

## Capítulo 2

### Revisión de Estadística y probabilidad

#### 2.1 Introducción

Comenzamos con una breve revisión de la estadística, para presentar algunos de los conceptos claves y la notación que usamos en este y los siguientes capítulos. La Estadística nos ayuda con la **recopilación y análisis de datos** con el fin de obtener información, dibujar, obtener conclusiones y apoyo a la toma de decisiones. Los métodos estadísticos son necesarios cuando tenemos información incompleta sobre un fenómeno. Típicamente tenemos incompleta la información porque no podemos recopilar datos de todos los miembros de una población o si hay incertidumbre en las observaciones que hacemos (por ejemplo, debido a la medición y el ruido). Cuando no podemos encuestar a una población completa, una muestra elegida al azar puede estudiarse en su lugar, y podemos usar métodos estadísticos y calcular estadística descriptiva con (parámetros como la media y la desviación estándar) para hacer inferencias sobre las propiedades de toda la población (también llamado espacio muestral) de manera sistemática.

##### 2.1.1 Población y muestra

La población es el conjunto total de individuos, objetos, eventos, o medidas de consideración que tienen las mismas características y sobre el cual estamos interesados en obtener conclusiones.

“Es demasiado grande para poder estudiarlos”

La muestra es un subconjunto de la población al que tenemos accesos y sobre el cual realmente deberemos estudiar las características de interés de la población.

“la muestra debería ser representativa”

##### 2.1.2 Los Datos.

Hanks, puntualiza que a menudo pasamos por alto el hecho de que los datos significan algo (“La materia prima de la estadística y la simulación”) y que es importante entender su significado. Tenemos que mirar más allá de los números y comprender lo que representan si vamos a dar valor agregado a la resolución del problema, “no tiene nada que ver con algoritmos o ingeniería ni nada de eso”, la comprensión de los datos es un arte y muy importante.

Aquí es donde los analista de datos, científicos de datos, abren su caja de herramientas para encontrar el enfoque de análisis y algoritmos para trabajar con los datos. Hay cientos de técnicas para procesar los datos y obtener información, para la toma de decisiones. La investigación de Operaciones, la teoría de decisión, teoría de los juegos, la Simulación, Data Mining, Machine Learning y todas las metodologías que han existido desde hace tiempo. “Una vez que entiendes los datos y conoces el problema que estamos tratando de resolver, es cuando se puede elegir el algoritmo y obtener la solución óptima”.

Wikipedia, nos dice que un **dato** es una representación simbólica (numérica, alfabética, algorítmica, espacial, etc.) de un atributo o variable cuantitativa o cualitativa. Los Datos cualitativos nos describen el aspecto de un objeto o fenómeno mediante alguna característica (cualidad) específica (Ejemplo el estado civil de las personas, sexo, estrato social). Los datos cuantitativos, podemos expresar de forma numérica la magnitud de cierta característica de un objeto o fenómeno.

### 2.1.3 Fuentes de datos

Los datos pueden tener diversas fuentes a las que se puede acceder directa o indirectamente. La calidad, pertinencia y veracidad de un dato depende en gran medida de su origen y el modo en el que éste se obtiene. Una de las bases del método científico es la de poder identificar, aislar y de ser posible, reproducir la fuente de los datos que son objetos de un estudio.

## 2.2. Análisis estadísticos de datos.

No es objetivo específico de este libro detallar con profundidad la estadística pero básicamente todo técnico en simulación debe conocer cómo realizar un trabajo estadístico o una investigación estadística:

1. **Diseño formulación del análisis estadístico:** en esta actividad se planifica las tareas que debemos realizar y en muchos de los casos se debe definir la hipótesis.
2. **Observación estadística y toma de datos:** se realiza una exploración exhaustiva de todos los elementos de la población, la observación parcial solo se observa una parte de los elementos de la población “Muestra”, se utiliza cuando es necesario disponer de una parte representativa de la población.
3. **Exploración de los datos:** aquí realizamos la reducción estadística que describimos, resumimos, tabulamos, graficamos los datos para mirarlos desde diferentes ángulos, permitiendo tener datos de calidad.
4. **Modelado de los datos:** una vez que hemos generado los diferentes estadísticos ya podemos observar el comportamiento de los datos y podemos utilizar los diferentes modelos estadísticos dando así respuestas de predicciones, relaciones y correlaciones de los diferentes atributos de los datos.
5. **Realizar estimaciones:** aquí realizamos generalizaciones, estimaciones, pronósticos y otras acciones con los atributos de los datos permitiendo tener información variada para poder realizar conclusiones y análisis de resultados.
6. **Análisis de Resultados:** realizamos un análisis de los datos de los informes y estimaciones para determinar si son correctos y confirmar o rechazar la hipótesis.

### 2.2.1 Datos continuos y Datos discretos.

Los datos cuantitativos se dividen en datos continuos que puede admitir cualquier valor intermedio dentro de un intervalo de los números reales. Por ejemplo el rendimiento académico, la estatura, el peso, salario, tiempo, volumen que pueden ser obtenidos por medición. Los datos discretos son aquella que admite interrupciones verdaderas en su medición y por lo tanto no admite valores intermedios. Por ejemplo, número de alumnos de la clase, el número de hijos de una familia, el número de clientes en espera en un almacén.

## 2.3 Estadística descriptiva.

Técnicas para resumir y describir datos cuantitativos: Esta descripción le informará de la localización, dispersión, forma de la distribución de sus datos, utilizando técnicas como Tablas de frecuencia, Gráficos: Barras. sectorial. polígono de frecuencias. curva normal. Descripciones numéricas. Promedios (media.

mediana y moda), medidas de variabilidad (desviación estándar, varianza), medida de la relación entre las variables, etc.

Para el análisis de datos del estudio, se tomará la información de los registros diarios de la atención en emergencias del Hospital Rodríguez Zambrano en los meses de abril y mayo del 2017. Tomando en cuenta el total de pacientes que ingresan, los atendidos y no atendidos.

## **Tabla 2.1 fuentes de datos**

| Día<br>Semana | FECHA      | TOTAL_PACIENTES | ATENDIDOS | NO-ATENDIDOS |
|---------------|------------|-----------------|-----------|--------------|
| Sábado        | 01/04/2017 | 93              | 62        | 31           |
| Domingo       | 02/04/2017 | 111             | 86        | 25           |
| Lunes         | 03/04/2017 | 147             | 79        | 68           |
| Martes        | 04/04/2017 | 161             | 123       | 38           |
| Miercoles     | 05/04/2017 | 124             | 100       | 24           |
| Jueves        | 06/04/2017 | 159             | 130       | 29           |
| Viernes       | 07/04/2017 | 148             | 113       | 35           |
| Sábado        | 08/04/2017 | 89              | 75        | 14           |
| Domingo       | 09/04/2017 | 68              | 56        | 12           |
| Lunes         | 10/04/2017 | 167             | 116       | 51           |
| Martes        | 11/04/2017 | 136             | 100       | 35           |
| Miercoles     | 12/04/2017 | 128             | 117       | 11           |
| Jueves        | 13/04/2017 | 110             | 93        | 17           |
| Viernes       | 14/04/2017 | 88              | 58        | 30           |
| Sábado        | 15/04/2017 | 123             | 89        | 34           |
| Domingo       | 16/04/2017 | 136             | 116       | 20           |
| Lunes         | 17/04/2017 | 189             | 149       | 40           |
| Martes        | 18/04/2017 | 151             | 111       | 40           |
| Miercoles     | 19/04/2017 | 130             | 95        | 35           |
| Jueves        | 20/04/2017 | 122             | 108       | 14           |
| Viernes       | 21/04/2017 | 110             | 84        | 26           |
| Sábado        | 22/04/2017 | 94              | 62        | 32           |
| Domingo       | 23/04/2017 | 102             | 71        | 31           |
| Lunes         | 24/04/2017 | 166             | 130       | 36           |
| Martes        | 25/04/2017 | 122             | 86        | 36           |
| Miercoles     | 26/04/2017 | 108             | 89        | 19           |
| Jueves        | 27/04/2017 | 120             | 99        | 21           |
| Viernes       | 28/04/2017 | 116             | 91        | 25           |
| Sábado        | 29/04/2017 | 90              | 63        | 27           |
| Domingo       | 30/04/2017 | 98              | 64        | 34           |
| Lunes         | 01/05/2017 | 150             | 121       | 29           |
| Martes        | 02/05/2017 | 159             | 118       | 40           |
| Miercoles     | 03/05/2017 | 180             | 143       | 37           |
| Jueves        | 04/05/2017 | 194             | 141       | 53           |
| Viernes       | 05/05/2017 | 150             | 109       | 41           |
| Sábado        | 06/05/2017 | 148             | 106       | 42           |
| Domingo       | 07/05/2017 | 137             | 105       | 32           |
| Lunes         | 08/05/2017 | 182             | 140       | 42           |
| Martes        | 09/05/2017 | 176             | 143       | 32           |
| Miercoles     | 10/05/2017 | 164             | 125       | 38           |
| Jueves        | 11/05/2017 | 182             | 142       | 40           |
| Viernes       | 12/05/2017 | 182             | 151       | 31           |
| Sábado        | 13/05/2017 | 146             | 114       | 32           |
| Domingo       | 14/05/2017 | 99              | 71        | 28           |
| Lunes         | 15/05/2017 | 188             | 134       | 52           |
| Martes        | 16/05/2017 | 178             | 137       | 41           |
| Miercoles     | 17/05/2017 | 166             | 121       | 44           |
| Jueves        | 18/05/2017 | 161             | 124       | 36           |
| Viernes       | 19/05/2017 | 173             | 109       | 64           |
| Sábado        | 20/05/2017 | 113             | 71        | 42           |
| Domingo       | 21/05/2017 | 94              | 61        | 32           |

|           |            |     |     |    |
|-----------|------------|-----|-----|----|
| Lunes     | 22/05/2017 | 178 | 146 | 32 |
| Martes    | 23/05/2017 | 207 | 152 | 55 |
| Miercoles | 24/05/2017 | 177 | 129 | 48 |
| Jueves    | 25/05/2017 | 111 | 86  | 25 |
| Viernes   | 26/05/2017 | 133 | 95  | 38 |
| Sábado    | 27/05/2017 | 137 | 97  | 40 |
| Domingo   | 28/05/2017 | 157 | 110 | 47 |
| Lunes     | 29/05/2017 | 162 | 119 | 43 |
| Martes    | 30/05/2017 | 140 | 122 | 18 |
| Miercoles | 31/05/2017 | 154 | 129 | 18 |

## Observación estadística de los datos.-

Por ahora tenemos en forma natural los datos obtenidos del registro de datos de la institución en la tabla 2.1 que no constituyen una fuente de mucha información, los registros en forma de tabla tienen los siguientes metadatos:

- Día de la semana
- FECHA
- TOTAL\_PACIENTES
- ATENDIDOS
- NO-ATENDIDOS

La **Distribución de frecuencias** separa los datos en clases y muestra el número de ocurrencias en cada clase, o frecuencia de clase, este tipo de agrupación facilita la interpretación y la presentación en cuadros numéricos, que luego se representan en gráficos.

Una de las mejores maneras de describir una variable es representar los valores que aparecen en el conjunto de datos y el número de veces que aparece cada valor. La representación más común de una distribución es un histograma, que es un gráfico que muestra la frecuencia de cada valor.

## Histogramas

En Python, podemos graficar fácilmente un histograma con la ayuda de la función **hist** de matplotlib, simplemente debemos pasarle los datos y la cantidad de contenedores en los que queremos dividirlos.

Aquí vamos a construir un Histograma de frecuencias:

- importamos:
- pandas
- matplotlib
- numpy
- Creamos un DataFrame con Pandas
- Presentamos los datos

## La biblioteca Matplotlib.

Matplotlib es la biblioteca de graficación basada en Python más popular y sobre la que una gran cantidad de proyectos se basan para despliegue de gráficos y visualización de datos.

```
In [1]: 1 # importamos la libreria Pandas, matplotlib y numpy que van a ser de mucha utilidad
2
3 import pandas as pd
4 import matplotlib.pyplot as plt
5 import numpy as np
6 # leemos los datos de la tabla del directorio Data de trabajo
7 datos = pd.read_csv('C:/Users/JORGEANIBAL/LIBRO/Data/datos.csv')
8 #Presentamos los datos en un DataFrame de Pandas
9 datos
10
```

Out[1]:

|     | DIASEMANA | FECHA      | TOTAL_PACIENTES | ATENDIDOS | NO_ATENDIDOS |
|-----|-----------|------------|-----------------|-----------|--------------|
| 0   | Sábado    | 01/04/2017 | 93              | 62        | 31           |
| 1   | Domingo   | 02/04/2017 | 111             | 86        | 25           |
| 2   | Lunes     | 03/04/2017 | 147             | 79        | 68           |
| 3   | Martes    | 04/04/2017 | 161             | 123       | 38           |
| 4   | Miercoles | 05/04/2017 | 124             | 100       | 24           |
| 5   | Jueves    | 06/04/2017 | 159             | 130       | 29           |
| 6   | Viernes   | 07/04/2017 | 148             | 113       | 35           |
| 7   | Sábado    | 08/04/2017 | 89              | 75        | 14           |
| 8   | Domingo   | 09/04/2017 | 68              | 56        | 12           |
| 9   | Lunes     | 10/04/2017 | 167             | 116       | 51           |
| 10  | Martes    | 11/04/2017 | 136             | 100       | 35           |
| 11  | Miercoles | 12/04/2017 | 128             | 117       | 11           |
| 12  | Jueves    | 13/04/2017 | 110             | 93        | 17           |
| 13  | Viernes   | 14/04/2017 | 88              | 58        | 30           |
| 14  | Sábado    | 15/04/2017 | 123             | 89        | 34           |
| 15  | Domingo   | 16/04/2017 | 136             | 116       | 20           |
| 16  | Lunes     | 17/04/2017 | 189             | 149       | 40           |
| 17  | Martes    | 18/04/2017 | 151             | 111       | 40           |
| 18  | Miercoles | 19/04/2017 | 130             | 95        | 35           |
| 19  | Jueves    | 20/04/2017 | 122             | 108       | 14           |
| 20  | Viernes   | 21/04/2017 | 110             | 84        | 26           |
| 21  | Sábado    | 22/04/2017 | 94              | 62        | 32           |
| 22  | Domingo   | 23/04/2017 | 102             | 71        | 31           |
| 23  | Lunes     | 24/04/2017 | 166             | 130       | 36           |
| 24  | Martes    | 25/04/2017 | 122             | 86        | 36           |
| 25  | Miercoles | 26/04/2017 | 108             | 89        | 19           |
| 26  | Jueves    | 27/04/2017 | 120             | 99        | 21           |
| 27  | Viernes   | 28/04/2017 | 116             | 91        | 25           |
| 28  | Sábado    | 29/04/2017 | 90              | 63        | 27           |
| 29  | Domingo   | 30/04/2017 | 98              | 64        | 34           |
| ... | ...       | ...        | ...             | ...       | ...          |

|    | DIASEMANA | FECHA      | TOTAL_PACIENTES | ATENDIDOS | NO_ATENDIDOS |
|----|-----------|------------|-----------------|-----------|--------------|
| 31 | Martes    | 02/05/2017 | 159             | 118       | 40           |
| 32 | Miercoles | 03/05/2017 | 180             | 143       | 37           |
| 33 | Jueves    | 04/05/2017 | 194             | 141       | 53           |
| 34 | Viernes   | 05/05/2017 | 150             | 109       | 41           |
| 35 | Sábado    | 06/05/2017 | 148             | 106       | 42           |
| 36 | Domingo   | 07/05/2017 | 137             | 105       | 32           |
| 37 | Lunes     | 08/05/2017 | 182             | 140       | 42           |
| 38 | Martes    | 09/05/2017 | 176             | 143       | 32           |
| 39 | Miercoles | 10/05/2017 | 164             | 125       | 38           |
| 40 | Jueves    | 11/05/2017 | 182             | 142       | 40           |
| 41 | Viernes   | 12/05/2017 | 182             | 151       | 31           |
| 42 | Sábado    | 13/05/2017 | 146             | 114       | 32           |
| 43 | Domingo   | 14/05/2017 | 99              | 71        | 28           |
| 44 | Lunes     | 15/05/2017 | 188             | 134       | 52           |
| 45 | Martes    | 16/05/2017 | 178             | 137       | 41           |
| 46 | Miercoles | 17/05/2017 | 166             | 121       | 44           |
| 47 | Jueves    | 18/05/2017 | 161             | 124       | 36           |
| 48 | Viernes   | 19/05/2017 | 173             | 109       | 64           |
| 49 | Sábado    | 20/05/2017 | 113             | 71        | 42           |
| 50 | Domingo   | 21/05/2017 | 94              | 61        | 32           |
| 51 | Lunes     | 22/05/2017 | 178             | 146       | 32           |
| 52 | Martes    | 23/05/2017 | 207             | 152       | 55           |
| 53 | Miercoles | 24/05/2017 | 177             | 129       | 48           |
| 54 | Jueves    | 25/05/2017 | 111             | 86        | 25           |
| 55 | Viernes   | 26/05/2017 | 133             | 95        | 38           |
| 56 | Sábado    | 27/05/2017 | 137             | 97        | 40           |
| 57 | Domingo   | 28/05/2017 | 157             | 110       | 47           |
| 58 | Lunes     | 29/05/2017 | 162             | 119       | 43           |
| 59 | Martes    | 30/05/2017 | 140             | 122       | 18           |
| 60 | Miercoles | 31/05/2017 | 154             | 129       | 18           |

61 rows × 5 columns

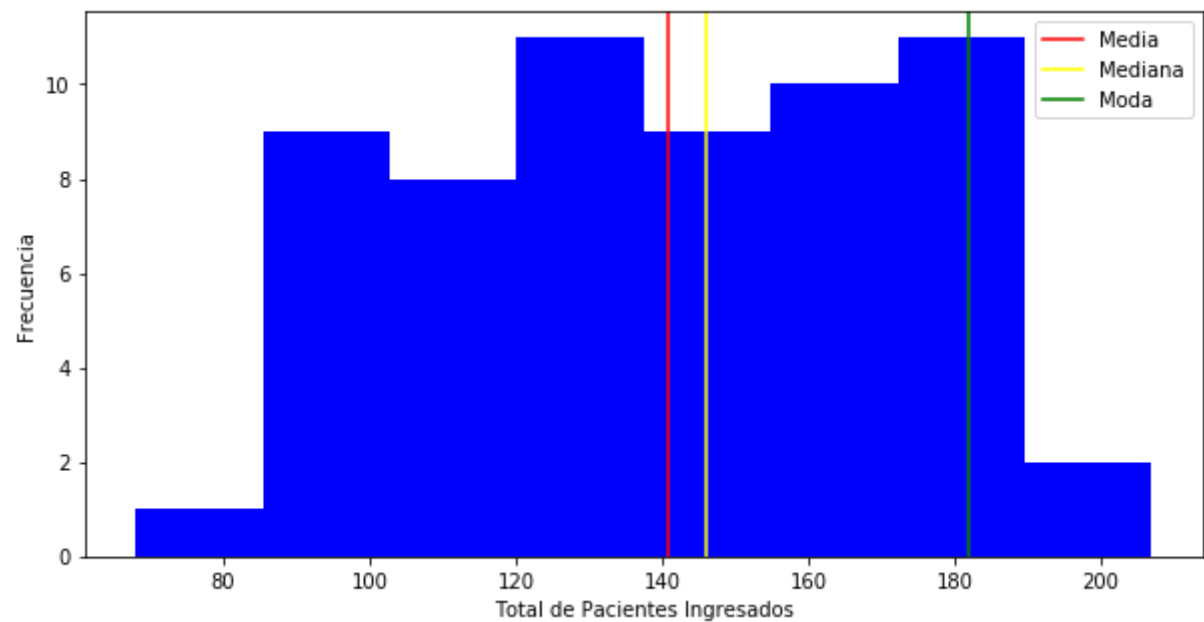
## Preparamos el Gráfico

Utilizamos la biblioteca `matplotlib.pyplot.hist(x,bins,Range=None,Color=None....)`

Donde **x\*\*** son los datos que vamos a graficar, **\*\*bins=8** el número de clases, y además el **color=blue**, en el gráfico también incorporamos la media **xmean()**, la mediana **x.median()**, y la moda **x.mode()[0]**, utilizando **plt.axvline**

In [4]:

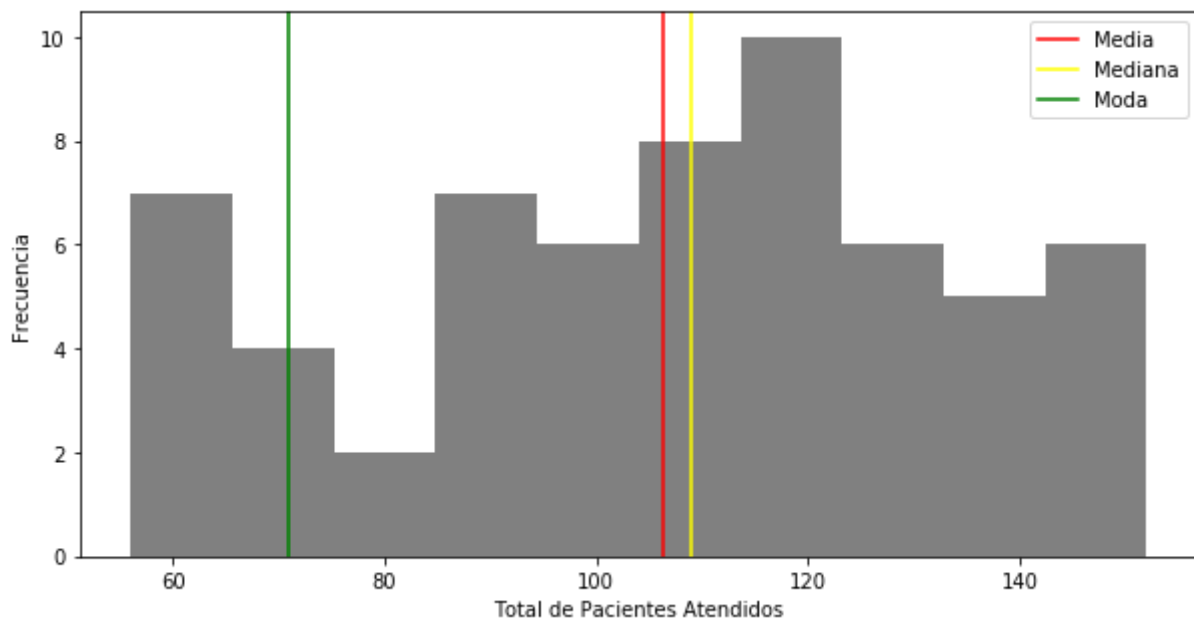
```
1 # Preparando para el grafico para la columna TOTAL PACIENTES
2 x=datos["TOTAL_PACIENTES"]
3 plt.figure(figsize=(10,5))
4 plt.hist(x,bins=8,color='blue')
5 plt.axvline(x.mean(),color='red',label='Media')
6 plt.axvline(x.median(),color='yellow',label='Mediana')
7 plt.axvline(x.mode()[0],color='green',label='Moda')
8 plt.xlabel('Total de Pacientes Ingresados')
9 plt.ylabel('Frecuencia')
10 plt.legend()
11 plt.show()
12
```





In [5]:

```
1 # Preparando para el grafico PARA LA COLUMNA ATENDIDOS
2 # Aqui cambiamos el color del histograma, y para las clases dejamos que matplotlib
3
4 x=datos["ATENDIDOS"]
5 plt.figure(figsize=(10,5))
6 plt.hist(x,bins=None,color='grey')
7 plt.axvline(x.mean(),color='red',label='Media')
8 plt.axvline(x.median(),color='yellow',label='Mediana')
9 plt.axvline(x.mode()[0],color='green',label='Moda')
10 plt.xlabel('Total de Pacientes Atendidos')
11 plt.ylabel('Frecuencia')
12 plt.legend()
13 plt.show()
14
```



## MEDIDAS DE TENDENCIA CENTRAL

### MEDIA, MEDIANA, MODA

### calculo de la media

es una medida de uso común, en una muestra de  $n$  datos, la media muestra el promedio aritmético simple de los datos utilizamos `mean()` para su calculo

```
In [6]: 1 # enviando las medias a t1, t2, t3 para su utilización
2 print("Media:", )
3
4 t1, t2, t3 = datos.mean()
5 print( "la Media de TOTAL PACIENTES:  ", t1)
6 print( "la Media de ATENDIDOS      :   ", t2)
7 print( "la Media de NO ATENDIDOS   :   ", t3)
8
9 print("DIRECTAMENTE DEL DATAFRAME ")
10 print()
11 datos.mean()
12
```

Media:  
la Media de TOTAL PACIENTES: 140.72131147540983  
la Media de ATENDIDOS : 106.32786885245902  
la Media de NO ATENDIDOS : 34.131147540983605  
DIRECTAMENTE DEL DATAFRAME

```
Out[6]: TOTAL_PACIENTES    140.721311
ATENDIDOS                106.327869
NO_ATENDIDOS             34.131148
dtype: float64
```

## calcula de la mediana

la mediana es el valor ubicado en el centro de los datos ordenados

```
In [7]: 1 # enviando las medias a m1, m2, m3 para su utilización
2 print("Mediana:", )
3
4 m1, m2, m3 = datos.median()
5
6 print( "la Mediana de TOTAL PACIENTES:  ", m1)
7 print( "la Mediana de ATENDIDOS      :   ", m2)
8 print( "la Mediana de NO ATENDIDOS   :   ", m3)
9
10 print("DIRECTAMENTE DEL DATAFRAME ")
11 print()
12 datos.median()
13
14
```

Mediana:  
la Mediana de TOTAL PACIENTES: 146.0  
la Mediana de ATENDIDOS : 109.0  
la Mediana de NO ATENDIDOS : 34.0  
DIRECTAMENTE DEL DATAFRAME

```
Out[7]: TOTAL_PACIENTES    146.0
ATENDIDOS                109.0
NO_ATENDIDOS             34.0
dtype: float64
```

## calcula de la moda

La moda es el dato que ocurre con más frecuencia en la muestra, puede existir o no, o también puede existir más de una moda

```
In [8]: 1 print("Moda:")
2
3 mo1 = datos["TOTAL_PACIENTES"].mode()
4 mo2 = datos["ATENDIDOS"].mode()
5 mo3 = datos["NO_ATENDIDOS"].mode()
6
7 print( "la Moda de TOTAL PACIENTES: ", mo1)
8 print( "la Moda de ATENDIDOS      : ", mo2)
9 print( "la Moda de NO ATENDIDOS   : ", mo3)
10
11
12 pd.DataFrame(mo1,mo2,mo3)
13
```

```
Moda:
la Moda de TOTAL PACIENTES:    0    182
dtype: int64
la Moda de ATENDIDOS      :    0    71
1    86
dtype: int64
la Moda de NO ATENDIDOS    :    0    32
dtype: int64
```

```
Out[8]:      32
71  NaN
86  NaN
```

Aqui observamos que para la variable ATENDIDOS tiene 2 modas la una de 71 y 86

## Resumen de medidas de tendencia central

los DataFrame de pandas permite entregar un informe completo de todas las medidas de tendencia central utilizando la función **describe()** donde nos presentará todas las medidas de tendencia central

```
In [9]: 1 # Tomamos Los datos de Las columnas
2 datos[['TOTAL_PACIENTES', 'ATENDIDOS', 'NO_ATENDIDOS']].describe()
3 # describe(), nos presenta directamente la media, desviación standar, el valor míni
4
```

```
Out[9]:
```

|       | TOTAL_PACIENTES | ATENDIDOS  | NO_ATENDIDOS |
|-------|-----------------|------------|--------------|
| count | 61.000000       | 61.000000  | 61.000000    |
| mean  | 140.721311      | 106.327869 | 34.131148    |
| std   | 32.517242       | 26.840095  | 11.811683    |
| min   | 68.000000       | 56.000000  | 11.000000    |
| 25%   | 113.000000      | 86.000000  | 27.000000    |
| 50%   | 146.000000      | 109.000000 | 34.000000    |
| 75%   | 166.000000      | 125.000000 | 40.000000    |
| max   | 207.000000      | 152.000000 | 68.000000    |

**Seleccionamos los datos del mes de abril y mayo para analizar los datos por mes**

Para seleccionar las columnas de un dataframe se usa una sintaxis de rangos por medio de dos puntos ::.

[::] Donde: corresponde al índice a partir del cual se iniciará el rango. corresponde al índice final del rango, el cual no será incluido en el resultado. corresponde al tamaño incrementos/decrementos que se aplicará al rango.

en el caso de nuestro ejercicio utilizaremos de la siguiente forma:

```
In [10]: 1 # seleccionamos los datos del mes de abril 30 registros para calcular estadísticos
2 df_1 = datos[:30]
3 print ("Estadísticos del mes de abril")
4 print ("-----")
5 df_1[['TOTAL_PACIENTES', 'ATENDIDOS', 'NO_ATENDIDOS']].describe()
6
```

Estadísticos del mes de abril

-----

Out[10]:

|       | TOTAL_PACIENTES | ATENDIDOS  | NO_ATENDIDOS |
|-------|-----------------|------------|--------------|
| count | 30.000000       | 30.000000  | 30.000000    |
| mean  | 123.533333      | 93.833333  | 29.666667    |
| std   | 28.369159       | 24.176197  | 11.905760    |
| min   | 68.000000       | 56.000000  | 11.000000    |
| 25%   | 103.500000      | 76.000000  | 21.750000    |
| 50%   | 122.000000      | 92.000000  | 30.500000    |
| 75%   | 144.250000      | 112.500000 | 35.000000    |
| max   | 189.000000      | 149.000000 | 68.000000    |

Para el caso del mes de mayo seleccionamos desde el rango de datos del mes

In [11]:

```
1 df_2 = datos[30:61]
2 df_2
3
```

Out[11]:

|    | DIASEMANA | FECHA      | TOTAL_PACIENTES | ATENDIDOS | NO_ATENDIDOS |
|----|-----------|------------|-----------------|-----------|--------------|
| 30 | Lunes     | 01/05/2017 | 150             | 121       | 29           |
| 31 | Martes    | 02/05/2017 | 159             | 118       | 40           |
| 32 | Miercoles | 03/05/2017 | 180             | 143       | 37           |
| 33 | Jueves    | 04/05/2017 | 194             | 141       | 53           |
| 34 | Viernes   | 05/05/2017 | 150             | 109       | 41           |
| 35 | Sábado    | 06/05/2017 | 148             | 106       | 42           |
| 36 | Domingo   | 07/05/2017 | 137             | 105       | 32           |
| 37 | Lunes     | 08/05/2017 | 182             | 140       | 42           |
| 38 | Martes    | 09/05/2017 | 176             | 143       | 32           |
| 39 | Miercoles | 10/05/2017 | 164             | 125       | 38           |
| 40 | Jueves    | 11/05/2017 | 182             | 142       | 40           |
| 41 | Viernes   | 12/05/2017 | 182             | 151       | 31           |
| 42 | Sábado    | 13/05/2017 | 146             | 114       | 32           |
| 43 | Domingo   | 14/05/2017 | 99              | 71        | 28           |
| 44 | Lunes     | 15/05/2017 | 188             | 134       | 52           |
| 45 | Martes    | 16/05/2017 | 178             | 137       | 41           |
| 46 | Miercoles | 17/05/2017 | 166             | 121       | 44           |
| 47 | Jueves    | 18/05/2017 | 161             | 124       | 36           |
| 48 | Viernes   | 19/05/2017 | 173             | 109       | 64           |
| 49 | Sábado    | 20/05/2017 | 113             | 71        | 42           |
| 50 | Domingo   | 21/05/2017 | 94              | 61        | 32           |
| 51 | Lunes     | 22/05/2017 | 178             | 146       | 32           |
| 52 | Martes    | 23/05/2017 | 207             | 152       | 55           |
| 53 | Miercoles | 24/05/2017 | 177             | 129       | 48           |
| 54 | Jueves    | 25/05/2017 | 111             | 86        | 25           |
| 55 | Viernes   | 26/05/2017 | 133             | 95        | 38           |
| 56 | Sábado    | 27/05/2017 | 137             | 97        | 40           |
| 57 | Domingo   | 28/05/2017 | 157             | 110       | 47           |
| 58 | Lunes     | 29/05/2017 | 162             | 119       | 43           |
| 59 | Martes    | 30/05/2017 | 140             | 122       | 18           |
| 60 | Miercoles | 31/05/2017 | 154             | 129       | 18           |

\*\*Igualmente realizamos de la misma forma para el mes de mayo

```
In [12]: 1 print ("Estadísticos del mes de Mayo ")
2 print ("-----")
3
4 df_2[['TOTAL_PACIENTES', 'ATENDIDOS', 'NO_ATENDIDOS']].describe()
```

Estadísticos del mes de Mayo  
-----

Out[12]:

|       | TOTAL_PACIENTES | ATENDIDOS  | NO_ATENDIDOS |
|-------|-----------------|------------|--------------|
| count | 31.000000       | 31.000000  | 31.000000    |
| mean  | 157.354839      | 118.419355 | 38.451613    |
| std   | 27.485206       | 23.845019  | 10.138503    |
| min   | 94.000000       | 61.000000  | 18.000000    |
| 25%   | 143.000000      | 107.500000 | 32.000000    |
| 50%   | 161.000000      | 121.000000 | 40.000000    |
| 75%   | 178.000000      | 138.500000 | 42.500000    |
| max   | 207.000000      | 152.000000 | 64.000000    |

## OTROS GRÁFICOS ADICIONALES

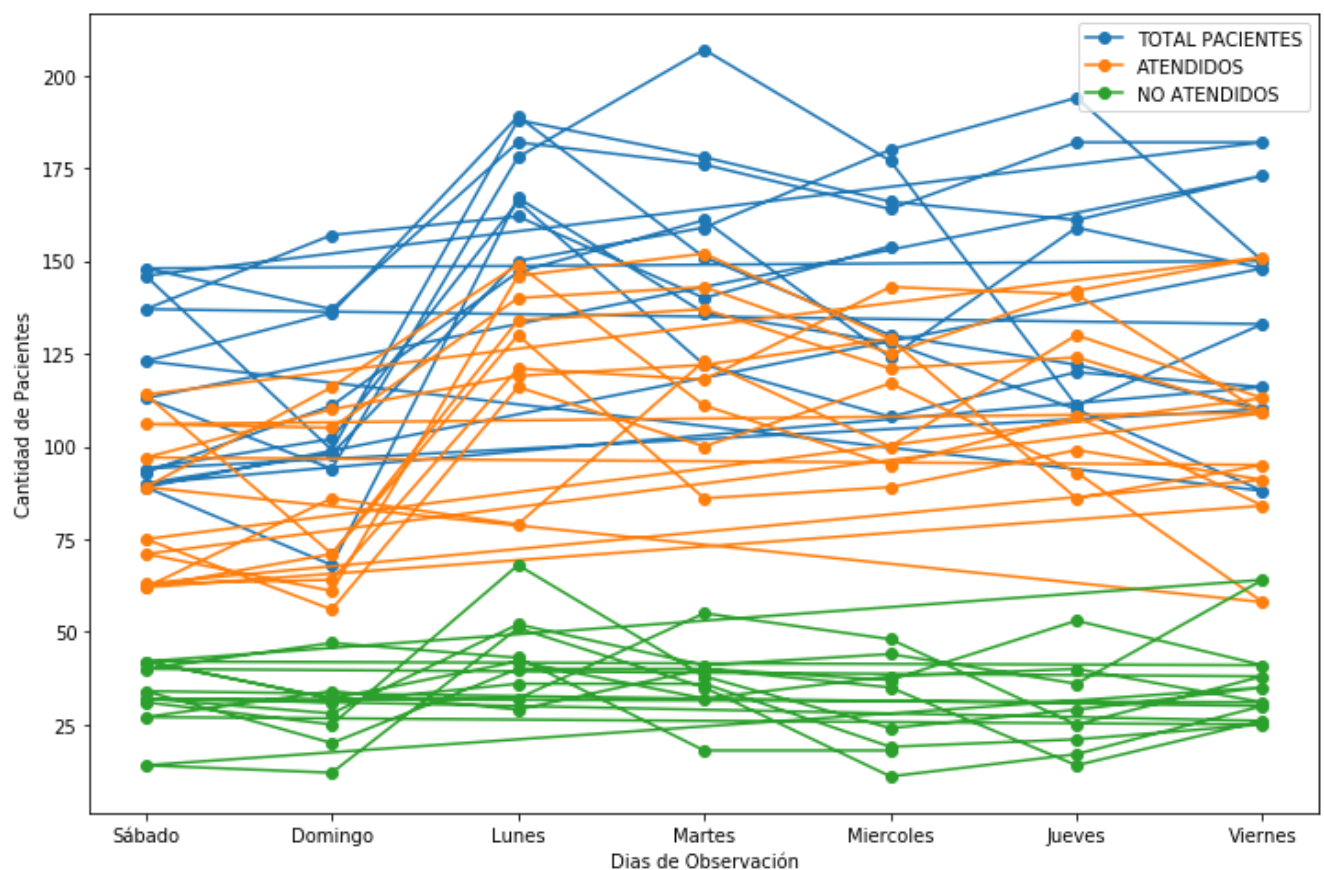
### Gráficos lineales

Nos permite mostrar la tendencia de los datos, para el caso del ejercicio el gráfico muestra la tendencia de los datos de las semanas del mes por cada una de las variables.

In [13]:

```
1 #x = np.arange(61)
2
3 x = datos["DIASEMANA"]
4 t1 = datos["TOTAL_PACIENTES"]
5 t2 = datos["ATENDIDOS"]
6 t3 = datos["NO_ATENDIDOS"]
7
8 plt.figure(figsize=(12,8))
9 plt.plot(x,t1,x,t2,x,t3,marker='o')
10 plt.xlabel('Dias de Observación')
11 plt.ylabel('Cantidad de Pacientes')
12
13
14 plt.legend(('TOTAL PACIENTES','ATENDIDOS', 'NO ATENDIDOS'), prop = {'size':10},loc=
15
```

Out[13]: <matplotlib.legend.Legend at 0x193ec1fc780>



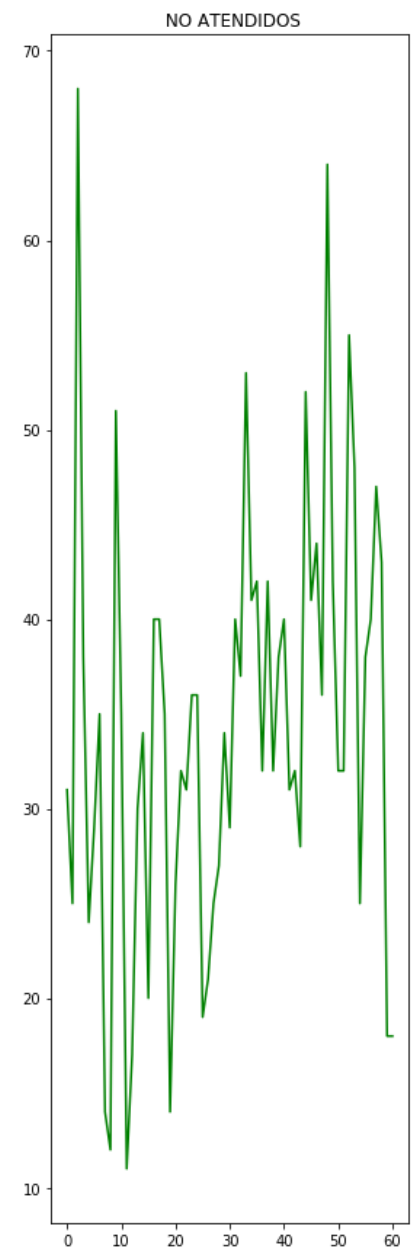
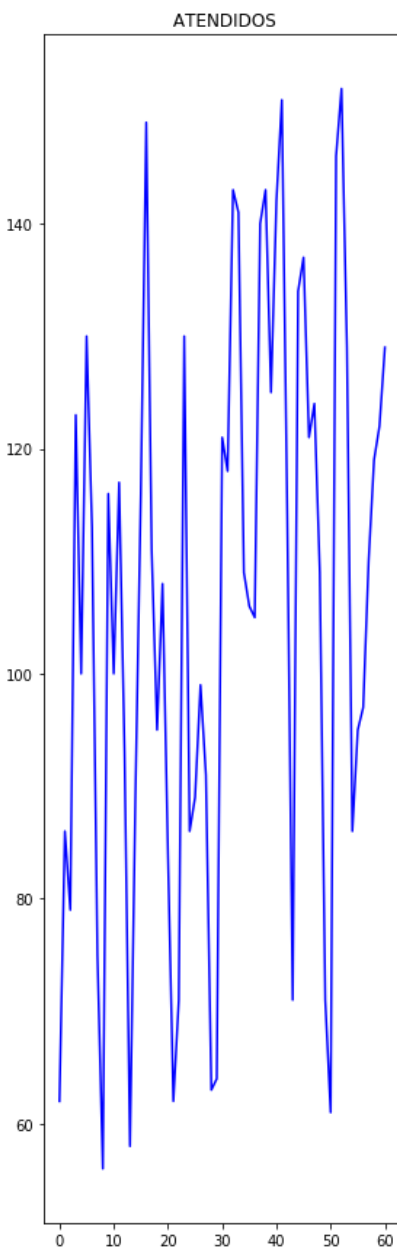
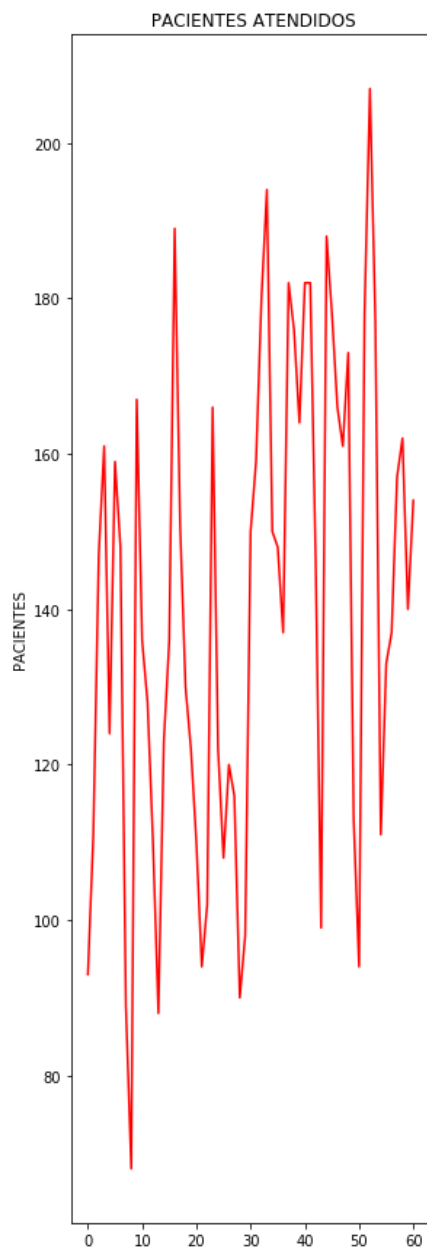
## Gráfico en varias ventanas

utilizamos subplot para hacer varios cuadros estadístico separadas de diferentes variables del problema. Esto nos permite tener una mejor apreciación de los datos

```
In [14]: 1
2 x = range(61)
3 plt.figure(figsize=(15,15))
4 plt.subplot(131)
5 t1 = datos["TOTAL_PACIENTES"]
6 p1, = plt.plot(x,t1,'r-')
7 plt.ylabel('PACIENTES')
8 plt.title(' PACIENTES ATENDIDOS')
9
10
11 plt.subplot(132)
12 t2 = datos["ATENDIDOS"]
13 p1, = plt.plot(x,t2,'b-')
14
15 plt.title(' ATENDIDOS')
16
17 plt.subplot(133)
18 t3 = datos["NO_ATENDIDOS"]
19 p1, = plt.plot(x,t3,'g-')
20
21 plt.title(' NO ATENDIDOS')
22
23
```

Out[14]: Text(0.5, 1.0, ' NO ATENDIDOS')





## TABLA DE FRECUENCIAS

In [15]:

```
1 # OBTENER LOS DATOS UNICOS DE LA TABLA
2 lis = datos["TOTAL_PACIENTES"].unique()
3 lis
4 dfclases=pd.DataFrame(lis,columns=["TOTAL_PACIENTES"])
5 dfclases
6
```

Out[15]:

| TOTAL_PACIENTES |     |
|-----------------|-----|
| 0               | 93  |
| 1               | 111 |
| 2               | 147 |
| 3               | 161 |
| 4               | 124 |
| 5               | 159 |
| 6               | 148 |
| 7               | 89  |
| 8               | 68  |
| 9               | 167 |
| 10              | 136 |
| 11              | 128 |
| 12              | 110 |
| 13              | 88  |
| 14              | 123 |
| 15              | 189 |
| 16              | 151 |
| 17              | 130 |
| 18              | 122 |
| 19              | 94  |
| 20              | 102 |
| 21              | 166 |
| 22              | 108 |
| 23              | 120 |
| 24              | 116 |
| 25              | 90  |
| 26              | 98  |
| 27              | 150 |
| 28              | 180 |
| 29              | 194 |
| 30              | 137 |
| 31              | 182 |
| 32              | 176 |
| 33              | 164 |
| 34              | 146 |

| TOTAL_PACIENTES |     |
|-----------------|-----|
| 35              | 99  |
| 36              | 188 |
| 37              | 178 |
| 38              | 173 |
| 39              | 113 |
| 40              | 207 |
| 41              | 177 |
| 42              | 133 |
| 43              | 157 |
| 44              | 162 |
| 45              | 140 |
| 46              | 154 |

```
In [16]: 1 #TABLA DE FRECUENCIAS ABSOLUTAS
2 # OBTENER FRECUENCIAS ABSOLUTAS DE CADA CLASE
3 datafi=pd.crosstab(index=datos["TOTAL_PACIENTES"], columns = "fi")
4
5 # Creamos una lista con los valores de las frecuencias
6
7 li = datafi.values
8 # agregamos una columna al dataframe
9 dfclases["fi"] = li
10 #observamos dfclase
11 dfclases =dfclases.sort_values('TOTAL_PACIENTES')
12
13
```

```
In [19]: 1 total = dfclases.sum(axis=0)
2 total
3
```

```
Out[19]: TOTAL_PACIENTES    6548
fi                        61
dtype: int64
```

```
In [20]: 1 total
2
```

```
Out[20]: TOTAL_PACIENTES    6548
fi                        61
dtype: int64
```

```

In [21]: 1 # Columna de Frecuencia relativa
          2
          3 total = dfclases.sum(axis=0)
          4
          5 datahi = dfclases["fi"]/total["fi"] # aqui calculamos la frecuencia
          6
          7 datahi.values
          8 # agregamos nueva columna de frecuencia relativa
          9 dfclases["hi"] = datahi
         10 dfclases
         11
         12

```

```

Out[21]:

```

|    | TOTAL_PACIENTES | fi | hi       |
|----|-----------------|----|----------|
| 8  | 68              | 1  | 0.016393 |
| 13 | 88              | 1  | 0.016393 |
| 7  | 89              | 1  | 0.016393 |
| 25 | 90              | 1  | 0.016393 |
| 0  | 93              | 1  | 0.016393 |
| 19 | 94              | 1  | 0.016393 |
| 26 | 98              | 2  | 0.032787 |
| 35 | 99              | 2  | 0.032787 |
| 20 | 102             | 1  | 0.016393 |
| 22 | 108             | 2  | 0.032787 |
| 12 | 110             | 1  | 0.016393 |
| 1  | 111             | 1  | 0.016393 |
| 39 | 113             | 1  | 0.016393 |
| 24 | 116             | 1  | 0.016393 |
| 23 | 120             | 1  | 0.016393 |
| 18 | 122             | 1  | 0.016393 |
| 14 | 123             | 1  | 0.016393 |
| 4  | 124             | 1  | 0.016393 |
| 11 | 128             | 2  | 0.032787 |
| 17 | 130             | 1  | 0.016393 |
| 42 | 133             | 3  | 0.049180 |
| 10 | 136             | 2  | 0.032787 |
| 30 | 137             | 1  | 0.016393 |
| 45 | 140             | 1  | 0.016393 |
| 34 | 146             | 1  | 0.016393 |
| 2  | 147             | 1  | 0.016393 |
| 6  | 148             | 1  | 0.016393 |
| 27 | 150             | 2  | 0.032787 |
| 16 | 151             | 1  | 0.016393 |
| 46 | 154             | 1  | 0.016393 |
| 43 | 157             | 1  | 0.016393 |

|    | TOTAL_PACIENTES | fi | hi       |
|----|-----------------|----|----------|
| 5  | 159             | 2  | 0.032787 |
| 3  | 161             | 1  | 0.016393 |
| 44 | 162             | 1  | 0.016393 |
| 33 | 164             | 1  | 0.016393 |
| 21 | 166             | 2  | 0.032787 |
| 9  | 167             | 1  | 0.016393 |
| 38 | 173             | 1  | 0.016393 |
| 32 | 176             | 2  | 0.032787 |
| 41 | 177             | 1  | 0.016393 |
| 37 | 178             | 1  | 0.016393 |
| 28 | 180             | 1  | 0.016393 |
| 31 | 182             | 2  | 0.032787 |
| 36 | 188             | 1  | 0.016393 |
| 15 | 189             | 2  | 0.032787 |
| 29 | 194             | 1  | 0.016393 |
| 40 | 207             | 2  | 0.032787 |

```
In [22]: 1 total1 = dfclases.sum(axis=0) # totales
```

```
In [23]: 1 total1
          2
```

```
Out[23]: TOTAL_PACIENTES    6548.0
          fi                61.0
          hi                 1.0
          dtype: float64
```

In [24]:

```
1 # la suma de Las frecuencias Relativas nos da 1
2 # aqui vamos a calcular la frecuencia absoluta
3 FA = dfclases["fi"].values
4 # obtenemos FA
5 a=[]
6 b=0
7 for c in FA:
8     b = c + b
9     a.append(b)
10 dfclases["FA"] = a
11
12 HI = dfclases["hi"].values
13 # obtenemos HI
14 a=[]
15 b=0
16 for c in HI:
17     b = c + b
18     a.append(b)
19 dfclases["HI"] = a
20
21 dfclases
22
23
```

Out[24]:

|    | TOTAL_PACIENTES | fi | hi       | FA | HI       |
|----|-----------------|----|----------|----|----------|
| 8  | 68              | 1  | 0.016393 | 1  | 0.016393 |
| 13 | 88              | 1  | 0.016393 | 2  | 0.032787 |
| 7  | 89              | 1  | 0.016393 | 3  | 0.049180 |
| 25 | 90              | 1  | 0.016393 | 4  | 0.065574 |
| 0  | 93              | 1  | 0.016393 | 5  | 0.081967 |
| 19 | 94              | 1  | 0.016393 | 6  | 0.098361 |
| 26 | 98              | 2  | 0.032787 | 8  | 0.131148 |
| 35 | 99              | 2  | 0.032787 | 10 | 0.163934 |
| 20 | 102             | 1  | 0.016393 | 11 | 0.180328 |
| 22 | 108             | 2  | 0.032787 | 13 | 0.213115 |
| 12 | 110             | 1  | 0.016393 | 14 | 0.229508 |
| 1  | 111             | 1  | 0.016393 | 15 | 0.245902 |
| 39 | 113             | 1  | 0.016393 | 16 | 0.262295 |
| 24 | 116             | 1  | 0.016393 | 17 | 0.278689 |
| 23 | 120             | 1  | 0.016393 | 18 | 0.295082 |
| 18 | 122             | 1  | 0.016393 | 19 | 0.311475 |
| 14 | 123             | 1  | 0.016393 | 20 | 0.327869 |
| 4  | 124             | 1  | 0.016393 | 21 | 0.344262 |
| 11 | 128             | 2  | 0.032787 | 23 | 0.377049 |
| 17 | 130             | 1  | 0.016393 | 24 | 0.393443 |
| 42 | 133             | 3  | 0.049180 | 27 | 0.442623 |
| 10 | 136             | 2  | 0.032787 | 29 | 0.475410 |
| 30 | 137             | 1  | 0.016393 | 30 | 0.491803 |
| 45 | 140             | 1  | 0.016393 | 31 | 0.508197 |

|    | TOTAL_PACIENTES | fi | hi       | FA | HI       |
|----|-----------------|----|----------|----|----------|
| 34 | 146             | 1  | 0.016393 | 32 | 0.524590 |
| 2  | 147             | 1  | 0.016393 | 33 | 0.540984 |
| 6  | 148             | 1  | 0.016393 | 34 | 0.557377 |
| 27 | 150             | 2  | 0.032787 | 36 | 0.590164 |
| 16 | 151             | 1  | 0.016393 | 37 | 0.606557 |
| 46 | 154             | 1  | 0.016393 | 38 | 0.622951 |
| 43 | 157             | 1  | 0.016393 | 39 | 0.639344 |
| 5  | 159             | 2  | 0.032787 | 41 | 0.672131 |
| 3  | 161             | 1  | 0.016393 | 42 | 0.688525 |
| 44 | 162             | 1  | 0.016393 | 43 | 0.704918 |
| 33 | 164             | 1  | 0.016393 | 44 | 0.721311 |
| 21 | 166             | 2  | 0.032787 | 46 | 0.754098 |
| 9  | 167             | 1  | 0.016393 | 47 | 0.770492 |
| 38 | 173             | 1  | 0.016393 | 48 | 0.786885 |
| 32 | 176             | 2  | 0.032787 | 50 | 0.819672 |
| 41 | 177             | 1  | 0.016393 | 51 | 0.836066 |
| 37 | 178             | 1  | 0.016393 | 52 | 0.852459 |
| 28 | 180             | 1  | 0.016393 | 53 | 0.868852 |
| 31 | 182             | 2  | 0.032787 | 55 | 0.901639 |
| 36 | 188             | 1  | 0.016393 | 56 | 0.918033 |
| 15 | 189             | 2  | 0.032787 | 58 | 0.950820 |
| 29 | 194             | 1  | 0.016393 | 59 | 0.967213 |
| 40 | 207             | 2  | 0.032787 | 61 | 1.000000 |

In [25]: 1 total1 = dfclases.sum(axis=0)

In [26]: 1 total1  
2

Out[26]: TOTAL\_PACIENTES 6548.000000  
fi 61.000000  
hi 1.000000  
FA 1422.000000  
HI 23.311475  
dtype: float64

In [27]:

```
1 datos.info()  
2
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 61 entries, 0 to 60  
Data columns (total 5 columns):  
DIASEMANA           61 non-null object  
FECHA               61 non-null object  
TOTAL_PACIENTES     61 non-null int64  
ATENDIDOS           61 non-null int64  
NO_ATENDIDOS        61 non-null int64  
dtypes: int64(3), object(2)  
memory usage: 2.5+ KB
```

## PRONOSTICA UN AÑO DE ATENCION (365) DÍAS

In [ ]:

```
1
```