0
2                          10                   7                        1
4                           0                   0                        0

                YearsWithCurrManager
EmployeeNumber
1                                  5
2                                  7
4                                  0
```

```
[3 rows x 34 columns]
```

Name of the columns in the imported data.

```
[40]: data.columns
```

```
[40]: Index(['Age', 'Attrition', 'BusinessTravel', 'DailyRate', 'Department',
             'DistanceFromHome', 'Education', 'EducationField', 'EmployeeCount',
             'EnvironmentSatisfaction', 'Gender', 'HourlyRate', 'JobInvolvement',
             'JobLevel', 'JobRole', 'JobSatisfaction', 'MaritalStatus',
             'MonthlyIncome', 'MonthlyRate', 'NumCompaniesWorked', 'Over18',
             'OverTime', 'PercentSalaryHike', 'PerformanceRating',
             'RelationshipSatisfaction', 'StandardHours', 'StockOptionLevel',
             'TotalWorkingYears', 'TrainingTimesLastYear', 'WorkLifeBalance',
             'YearsAtCompany', 'YearsInCurrentRole', 'YearsSinceLastPromotion',
             'YearsWithCurrManager'],
            dtype='object')
```

The preview of the variable `Attrition`

```
[44]: data['Attrition'].head()
```

```
[44]: EmployeeNumber
      1     Yes
      2      No
      4     Yes
      5      No
      7      No
      Name: Attrition, dtype: object
```

5 Data Manipulation

Selecting some variables from the original data and displaying a preview.

```
[45]: data[['Age', 'Gender','YearsAtCompany']].head()
```

```
[45]:                 Age  Gender  YearsAtCompany
      EmployeeNumber
      1                41  Female               6
      2                49    Male              10
      4                37    Male               0
      5                33  Female               8
      7                27    Male               2
```

Creating a new variables. Transforming the Age in years to the Age in months.

```
[46]: data['AgeInMonths'] = 12*data['Age']
      data['AgeInMonths'].head()
```

```
[46]: EmployeeNumber
1    492
2    588
4    444
5    396
7    324
Name: AgeInMonths, dtype: int64
```

Deleting the new created variable

```
[47]: del data['AgeInMonths']
```

```
[48]: data.columns
```

```
[48]: Index(['Age', 'Attrition', 'BusinessTravel', 'DailyRate', 'Department',
       'DistanceFromHome', 'Education', 'EducationField', 'EmployeeCount',
       'EnvironmentSatisfaction', 'Gender', 'HourlyRate', 'JobInvolvement',
       'JobLevel', 'JobRole', 'JobSatisfaction', 'MaritalStatus',
       'MonthlyIncome', 'MonthlyRate', 'NumCompaniesWorked', 'Over18',
       'OverTime', 'PercentSalaryHike', 'PerformanceRating',
       'RelationshipSatisfaction', 'StandardHours', 'StockOptionLevel',
       'TotalWorkingYears', 'TrainingTimesLastYear', 'WorkLifeBalance',
       'YearsAtCompany', 'YearsInCurrentRole', 'YearsSinceLastPromotion',
       'YearsWithCurrManager'],
      dtype='object')
```

Extracting the some observations from on specific variable

```
[50]: data['BusinessTravel'][10:15]
```

```
[50]: EmployeeNumber
14    Travel_Rarely
15    Travel_Rarely
16    Travel_Rarely
18    Travel_Rarely
19    Travel_Rarely
Name: BusinessTravel, dtype: object
```

Extracting some rows from the whole dataframe

```
[52]: data[10:15]
```

```
[52]:                 Age Attrition BusinessTravel  DailyRate  \
EmployeeNumber
14               35        No  Travel_Rarely        809
15               29        No  Travel_Rarely        153
16               31        No  Travel_Rarely        670
18               34        No  Travel_Rarely       1346
19               28       Yes  Travel_Rarely        103
```

```
                          Department  DistanceFromHome  Education  \
EmployeeNumber
14              Research & Development                16          3
15              Research & Development                15          2
16              Research & Development                26          1
18              Research & Development                19          2
19              Research & Development                24          3

               EducationField  EmployeeCount  EnvironmentSatisfaction  ...  \
EmployeeNumber                                                         ...
14                    Medical              1                        1  ...
15              Life Sciences              1                        4  ...
16              Life Sciences              1                        1  ...
18                    Medical              1                        2  ...
19              Life Sciences              1                        3  ...

               RelationshipSatisfaction  StandardHours  StockOptionLevel  \
EmployeeNumber
14                                    3             80                 1
15                                    4             80                 0
16                                    4             80                 1
18                                    3             80                 1
19                                    2             80                 0

                TotalWorkingYears TrainingTimesLastYear  WorkLifeBalance  \
EmployeeNumber
14                              6                     5                3
15                             10                     3                3
16                              5                     1                2
18                              3                     2                3
19                              6                     4                3

               YearsAtCompany  YearsInCurrentRole  YearsSinceLastPromotion  \
EmployeeNumber
14                          5                   4                        0
15                          9                   5                        0
16                          5                   2                        4
18                          2                   2                        1
19                          4                   2                        0

                YearsWithCurrManager
EmployeeNumber
14                                 3
15                                 8
16                                 3
18                                 2
```

```
19                                  3

[5 rows x 34 columns]
```

Selecting specific rows from the index variable `EmployeeNumbers`

[57]:
```python
selected_EmployeeNumbers = [15, 94, 337, 1120]
```

[58]:
```python
data['YearsAtCompany']
```

[58]:
```
EmployeeNumber
1        6
2       10
4        0
5        8
7        2
        ..
2061     5
2062     7
2064     6
2065     9
2068     4
Name: YearsAtCompany, Length: 1470, dtype: int64
```

[59]:
```python
data['YearsAtCompany'].loc[selected_EmployeeNumbers]
```

[59]:
```
EmployeeNumber
15      9
94      5
337     2
1120    7
Name: YearsAtCompany, dtype: int64
```

[60]:
```python
data.loc[selected_EmployeeNumbers]
```

[60]:
```
                Age Attrition     BusinessTravel  DailyRate  \
EmployeeNumber
15               29        No      Travel_Rarely        153
94               29        No      Travel_Rarely       1328
337              31        No  Travel_Frequently       1327
1120             29        No      Travel_Rarely       1107

                            Department  DistanceFromHome  Education  \
EmployeeNumber
15              Research & Development                15          2
94              Research & Development                 2          3
337             Research & Development                 3          4
1120            Research & Development                28          4
```

```
               EducationField  EmployeeCount  EnvironmentSatisfaction  ...  \
EmployeeNumber                                                          ...
15               Life Sciences              1                        4  ...
94               Life Sciences              1                        3  ...
337                    Medical              1                        2  ...
1120             Life Sciences              1                        3  ...

               RelationshipSatisfaction  StandardHours  StockOptionLevel  \
EmployeeNumber
15                                    4             80                 0
94                                    4             80                 1
337                                   1             80                 1
1120                                  1             80                 1

                TotalWorkingYears TrainingTimesLastYear  WorkLifeBalance  \
EmployeeNumber
15                             10                     3                3
94                              6                     3                3
337                             9                     3                3
1120                           11                     1                3

               YearsAtCompany  YearsInCurrentRole  YearsSinceLastPromotion  \
EmployeeNumber
15                          9                   5                        0
94                          5                   4                        0
337                         2                   2                        2
1120                        7                   5                        1

                YearsWithCurrManager
EmployeeNumber
15                                 8
94                                 4
337                                2
1120                               7

[4 rows x 34 columns]
```

What's the `YearsAtCompany` of the row with `EmployeeNumber` equal to 94?

[62]:
```
data.loc[94,'YearsAtCompany']
```

[62]:
```
5
```

Frequency of the variable `Department`

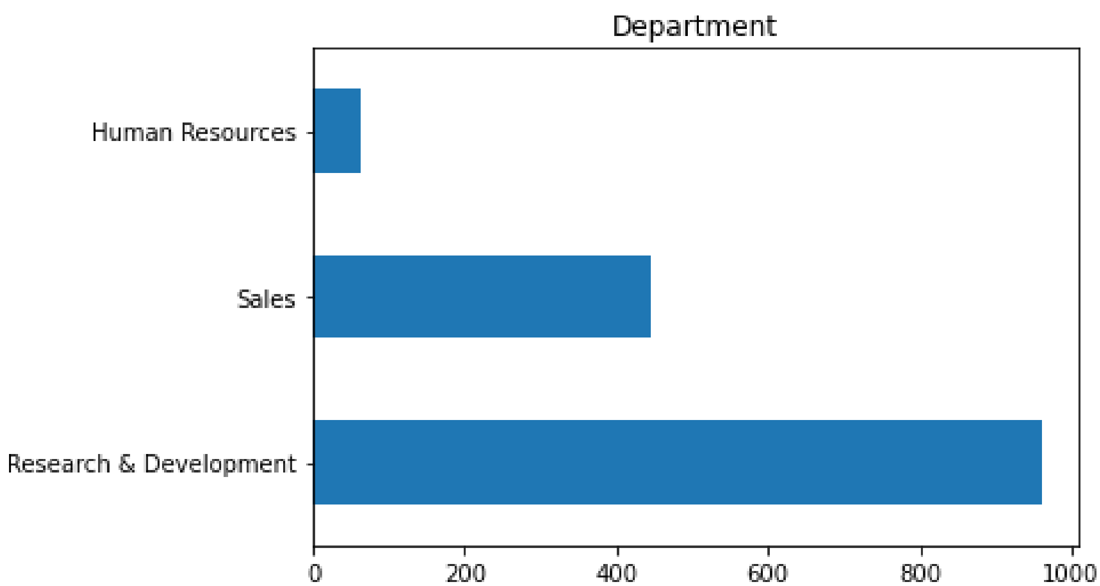
[64]:
```
data['Department'].value_counts()
```

[64]:
```
Research & Development    961
Sales                     446
Human Resources            63
Name: Department, dtype: int64
```

A barplot of the variable `Department`

[66]:
```
data['Department'].value_counts().plot(kind='barh', title='Department')
```

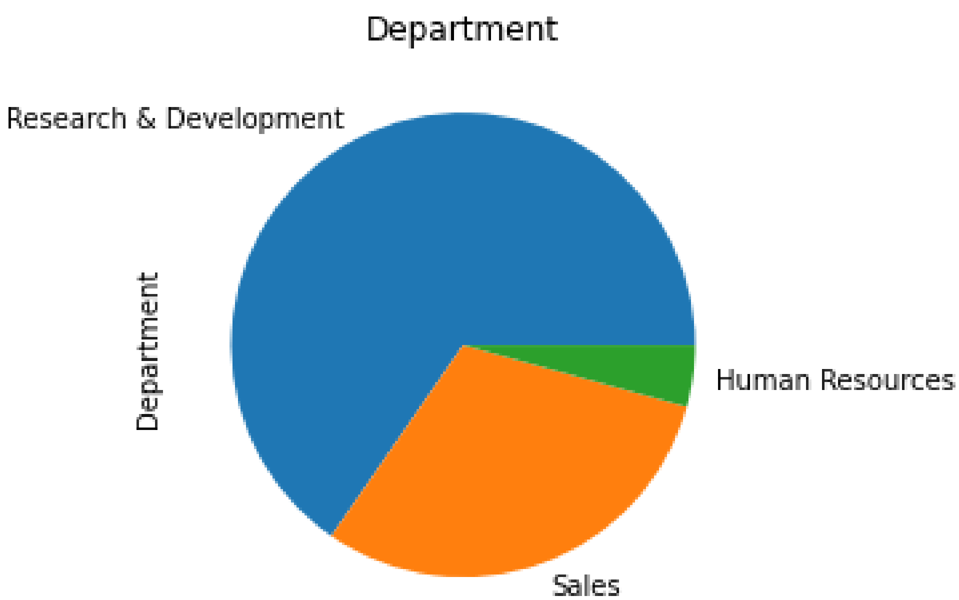
[66]:
```
<AxesSubplot:title={'center':'Department'}>
```



Creating a pie chart

[67]:
```
data['Department'].value_counts().plot(kind='pie', title='Department')
```

[67]:
```
<AxesSubplot:title={'center':'Department'}, ylabel='Department'>
```



Frequency of the variable `Attrition`

[70]:
```
data['Attrition'].value_counts()
```

[70]:
```
No     1233
Yes     237
Name: Attrition, dtype: int64
```

Frequency in percentage

[72]:
```
data['Attrition'].value_counts(normalize=True)
```

[72]:
```
No     0.838776
Yes    0.161224
Name: Attrition, dtype: float64
```

Compute the average of the variable `HourlyRate`

[73]:
```
data['HourlyRate'].mean()
```

[73]:
```
65.89115646258503
```

What's the overall statisfaction of the Employees?

[75]:
```
data['JobSatisfaction'].head()
```

[75]:
```
EmployeeNumber
1    4
```

```
2    2
4    3
5    3
7    2
Name: JobSatisfaction, dtype: int64
```

Let us change the levels of the variable satisfaction by creating first a disctionary

[77]:
```
JobSatisfaction_cat = {
    1: 'Low',
    2: 'Medium',
    3: 'High',
    4: 'Very High'
}
```

[78]:
```
data['JobSatisfaction'] = data['JobSatisfaction'].map(JobSatisfaction_cat)
data['JobSatisfaction'].head()
```

[78]:
```
EmployeeNumber
1    Very High
2       Medium
4         High
5         High
7       Medium
Name: JobSatisfaction, dtype: object
```

[79]:
```
data['JobSatisfaction'].value_counts()
```

[79]:
```
Very High    459
High         442
Low          289
Medium       280
Name: JobSatisfaction, dtype: int64
```

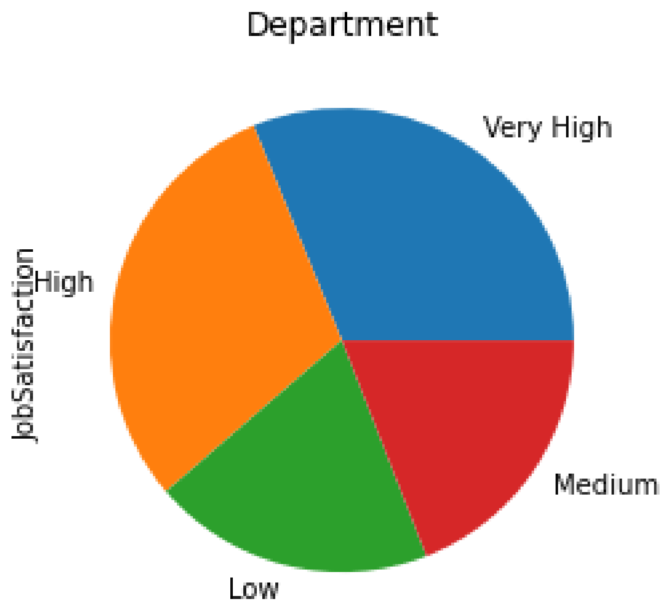
Computing percentages

[81]:
```
100*data['JobSatisfaction'].value_counts(normalize=True)
```

[81]:
```
Very High    31.224490
High         30.068027
Low          19.659864
Medium       19.047619
Name: JobSatisfaction, dtype: float64
```

[82]:
```
data['JobSatisfaction'].value_counts(normalize=True).plot(kind='pie',␣
 ↪title='Department')
```

[82]:
```
<AxesSubplot:title={'center':'Department'}, ylabel='JobSatisfaction'>
```




```
[88]: from pandas.api.types import CategoricalDtype
cats=['Low', 'Medium', 'High', 'Very High']
cat_type = CategoricalDtype(categories=cats, ordered=True)
data['JobSatisfaction'] = data['JobSatisfaction'].astype(cat_type)
```

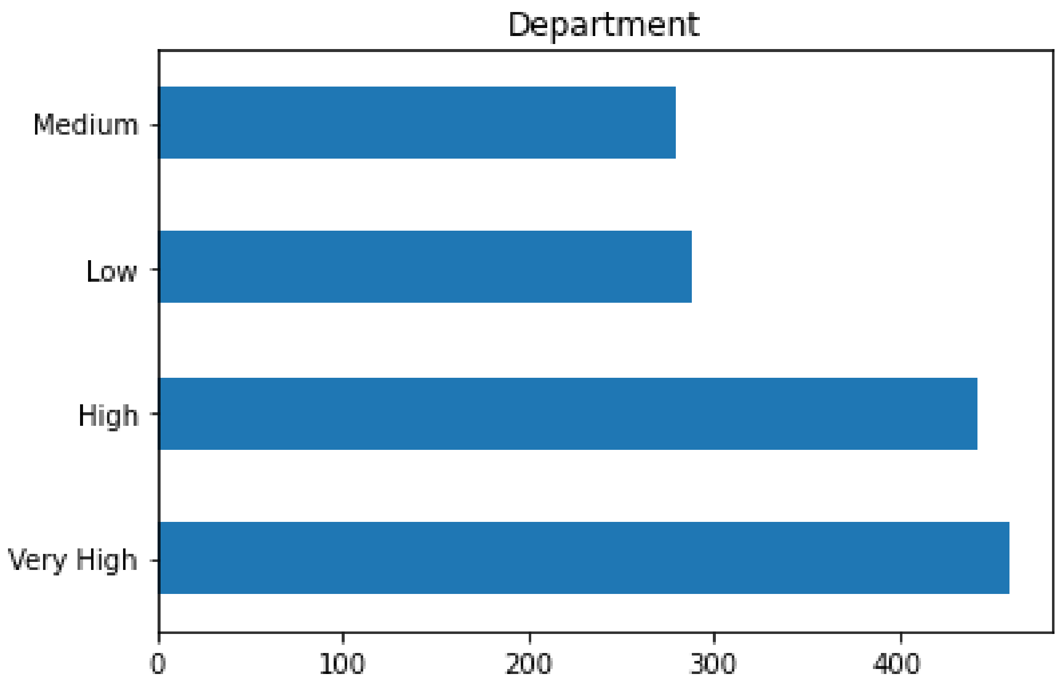
```
[89]: data['JobSatisfaction'].head()
```

```
[89]: EmployeeNumber
1    Very High
2       Medium
4         High
5         High
7       Medium
Name: JobSatisfaction, dtype: category
Categories (4, object): ['Low' < 'Medium' < 'High' < 'Very High']
```

Sorting by frequencies (it's the default option)

```
[91]: data['JobSatisfaction'].value_counts().plot(kind='barh', title='Department')
```

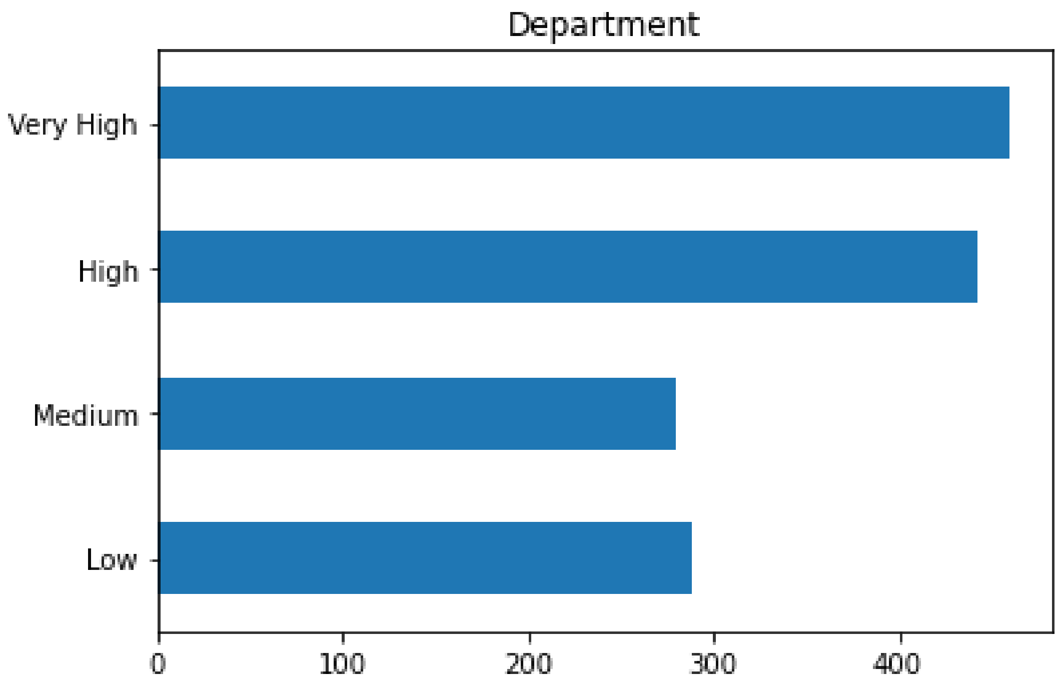
```
[91]: <AxesSubplot:title={'center':'Department'}>
```



Canceling the default sorting option and the bars will be sorted according to the categories

[92]:
```
data['JobSatisfaction'].value_counts(sort=False).plot(kind='barh',␣
↪title='Department')
```

[92]: `<AxesSubplot:title={'center':'Department'}>`



[93]:
```
data['JobSatisfaction'] == 'Low'
```

[93]:
```
EmployeeNumber
1       False
2       False
4       False
5       False
7       False
        ...
2061    False
2062     True
2064    False
2065    False
2068    False
Name: JobSatisfaction, Length: 1470, dtype: bool
```

[94]:
```
data.loc[data['JobSatisfaction'] == 'Low'].index
```

[94]:
```
Int64Index([  10,   20,   27,   31,   33,   38,   51,   52,   54,   68,
            ...
            1975, 1980, 1998, 2021, 2023, 2038, 2054, 2055, 2057, 2062],
           dtype='int64', name='EmployeeNumber', length=289)
```

[95]:
```
data['JobInvolvement'].head()
```

[95]:
```
EmployeeNumber
1    3
2    2
4    2
5    3
7    3
Name: JobInvolvement, dtype: int64
```

Selecting observation of a specific interest: Those with either "Low" or "Very High" Job statisfaction

[107]:
```
subset_of_interest = data.loc[(data['JobSatisfaction'] == "Low") |␣
↪(data['JobSatisfaction'] == "Very High")]
subset_of_interest.shape
```

[107]:
```
(748, 34)
```

[108]:
```
subset_of_interest.head()
```

[108]:
```
                Age Attrition     BusinessTravel  DailyRate  \
EmployeeNumber
1                41       Yes      Travel_Rarely       1102
8                32        No  Travel_Frequently       1005
10               59        No      Travel_Rarely       1324
18               34        No      Travel_Rarely       1346
20               29        No      Travel_Rarely       1389

                            Department  DistanceFromHome  Education  \
EmployeeNumber
1                                Sales                 1          2
8               Research & Development                 2          2
10              Research & Development                 3          3
18              Research & Development                19          2
20              Research & Development                21          4

               EducationField  EmployeeCount  EnvironmentSatisfaction  ...  \
EmployeeNumber                                                         ...
1               Life Sciences              1                        2  ...
8               Life Sciences              1                        4  ...
10                    Medical              1                        3  ...
18                    Medical              1                        2  ...
20              Life Sciences              1                        2  ...

               RelationshipSatisfaction  StandardHours  StockOptionLevel  \
EmployeeNumber
1                                     1             80                 0
8                                     3             80                 0
10                                    1             80                 3
```

```
18                                    3           80                  1
20                                    3           80                  1

                TotalWorkingYears TrainingTimesLastYear WorkLifeBalance  \
EmployeeNumber
1                               8                     0               1
8                               8                     2               2
10                             12                     3               2
18                              3                     2               3
20                             10                     1               3

               YearsAtCompany  YearsInCurrentRole  YearsSinceLastPromotion  \
EmployeeNumber
1                           6                   4                        0
8                           7                   7                        3
10                          1                   0                        0
18                          2                   2                        1
20                         10                   9                        8

                YearsWithCurrManager
EmployeeNumber
1                                  5
8                                  6
10                                 0
18                                 2
20                                 8

[5 rows x 34 columns]
```

[109]:
```
subset_of_interest['JobSatisfaction'].value_counts()
```

[109]:
```
Very High    459
Low          289
Medium         0
High           0
Name: JobSatisfaction, dtype: int64
```

Let's then remove the categories or levels that we won't use

[110]:
```
subset_of_interest['JobSatisfaction'].cat.remove_unused_categories(inplace=True)
```

```
C:\ProgramData\Anaconda3\lib\site-
packages\pandas\core\arrays\categorical.py:2631: FutureWarning: The `inplace`
parameter in pandas.Categorical.remove_unused_categories is deprecated and will
be removed in a future version.
  res = method(*args, **kwargs)
```

The categories 'Medium' and 'High' won't be displayed

[112]:
```
subset_of_interest['JobSatisfaction'].value_counts()
```

[112]:
```
Very High    459
Low          289
Name: JobSatisfaction, dtype: int64
```

[113]:
```
grouped = subset_of_interest.groupby('JobSatisfaction')
```

[116]:
```
grouped.head()
```

[116]:
```
                Age Attrition     BusinessTravel  DailyRate  \
EmployeeNumber
1                41       Yes      Travel_Rarely       1102
8                32        No  Travel_Frequently       1005
10               59        No      Travel_Rarely       1324
18               34        No      Travel_Rarely       1346
20               29        No      Travel_Rarely       1389
22               22        No         Non-Travel       1123
23               53        No      Travel_Rarely       1219
27               36       Yes      Travel_Rarely       1218
31               34       Yes      Travel_Rarely        699
33               32       Yes  Travel_Frequently       1125

                            Department  DistanceFromHome  Education  \
EmployeeNumber
1                                Sales                 1          2
8               Research & Development                 2          2
10              Research & Development                 3          3
18              Research & Development                19          2
20              Research & Development                21          4
22              Research & Development                16          2
23                               Sales                 2          4
27                               Sales                 9          4
31              Research & Development                 6          1
33              Research & Development                16          1

               EducationField  EmployeeCount  EnvironmentSatisfaction  ...  \
EmployeeNumber                                                         ...
1               Life Sciences              1                        2  ...
8               Life Sciences              1                        4  ...
10                    Medical              1                        3  ...
18                    Medical              1                        2  ...
20              Life Sciences              1                        2  ...
22                    Medical              1                        4  ...
23              Life Sciences              1                        1  ...
27              Life Sciences              1                        3  ...
31                    Medical              1                        2  ...
```

```
33               Life Sciences             1                         2  ...

               RelationshipSatisfaction  StandardHours  StockOptionLevel  \
EmployeeNumber
1                                     1             80                 0
8                                     3             80                 0
10                                    1             80                 3
18                                    3             80                 1
20                                    3             80                 1
22                                    2             80                 2
23                                    3             80                 0
27                                    2             80                 0
31                                    3             80                 0
33                                    2             80                 0

                TotalWorkingYears TrainingTimesLastYear WorkLifeBalance  \
EmployeeNumber
1                               8                     0               1
8                               8                     2               2
10                             12                     3               2
18                              3                     2               3
20                             10                     1               3
22                              1                     2               2
23                             31                     3               3
27                             10                     4               3
31                              8                     2               3
33                             10                     5               3

               YearsAtCompany  YearsInCurrentRole  YearsSinceLastPromotion  \
EmployeeNumber
1                           6                   4                        0
8                           7                   7                        3
10                          1                   0                        0
18                          2                   2                        1
20                         10                   9                        8
22                          1                   0                        0
23                         25                   8                        3
27                          5                   3                        0
31                          4                   2                        1
33                         10                   2                        6

                YearsWithCurrManager
EmployeeNumber
1                                  5
8                                  6
10                                 0
18                                 2
```

```
20                              8
22                              0
23                              7
27                              3
31                              3
33                              7

[10 rows x 34 columns]
```

[114]:
```
grouped.groups
```

[114]:
```
{'Low': [10, 20, 27, 31, 33, 38, 51, 52, 54, 68, 70, 74, 75, 81, 86, 88, 100,
101, 113, 124, 133, 134, 145, 153, 170, 190, 197, 199, 200, 235, 239, 240, 241,
244, 250, 267, 274, 282, 288, 297, 299, 303, 328, 334, 339, 340, 347, 351, 362,
369, 374, 382, 390, 396, 412, 424, 425, 429, 451, 454, 474, 486, 510, 515, 517,
522, 524, 530, 532, 534, 536, 538, 549, 567, 573, 590, 605, 615, 625, 630, 648,
650, 662, 664, 667, 682, 684, 702, 705, 725, 728, 729, 732, 733, 742, 758, 764,
771, 775, 776, ...], 'Very High': [1, 8, 18, 22, 23, 24, 30, 36, 39, 40, 42, 45,
49, 53, 57, 62, 63, 72, 73, 76, 78, 79, 97, 98, 104, 106, 107, 112, 116, 117,
118, 120, 137, 139, 140, 143, 144, 148, 152, 154, 155, 158, 165, 169, 174, 179,
184, 192, 195, 198, 207, 215, 217, 221, 223, 228, 230, 242, 243, 245, 246, 262,
264, 273, 275, 281, 283, 286, 287, 291, 298, 302, 306, 309, 311, 312, 315, 316,
319, 323, 325, 327, 333, 335, 336, 338, 346, 349, 353, 361, 367, 372, 373, 377,
378, 380, 388, 389, 391, 393, ...]}
```

The Low statisfaction group

[115]:
```
grouped.get_group('Low').head()
```

[115]:
```
                Age Attrition     BusinessTravel  DailyRate  \
EmployeeNumber
10               59        No      Travel_Rarely       1324
20               29        No      Travel_Rarely       1389
27               36       Yes      Travel_Rarely       1218
31               34       Yes      Travel_Rarely        699
33               32       Yes  Travel_Frequently       1125

                            Department  DistanceFromHome  Education  \
EmployeeNumber
10              Research & Development                 3          3
20              Research & Development                21          4
27                               Sales                 9          4
31              Research & Development                 6          1
33              Research & Development                16          1

               EducationField  EmployeeCount  EnvironmentSatisfaction  ...  \
EmployeeNumber                                                         ...
10                    Medical              1                        3  ...
```

```
20             Life Sciences             1                        2  ...
27             Life Sciences             1                        3  ...
31                   Medical             1                        2  ...
33             Life Sciences             1                        2  ...

              RelationshipSatisfaction  StandardHours  StockOptionLevel  \
EmployeeNumber
10                                   1             80                 3
20                                   3             80                 1
27                                   2             80                 0
31                                   3             80                 0
33                                   2             80                 0

               TotalWorkingYears TrainingTimesLastYear WorkLifeBalance  \
EmployeeNumber
10                            12                     3               2
20                            10                     1               3
27                            10                     4               3
31                             8                     2               3
33                            10                     5               3

              YearsAtCompany  YearsInCurrentRole  YearsSinceLastPromotion  \
EmployeeNumber
10                         1                   0                        0
20                        10                   9                        8
27                         5                   3                        0
31                         4                   2                        1
33                        10                   2                        6

               YearsWithCurrManager
EmployeeNumber
10                                0
20                                8
27                                3
31                                3
33                                7

[5 rows x 34 columns]
```

and the Very High satisfaction group

[104]:
```python
grouped.get_group('Very High').head()
```

[104]:
```
               Age Attrition     BusinessTravel  DailyRate  \
EmployeeNumber
1               41       Yes      Travel_Rarely       1102
8               32        No  Travel_Frequently       1005
18              34        No      Travel_Rarely       1346
```

```
22                22         No         Non-Travel       1123
23                53         No      Travel_Rarely       1219

                            Department  DistanceFromHome  Education  \
EmployeeNumber
1                                Sales                 1          2
8               Research & Development                 2          2
18              Research & Development                19          2
22              Research & Development                16          2
23                               Sales                 2          4

               EducationField  EmployeeCount  EnvironmentSatisfaction  ...  \
EmployeeNumber                                                         ...
1               Life Sciences              1                        2  ...
8               Life Sciences              1                        4  ...
18                    Medical              1                        2  ...
22                    Medical              1                        4  ...
23              Life Sciences              1                        1  ...

               RelationshipSatisfaction  StandardHours  StockOptionLevel  \
EmployeeNumber
1                                     1             80                 0
8                                     3             80                 0
18                                    3             80                 1
22                                    2             80                 2
23                                    3             80                 0

                TotalWorkingYears TrainingTimesLastYear WorkLifeBalance  \
EmployeeNumber
1                               8                     0               1
8                               8                     2               2
18                              3                     2               3
22                              1                     2               2
23                             31                     3               3

               YearsAtCompany  YearsInCurrentRole  YearsSinceLastPromotion  \
EmployeeNumber
1                           6                   4                        0
8                           7                   7                        3
18                          2                   2                        1
22                          1                   0                        0
23                         25                   8                        3

                YearsWithCurrManager
EmployeeNumber
1                                  5
8                                  6
```

```
18                                 2
22                                 0
23                                 7

[5 rows x 34 columns]
```

The average of the Age of each group

[120]:
```
grouped[['Age','JobSatisfaction']].head()
```

[120]:
```
                Age JobSatisfaction
EmployeeNumber
1                41       Very High
8                32       Very High
10               59             Low
18               34       Very High
20               29             Low
22               22       Very High
23               53       Very High
27               36             Low
31               34             Low
33               32             Low
```

[121]:
```
grouped['Age'].mean()
```

[121]:
```
JobSatisfaction
Low          36.916955
Very High    36.795207
Name: Age, dtype: float64
```

[122]:
```
grouped['Age'].describe()
```

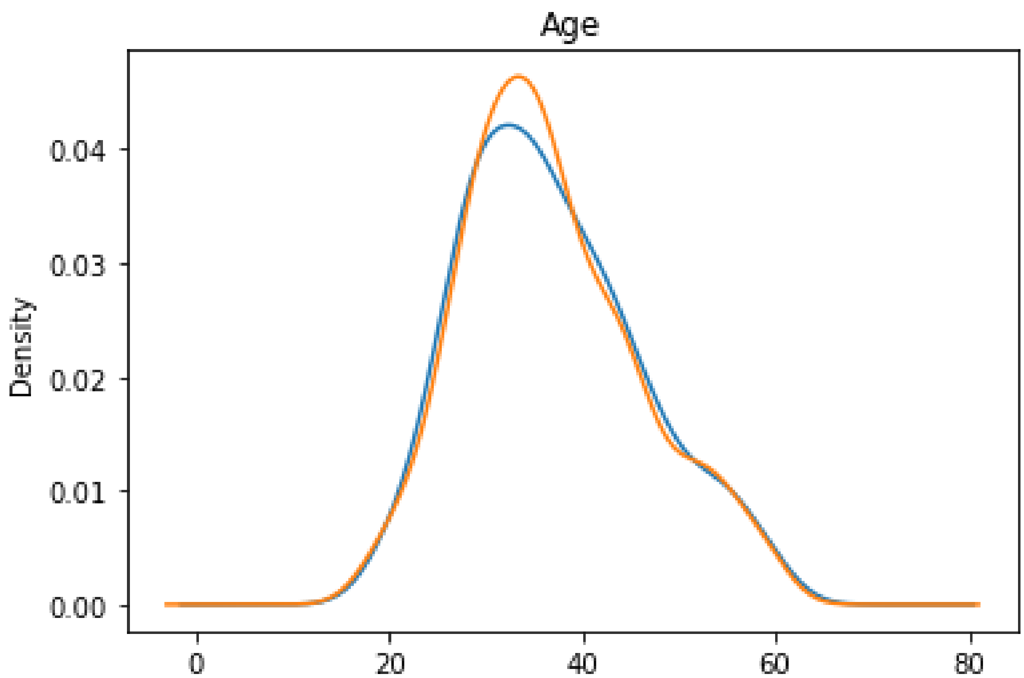
[122]:
```
                 count       mean       std   min   25%   50%   75%   max
JobSatisfaction
Low              289.0  36.916955  9.245496  19.0  30.0  36.0  42.0  60.0
Very High        459.0  36.795207  9.125609  18.0  30.0  35.0  43.0  60.0
```

[ ]:
```
grouped['Age'].describe().unstack()
```

Comparing densities

[124]:
```
grouped['Age'].plot(kind='density', title='Age')
```

[124]:
```
JobSatisfaction
Low          AxesSubplot(0.125,0.125;0.775x0.755)
Very High    AxesSubplot(0.125,0.125;0.775x0.755)
Name: Age, dtype: object
```



By Department

[125]:
```
grouped['Department'].value_counts().unstack()
```

[125]:
```
Department       Human Resources  Research & Development  Sales
JobSatisfaction
Low                           11                     192     86
Very High                     17                     295    147
```

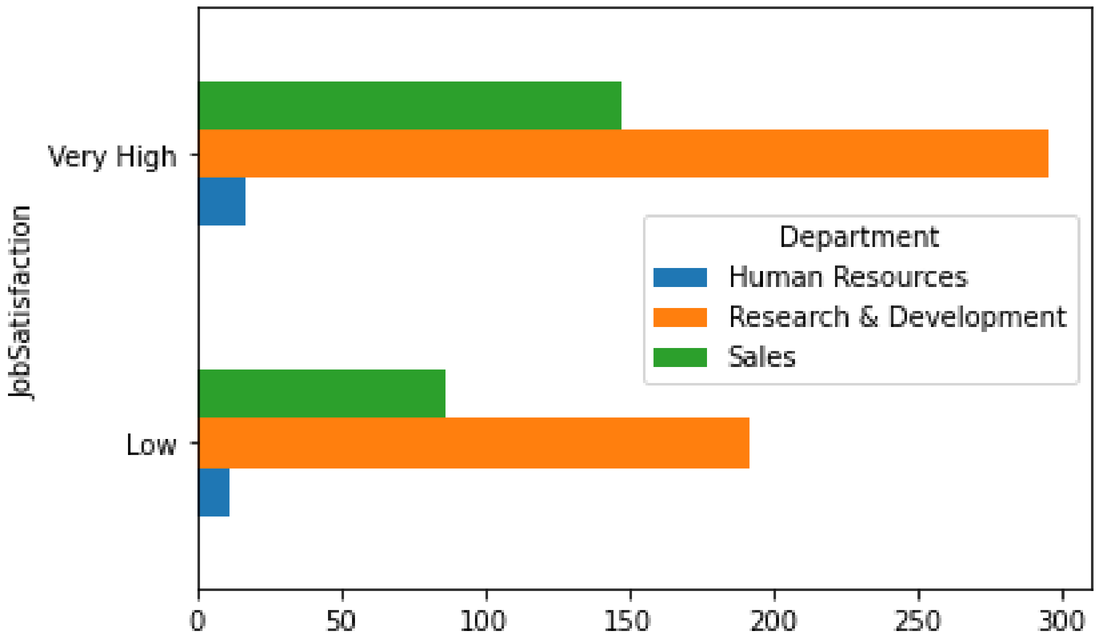
We can normalize it

[126]:
```
grouped['Department'].value_counts(normalize=True).unstack()
```

[126]:
```
Department       Human Resources  Research & Development     Sales
JobSatisfaction
Low                     0.038062                0.664360  0.297578
Very High               0.037037                0.642702  0.320261
```

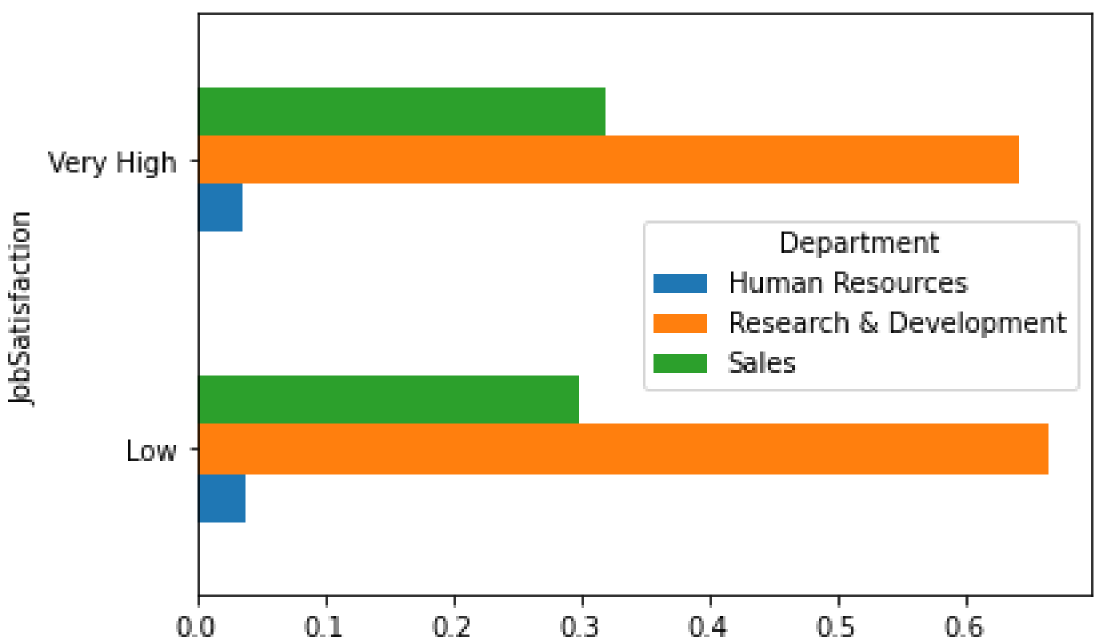
[127]:
```
grouped['Department'].value_counts().unstack().plot(kind="barh")
```

[127]:
```
<AxesSubplot:ylabel='JobSatisfaction'>
```



[128]:
```
grouped['Department'].value_counts(normalize=True).unstack().plot(kind="barh")
```

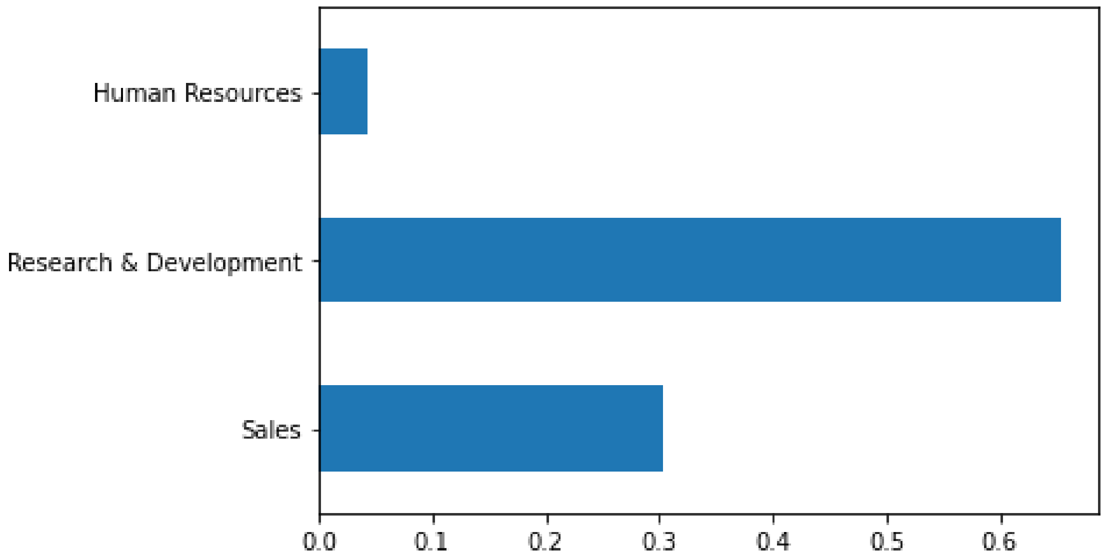
[128]: <AxesSubplot:ylabel='JobSatisfaction'>



We can compare it with the whole sample

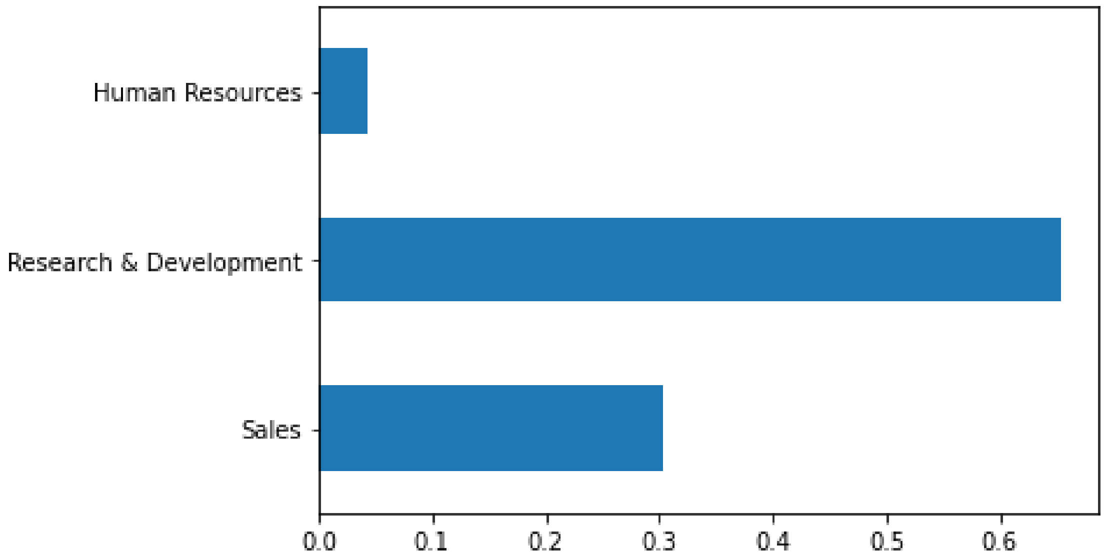
[129]:
```
data['Department'].value_counts(normalize=True,sort=False).plot(kind="barh")
```

[129]: <AxesSubplot:>



[132]:
```
data['Department'].value_counts(normalize=True,sort=False).plot(kind="barh")
```

[132]: <AxesSubplot:>



[133]:
```
grouped['DistanceFromHome'].describe().unstack()
```

[133]:

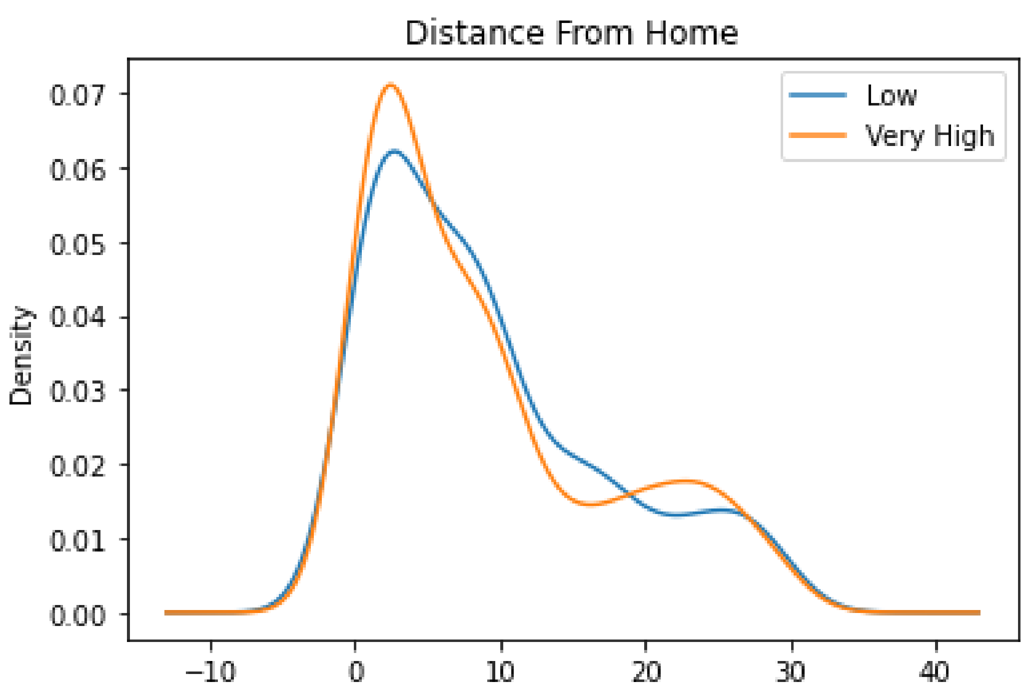
```
         JobSatisfaction
count    Low                289.000000
         Very High          459.000000
mean     Low                  9.190311
         Very High            9.030501
std      Low                  8.045127
         Very High            8.257004
min      Low                  1.000000
         Very High            1.000000
25%      Low                  2.000000
         Very High            2.000000
50%      Low                  7.000000
         Very High            7.000000
75%      Low                 14.000000
         Very High           14.000000
max      Low                 29.000000
         Very High           29.000000
dtype: float64
```

[134]:

```
grouped['DistanceFromHome'].plot(kind='density', title='Distance From␣
 ↪Home',legend=True)
```

[134]:

```
JobSatisfaction
Low          AxesSubplot(0.125,0.125;0.775x0.755)
Very High    AxesSubplot(0.125,0.125;0.775x0.755)
Name: DistanceFromHome, dtype: object
```

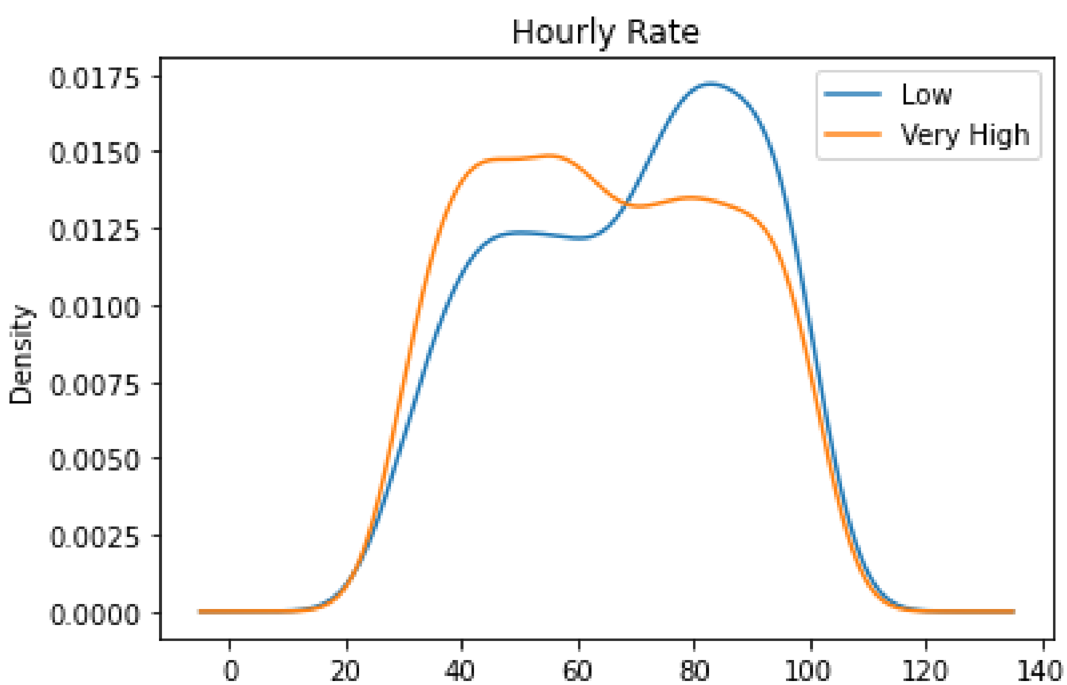


[135]:
```
grouped['HourlyRate'].describe()
```

[135]:
```
                 count       mean        std   min   25%   50%   75%    max
JobSatisfaction
Low              289.0  68.636678  20.439515  30.0  52.0  72.0  86.0  100.0
Very High        459.0  64.681917  20.647571  30.0  47.0  64.0  82.5  100.0
```

[136]:
```
grouped['HourlyRate'].plot(kind='density', title='Hourly Rate',legend=True)
```

[136]:
```
JobSatisfaction
Low          AxesSubplot(0.125,0.125;0.775x0.755)
Very High    AxesSubplot(0.125,0.125;0.775x0.755)
Name: HourlyRate, dtype: object
```



[137]:
```
grouped['MonthlyIncome'].describe()
```

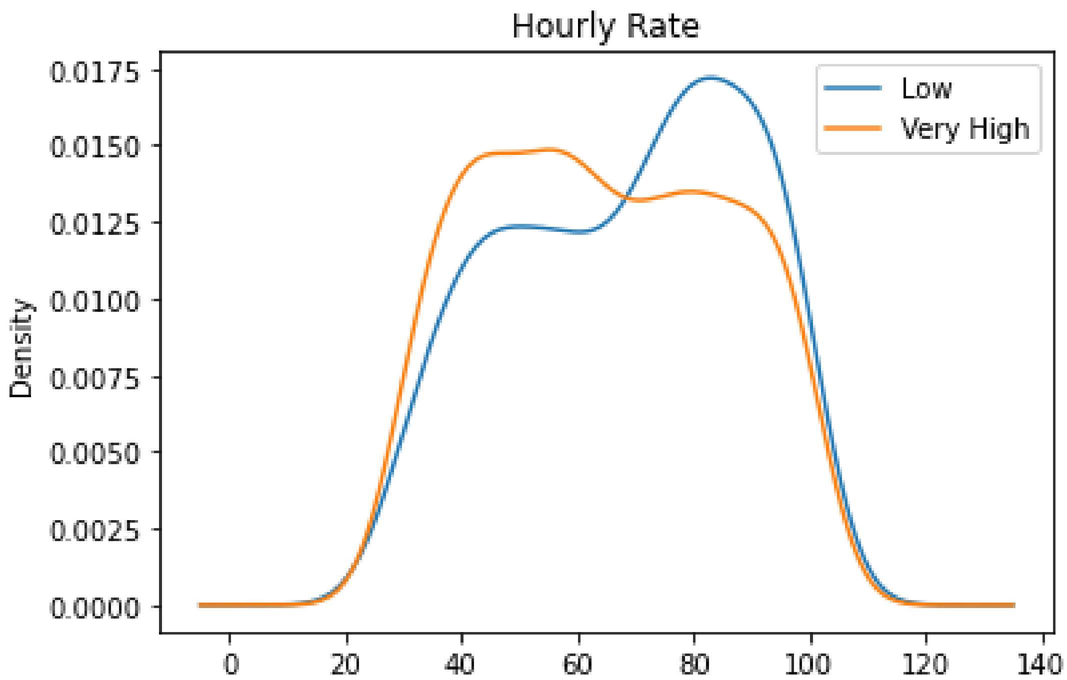
[137]:
```
                 count         mean          std     min     25%     50%  \
JobSatisfaction
Low              289.0  6561.570934  4645.170134  1091.0  3072.0  4968.0
Very High        459.0  6472.732026  4573.906428  1051.0  2927.5  5126.0

                    75%      max
JobSatisfaction
```

```
Low             8564.0  19943.0
Very High       7908.0  19845.0
```

[138]:
```
grouped['HourlyRate'].plot(kind='density', title='Hourly Rate',legend=True)
```

[138]:
```
JobSatisfaction
Low          AxesSubplot(0.125,0.125;0.775x0.755)
Very High    AxesSubplot(0.125,0.125;0.775x0.755)
Name: HourlyRate, dtype: object
```



[13]:
```
!pip install numpy
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: numpy in
c:\users\dhafe\appdata\roaming\python\python310\site-packages (1.22.1)
```

[ ]: