Project Simulation

May 27, 2021

[8]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

[9]:
```
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)
```

[10]:
```
dataset = pd.read_excel('Jupyter test.xlsx')
dataset.shape
```

[10]:
```
(301, 2)
```

[11]:
```
dataset.head()
```

[11]:
```
   Price   Distance
0   37.9   84.87882
1   42.2  306.59470
2   47.3  561.98450
3   54.8  561.98450
4   43.1  390.56840
```

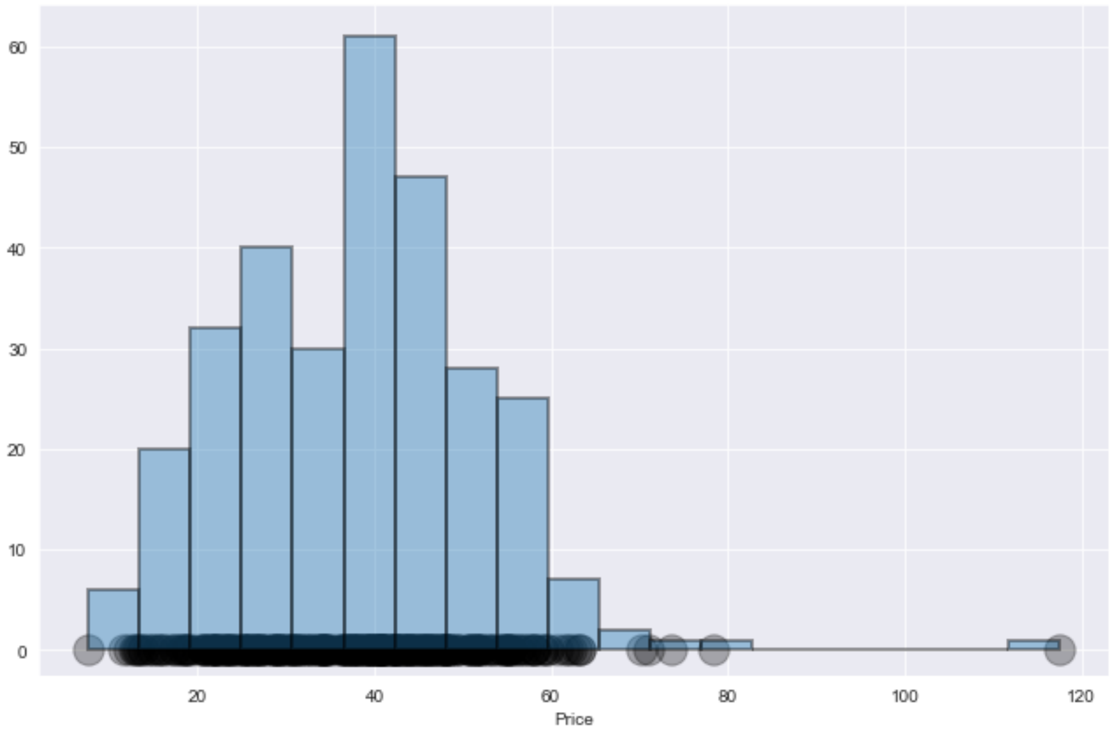
Air passenger transport

[12]:
```
Pr = dataset.Price
Dt = dataset.Distance
```

[13]:
```
sns.set_style('darkgrid')
```

0.1 Estimation par histogramme

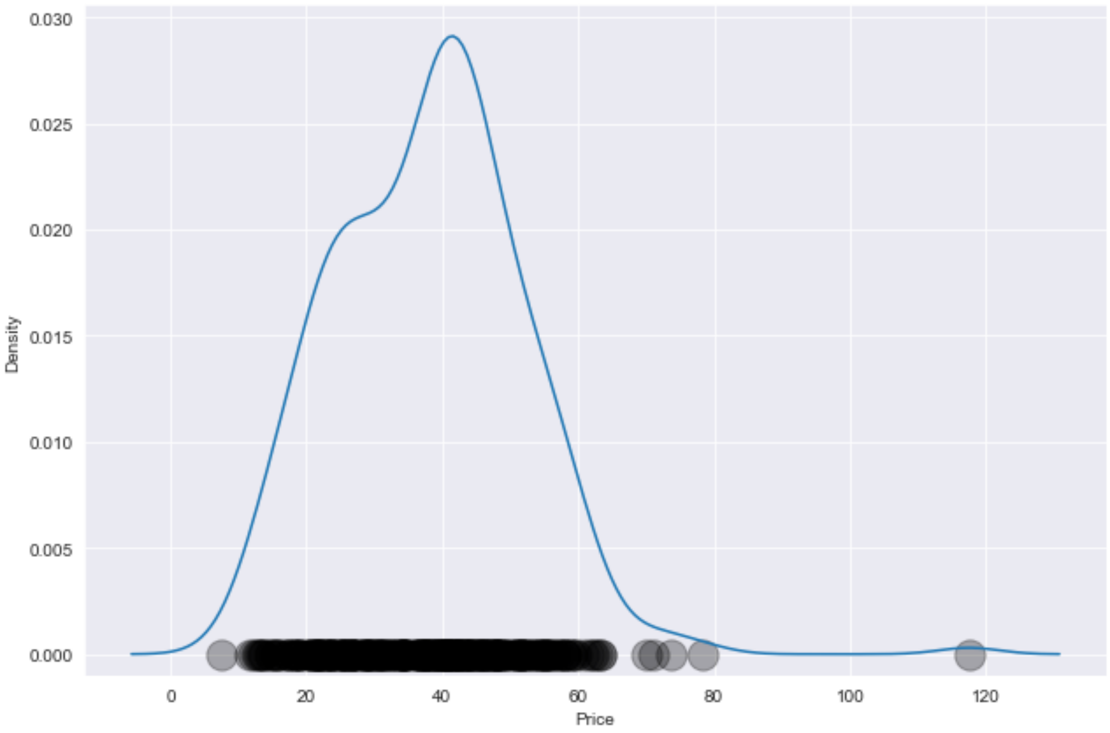
[31]:
```
plt.figure(figsize=(9,6))

plt.scatter(Pr, np.zeros_like(Pr), s=300, marker='o', alpha = 0.3, c='k')
sns.distplot(Pr, hist_kws=dict(edgecolor="k", linewidth=2), kde=False)

plt.ylim(-2.5,)
plt.tight_layout();
```



0.2 Estimation par noyau

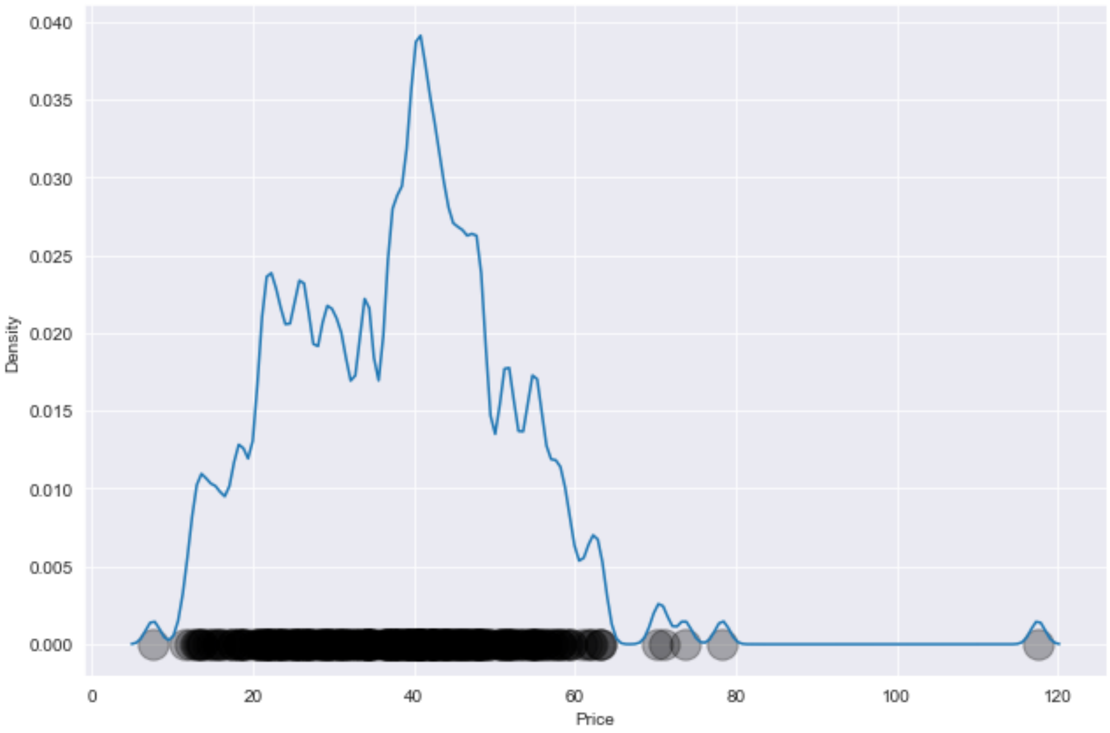
0.2.1 le noyau est gaussien par défaut

[22]:
```
plt.figure(figsize=(9,6))

sns.kdeplot(Pr)
plt.scatter(Pr, np.zeros_like(Pr), s=300, marker='o', alpha = 0.3, c='k')

plt.ylim(-1e-3,)
plt.tight_layout();
```



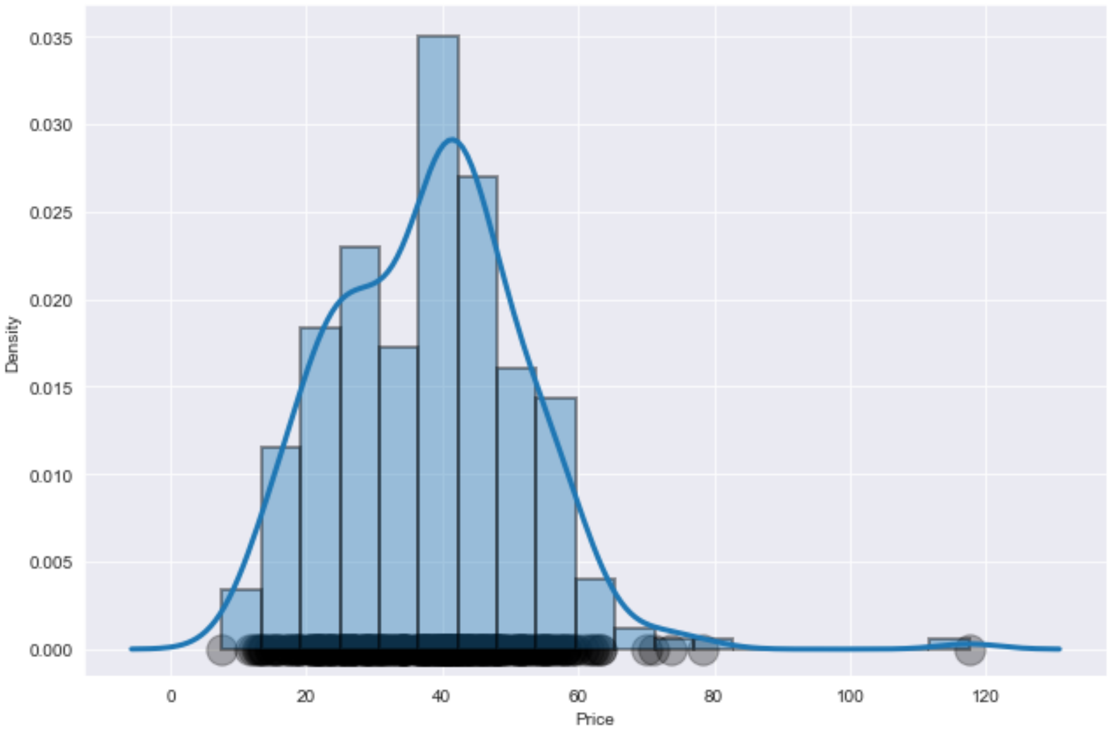
0.3 Changement de h (bandwidth)

[39]:
```
plt.figure(figsize=(9,6))

sns.kdeplot(Pr, bw_adjust=.2);
plt.scatter(Pr, np.zeros_like(Pr), s=300, marker='o', alpha = 0.3, c='k')

plt.ylim(-0.002,)
plt.tight_layout();
```



0.4 Estimation par histogramme et par noyau

[43]:
```
plt.figure(figsize=(9,6))

sns.distplot(Pr, kde_kws={"lw": 3}, hist_kws=dict(edgecolor="k", linewidth=2))
plt.scatter(Pr, np.zeros_like(Pr), s=300, marker='o', alpha = 0.3, c='k')

plt.ylim(-0.0015,)
plt.tight_layout();
```


0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
Density
0
20
40
60
80
100
120
Price

Optimized bandwidth

May 27, 2021

[1]:
```python
import numpy as np
import scipy as sp
from scipy.stats import gaussian_kde
from scipy.stats import norm
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

[2]:
```python
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)
```

[3]:
```python
import pandas as pd
dataset = pd.read_excel('Jupyter test.xlsx')
dataset.shape
```

[3]:
```
(301, 2)
```

[4]:
```python
dataset.head()
```

[4]:
```
   Price   Distance
0   37.9   84.87882
1   42.2  306.59470
2   47.3  561.98450
3   54.8  561.98450
4   43.1  390.56840
```

[5]:
```python
Pr = dataset.Price
```

[6]:
```python
min(Pr)
```

[6]:
```
7.6
```

[7]:
```python
max(Pr)
```

[7]:
```
117.5
```

[8]:
```python
sns.set_style('darkgrid')
```

[9]:

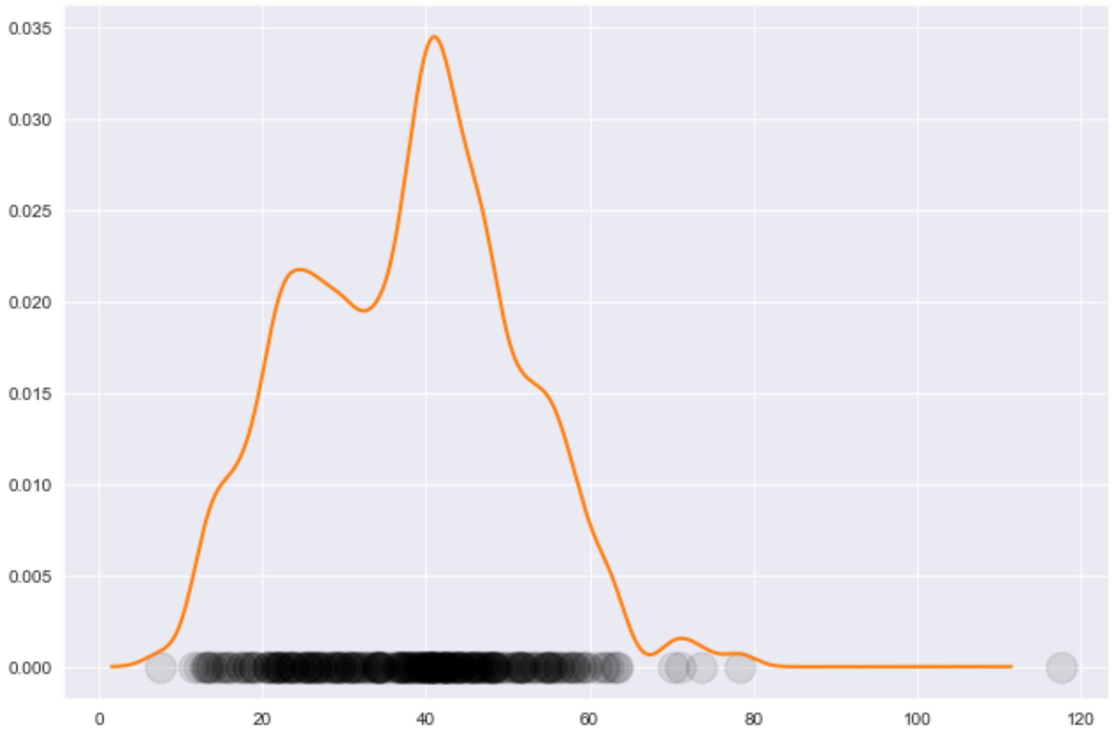
```
def computed_kde(data, width=2, gridsize=1000, normalized=True, bounds=None):

    if bounds:
        xmin, xmax = bounds
    else:
        xmin = min(data) -3 * width
        xmax = max(data) -3 * width

    x = np.linspace(xmin, xmax, gridsize)

    kde = np.zeros(gridsize)
    for val in data:
        kde += norm.pdf(x, loc=val, scale=width)
    kde /= len(data)

    return x, kde
```

[10]:

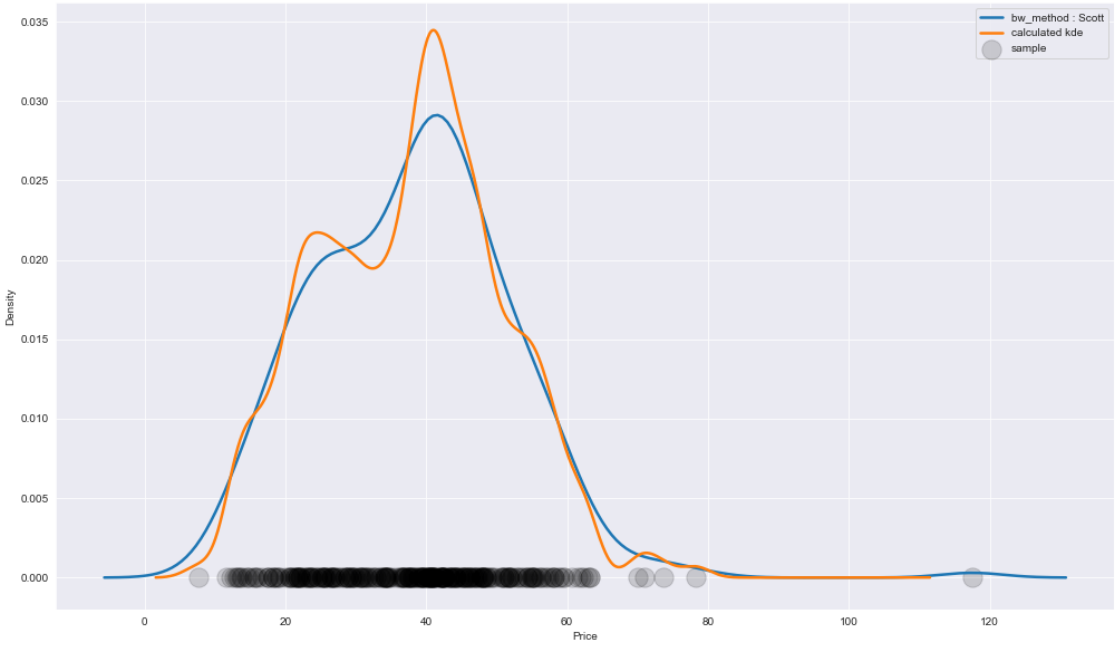
```
x, kde = computed_kde(Pr, gridsize=1000)
```

[13]:

```
plt.figure(figsize=(9,6))

plt.plot(x, kde, lw = 2, color='C1', label="calculated kde")
plt.scatter(Pr, np.zeros_like(Pr), s=300, marker='o', alpha=.1, c='k',␣
 ↪label="sample")

plt.ylim()
plt.tight_layout()
```



[12]:
```
plt.figure(figsize=(17,10))

sns.kdeplot(Pr, lw = 2.5, bw_method="scott", label="bw_method : Scott")
plt.plot(x, kde, lw = 2.5, label="calculated kde")
plt.scatter(Pr, np.zeros_like(Pr), s=300, marker='o', alpha=.15, c='k',␣
 ↪label="sample")

plt.ylim(-.002)
plt.legend();
```


bw_method : Scott
calculated kde
sample
Density
0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
0
20
40
60
80
100
120
Price