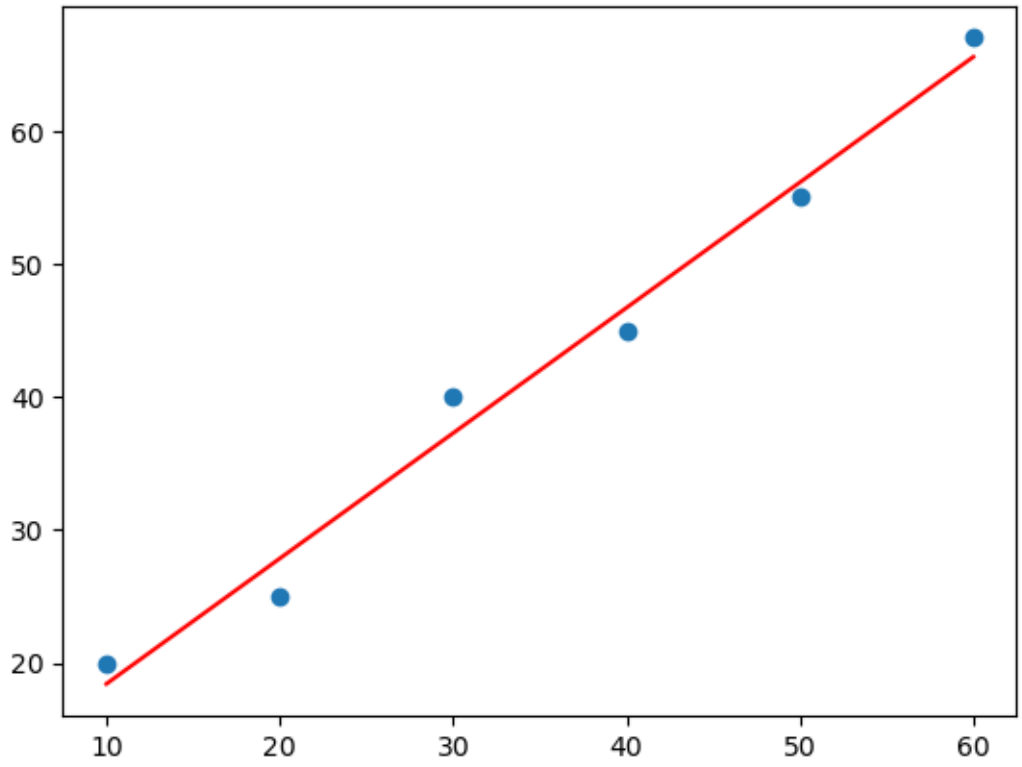
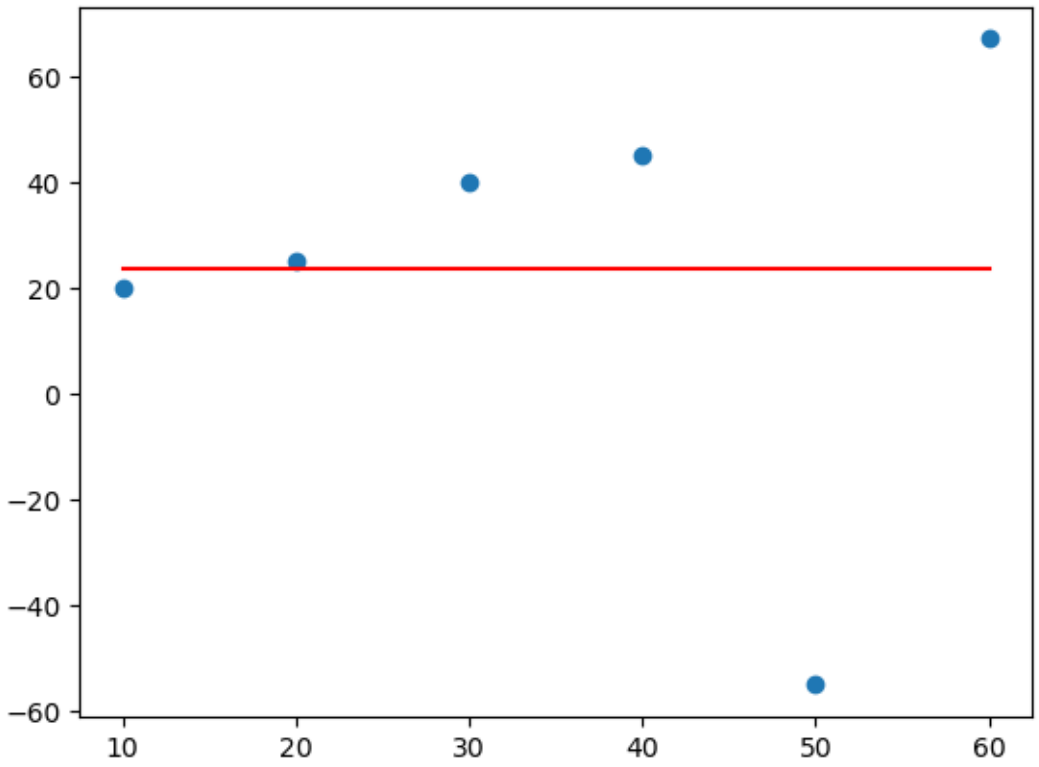
Lesson-4 Dected and handle ourlier

June 14, 2024

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
# create dataframe
data = [[10, 20], [20, 25], [30, 40], [40, 45], [50, 55], [60, 67]]
# convert to dataframee
data = pd.DataFrame(data, columns = ['Input', 'Output'])
# find the slop and y intercept
m, b = np.polyfit(data['Input'], data['Output'], 1)
# plot scatter and line chart
plt.scatter(data['Input'], data['Output'])
plt.plot(data['Input'], m * data['Input'] + b, c='r')
plt.show()
```



[2]:
```
# create dataframe with an outlier
data = [[10, 20], [20, 25], [30, 40], [40, 45], [50, -55], [60, 67]]
# convert to dataframee
data = pd.DataFrame(data, columns = ['Input', 'Output'])
# find the slop and y intercept
m, b = np.polyfit(data['Input'], data['Output'], 1)
# plot scatter and line chart
plt.scatter(data['Input'], data['Output'])
plt.plot(data['Input'], m * data['Input'] + b, c='r')
plt.show()
```

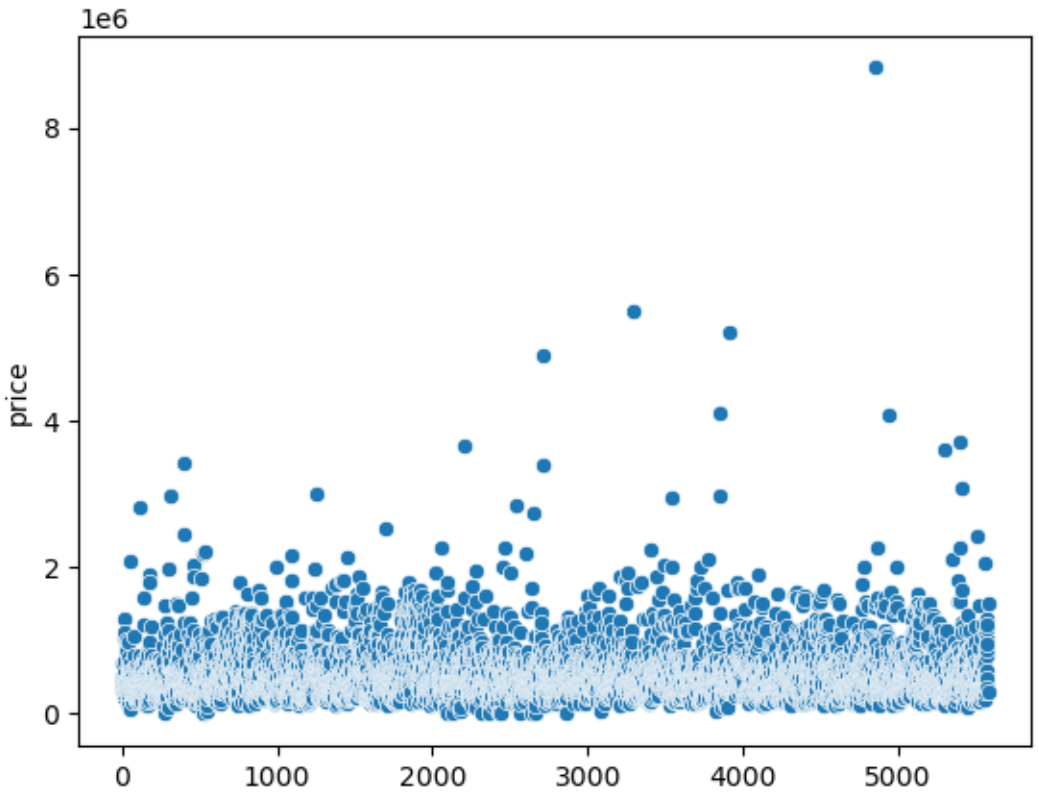


[ ]:
```
```

[3]:
```
import pandas as pd
data = pd.read_csv("/home/uca/Documents/preply/Preply/Andrew/Lesson-4 and 5/
 ↪house.csv")
```

[4]:
```
# import seaborn module
import seaborn as sns
sns.scatterplot(data, x=data.index, y=data.price)
```

[4]: <AxesSubplot:ylabel='price'>



[5]:
```
whiskers
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[5], line 1
----> 1 whiskers

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

[ ]:

[6]:
```
import numpy as np
data_mean = np.mean(data['price'])
data_std = np.std(data['price'])
```

[7]:
```
cut_off = data_std * 3
lower = data_mean - cut_off
upper = data_mean + cut_off
```

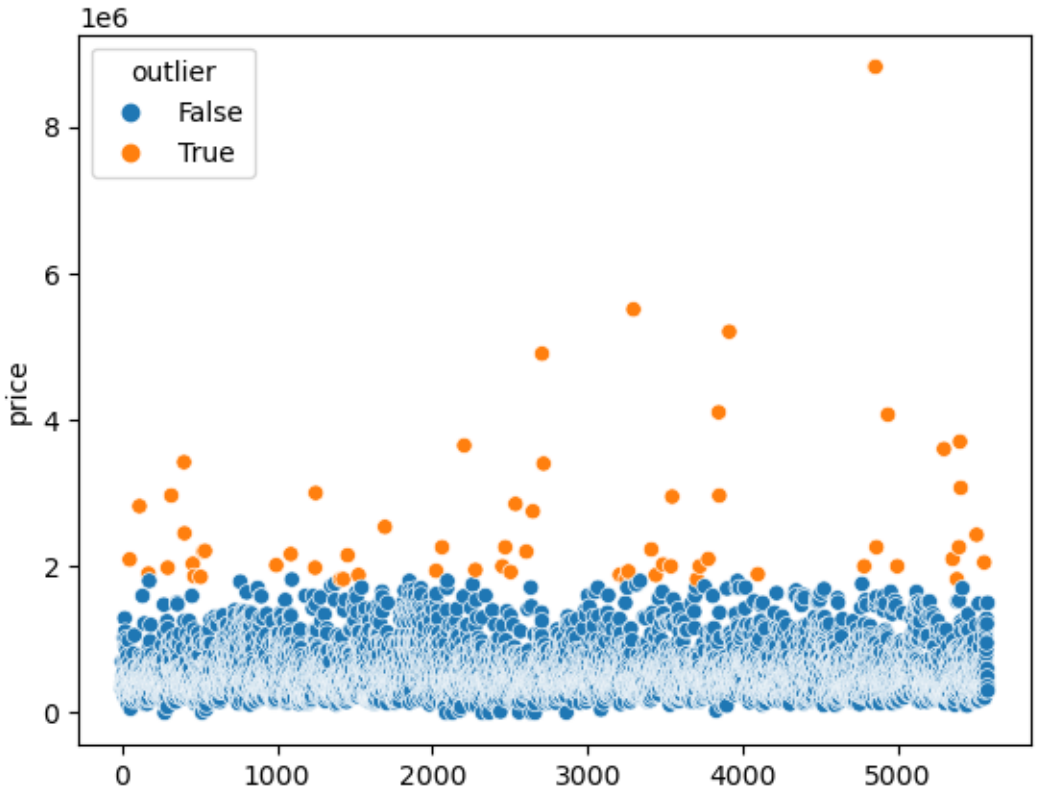
[8]:
```
outlier = []
# check each data point
for i in data['price']:
    if i > upper:
        outlier.append(i)
```

[10]:
```
len(outlier)
```

[10]: 63

[13]:
```
data['outlier'] = data['price'].isin(outlier)
sns.scatterplot(data= data, x=data.index, y='price', hue='outlier')
```

[13]: <AxesSubplot:ylabel='price'>



[14]:
```
# dropping the outliers
data.drop(data.price[data.price < lower].index, inplace=True)
data.drop(data.price[data.price > upper].index, inplace=True)
```

[ ]: