

3-Plotting

April 11, 2020

1 Plotting

```
[50]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

data = pd.read_csv(r"customer-churn-model/Customer Churn Model.txt")
data.head()
```

```
[50]: State Account Length Area Code Phone Int'l Plan VMail Plan \
0 KS 128 415 382-4657 no yes
1 OH 107 415 371-7191 no yes
2 NJ 137 415 358-1921 no no
3 OH 84 408 375-9999 yes no
4 OK 75 415 330-6626 yes no

VMail Message Day Mins Day Calls Day Charge ... Eve Calls Eve Charge \
0 25 265.1 110 45.07 ... 99 16.78
1 26 161.6 123 27.47 ... 103 16.62
2 0 243.4 114 41.38 ... 110 10.30
3 0 299.4 71 50.90 ... 88 5.26
4 0 166.7 113 28.34 ... 122 12.61

Night Mins Night Calls Night Charge Intl Mins Intl Calls Intl Charge \
0 244.7 91 11.01 10.0 3 2.70
1 254.4 103 11.45 13.7 3 3.70
2 162.6 104 7.32 12.2 5 3.29
3 196.9 89 8.86 6.6 7 1.78
4 186.9 121 8.41 10.1 3 2.73

CustServ Calls Churn?
0 1 False.
1 1 False.
2 0 False.
3 2 False.
4 3 False.
```

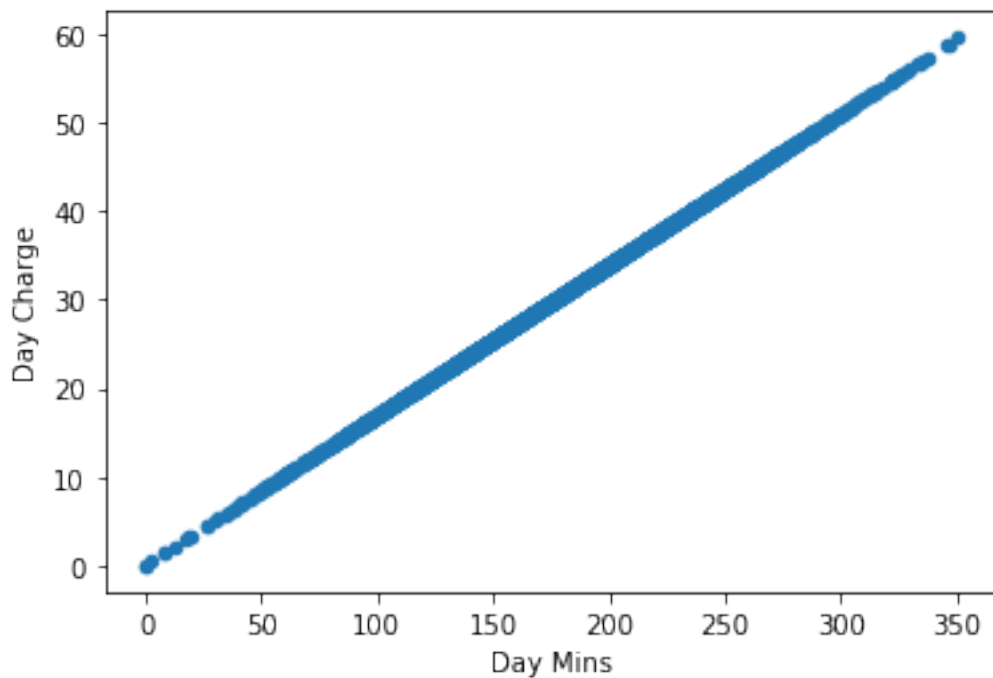
[5 rows x 21 columns]

```
[3]: %matplotlib inline
```

1.1 Scatter Plot

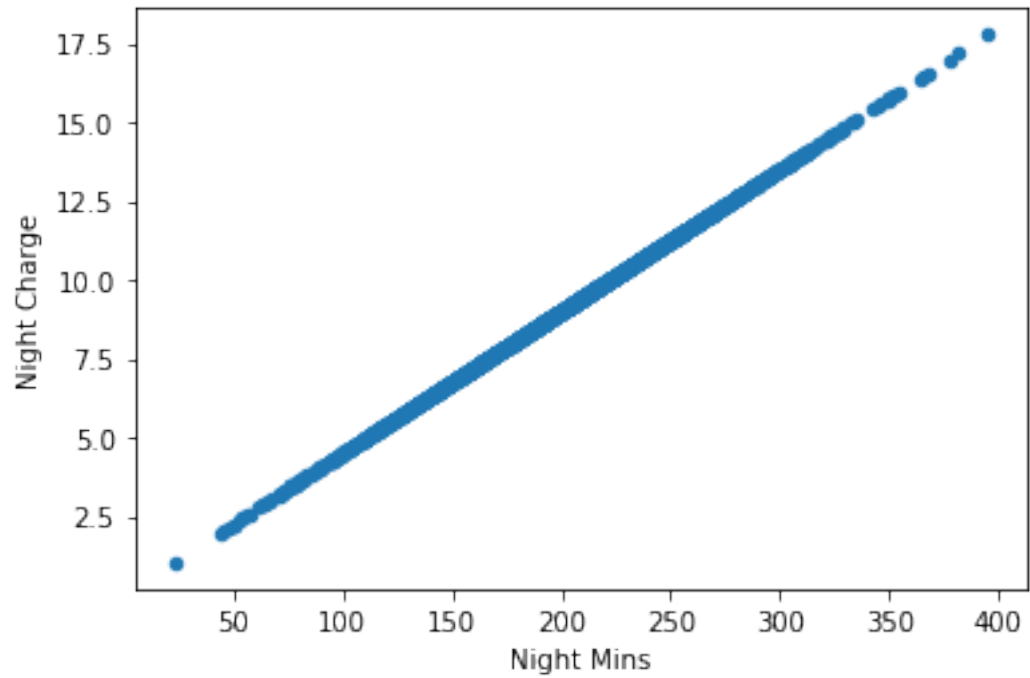
```
[20]: data.plot(kind='scatter', x='Day Mins', y='Day Charge')
```

```
[20]: <matplotlib.axes._subplots.AxesSubplot at 0x1f02845b6a0>
```



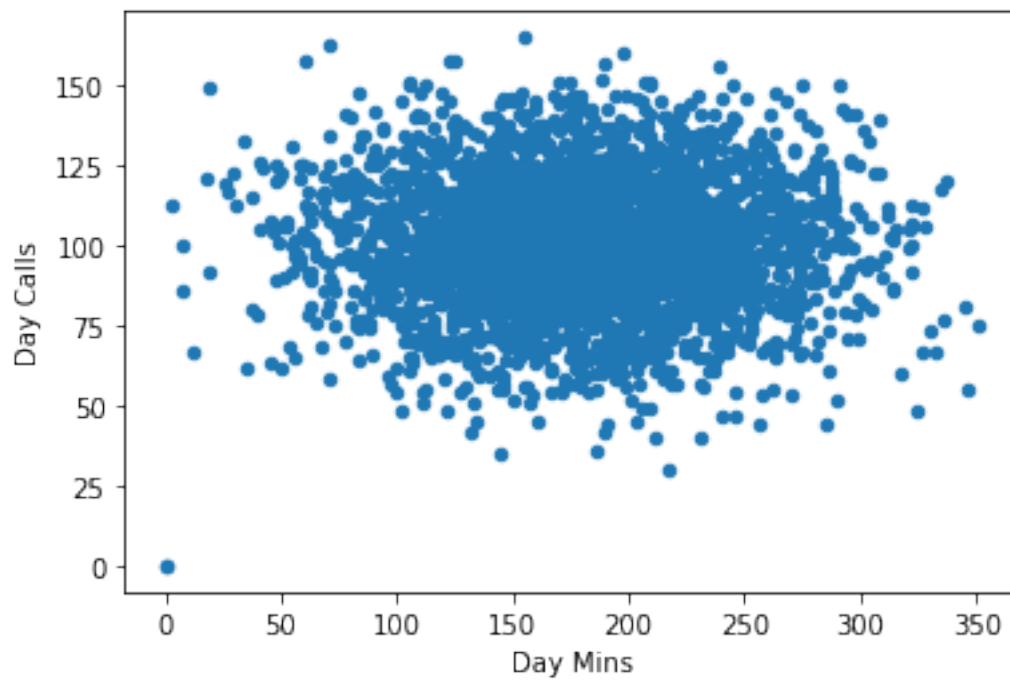
```
[14]: data.plot(kind='scatter', x='Night Mins', y='Night Charge')
```

```
[14]: <matplotlib.axes._subplots.AxesSubplot at 0x1f027168a90>
```



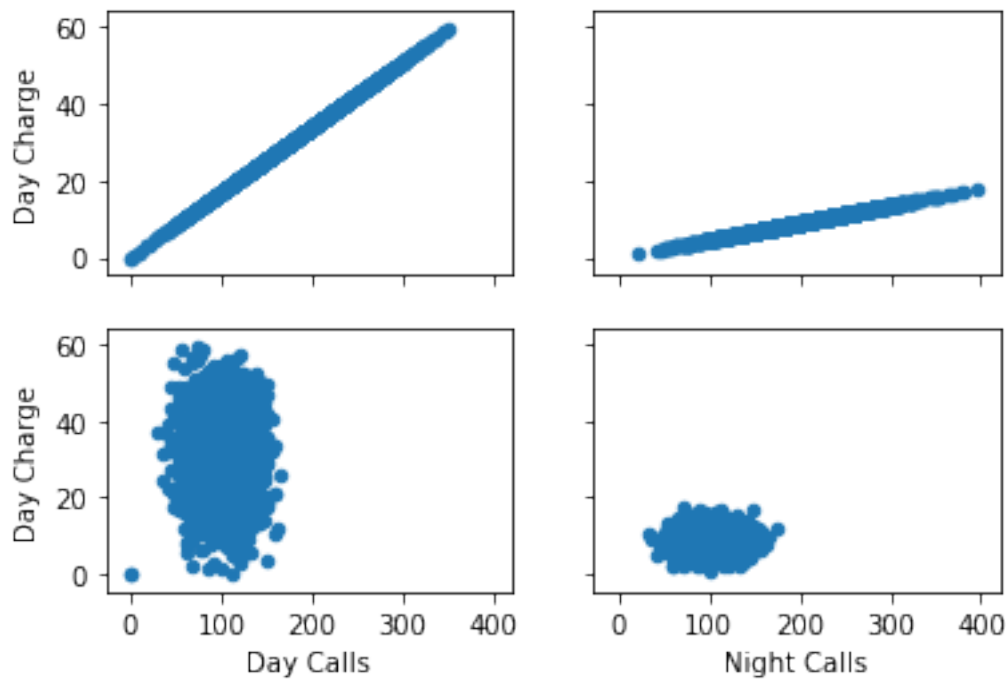
```
[15]: data.plot(kind='scatter', x='Day Mins', y='Day Calls')
```

```
[15]: <matplotlib.axes._subplots.AxesSubplot at 0x1f027223cf8>
```



```
[19]: figure, axes = plt.subplots(2,2, sharey=True, sharex=True)
data.plot(kind='scatter', x='Day Mins', y='Day Charge',ax= axes[0][0])
data.plot(kind='scatter', x='Night Mins', y='Night Charge', ax=axes[0][1])
data.plot(kind='scatter', x='Day Calls', y='Day Charge',ax= axes[1][0])
data.plot(kind='scatter', x='Night Calls', y='Night Charge', ax=axes[1][1])
```

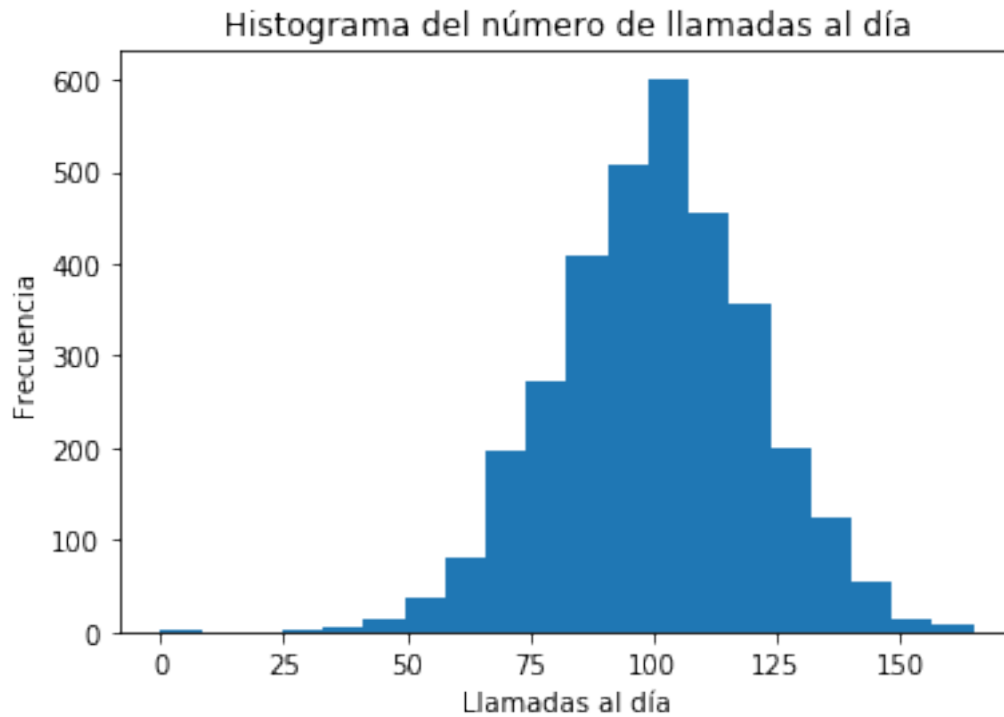
[19]: <matplotlib.axes._subplots.AxesSubplot at 0x1f02707b710>



1.2 Histograma de frecuencias

```
[26]: plt.hist(data['Day Calls'], bins=20)
plt.xlabel('Llamadas al día')
plt.ylabel('Frecuencia')
plt.title('Histograma del número de llamadas al día')
```

[26]: Text(0.5, 1.0, 'Histograma del número de llamadas al día')



Para saber cuantas divisiones se hacen en un histograma(bins), se sigue la regla de Sturges
Este número viene dado por “c” en la siguiente expresión:

$c = 1 + \log_2(M)$,donde M es el tamaño de la muestra

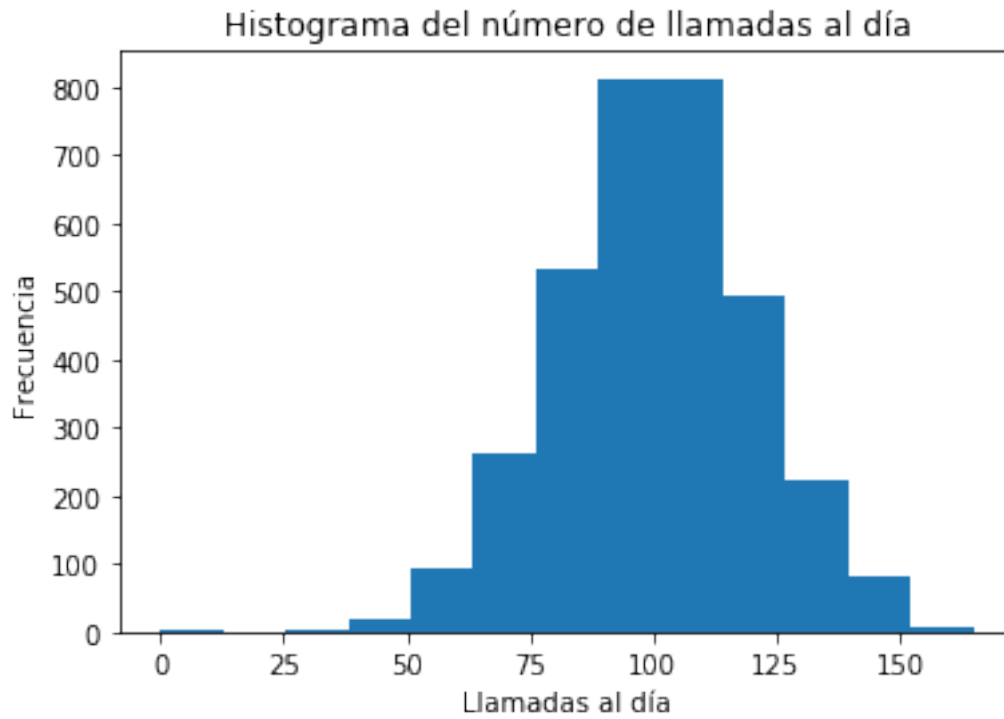
Por lo tanto el histograma pasado quedaría: $c = 1 + \log_2(3333)$

```
[43]: c= 1 + np.log2(3333)
      c=int(c.round())
      print(c)
```

13

```
[45]: plt.hist(data['Day Calls'], bins=c) #bins =c
      plt.xlabel('Llamadas al día')
      plt.ylabel('Frecuencia')
      plt.title('Histograma del número de llamadas al día')
```

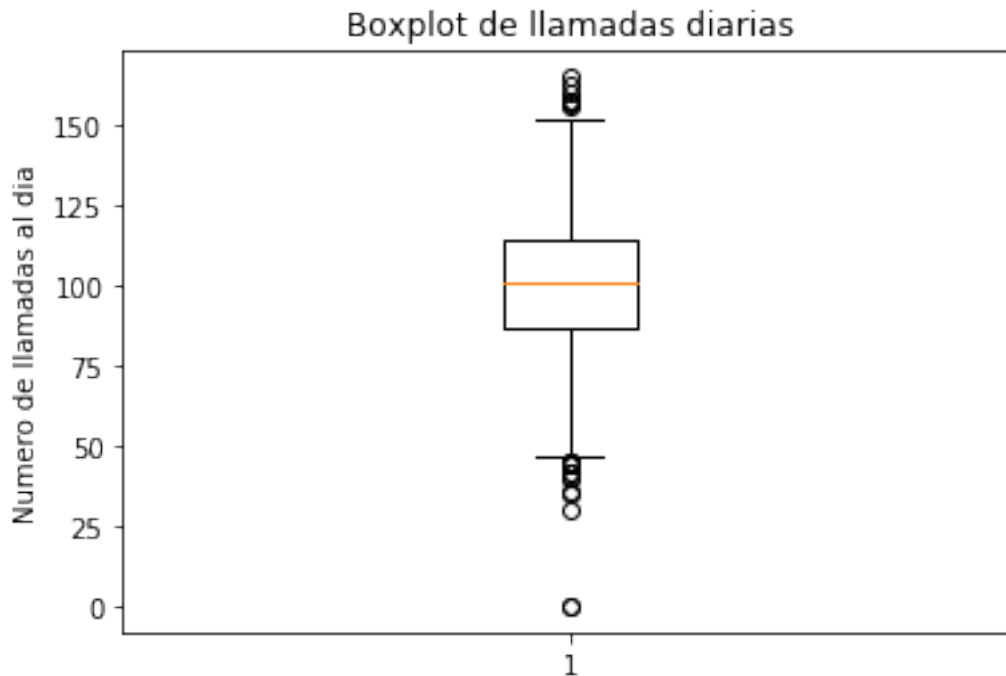
```
[45]: Text(0.5, 1.0, 'Histograma del número de llamadas al día')
```



1.3 Boxplot

```
[48]: plt.boxplot(data['Day Calls'])  
      plt.ylabel('Numero de llamadas al dia')  
      plt.title('Boxplot de llamadas diarias')
```

```
[48]: Text(0.5, 1.0, 'Boxplot de llamadas diarias')
```



- *Base de la caja:* De ahí para abajo son el 25% de los datos más chicos (cuantil 25%)
- *Tapa de la caja:* De ahí para arriba son el 25% de los datos más grandes (cuantil 75%)
- *Mitad de la caja:* La mediana de los datos
- *Ancho de la caja:* Abarca donde se concentran el 50% de los datos
- *Circulos:* Outliers (Pocos valores fuera de lo común)

```
[49]: data['Day Calls'].describe()
```

```
[49]: count    3333.000000
      mean      100.435644
      std       20.069084
      min        0.000000
      25%       87.000000
      50%      101.000000
      75%      114.000000
      max      165.000000
      Name: Day Calls, dtype: float64
```

```
[ ]:
```