Evidencia Final "El arte de la analítica"

Equipo 6:

A01706095 - Naomi Estefanía Nieto Vega

A01706189 - Alejandro Angel Calderon Berges

A01706596 - Carlos Soria de la Cabada

Entregable 1 - Obtención de Estadísticas Descriptivas

Paso 1. Cargar los datos de nuestra base de datos con ayuda de pandas

En esta sección únicamente se leen los datos, como podemos ver es acerca de unos artículos publicados en la web.

In [7]:
```
import pandas as pd
data = pd.read_csv("heart.csv")
```

In [8]:
```
data
```

Out[8]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	targe
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	.
298	57	0	0	140	241	0	1	123	1	0.2	1	0	3	
299	45	1	3	110	264	0	1	132	0	1.2	1	0	3	
300	68	1	0	144	193	1	1	141	0	3.4	1	2	3	
301	57	1	0	130	131	0	1	115	1	1.2	1	1	3	
302	57	0	1	130	236	0	0	174	0	0.0	1	1	2	

303 rows × 14 columns

Significado de las variables:

- age - edad en años
- sex - (1 = hombre; 0 = mujer)
- cp - tipo de dolor en el pecho previo al infarto
- trestbps - presión sanguínea en mmHg
- chol - colesterol en sangre en mg/dl
- fbs - azucar en sangre mayor a 120 mg/dl (1 = si; 0 = no)

- restecg - resultados de electrocardiograma
- thalach - máxima frecuencia cardiaca
- oldpeak - depresión del segmento ST
- slope - pendiente del segmento ST
- target - tuvo infarto o no (1=si, 0=no)

In [9]:
```
import pandas as pd #Aqui importamos la libreria
import csv

df=pd.read_csv("heart.csv") #Aqui leemos el archivo
df.describe() #Aqui le decimos que nos muestre los analiticos en la tabla de aba
```

Out[9]:

	age	sex	cp	trestbps	chol	fbs	restecg
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000

Paso 2. Verificar la cantidad de datos, las variables que contiene cada vector de datos e identifica el tipo de variables.

En general tenemos 302 filas con datos, de los cuales existen las variables o columas de edad, sexo, tipo de dolor de pecho, colesterol, azúcar en sangre, resultados de electrocardiograma, pulso máximo, ejercicio, depresión del segmento ST y objetivo.

El tipo de las variables es float y entero en su mayoría

Paso 3. Analizar las variables para saber que representa cada una y en que rangos se encuentran.

En este caso las variables como ya mencioné antes representan una variable de las veces que ha ocurrido un evento, en este caso relacionado con los infartos, para esto se analizan los datos relevantes como la frecuencia, el colesterol, azúcar en sangre, entre otros factores que nos permiten predecir cuando alguien podría tener un infarto basado en los datos analizados.

In [16]:
```
data = [['age',29.000000,77.000000,303.000000,54.366337,9.082101],
        ['sex',0.000000,1.000000,303.000000,0.683168,0.466011],
        ['cp',0.000000,3.000000,303.000000,0.966997,1.032052],
        ['trestbps',94.000000,200.000000,303.000000,0.966997,1.032052],
        ['chol',126.000000,564.124224,303.000000,246.264026,51.830751],
        ['fbs',0.000000,1.000000,303.000000,0.148515,0.356198],
        ['resteccg',0.000000,2.000000,303.000000,0.528053,0.525860],
        ['thalach',71.000000,202.000000,303.000000,149.646865,22.905161],
        ['exang',0.000000,1.000000,303.000000,0.326733,0.469794],
        ['oldpeak',0.000000,6.2000000,303.000000,1.039604,1.161075],
        ['slope',0.000000,2.000000,303.000000,1.399340,0.616226],
        ['ca',0.000000,4.000000,303.000000,0.729373,1.022606],
```

```
        ['thal',0.000000,3.000000,303.000000,2.313531,0.612277],
        ['target',0.000000,1.000000,303.000000,0.544554,0.498835]]

df = pd.DataFrame(data, columns = ['Variables', 'Min Range','Max Range','Count',
print(df)
```

```
   Variables  Min Range   Max Range  Count        Mean        std
0        age       29.0   77.000000  303.0   54.366337   9.082101
1        sex        0.0    1.000000  303.0    0.683168   0.466011
2         cp        0.0    3.000000  303.0    0.966997   1.032052
3   trestbps       94.0  200.000000  303.0    0.966997   1.032052
4       chol      126.0  564.124224  303.0  246.264026  51.830751
5        fbs        0.0    1.000000  303.0    0.148515   0.356198
6   resteccg        0.0    2.000000  303.0    0.528053   0.525860
7    thalach       71.0  202.000000  303.0  149.646865  22.905161
8      exang        0.0    1.000000  303.0    0.326733   0.469794
9    oldpeak        0.0    6.200000  303.0    1.039604   1.161075
10     slope        0.0    2.000000  303.0    1.399340   0.616226
11        ca        0.0    4.000000  303.0    0.729373   1.022606
12      thal        0.0    3.000000  303.0    2.313531   0.612277
13    target        0.0    1.000000  303.0    0.544554   0.498835
```

Paso 4. Conclusiones

Como conclusión de acuerdo a las variables podemos asumir que hay un promedio de cantidad de veces que se comparte un artículo bastante alto, siendo este de 27,948; esto podría decirnos que a la mayoría de los usuarios que interactuan les gusta ese artículo. Además es importante observar que tenemos un número alto en la desviación estándar esto quiere decir que existe una gran dispersión en la población de los datos con respecto a la media. La cuenta en general para las variables es de 161, a excepción de la cantidad de comentarios que es de 129. Asimismo tenemos una media aritmética de 1808, esto es la cantidad promedio de palabras que están escritas en los artículos.

Entregable 2. Mapas de calor y boxplots

Paso 1. Cargar los datos usando tu lector de csv o con pandas.

In [17]:
```
import pandas as pd

df = pd.read_csv('heart.csv')
df.head()
```

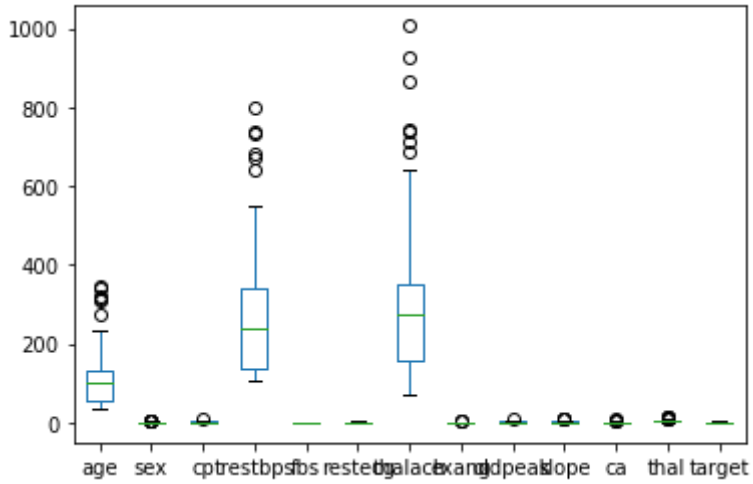
Out[17]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

a) Diagrama de cajas y bigotes

In [22]:
```
import matplotlib.pyplot as plit
import seaborn as sns
df.groupby('chol').sum().plot(kind='box',legend='Reverse')
```

Out[22]: <AxesSubplot:>



En esta gráfica podemos ver el diagrama un poco disperso únicamente en 3 variables porque son las que cuentan con números más grandes o mayor variedad que se puede observar al graficarse a diferencia de las variables que tienen solo datos booleanos no hay mucho que observar. De acuerdo con esto las variables más significativas son la edad, la presión sanguínea y la frecuencia cardiaca.

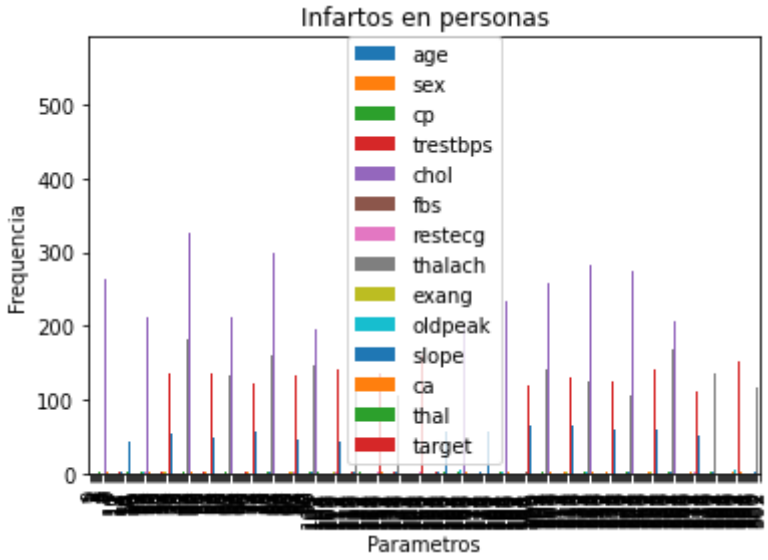
b) Histogramas

En esta sección podemos ver el histograma, que nos muestra la frecuencia con la que ciertos hechos sucedem en este caso la frecuencia de las variables relevantes para los infartos.

In [24]:
```
import matplotlib.pyplot as plt
import pandas as pd

df.plot(kind = 'bar')
plt.ylabel('Frequencia')
plt.xlabel('Parametros')
plt.title('Infartos en personas')

plt.show()
```

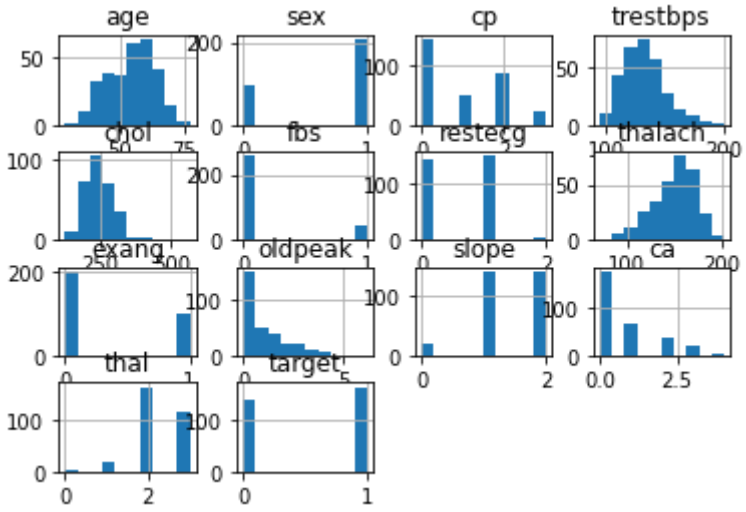


In [25]:
```
import pandas as pd
```

```
datos = pd.read_csv('heart.csv')
df = pd.DataFrame(datos)

data.drop ([0,1]).hist()
```

Out[25]:
```
array([[<AxesSubplot:title={'center':'age'}>,
        <AxesSubplot:title={'center':'sex'}>,
        <AxesSubplot:title={'center':'cp'}>,
        <AxesSubplot:title={'center':'trestbps'}>],
       [<AxesSubplot:title={'center':'chol'}>,
        <AxesSubplot:title={'center':'fbs'}>,
        <AxesSubplot:title={'center':'restecg'}>,
        <AxesSubplot:title={'center':'thalach'}>],
       [<AxesSubplot:title={'center':'exang'}>,
        <AxesSubplot:title={'center':'oldpeak'}>,
        <AxesSubplot:title={'center':'slope'}>,
        <AxesSubplot:title={'center':'ca'}>],
       [<AxesSubplot:title={'center':'thal'}>,
        <AxesSubplot:title={'center':'target'}>, <AxesSubplot:>,
        <AxesSubplot:>]], dtype=object)
```

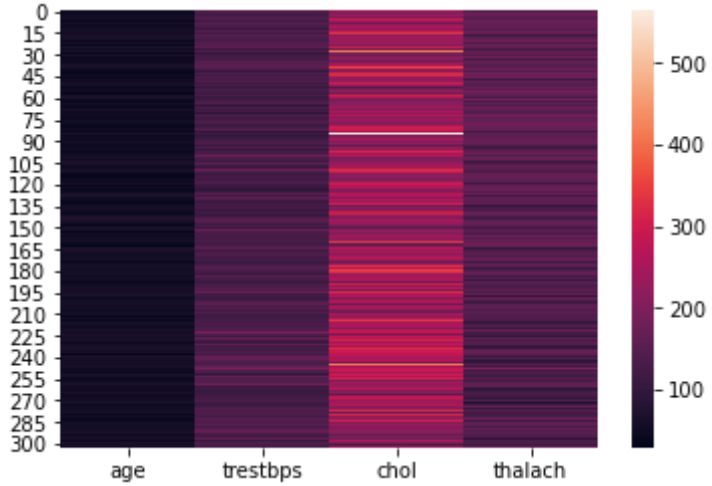


3.- Mapas de calor

El mapa de color es una ayuda visual, que nos permite ver en conjunto los valores que estan asociados con colores para que sea más perceptible notar patrones. De este se eliminaron los datos con valores mínimos o booleanos ya que no mostraban un cambio relevante en la gráfica final por lo que en la línea 6 se excluyen.

In [28]:
```
import pandas as pd
import seaborn as sb
import matplotlib.pyplot as plt
data = pd.read_csv('heart.csv')

uniform_data = data.drop(['sex','cp','fbs','restecg','exang','oldpeak','slope','
ax = sb.heatmap(uniform_data)
```



Conclusiones

1. ¿Hay alguna variable que no aporta información?

Si, existen variables que no se consideran muy relevantes por el tipo de dato que tienen en sus valores, esto nos impide analizarlo de esta forma y lo que debemos hacer para considerarlo es el proceso de normalización de la información para tener una base de datos limpia, sin valores extraños o valores NULL.

2. Si tuvieras que eliminar variables, ¿cuáles quitarías y por qué?

Quitaría las variables de resultados de electrocardiograma, depresión y pendiente del segmento ST del ciclo cardiaco de la persona. Las borraría porque considero que los datos que aportan son muy mínimos y no tan necesarios para el alcance de análisis que tendrá nuestro proyecto en este momento.

3. ¿Existen variables que tengan datos extraños?

No existen valores con caracteres extraños, unicamente los booleanos que nos complican un poco el análisis pero fuera de eso al momento de buscar la base de datos intentamos que tuviera mayormente valores númericos. Las variables con valores bool son fbs, target y el sexo.

4. Si comparas las variables, ¿todas están en rangos similares? ¿Crees que esto afecte?

No todas están en rangos similares pero hay algunas que si comparten similitudes como la presión sanguínea o el colesterol en sangre ya que estos son indicios principales que nos dan una posible alerta de que esa persona puede sufrir un infarto. El hecho de que estén en rangos similares si afecta un poco ya que al estar los datos tan parecidos y no tener variedad es posible que se convierta en un patrón repetitivo de datos y esto nos limita en cuanto a predicciones.

5. ¿Puedes encontrar grupos qué se parezcan? ¿Qué grupos son estos?

Sí, como ya mencioné antes la presión sanguínea en milimetros de mercurio de las personas y el el colesterol en sangre.

Entregable 3. Patrones con K-means

In [39]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
import sklearn
from sklearn.cluster import KMeans
from mpl_toolkits.mplot3d import Axes3D
from sklearn.preprocessing import scale
import sklearn.metrics as sm
from sklearn import datasets
from sklearn.metrics import confusion_matrix, classification_report

df=pd.read_csv("heart.csv") #Aqui leemos el archivo

countFemale = len(df[df.sex == 0])
countMale = len(df[df.sex == 1])
```

In [45]:
```
df=pd.read_csv("heart.csv")
X = scale(df)
y= pd.DataFrame(df)
variable_names = df
X[0:10,]
```

Out[45]:
```
array([[ 0.9521966 ,  0.68100522,  1.97312292,  0.76395577, -0.25633371,
         2.394438  , -1.00583187,  0.01544279, -0.69663055,  1.08733806,
        -2.27457861, -0.71442887, -2.14887271,  0.91452919],
       [-1.91531289,  0.68100522,  1.00257707, -0.09273778,  0.07219949,
        -0.41763453,  0.89896224,  1.63347147, -0.69663055,  2.12257273,
        -2.27457861, -0.71442887, -0.51292188,  0.91452919],
       [-1.47415758, -1.46841752,  0.03203122, -0.09273778, -0.81677269,
        -0.41763453, -1.00583187,  0.97751389, -0.69663055,  0.31091206,
         0.97635214, -0.71442887, -0.51292188,  0.91452919],
       [ 0.18017482,  0.68100522,  0.03203122, -0.66386682, -0.19835726,
        -0.41763453,  0.89896224,  1.23989692, -0.69663055, -0.20670527,
         0.97635214, -0.71442887, -0.51292188,  0.91452919],
       [ 0.29046364, -1.46841752, -0.93851463, -0.66386682,  2.08204965,
        -0.41763453,  0.89896224,  0.58393935,  1.43548113, -0.37924438,
         0.97635214, -0.71442887, -0.51292188,  0.91452919],
       [ 0.29046364,  0.68100522, -0.93851463,  0.47839125, -1.04867848,
        -0.41763453,  0.89896224, -0.07201822, -0.69663055, -0.55178349,
        -0.64911323, -0.71442887, -2.14887271,  0.91452919],
       [ 0.18017482, -1.46841752,  0.03203122,  0.47839125,  0.92252071,
        -0.41763453, -1.00583187,  0.1466343 , -0.69663055,  0.22464251,
        -0.64911323, -0.71442887, -0.51292188,  0.91452919],
       [-1.1432911 ,  0.68100522,  0.03203122, -0.66386682,  0.32343076,
        -0.41763453,  0.89896224,  1.0212444 , -0.69663055, -0.89686172,
         0.97635214, -0.71442887,  1.12302895,  0.91452919],
       [-0.26098049,  0.68100522,  1.00257707,  2.30600417, -0.91340011,
         2.394438  ,  0.89896224,  0.54020884, -0.69663055, -0.46551394,
         0.97635214, -0.71442887,  1.12302895,  0.91452919],
       [ 0.29046364,  0.68100522,  1.00257707,  1.04952029, -1.51249006,
        -0.41763453,  0.89896224,  1.0649749 , -0.69663055,  0.48345117,
         0.97635214, -0.71442887, -0.51292188,  0.91452919]])
```

In [46]:
```
clustering = KMeans(n_clusters=3, random_state = 5)
clustering.fit(X)
```

Out[46]:
```
KMeans(n_clusters=3, random_state=5)
```

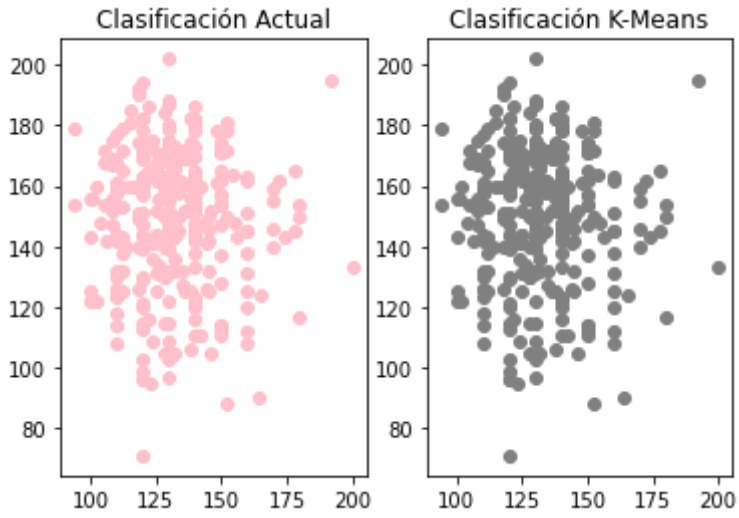
In [52]:
```
data_df = pd.DataFrame(df)
data_df.columns=['age', 'sex', 'cp', 'trestbps','chol','fbs','resteccg','thalach
#y.columns=['index']
```

In [72]:
```
import matplotlib.pyplot as plt
color_theme = np.array(['darkgray', 'lightsalmon','powderblue'])
```

```
plt.subplot(1,2,1)
plt.scatter(x=data_df.trestbps, y=data_df.thalach, c="pink")
plt.title("Clasificación Actual")

plt.subplot(1,2,2)
plt.scatter(x=data_df.trestbps, y=data_df.thalach, c="gray")
plt.title("Clasificación K-Means")
```

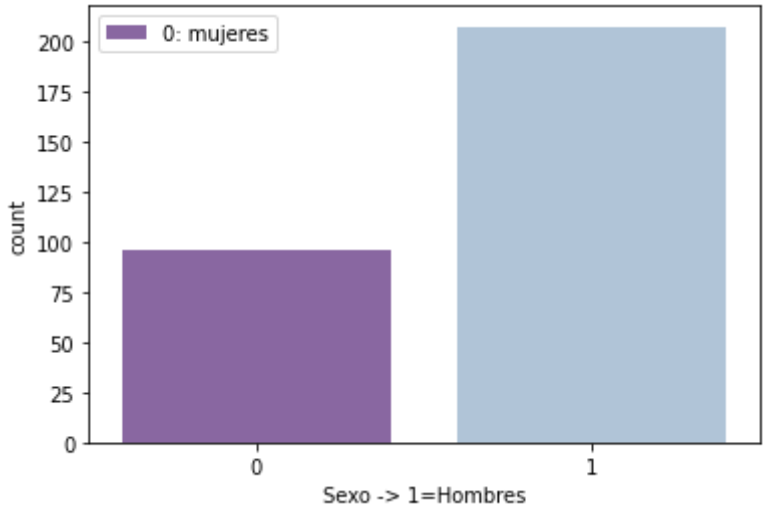
Out[72]: Text(0.5, 1.0, 'Clasificación K-Means')



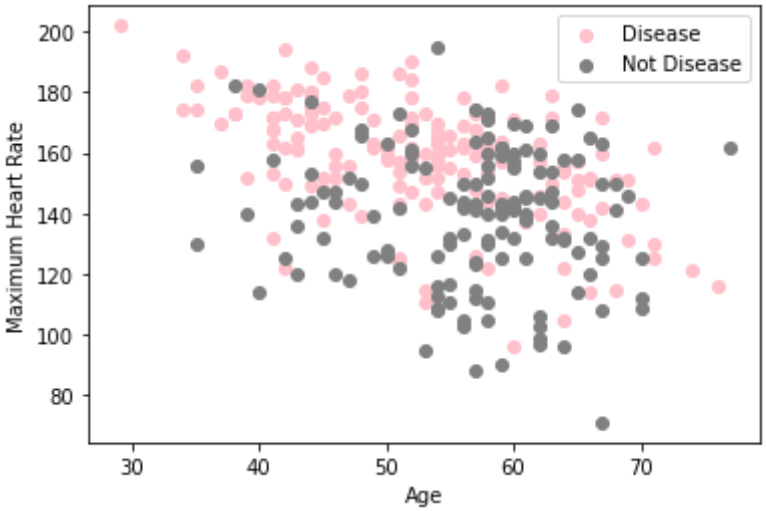
In [67]:
```
print("Porcentaje de pacientes mujeres: {:.2f}%".format((countFemale / (len(df.s
print("Porcentaje de pacientes hombres: {:.2f}%".format((countMale / (len(df.sex
```

```
Porcentaje de pacientes mujeres: 31.68%
Porcentaje de pacientes hombres: 68.32%
```

In [91]:
```
sns.countplot(x='sex', data=df, palette="BuPu_r")
plt.xlabel("Sexo -> 1=Hombres")
plt.legend(["0: mujeres"])
plt.show()
```



In [75]:
```
plt.scatter(x=df.age[df.target==1], y=df.thalach[(df.target==1)], c="pink")
plt.scatter(x=df.age[df.target==0], y=df.thalach[(df.target==0)], c="gray")
plt.legend(["Disease", "Not Disease"])
plt.xlabel("Age")
plt.ylabel("Maximum Heart Rate")
plt.show()
```



Repositorio en GitHub

Link del repositorio : https://github.com/naominietov/TC1002600/tree/SemanaTec6