Coronary Artery Disease (CAD) -- Data Analysis

https://archive.ics.uci.edu/ml/datasets/Heart+Disease

In [ ]:
```
#from IPython.display import Image
#Image("../input/heart-pics/Heart_pumping1.gif")
```

In [ ]:
```
#from IPython.display import Image
#Image("../input/heart-pics/Heart_pumping2.gif")
```

Pics Link: https://en.wikipedia.org/wiki/Heart_valve

In [1]:
```
import pandas as pd
import matplotlib
import matplotlib.pyplot as plt
import matplotlib.cm as cm
import seaborn as sns

%matplotlib inline
```

In [2]:
```
cleveland_df = pd.read_csv('heart.csv')
```

In [3]:
```
cleveland_df.head()
```

Out[3]:

	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

Dataset dimensionality

In [4]:
```
cleveland_df.shape
```

Out[4]:
```
(303, 14)
```

In [7]:
```
cleveland_df.age.max()
```

Out[7]:
```
77
```

Assigning feature names to the dataset

In [5]:
```
features = ['age','gender','chest_pain','rest_bp','cholestrol','fst_bs','rest_ecg','
```

In [6]:
```
cleveland_df.columns = features
```

In [7]:
```
cleveland_df.head()
```

Out[7]:

	age	gender	chest_pain	rest_bp	cholestrol	fst_bs	rest_ecg	max_hrt_rate	ex_angina	oldpeak
0	63	1	3	145	233	1	0	150	0	2.3

	age	gender	chest_pain	rest_bp	cholestrol	fst_bs	rest_ecg	max_hrt_rate	ex_angina	oldpeak
1	37	1	2	130	250	0	1	187	0	3.5
2	41	0	1	130	204	0	0	172	0	1.4
3	56	1	1	120	236	0	1	178	0	0.8
4	57	0	0	120	354	0	1	163	1	0.6

Data Pre-processing

Step-1: Class handling in 'num'

Handling the multivariate predicted attribute i.e. 'num', in which records with values >1 are effected with CAD and <1 are non-CAD.

Checking the counts before applying any operation

In [8]:
```
pd.DataFrame(cleveland_df['num'].value_counts())
```

Out[8]:

	num
1	165
0	138

Finding the count of records with blockage > 50%

In [9]:
```
cleveland_df[cleveland_df['num'] > 0]['num'].count()
```

Out[9]: 165

Categorizing the 'num' feature into two classes 0(Non-CAD i.e. blockage < 50%) and 1(CAD i.e. blockage >50%)

In [10]:
```
cleveland_df['result'] = cleveland_df['num'].apply(lambda val : val if val == 0 else
```

In [11]:
```
cleveland_df.head()
```

Out[11]:

	age	gender	chest_pain	rest_bp	cholestrol	fst_bs	rest_ecg	max_hrt_rate	ex_angina	oldpeak
0	63	1	3	145	233	1	0	150	0	2.3
1	37	1	2	130	250	0	1	187	0	3.5
2	41	0	1	130	204	0	0	172	0	1.4
3	56	1	1	120	236	0	1	178	0	0.8
4	57	0	0	120	354	0	1	163	1	0.6

In [12]:
```
cleveland_df['result'].value_counts()
```

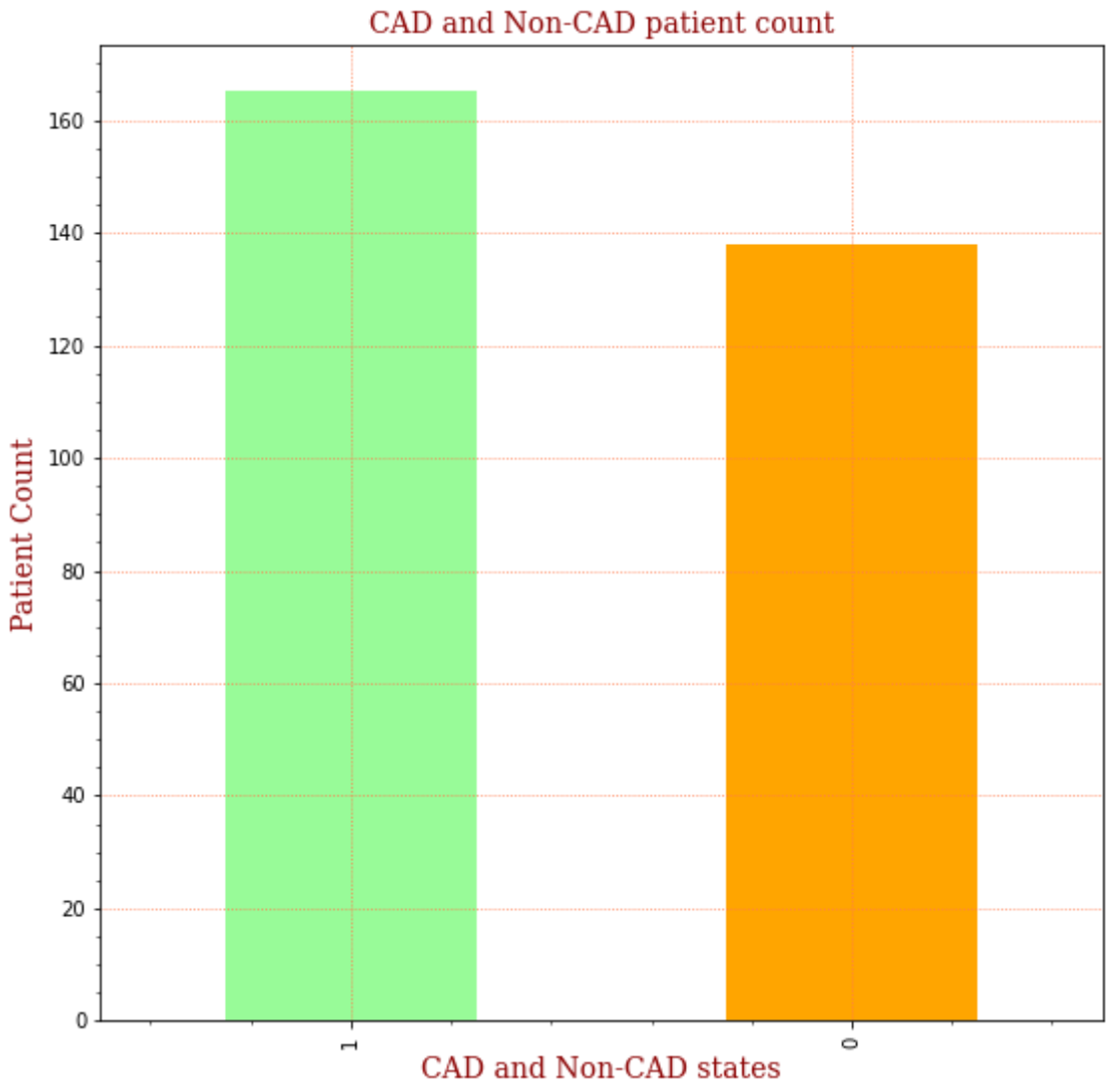
Out[12]:
```
1    165
0    138
Name: result, dtype: int64
```

Visualize the CAD and Non-CAD records

In [13]:
```
bar_font = {'family': 'serif',
        'color':  'darkred',
        'weight': 'normal',
        'size': 14,
        }
```

In [14]:
```
cleveland_df['result'].value_counts().plot(kind='bar', figsize=(9,9), color= ['paleg
plt.minorticks_on()
plt.grid(which='major',color='coral',linestyle=':')
plt.xlabel('CAD and Non-CAD states',fontdict=bar_font)
plt.ylabel('Patient Count',fontdict=bar_font)
plt.title('CAD and Non-CAD patient count',fontdict=bar_font)
```

Out[14]:
```
Text(0.5, 1.0, 'CAD and Non-CAD patient count')
```



Finding the missing values

As missing values are marked as '?'. So replacing such values with None.

In [15]:
```
cleveland_df = cleveland_df.applymap(lambda val : None if val == '?' else val)
```

Counting the missing values

In [16]:
```
cleveland_df.isnull().sum()
```

Out[16]:
```
age             0
```

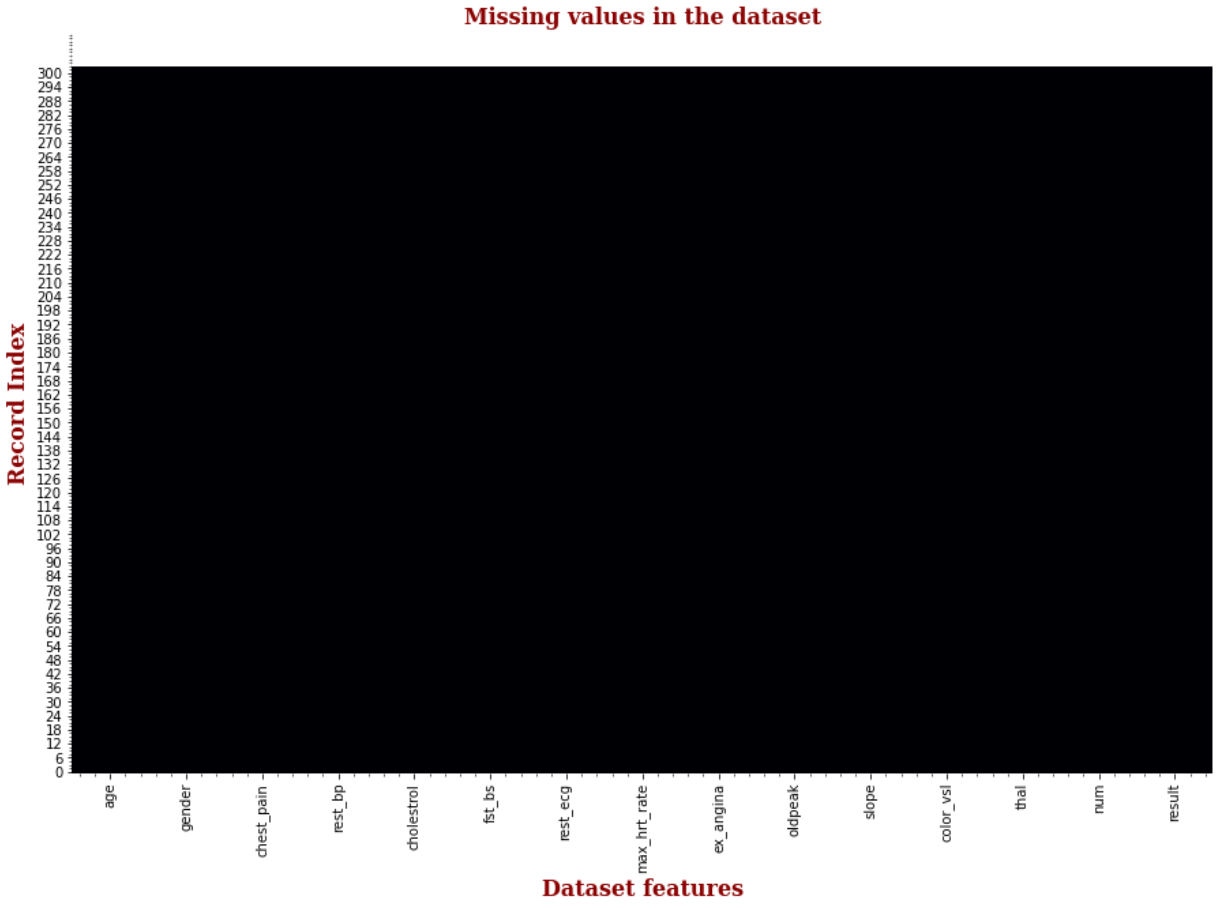
```
gender          0
chest_pain      0
rest_bp         0
cholestrol      0
fst_bs          0
rest_ecg        0
max_hrt_rate    0
ex_angina       0
oldpeak         0
slope           0
color_vsl       0
thal            0
num             0
result          0
dtype: int64
```

Visualize the missing records

In [17]:
```
font = {'family': 'serif',
        'color':  'darkred',
        'weight': 'bold',
        'size': 16,
        }
```

In [18]:
```
plt.figure(figsize = (15,10))
sns.heatmap(cleveland_df.isnull(),cbar=False,cmap='inferno')
plt.axis(ymin=0,ymax=318)
plt.minorticks_on()
plt.xlabel("Dataset features",fontdict=font)
plt.ylabel("Record Index",fontdict=font)
plt.title("Missing values in the dataset",fontdict=font)
```

Out[18]:
```
Text(0.5, 1.0, 'Missing values in the dataset')
```



Records with NULL Color_vsl or ca

In [19]:
```
cleveland_df[cleveland_df['color_vsl'].isnull()]
```

Out[19]:

age	gender	chest_pain	rest_bp	cholestrol	fst_bs	rest_ecg	max_hrt_rate	ex_angina	oldpeak

Count the records for color_vsl categories

In [20]:
```
cleveland_df['color_vsl'].value_counts()
```

Out[20]:
```
0    175
1     65
2     38
3     20
4      5
Name: color_vsl, dtype: int64
```

Count the CAD and Non-CAD records for every color_vsl category

In [21]:
```
pd.DataFrame(cleveland_df.groupby(['color_vsl','result'])['result'].count())
```

Out[21]:

		result
color_vsl	result	
0	0	45
	1	130
1	0	44
	1	21
2	0	31
	1	7
3	0	17
	1	3
4	0	1
	1	4

Filling the missing values in COLOR_VSL feature

Replacing NULL with MAX occurence of respective feature class based on TARGET column

In [22]:
```
cleveland_df['fix_color_vsl'] = cleveland_df['color_vsl'].fillna(value='0.0')
```

In [23]:
```
cleveland_df['fix_color_vsl'].value_counts()
```

Out[23]:
```
0    175
1     65
2     38
3     20
4      5
Name: fix_color_vsl, dtype: int64
```

Records with NULL THAL or THALASSEMIA

In [24]:
```
cleveland_df[cleveland_df['thal'].isnull()]
```

Out[24]:

age	gender	chest_pain	rest_bp	cholestrol	fst_bs	rest_ecg	max_hrt_rate	ex_angina	oldpeak

Count the records for thal categories

In [25]:
```
cleveland_df['thal'].value_counts()
```

Out[25]:
```
2    166
3    117
1     18
0      2
Name: thal, dtype: int64
```

Count the CAD and Non-CAD records for every thal category

In [26]:
```
thal_missing_val = pd.DataFrame(cleveland_df.groupby(['thal','result'])['result'].co
```

In [27]:
```
thal_missing_val.index.names = ['thal', 'result1']
```

In [28]:
```
thal_missing_val.sort_values(['thal'],ascending=True)
```

Out[28]:

		result
thal	**result1**	
0	**0**	1
	1	1
1	**0**	12
	1	6
2	**0**	36
	1	130
3	**0**	89
	1	28

Filling the missing values in THAL feature

Replacing NULL with MAX occurence of respective feature class based on TARGET column

In [29]:
```
cleveland_df[(cleveland_df['result'] == 0) & (cleveland_df['thal'].isna())]
```

Out[29]:

age	gender	chest_pain	rest_bp	cholestrol	fst_bs	rest_ecg	max_hrt_rate	ex_angina	oldpeak

In [30]:
```
cleveland_df['result'].dtype
```

Out[30]:
```
dtype('int64')
```

In [31]:
```
cleveland_df['fix_thal'] = cleveland_df[['thal','result']].apply(lambda val : '7.0'
                                                                 else '3.0' if val['
```

In [32]:
```
cleveland_df['thal'].value_counts()
```

Out[32]:
```
2    166
3    117
```

```
1     18
0      2
Name: thal, dtype: int64
```

In [33]:
```
cleveland_df['fix_thal'].value_counts()
```

Out[33]:
```
2    166
3    117
1     18
0      2
Name: fix_thal, dtype: int64
```

Visualizing missing values again

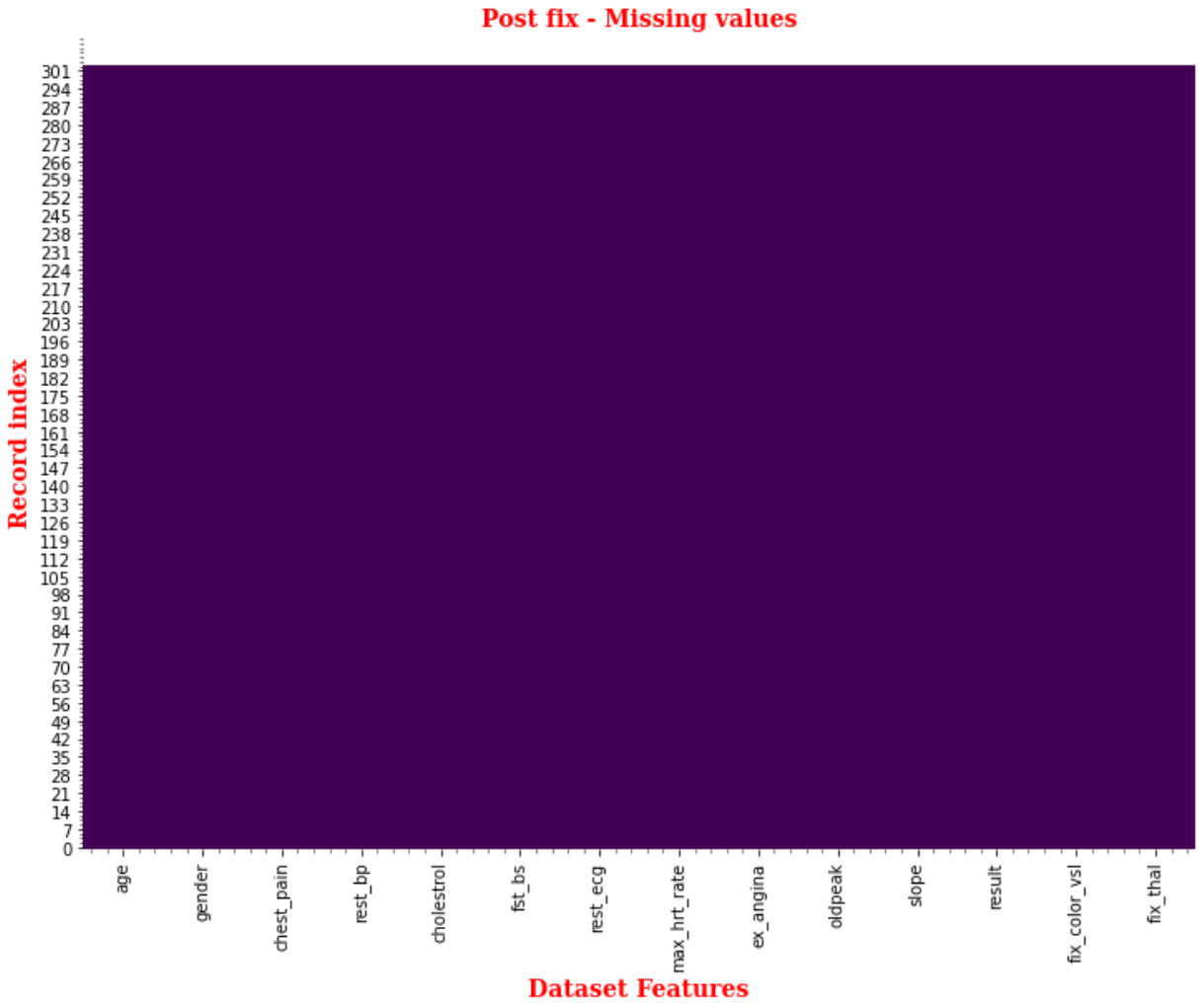
In [34]:
```
cleveland_df.columns
```

Out[34]:
```
Index(['age', 'gender', 'chest_pain', 'rest_bp', 'cholestrol', 'fst_bs',
       'rest_ecg', 'max_hrt_rate', 'ex_angina', 'oldpeak', 'slope',
       'color_vsl', 'thal', 'num', 'result', 'fix_color_vsl', 'fix_thal'],
      dtype='object')
```

In [35]:
```
missing_val_font = {'family':'serif',
                    'weight': 'bold',
                    'size': 14,
                    'color':'red'}
```

In [36]:
```
plt.figure(figsize=(12,9))
sns.heatmap(cleveland_df[['age', 'gender', 'chest_pain', 'rest_bp', 'cholestrol', 'f
       'rest_ecg', 'max_hrt_rate', 'ex_angina', 'oldpeak', 'slope', 'result', 'fix_c
plt.minorticks_on()
plt.axis(ymin=0,ymax=315)
plt.xlabel("Dataset Features",fontdict=missing_val_font)
plt.ylabel("Record index",fontdict=missing_val_font)
plt.title("Post fix - Missing values",fontdict=missing_val_font)
```

Out[36]:
```
Text(0.5, 1.0, 'Post fix - Missing values')
```



Datatype handling

Several columns are having definite values but are of float datatype

In [37]:
```
cleveland_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 17 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   age            303 non-null    int64  
 1   gender         303 non-null    int64  
 2   chest_pain     303 non-null    int64  
 3   rest_bp        303 non-null    int64  
 4   cholestrol     303 non-null    int64  
 5   fst_bs         303 non-null    int64  
 6   rest_ecg       303 non-null    int64  
 7   max_hrt_rate   303 non-null    int64  
 8   ex_angina      303 non-null    int64  
 9   oldpeak        303 non-null    float64
 10  slope          303 non-null    int64  
 11  color_vsl      303 non-null    int64  
 12  thal           303 non-null    int64  
 13  num            303 non-null    int64  
 14  result         303 non-null    int64  
 15  fix_color_vsl  303 non-null    int64  
 16  fix_thal       303 non-null    int64  
dtypes: float64(1), int64(16)
memory usage: 40.4 KB
```

Created a UDF for converting the datatypes of required columns

In [38]:
```
def handle_datatype(df_name,unchange_col=None):
    """
    Description: This function will change the datatype of the features in the datas

    Input parameter:
    *df_name*: It will only accept the DataFrame object.
    *unchange_col*: This is the column for which you don't want to change the dataty

    Return:
    It will returned the modified DataFrame object.
    """
    cols = ['age', 'gender', 'chest_pain', 'rest_bp', 'cholestrol', 'fst_bs',
            'rest_ecg', 'max_hrt_rate', 'ex_angina', 'oldpeak','slope', 'result', 'f
    for col in cols:
        if col != unchange_col:
            df_name[col] = df_name[col].astype('float')
            df_name[col] = df_name[col].astype('int')
    return df_name
```

In [39]:
```
handle_datatype(cleveland_df,'oldpeak')
```

Out[39]:

	age	gender	chest_pain	rest_bp	cholestrol	fst_bs	rest_ecg	max_hrt_rate	ex_angina	oldpea
0	63	1	3	145	233	1	0	150	0	2
1	37	1	2	130	250	0	1	187	0	3
2	41	0	1	130	204	0	0	172	0	1
3	56	1	1	120	236	0	1	178	0	0
4	57	0	0	120	354	0	1	163	1	0
...	...	...	...	...	...	...	...	...	...	
298	57	0	0	140	241	0	1	123	1	0
299	45	1	3	110	264	0	1	132	0	1
300	68	1	0	144	193	1	1	141	0	3
301	57	1	0	130	131	0	1	115	1	1
302	57	0	1	130	236	0	0	174	0	0

303 rows × 17 columns

Drop the non-required column

In [40]:
```
cleveland_df.drop(['color_vsl','thal','num'],axis=1,inplace=True)
```

Exploratory Data Analysis

Question-1: How many people of age group 29-48 have blockage greater than 50%?

In [41]:
```
age_grp_29_48 = pd.concat([cleveland_df[(cleveland_df['age'] >= 29.0) & (cleveland_d
           pd.DataFrame({'color':['palegreen','orange']})],axis=1)
```

In [42]:
```
age_grp_29_48.reset_index(inplace=True)
```

In [43]:
```
age_grp_29_48.columns = ['result','age','color']
```

In [44]:
```
age_grp_29_48
```

Out[44]:

	result	age	color
0	0	21	palegreen
1	1	55	orange

In [45]:
```
label_style={'family':'serif','color':'red','size':16}
age_grp_29_48.plot(kind='bar',x='result',y='age',figsize=(8,8),color=age_grp_29_48['
plt.minorticks_on()
plt.grid(which='major',linestyle=':',color='coral')
plt.xlabel('Heart disease result',fontdict=label_style)
plt.ylabel('Number of people',fontdict=label_style)
plt.title('Heart Disease Result of people from age group 29 - 48',fontdict=label_sty
```

Out[45]:
```
Text(0.5, 1.0, 'Heart Disease Result of people from age group 29 - 48')
```

Question-2: How many people of age group 48-56 have blockage greater than 50%?

In [46]:
```
age_grp_48_56 = pd.concat([cleveland_df[(cleveland_df['age'] >= 48.0) & (cleveland_d
           pd.DataFrame({'color':['palegreen','orange']})],axis=1)

age_grp_48_56.reset_index(inplace=True)
```

```
age_grp_48_56.columns = ['result','age','color']

age_grp_48_56
```

Out[46]:

	result	age	color
0	0	28	palegreen
1	1	48	orange

In [47]:

```
label_style={'family':'serif','color':'red','size':16}
age_grp_48_56.plot(kind='bar',x='result',y='age',figsize=(8,8),color=age_grp_48_56['
plt.minorticks_on()
plt.grid(which='major',linestyle=':',color='coral')
plt.xlabel('Heart disease result',fontdict=label_style)
plt.ylabel('Number of people',fontdict=label_style)
plt.title('Heart Disease Result of people from age group 48 - 56',fontdict=label_sty
```

Out[47]: Text(0.5, 1.0, 'Heart Disease Result of people from age group 48 - 56')

Question-3: How many people of age group 56-77 have blockage greater than 50%?

In [48]:

```
age_grp_56_77 = pd.concat([cleveland_df[(cleveland_df['age'] >= 56.0)][['age','resul
           pd.DataFrame({'color':['palegreen','orange']})],axis=1)

age_grp_56_77.reset_index(inplace=True)

age_grp_56_77.columns = ['result','age','color']
```

```
age_grp_56_77
```

Out[48]:

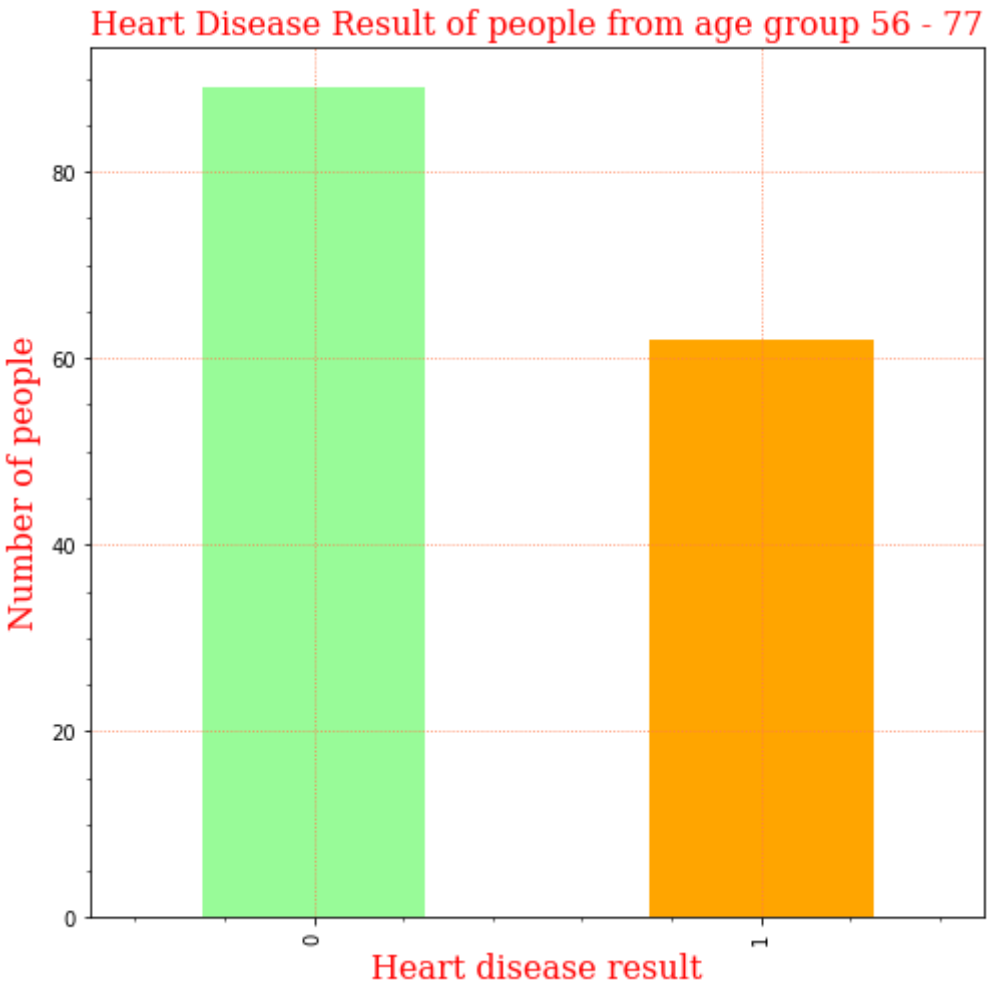
	result	age	color
0	0	89	palegreen
1	1	62	orange

In [49]:

```
label_style={'family':'serif','color':'red','size':16}
age_grp_56_77.plot(kind='bar',x='result',y='age',figsize=(8,8),color=age_grp_56_77['
plt.minorticks_on()
plt.grid(which='major',linestyle=':',color='coral')
plt.xlabel('Heart disease result',fontdict=label_style)
plt.ylabel('Number of people',fontdict=label_style)
plt.title('Heart Disease Result of people from age group 56 - 77',fontdict=label_sty
```

Out[49]:

```
Text(0.5, 1.0, 'Heart Disease Result of people from age group 56 - 77')
```



Question-4: How many MALE and FEMALE have heart disease?

In [50]:

```
pd.concat([pd.DataFrame(cleveland_df['gender'].value_counts()),pd.DataFrame({'gender
```

Out[50]:

	gender	gender_name
0	96	female
1	207	male

In [51]:
```
gender_dist = pd.DataFrame(cleveland_df.groupby(by=['gender','result'],axis=0)['age'

gender_dist.columns = ['Count of people']
gender_dist.index.names = ['Gender(0:Female,1:Male)','Heart Disease Result']
```

In [52]:
```
gender_dist
```

Out[52]:

		Count of people
Gender(0:Female,1:Male)	**Heart Disease Result**	
0	**0**	24
	1	72
1	**0**	114
	1	93

Question-5: How many patients suffered from various CHEST PAINS?

In [53]:
```
chest_pain_dist = pd.DataFrame(cleveland_df.groupby(by=['gender','result','chest_pai
```

In [54]:
```
chest_pain_dist.columns = ['Patient Count']
chest_pain_dist.index.names = ['Gender(0:Female,1:Male)','Heart Disease Result','Che
```

In [55]:
```
chest_pain_dist
```

Out[55]:

			Patient Count
Gender(0:Female,1:Male)	**Heart Disease Result**	**Chest Pain Type**	
0	**0**	**0**	21
		1	2
		2	1
	1	**0**	18
		1	16
		2	34
		3	4
1	**0**	**0**	83
		1	7
		2	17
		3	7
	1	**0**	21
		1	25
		2	35
		3	12

Question-6: Does high blood pressure at rest

corresponds to a CAD?

In [56]:
```
cleveland_df['rest_bp'].describe()
```

Out[56]:
```
count    303.000000
mean     131.623762
std       17.538143
min       94.000000
25%      120.000000
50%      130.000000
75%      140.000000
max      200.000000
Name: rest_bp, dtype: float64
```

BP Group1: (94-120]

In [57]:
```
rest_bp_94_120 = pd.DataFrame(cleveland_df[(cleveland_df['rest_bp'] >= 94.0) & (clev
rest_bp_94_120.reset_index(inplace=True)
rest_bp_94_120.columns = ['CAD Result','Patient_Count']
rest_bp_94_120['color'] = rest_bp_94_120['CAD Result'].apply(lambda val : 'palegreen
```

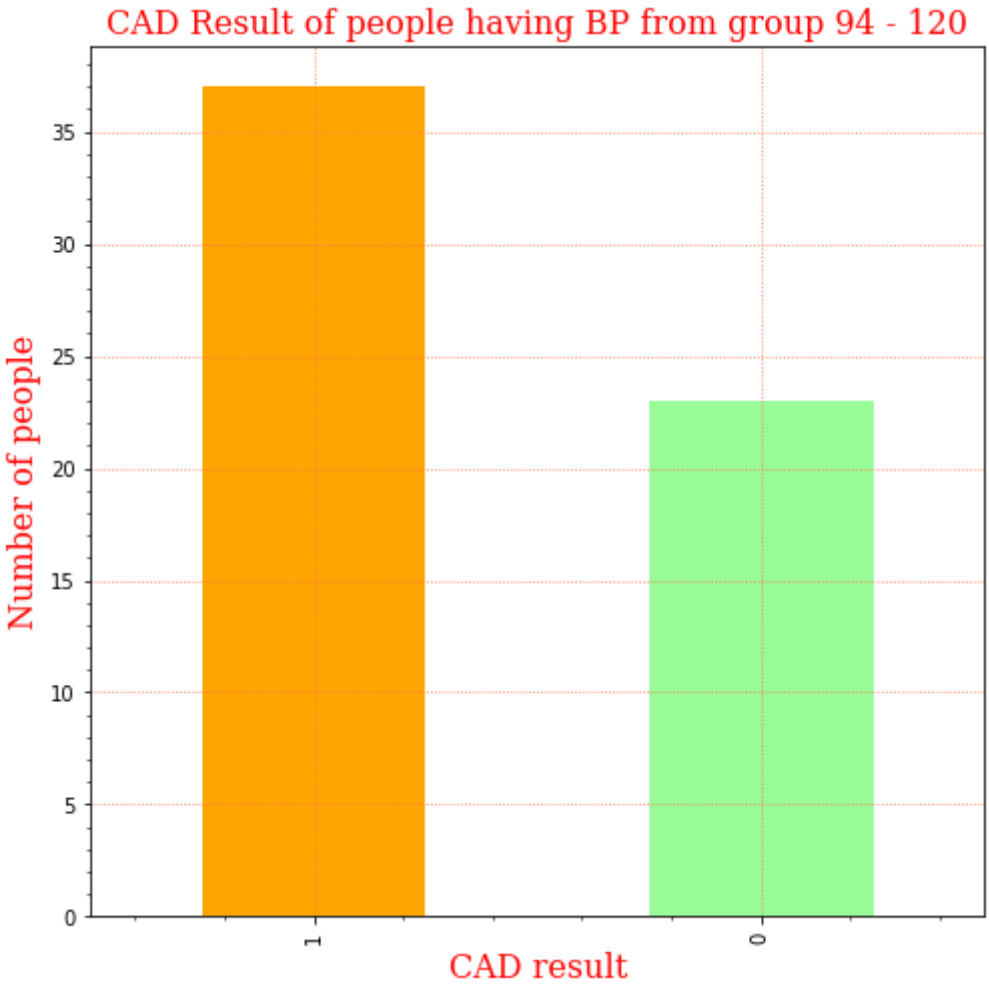
In [58]:
```
rest_bp_94_120
```

Out[58]:

	CAD Result	Patient_Count	color
0	1	37	orange
1	0	23	palegreen

In [59]:
```
label_style={'family':'serif','color':'red','size':16}
rest_bp_94_120.plot(kind='bar',y='Patient_Count',x='CAD Result',figsize=(8,8),color=
plt.minorticks_on()
plt.grid(which='major',linestyle=':',color='coral')
plt.xlabel('CAD result',fontdict=label_style)
plt.ylabel('Number of people',fontdict=label_style)
plt.title('CAD Result of people having BP from group 94 - 120',fontdict=label_style)
```

Out[59]:
```
Text(0.5, 1.0, 'CAD Result of people having BP from group 94 - 120')
```



BP Group2: (120-130]

In [60]:
```
rest_bp_120_130 = pd.DataFrame(cleveland_df[(cleveland_df['rest_bp'] >= 120.0) & (cl
rest_bp_120_130.reset_index(inplace=True)
rest_bp_120_130.columns = ['CAD Result','Patient_Count']
rest_bp_120_130['color'] = rest_bp_120_130['CAD Result'].apply(lambda val : 'palegre
```

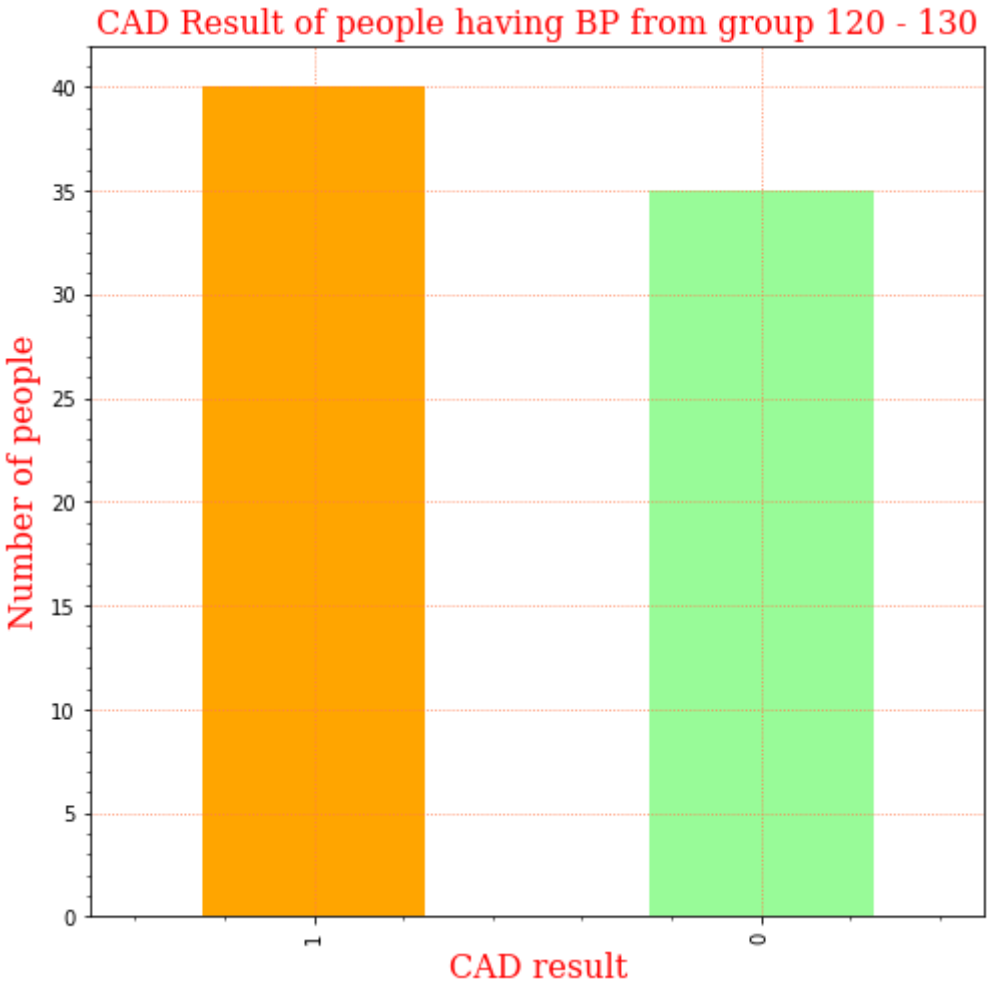
In [61]:
```
rest_bp_120_130
```

Out[61]:

	CAD Result	Patient_Count	color
0	1	40	orange
1	0	35	palegreen

In [62]:
```
label_style={'family':'serif','color':'red','size':16}
rest_bp_120_130.plot(kind='bar',y='Patient_Count',x='CAD Result',figsize=(8,8),color
plt.minorticks_on()
plt.grid(which='major',linestyle=':',color='coral')
plt.xlabel('CAD result',fontdict=label_style)
plt.ylabel('Number of people',fontdict=label_style)
plt.title('CAD Result of people having BP from group 120 - 130',fontdict=label_style
```

Out[62]:
```
Text(0.5, 1.0, 'CAD Result of people having BP from group 120 - 130')
```



BP Group3: (130-140]

In [63]:
```
rest_bp_130_140 = pd.DataFrame(cleveland_df[(cleveland_df['rest_bp'] >= 130.0) & (cl
rest_bp_130_140.reset_index(inplace=True)
rest_bp_130_140.columns = ['CAD Result','Patient_Count']
rest_bp_130_140['color'] = rest_bp_130_140['CAD Result'].apply(lambda val : 'palegre
```

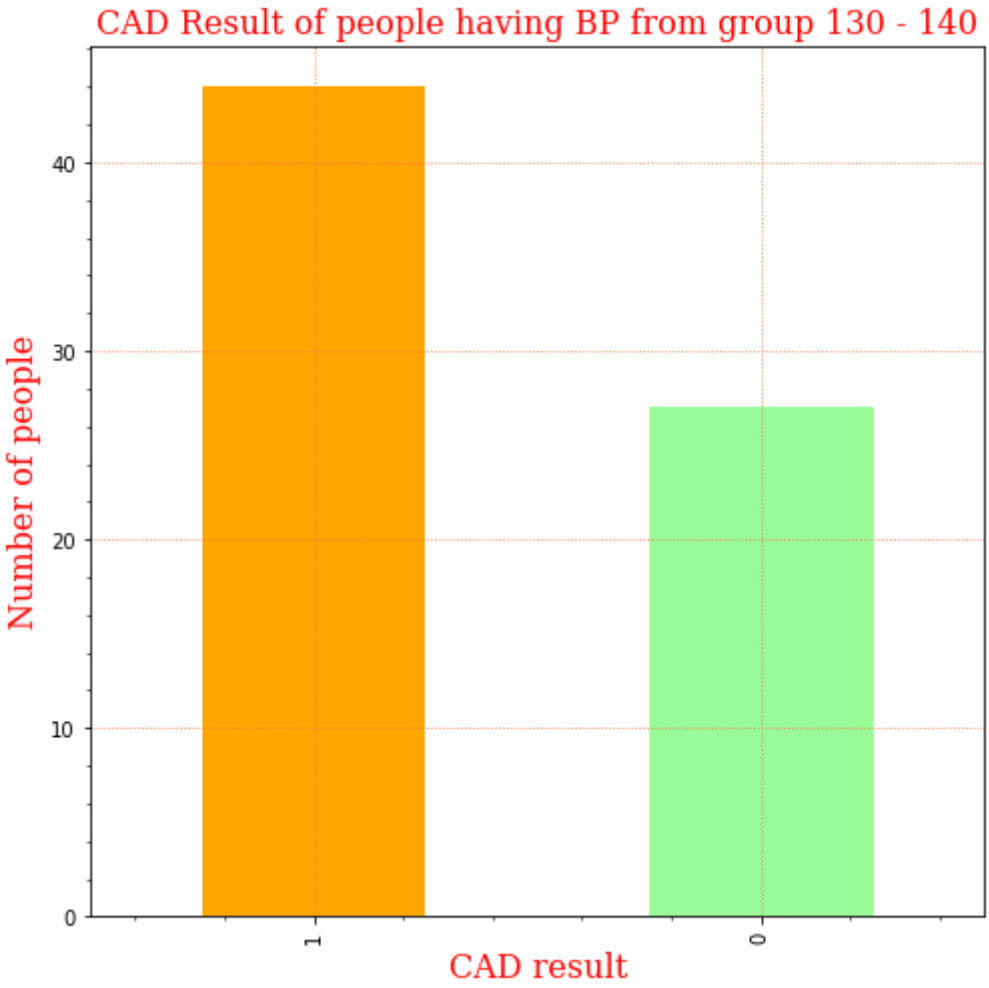
In [64]:
```
rest_bp_130_140
```

Out[64]:

	CAD Result	Patient_Count	color
0	1	44	orange
1	0	27	palegreen

In [65]:
```
label_style={'family':'serif','color':'red','size':16}
rest_bp_130_140.plot(kind='bar',y='Patient_Count',x='CAD Result',figsize=(8,8),color
plt.minorticks_on()
plt.grid(which='major',linestyle=':',color='coral')
plt.xlabel('CAD result',fontdict=label_style)
plt.ylabel('Number of people',fontdict=label_style)
plt.title('CAD Result of people having BP from group 130 - 140',fontdict=label_style
```

Out[65]:
```
Text(0.5, 1.0, 'CAD Result of people having BP from group 130 - 140')
```



BP Group4: 140 or more

In [66]:
```
rest_bp_140_more = pd.DataFrame(cleveland_df[(cleveland_df['rest_bp'] >= 140.0)]['re
rest_bp_140_more.reset_index(inplace=True)
rest_bp_140_more.columns = ['CAD Result','Patient_Count']
rest_bp_140_more['color'] = rest_bp_140_more['CAD Result'].apply(lambda val : 'paleg
```

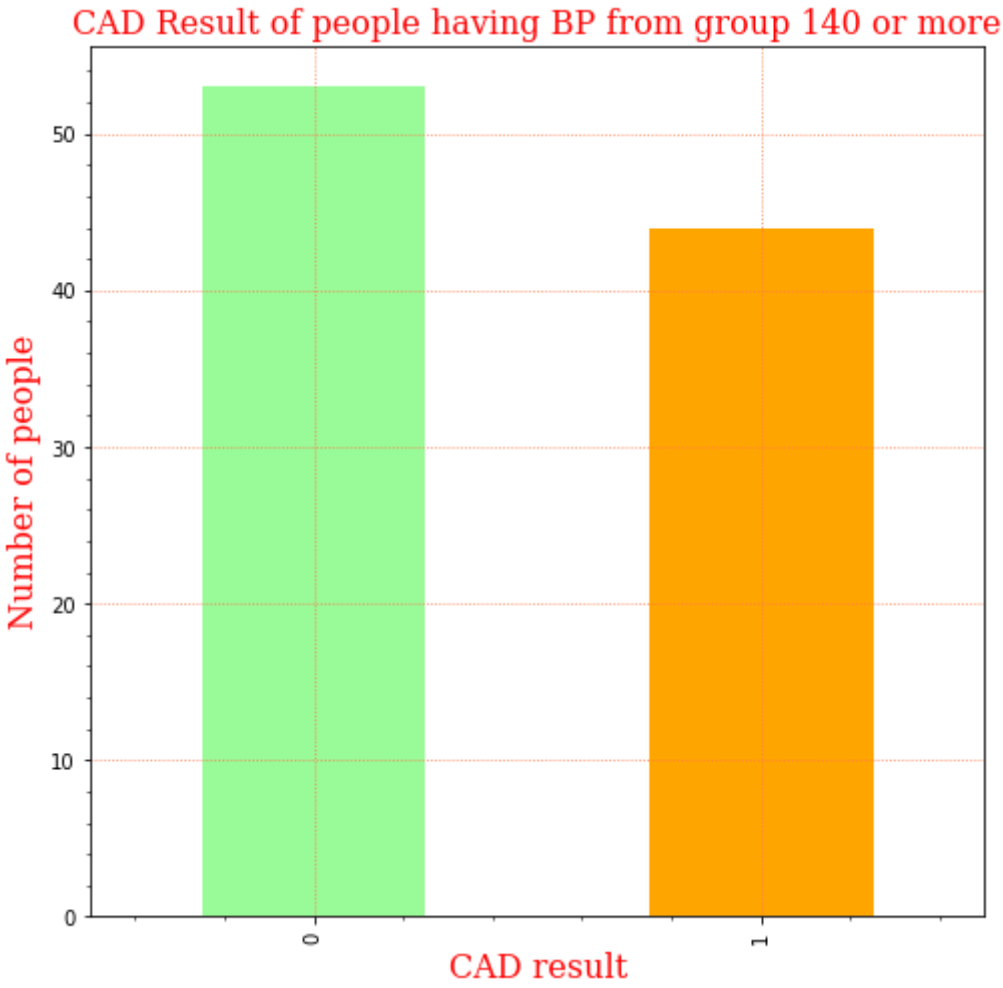
In [67]:
```
rest_bp_140_more
```

Out[67]:

	CAD Result	Patient_Count	color
0	0	53	palegreen
1	1	44	orange

In [68]:
```
label_style={'family':'serif','color':'red','size':16}
rest_bp_140_more.plot(kind='bar',y='Patient_Count',x='CAD Result',figsize=(8,8),colo
plt.minorticks_on()
plt.grid(which='major',linestyle=':',color='coral')
plt.xlabel('CAD result',fontdict=label_style)
plt.ylabel('Number of people',fontdict=label_style)
plt.title('CAD Result of people having BP from group 140 or more',fontdict=label_sty
```

Out[68]:
```
Text(0.5, 1.0, 'CAD Result of people having BP from group 140 or more')
```



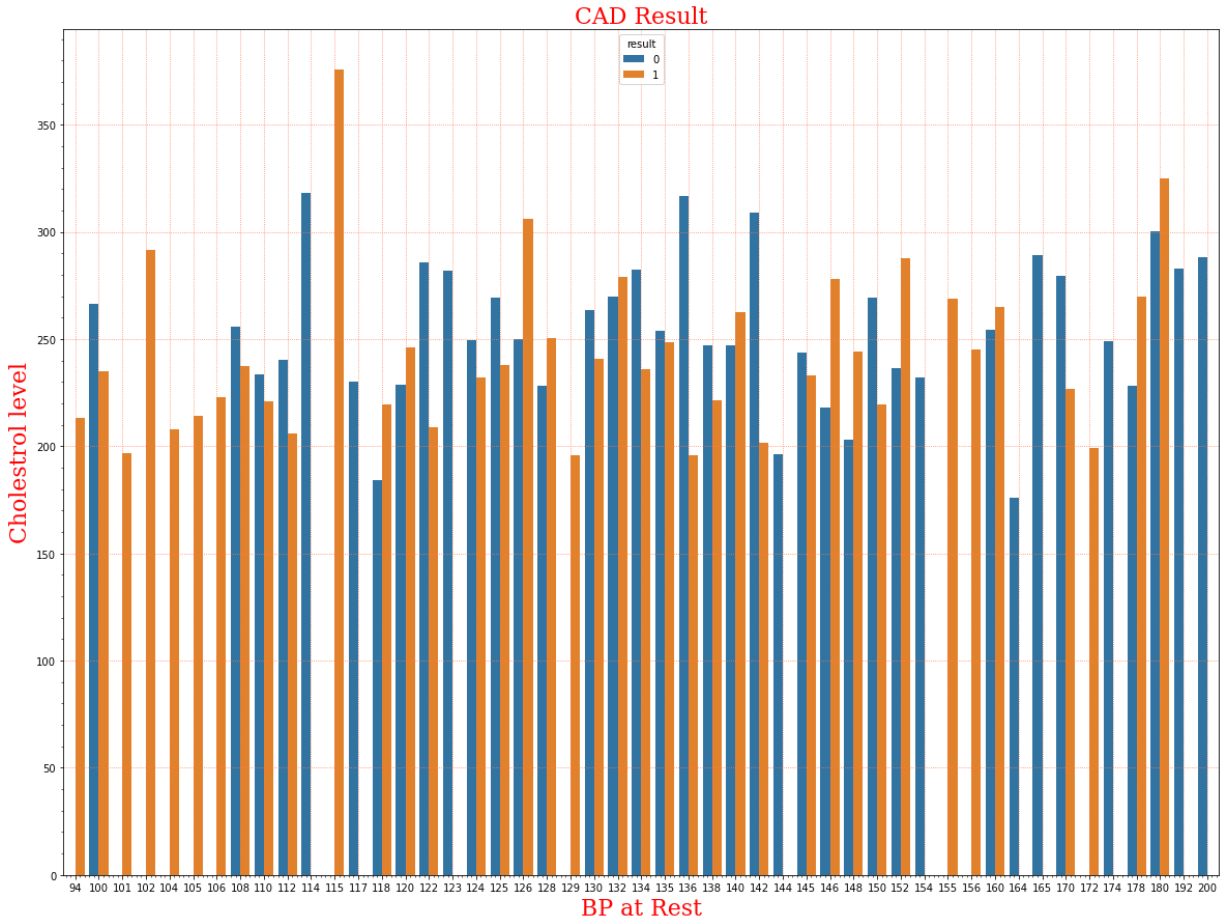
Question-7: Does high blood pressure corresponds to high serum cholestrol, also leads to CAD?

In [69]:
```
cleveland_df['cholestrol'].describe()
```

Out[69]:
```
count    303.000000
mean     246.264026
std       51.830751
min      126.000000
25%      211.000000
50%      240.000000
75%      274.500000
max      564.000000
Name: cholestrol, dtype: float64
```

In [70]:
```
label_style={'family':'serif','color':'red','size':22}
plt.figure(figsize=(20,15))
sns.barplot(x=cleveland_df['rest_bp'],y=cleveland_df['cholestrol'],hue=cleveland_df[
plt.minorticks_on()
plt.grid(which='major',linestyle=':',color='coral')
plt.xlabel('BP at Rest',fontdict=label_style)
plt.ylabel('Cholestrol level',fontdict=label_style)
plt.title('CAD Result',fontdict=label_style)
```

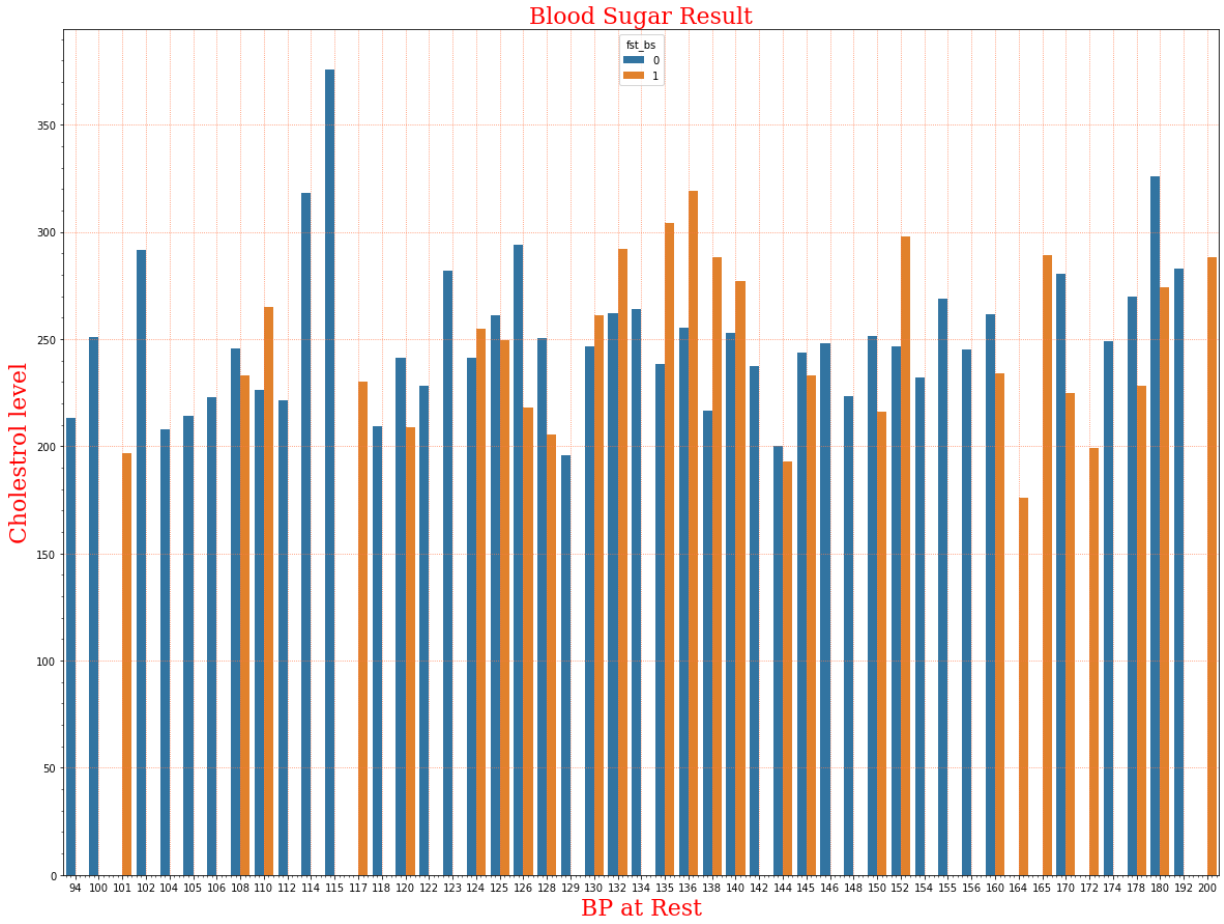
Out[70]:
```
Text(0.5, 1.0, 'CAD Result')
```



Question-8: Does high blood pressure corresponds to high serum cholestrol, also leads to high blood sugar?

In [71]:
```
label_style={'family':'serif','color':'red','size':22}
plt.figure(figsize=(20,15))
sns.barplot(x=cleveland_df['rest_bp'],y=cleveland_df['cholestrol'],hue=cleveland_df[
plt.minorticks_on()
plt.grid(which='major',linestyle=':',color='coral')
plt.xlabel('BP at Rest',fontdict=label_style)
plt.ylabel('Cholestrol level',fontdict=label_style)
plt.title('Blood Sugar Result',fontdict=label_style)
```

Out[71]:
```
Text(0.5, 1.0, 'Blood Sugar Result')
```



Question-9: Does high blood pressure corresponds to high blood sugar, also leads to CAD?

In [72]:
```
pd.DataFrame(cleveland_df[cleveland_df['rest_bp'] >=160]['fst_bs'].value_counts())
```

Out[72]:

	fst_bs
0	18
1	8

In [73]:
```
sugar_high_bp_relation = pd.DataFrame(cleveland_df[cleveland_df['rest_bp'] >=160].gr
sugar_high_bp_relation.columns = ['People count']
sugar_high_bp_relation.index.names = ['Blood Sugar(0:Low,1:High)','CAD Result']
sugar_high_bp_relation
```

Out[73]:

		People count
Blood Sugar(0:Low,1:High)	**CAD Result**	
0	**0**	11
	1	7
1	**0**	6
	1	2

Question-10: Does ST Wave Abnormality corresponds leads to CAD?

In [74]:
```
pd.DataFrame(cleveland_df.groupby('rest_ecg')['result'].count())
```

Out[74]:

rest_ecg	result
0	147
1	152
2	4

In [75]:
```
rest_ecg_CAD_relation = pd.DataFrame(cleveland_df.groupby(['rest_ecg','result'])['ag
rest_ecg_CAD_relation.columns = ['People Count']
rest_ecg_CAD_relation.index.names = ['Rest ECG(0:Normal, 1:ST Wave Abnormal, 2:Left
rest_ecg_CAD_relation
```

Out[75]:

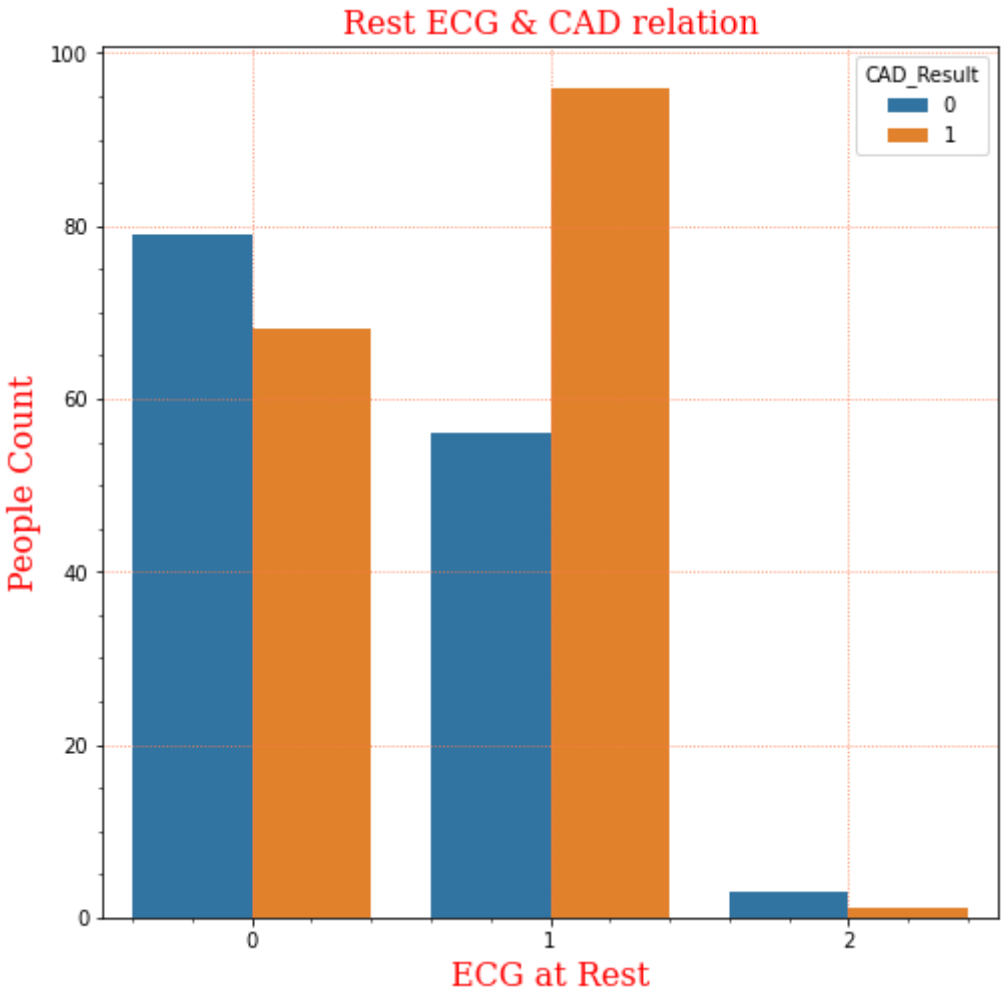
		People Count
Rest ECG(0:Normal, 1:ST Wave Abnormal, 2:Left Vent Hypertrophy)	CAD Result	
0	0	79
	1	68
1	0	56
	1	96
2	0	3
	1	1

In [76]:
```
rest_ecg_CAD_relation.reset_index(inplace=True)
```

In [77]:
```
rest_ecg_CAD_relation.columns = ['Rest_ECG','CAD_Result','People_Count']
```

In [78]:
```
label_style={'family':'serif','color':'red','size':16}
plt.figure(figsize=(8,8))
sns.barplot(x=rest_ecg_CAD_relation['Rest_ECG'],y=rest_ecg_CAD_relation['People_Coun
plt.minorticks_on()
plt.grid(which='major',linestyle=':',color='coral')
plt.xlabel('ECG at Rest',fontdict=label_style)
plt.ylabel('People Count',fontdict=label_style)
plt.title('Rest ECG & CAD relation',fontdict=label_style)
```

Out[78]:
```
Text(0.5, 1.0, 'Rest ECG & CAD relation')
```



Question-11: Does LEFT VENTRICULAR HYPERTROPHY has a realtion with Blood Pressure and Cholestrol?

In [79]:
```
cleveland_df[cleveland_df['rest_ecg'] == 2][['rest_bp','cholestrol']].describe()
```

Out[79]:

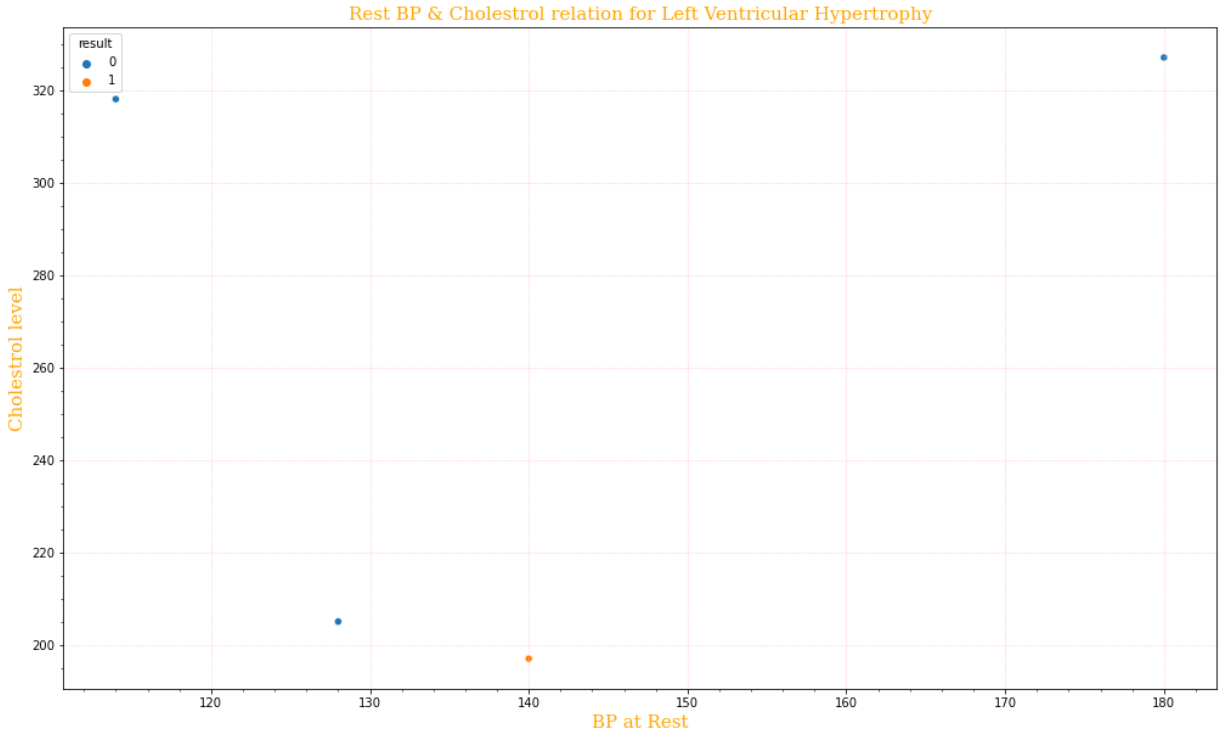
	rest_bp	cholestrol
count	4.000000	4.000000
mean	140.500000	261.750000
std	28.396009	70.320101
min	114.000000	197.000000
25%	124.500000	203.000000
50%	134.000000	261.500000
75%	150.000000	320.250000
max	180.000000	327.000000

In [80]:
```
left_vent_ht_bp_chol_relation = cleveland_df[cleveland_df['rest_ecg'] == 2][['rest_b
```

In [81]:
```
label_style={'family':'serif','color':'orange','size':15}
plt.figure(figsize=(17,10))
sns.scatterplot(x=left_vent_ht_bp_chol_relation['rest_bp'],y=left_vent_ht_bp_chol_re
plt.minorticks_on()
```

```
plt.grid(which='major',linestyle=':',color='pink')
plt.xlabel('BP at Rest',fontdict=label_style)
plt.ylabel('Cholestrol level',fontdict=label_style)
plt.title('Rest BP & Cholestrol relation for Left Ventricular Hypertrophy',fontdict=
```

Out[81]: Text(0.5, 1.0, 'Rest BP & Cholestrol relation for Left Ventricular Hypertrophy')



Question-12: Does LEFT VENTRICULAR HYPERTROPHY associated with High Blood Sugar, also leads to CAD?

In [82]:
```
left_vent_hyt_bs_cad = cleveland_df[cleveland_df['rest_ecg'] == 2][['age','fst_bs','
left_vent_hyt_bs_cad.index.names = ['Blood Sugar','CAD Result']
left_vent_hyt_bs_cad.columns = ['People count']
```

In [83]:
```
left_vent_hyt_bs_cad
```

Out[83]:

		People count
Blood Sugar	**CAD Result**	
0	**0**	3
	1	1

Question-13: Does MAX Heart Rate corresponds to BP at Rest, also leads to CAD?

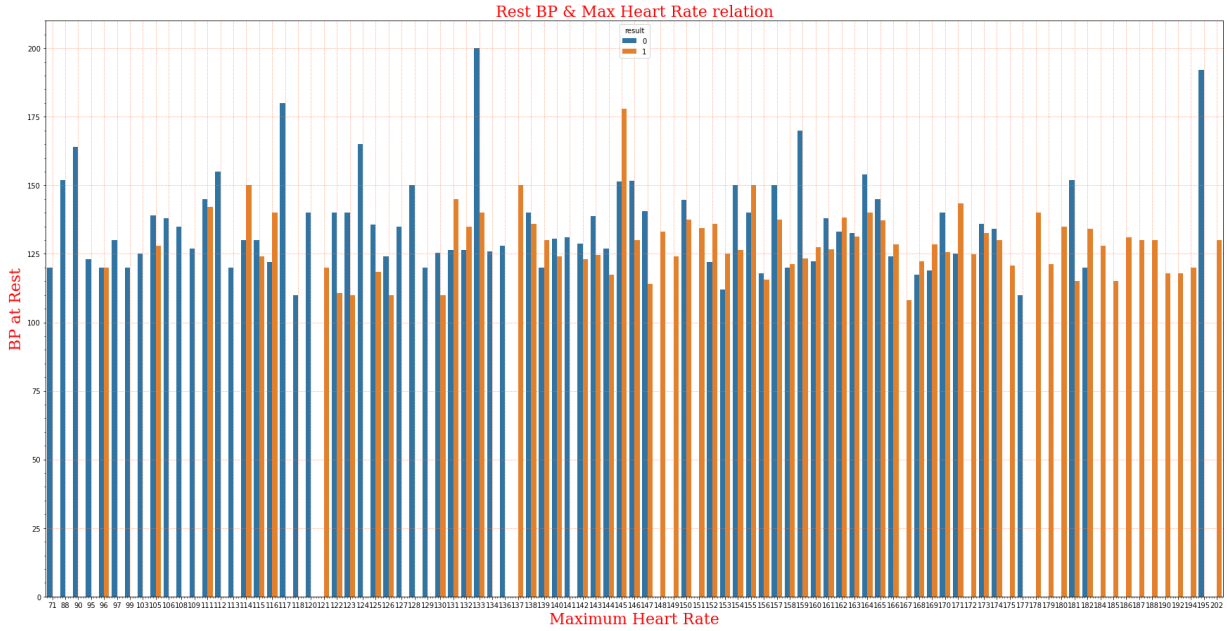
In [84]:
```
cleveland_df['max_hrt_rate'].describe()
```

Out[84]:
```
count    303.000000
mean     149.646865
std       22.905161
min       71.000000
25%      133.500000
50%      153.000000
75%      166.000000
```

```
max      202.000000
Name: max_hrt_rate, dtype: float64
```

In [85]:
```
label_style={'family':'serif','color':'red','size':22}
plt.figure(figsize=(30,15))
sns.barplot(x=cleveland_df['max_hrt_rate'],y=cleveland_df['rest_bp'],hue=cleveland_d
plt.minorticks_on()
plt.grid(which='major',linestyle=':',color='coral')
plt.xlabel('Maximum Heart Rate',fontdict=label_style)
plt.ylabel('BP at Rest',fontdict=label_style)
plt.title('Rest BP & Max Heart Rate relation',fontdict=label_style)
```

Out[85]:
```
Text(0.5, 1.0, 'Rest BP & Max Heart Rate relation')
```



Question-14: Does Exercise induced angina corresponds to CAD?

In [86]:
```
pd.DataFrame(cleveland_df['ex_angina'].value_counts())
```

Out[86]:

	ex_angina
0	204
1	99

In [87]:
```
exc_angina_cad = pd.DataFrame(cleveland_df.groupby(by=['ex_angina','result'],axis=0)
exc_angina_cad.index.names = ['Exc Angina(1:Yes, 0:No)', 'CAD Result']
exc_angina_cad.columns = ['People Count']
exc_angina_cad
```

Out[87]:

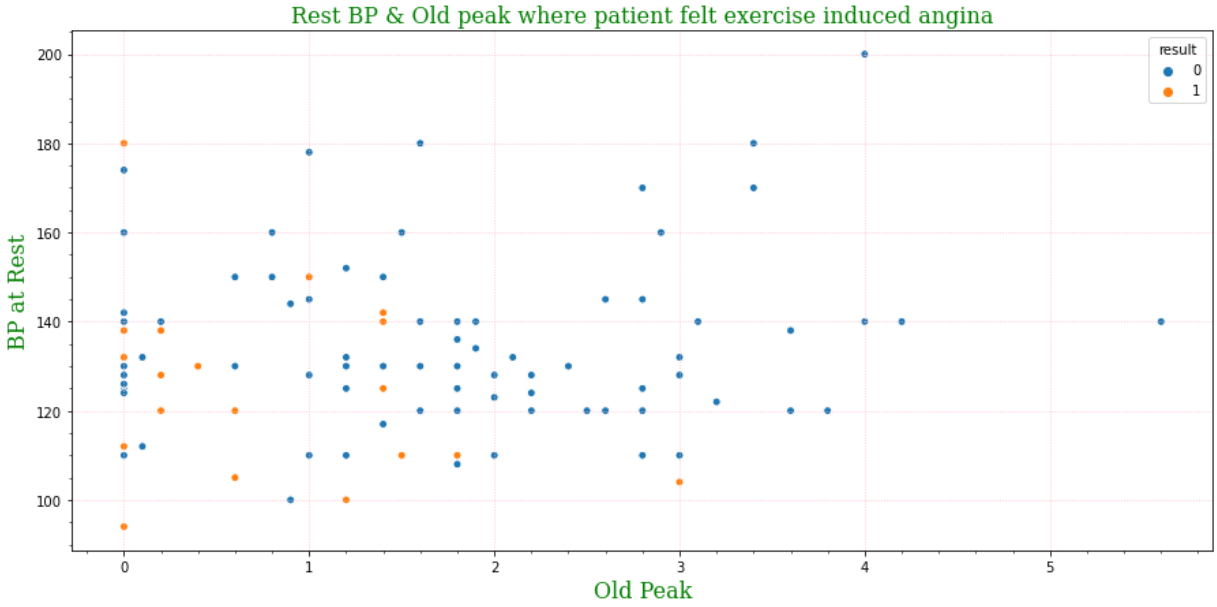
		People Count
Exc Angina(1:Yes, 0:No)	**CAD Result**	
0	**0**	62
	1	142
1	**0**	76
	1	23

Question-15: How exercise induced angina and oldpeak corresponds to the CAD result?

In [88]:
```
exangina_oldpk = cleveland_df[cleveland_df['ex_angina'] == 1][['rest_bp','oldpeak','
```

In [89]:
```
label_style={'family':'serif','color':'Green','size':16}
plt.figure(figsize=(15,7))
sns.scatterplot(x=exangina_oldpk['oldpeak'],y=exangina_oldpk['rest_bp'],hue=exangina
plt.minorticks_on()
plt.grid(which='major',linestyle=':',color='pink')
plt.xlabel('Old Peak',fontdict=label_style)
plt.ylabel('BP at Rest',fontdict=label_style)
plt.title('Rest BP & Old peak where patient felt exercise induced angina',fontdict=l
```

Out[89]:
```
Text(0.5, 1.0, 'Rest BP & Old peak where patient felt exercise induced angina')
```



Question-16: What kind of ST slope in exercise test corresponds more to CAD?

In [90]:
```
pd.DataFrame(cleveland_df['slope'].value_counts())
```

Out[90]:

	slope
2	142
1	140
0	21

In [91]:
```
pd.DataFrame(cleveland_df.groupby(['slope','result'])['age'].count())
```

Out[91]:

		age
slope	**result**	
0	**0**	12
	1	9
1	**0**	91
	1	49

		age
slope	**result**	
2	**0**	35
	1	107

Question-17: Does ST slope has a relationship with Oldpeak and Max heart rate?

In [92]:
```
cleveland_df[['rest_bp','oldpeak','slope']].head()
```

Out[92]:

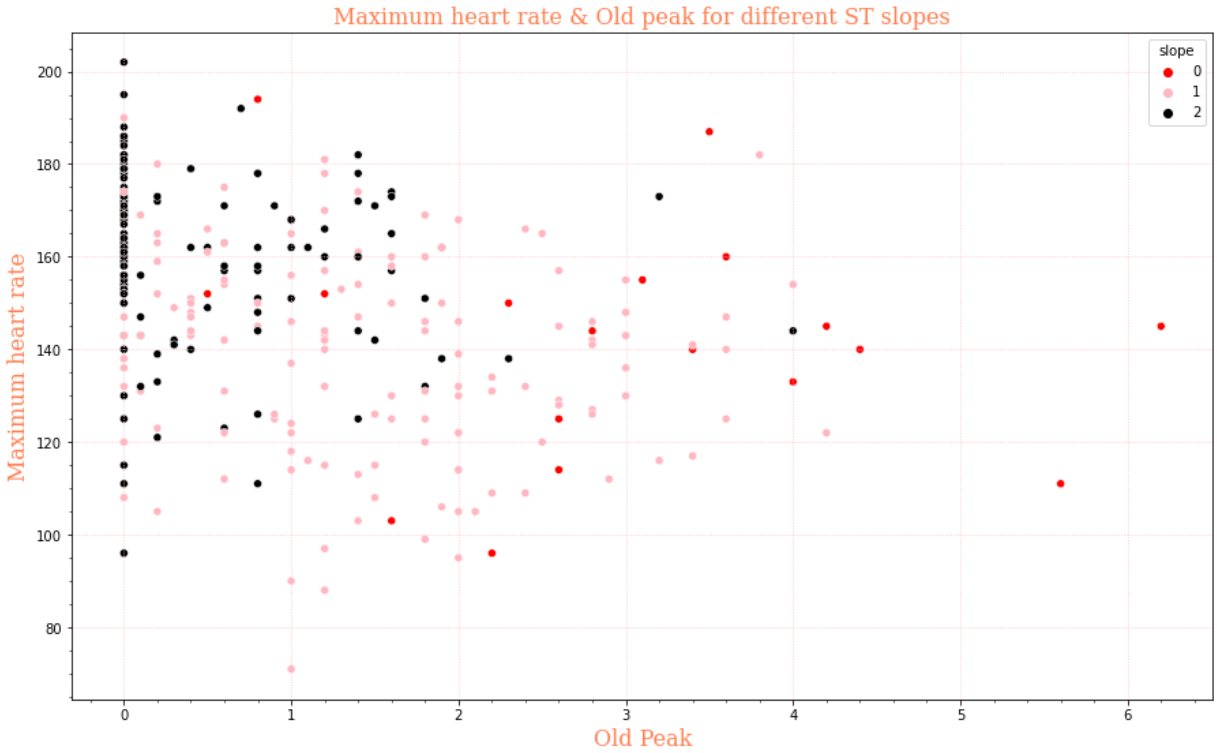
	rest_bp	oldpeak	slope
0	145	2.3	0
1	130	3.5	0
2	130	1.4	2
3	120	0.8	2
4	120	0.6	2

In [93]:
```
cleveland_df['slope'].unique()
```

Out[93]:
```
array([0, 2, 1])
```

In [94]:
```
label_style={'family':'serif','color':'coral','size':16}
plt.figure(figsize=(15,9))
sns.scatterplot(x=cleveland_df['oldpeak'],y=cleveland_df['max_hrt_rate'],hue=clevela
plt.minorticks_on()
plt.grid(which='major',linestyle=':',color='pink')
plt.xlabel('Old Peak',fontdict=label_style)
plt.ylabel('Maximum heart rate',fontdict=label_style)
plt.title('Maximum heart rate & Old peak for different ST slopes',fontdict=label_sty
```

Out[94]:
```
Text(0.5, 1.0, 'Maximum heart rate & Old peak for different ST slopes')
```



Question-18: How Color Vessels in Flouroscopy and Exercise induced angina corresponds to CAD?

In [95]:
```
pd.DataFrame(cleveland_df['fix_color_vsl'].value_counts())
```

Out[95]:

	fix_color_vsl
0	175
1	65
2	38
3	20
4	5

In [96]:
```
pd.DataFrame(cleveland_df.groupby(['fix_color_vsl','result'])['age'].count())
```

Out[96]:

		age
fix_color_vsl	**result**	
0	**0**	45
	1	130
1	**0**	44
	1	21
2	**0**	31
	1	7
3	**0**	17
	1	3
4	**0**	1

		age
fix_color_vsl	**result**	
	1	4

In [97]:
```
pd.DataFrame(cleveland_df.groupby(['fix_color_vsl','ex_angina','result'])['age'].cou
```

Out[97]:

			age
fix_color_vsl	**ex_angina**	**result**	
0	**0**	**0**	19
		1	112
	1	**0**	26
		1	18
1	**0**	**0**	19
		1	16
	1	**0**	25
		1	5
2	**0**	**0**	14
		1	7
	1	**0**	17
3	**0**	**0**	10
		1	3
	1	**0**	7
4	**0**	**1**	4
	1	**0**	1

Question-19: How Thalassemia corrsponds to CAD?

In [98]:
```
pd.DataFrame(cleveland_df['fix_thal'].value_counts())
```

Out[98]:

	fix_thal
2	166
3	117
1	18
0	2

In [99]:
```
pd.DataFrame(cleveland_df.groupby(['fix_thal','result'])['age'].count())
```

Out[99]:

		age
fix_thal	**result**	
0	**0**	1

		age
fix_thal	result	
	1	1
1	0	12
	1	6
2	0	36
	1	130
3	0	89
	1	28

Question-20: Does Thalassemia has any relationship with Age or Max Heart rate/BP/Cholestrol?

In [100…
```
label_style={'family':'serif','color':'coral','size':16}
plt.figure(figsize=(15,9))
sns.scatterplot(x=cleveland_df['age'],y=cleveland_df['max_hrt_rate'],hue=cleveland_d
plt.minorticks_on()
plt.grid(which='major',linestyle=':',color='pink')
plt.xlabel('Age',fontdict=label_style)
plt.ylabel('Maximum heart rate',fontdict=label_style)
plt.title('Maximum heart rate & Age for different effects of Thalassemia',fontdict=l
```

Out[100… Text(0.5, 1.0, 'Maximum heart rate & Age for different effects of Thalassemia')

Don't forget to upvote this notebook if you like the work..

Also, feel free to share any improvement ;)

In [ ]: