



# Programación Estadística con Python

## Sesión 3

**Describing nominal and quantitative data**

**Alberto Sanz, Ph.D**

`asanz@edem.es`

**MASTER EN DATA ANALYTICS PARA LA EMPRESA**

# Describing nominal variables (I)

2

```
#Create a dataframe with the table of frequencies  
mytable = wbr.groupby(['weathersit']).size()
```

```
# Transform frequencies to percentages  
# a) obtain n  
n=mytable.sum()
```

```
# b) divide by n in order to get  
# proportions, and multiply by 100
```

```
mytable = (mytable/n)*100
```

```
# Round to your pleasure  
mytable3 = round(mytable2,1)  
print (mytable3=
```

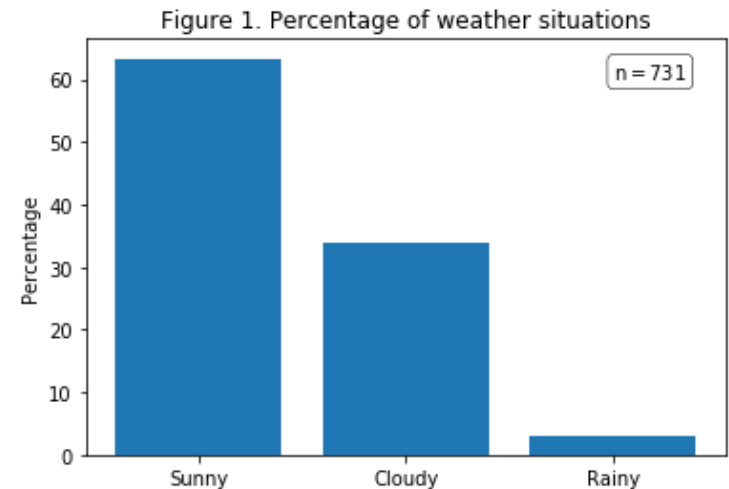
Table 1. Percentage of weather situations

Sunny	62
Cloudy	34
Rainy	4
(n)=731	

# Describing nominal variables (II)

3

```
#Barchart2
bar_list = ['Sunny', 'Cloudy', 'Rainy']
plt.bar(bar_list, mytable2, edgecolor='black')
plt.ylabel('Percentage')
plt.title('Figure 1. Percentage of weather situations')
```



```
#####
#Extra tip: Legend with sample size
# You need to have the sample size stored into n
props = dict(boxstyle='round', facecolor='white', lw=0.5)
textstr = '$\mathrm{n}=%.0f$'%(n)
plt.text (2,60, textstr ,   bbox=props)
#####
```

# Describing quantitative variables (I)

4

```
#Histogram Figure 1

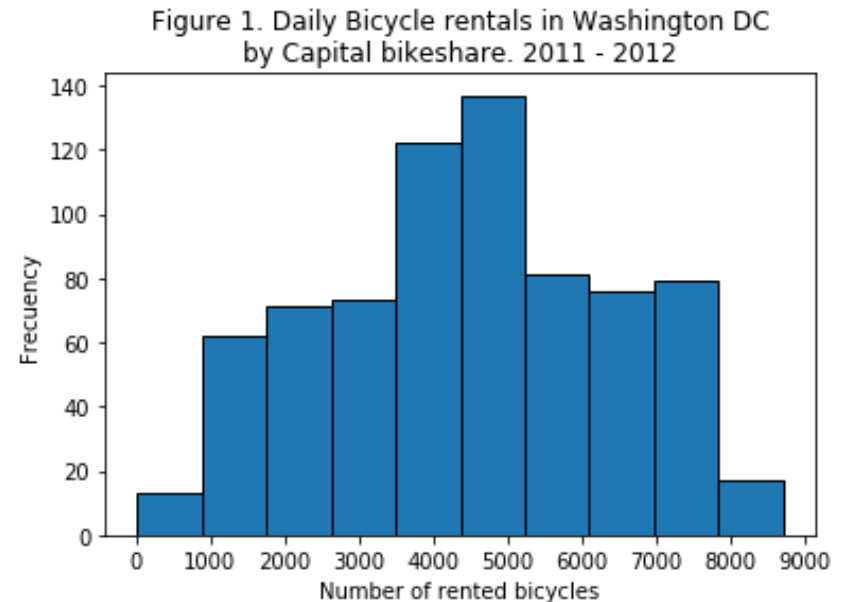
plt.hist(x, bins=10,
         edgecolor='black')

plt.xticks(np.arange(0, 10000,
                    step=1000))

plt.title('Figure 1. Daily Bicycle rentals
          in Washington DC'\n'
          'by Capital bikeshare.2011 - 2012')

plt.ylabel('Frecuency')

plt.xlabel('Number of rented bicycles')
```



# Describing quantitative variables (II)

5

```
#Histogram Figure 2
plt.hist(x, bins=10,          edgecolor='black')

plt.xticks(np.arange(0, 10000,
step=1000))

plt.title('Figure 1. Daily Bicycle rentals
in Washington DC'\n'
          'by Capital bikeshare.2011 - 2012')

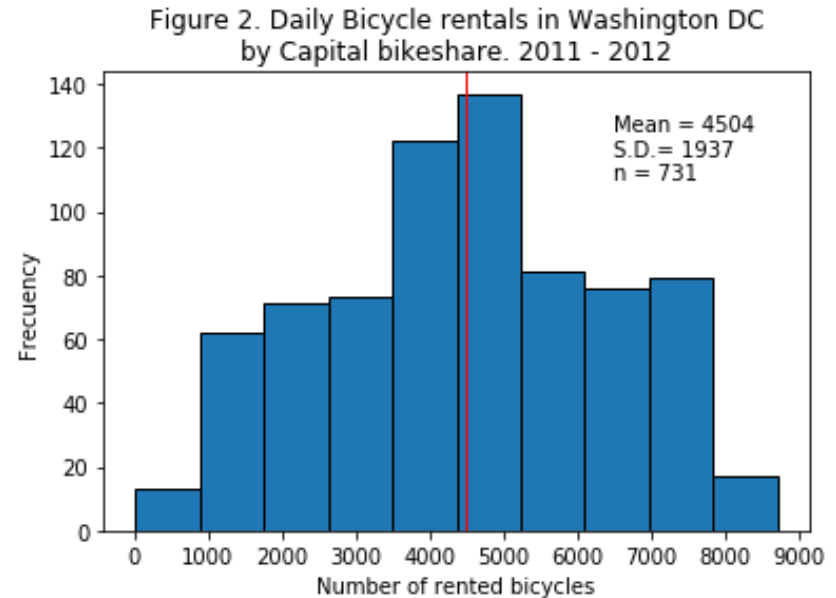
plt.ylabel('Frecuency')

plt.xlabel('Number of rented bicycles')

textstr = 'Mean = 4504\nS.D.= 1937 \nn = 731'

plt.text (6500,110, textstr)

# Add reference lines and store their names in
label for later legend
plt.axvline(x=4504,
            linewidth=1,
            linestyle= 'solid',
            color="red", label='Mean')
```



# Describing quantitative variables (III)

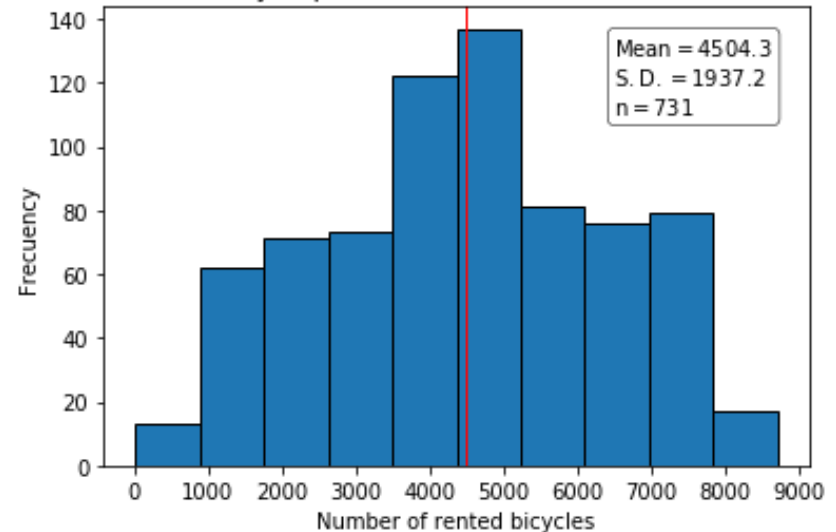
6

```
#histogram ver3
plt.hist(x, bins=10, edgecolor='black')
plt.xticks(np.arange(0, 10000, step=1000))
plt.title('Figure 3. Daily Bicycle rentals in Washington DC'
          '\n' 'by Capital bikeshare. 2011 - 2012')
plt.ylabel('Frecuency')
plt.xlabel('Number of rented bicycles')
```

```
props = dict(boxstyle='round', facecolor='white', lw=0.5)
textstr = '$\mathrm{Mean}=%.1f$\n$\mathrm{S.D.}=%.1f$\n$\mathrm{n}=%.0f$'%(m, sd, n)
plt.text(6500, 110, textstr, bbox=props)
```

```
plt.axvline(x=m,
            linewidth=1,
            linestyle= 'solid',
            color="red", label='Mean')
```

Figure 3. Daily Bicycle rentals in Washington DC by Capital bikeshare. 2011 - 2012



# Describing quantitative variables (IV)

7

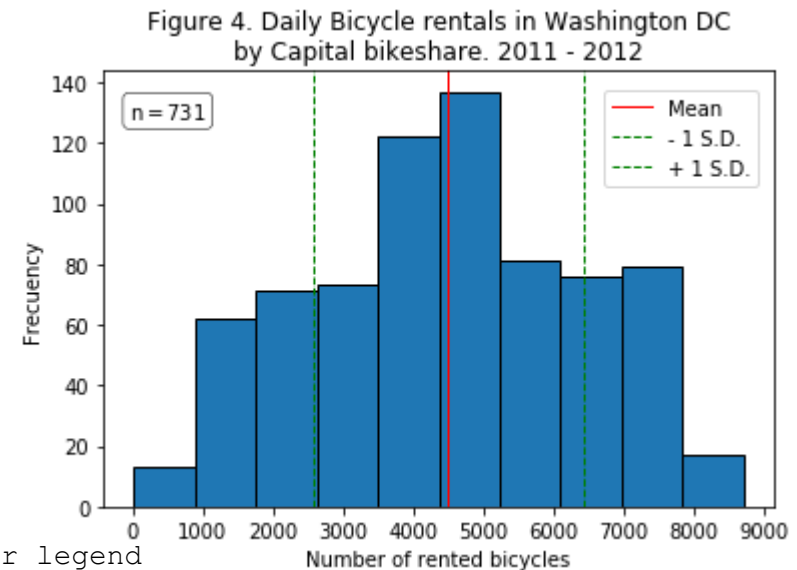
```
#histogram ver4
plt.hist(x, bins=10, edgecolor='black')
plt.xticks(np.arange(0, 10000, step=1000))
plt.title('Figure 1. Daily Bicycle rentals in Washington DC'
        '\n' 'by Capital bikeshare. 2011 - 2012')
plt.ylabel('Frecuency')
plt.xlabel('Number of rented bicycles')
```

```
props = dict(boxstyle='round', facecolor='white', lw=0.5)
textstr = '$\mathrm{n}=%.0f$'%(n)
plt.text(-50,128, textstr, bbox=props)
```

```
# Add reference lines and store their names in label for later legend
```

```
plt.axvline(x=m,
            linewidth=1,
            linestyle='solid',
            color="red", label='Mean')
plt.axvline(x=m-sd,
            linewidth=1,
            linestyle='dashed',
            color="green", label='- 1 S.D.')
plt.axvline(x=m + sd,
            linewidth=1,
            linestyle='dashed',
            color="green", label='+ 1 S.D.')
```

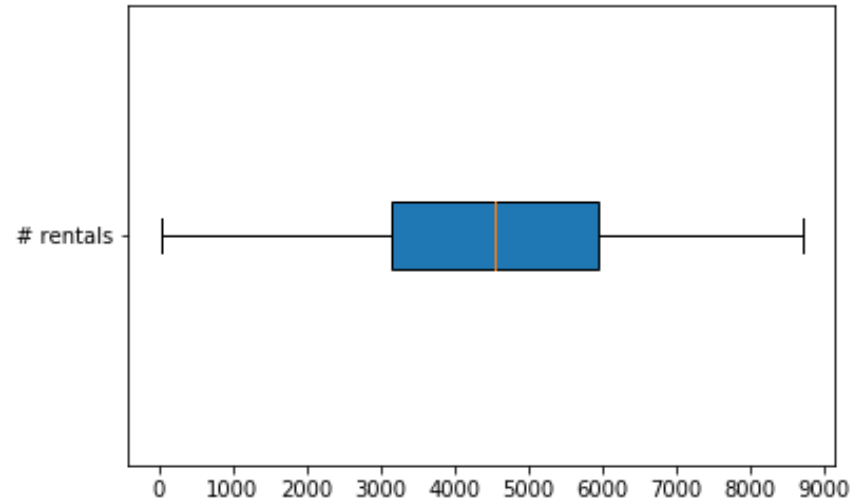
```
plt.legend(loc='upper left', bbox_to_anchor=(0.73, 0.98))
```



# Exploring quantitative variables (V)

8

```
#Boxplot
plt.boxplot(x,patch_artist=True,
            vert=False,
            labels=['# rentals'])
plt.xticks(np.arange(0, 10000, step=1000))
plt.show()
```





**Questions?**

**Thank you !**

Alberto Sanz  
asanz@edem.es