

In [11]:

```
import numpy as np
import pandas as pd
import seaborn as sb
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.metrics import confusion_matrix, accuracy_score
from sklearn.linear_model import LogisticRegression
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.cluster import KMeans
import plotly.express as px
import scipy.stats as stats
```

In [29]:

```
data = pd.read_csv(r'C:\Users\Mannahil Miftah\Downloads\heart_disease_uci.csv')
data
```

Out[29]:

	id	age	sex	dataset		cp	trestbps	chol	fb	restecg	thalch	exang	oldpeak	slope
0	1	63	Male	Cleveland	typical angina	145.0	233.0	True		lv hypertrophy	150.0	False	2.3	downsloping
1	2	67	Male	Cleveland	asymptomatic	160.0	286.0	False		lv hypertrophy	108.0	True	1.5	flat
2	3	67	Male	Cleveland	asymptomatic	120.0	229.0	False		lv hypertrophy	129.0	True	2.6	flat
3	4	37	Male	Cleveland	non-anginal	130.0	250.0	False		normal	187.0	False	3.5	downsloping
4	5	41	Female	Cleveland	atypical angina	130.0	204.0	False		lv hypertrophy	172.0	False	1.4	upsloping
...
915	916	54	Female	VA Long Beach	asymptomatic	127.0	333.0	True		st-t abnormality	154.0	False	0.0	NaN
916	917	62	Male	VA Long Beach	typical angina	NaN	139.0	False		st-t abnormality	NaN	NaN	NaN	NaN
917	918	55	Male	VA Long Beach	asymptomatic	122.0	223.0	True		st-t abnormality	100.0	False	0.0	NaN
918	919	58	Male	VA Long Beach	asymptomatic	NaN	385.0	True		lv hypertrophy	NaN	NaN	NaN	NaN
919	920	62	Male	VA Long Beach	atypical angina	120.0	254.0	False		lv hypertrophy	93.0	True	0.0	NaN

920 rows × 16 columns

Exploring Data

In [30]:

```
data.shape
```

Out[30]:

```
(920, 16)
```

In [31]:

```
data.columns
```

Out[31]:

```
Index(['id', 'age', 'sex', 'dataset', 'cp', 'trestbps', 'chol', 'fbs',
       'restecg', 'thalch', 'exang', 'oldpeak', 'slope', 'ca', 'thal', 'num'],
      dtype='object')
```

In [32]:

```
data.dtypes
```

Out[32]:

```
id          int64
age         int64
sex          object
dataset      object
cp           object
trestbps    float64
chol         float64
fbs          object
restecg      object
thalch       float64
exang         object
oldpeak     float64
slope         object
ca            float64
thal          object
num          int64
dtype: object
```

In [33]:

```
data.info
```

Out[33]:

```
<bound method DataFrame.info of
restbps  chol   fbs  \
0        1    63  Male  Cleveland  typical angina  145.0  233.0  True
1        2    67  Male  Cleveland  asymptomatic  160.0  286.0  False
2        3    67  Male  Cleveland  asymptomatic  120.0  229.0  False
3        4    37  Male  Cleveland  non-anginal  130.0  250.0  False
4        5    41 Female  Cleveland atypical angina  130.0  204.0  False
..      ...
915     916    54 Female  VA Long Beach  asymptomatic  127.0  333.0  True
916     917    62  Male  VA Long Beach  typical angina  NaN  139.0  False
917     918    55  Male  VA Long Beach  asymptomatic  122.0  223.0  True
918     919    58  Male  VA Long Beach  asymptomatic  NaN  385.0  True
919     920    62  Male  VA Long Beach atypical angina  120.0  254.0  False

               restecg  thalch  exang  oldpeak  slope  ca  \
0      lv hypertrophy  150.0  False    2.3  downsloping  0.0
1      lv hypertrophy  108.0  True    1.5    flat  3.0
2      lv hypertrophy  129.0  True    2.6    flat  2.0
3          normal    187.0  False   3.5  downsloping  0.0
4      lv hypertrophy  172.0  False   1.4  upsloping  0.0
..      ...
915  st-t abnormality  154.0  False   0.0    NaN  NaN
916  st-t abnormality    NaN  NaN  NaN  NaN  NaN
917  st-t abnormality  100.0  False   0.0    NaN  NaN
918      lv hypertrophy    NaN  NaN  NaN  NaN  NaN
919      lv hypertrophy   93.0  True   0.0    NaN  NaN

               thal  num
0      fixed defect    0
1          normal    2
2  reversable defect    1
3          normal    0
4          normal    0
..      ...
915      NaN  ...
916      ...  ...
917      ...  ...
918      ...  ...
919      ...  ...
```

```
916          NaN  0
917  fixed defect  2
918          NaN  0
919          NaN  1
```

[920 rows x 16 columns]>

In [34]:

```
data.describe()
```

Out[34]:

	id	age	trestbps	chol	thalch	oldpeak	ca	num
count	920.000000	920.000000	861.000000	890.000000	865.000000	858.000000	309.000000	920.000000
mean	460.500000	53.510870	132.132404	199.130337	137.545665	0.878788	0.676375	0.995652
std	265.725422	9.424685	19.066070	110.780810	25.926276	1.091226	0.935653	1.142693
min	1.000000	28.000000	0.000000	0.000000	60.000000	-2.600000	0.000000	0.000000
25%	230.750000	47.000000	120.000000	175.000000	120.000000	0.000000	0.000000	0.000000
50%	460.500000	54.000000	130.000000	223.000000	140.000000	0.500000	0.000000	1.000000
75%	690.250000	60.000000	140.000000	268.000000	157.000000	1.500000	1.000000	2.000000
max	920.000000	77.000000	200.000000	603.000000	202.000000	6.200000	3.000000	4.000000

Checking for missing/nan values

In [35]:

```
#printing values of all attributes to identify if any attribute has ?
```

```
allColumns = data.columns
for c in allColumns:
    missed = data[c].isin(['?']).sum()
    if missed > 0:
        print(c, "\t", missed)
```

In [36]:

```
data.isna().sum()
```

Out[36]:

```
id          0
age         0
sex         0
dataset     0
cp          0
trestbps   59
chol        30
fbs         90
restecg     2
thalch      55
exang       55
oldpeak    62
slope       309
ca          611
thal        486
num         0
dtype: int64
```

In [37]:

```
data.dropna(inplace = True)
```

In [39]:

```
data.isna().sum()
```

Out [39]:

```
id          0  
age         0  
sex         0  
dataset     0  
cp          0  
trestbps   0  
chol        0  
fbs         0  
restecg    0  
thalch      0  
exang       0  
oldpeak    0  
slope       0  
ca          0  
thal        0  
num         0  
dtype: int64
```

Checking if any column contains wrong/unnecessary value

In [40]:

```
data['age'].unique()
```

Out [40]:

```
array([63, 67, 37, 41, 56, 62, 57, 53, 44, 52, 48, 54, 49, 64, 58, 60, 50,  
66, 43, 40, 69, 59, 42, 55, 61, 65, 71, 51, 46, 45, 39, 68, 47, 34,  
35, 29, 70, 77, 38, 74, 76], dtype=int64)
```

In [41]:

```
data['sex'].unique()
```

Out [41]:

```
array(['Male', 'Female'], dtype=object)
```

In [42]:

```
data['cp'].unique()
```

Out [42]:

```
array(['typical angina', 'asymptomatic', 'non-anginal', 'atypical angina'],  
      dtype=object)
```

In [43]:

```
data['trestbps'].unique()
```

Out [43]:

```
array([145., 160., 120., 130., 140., 172., 150., 110., 132., 117., 135.,  
112., 105., 124., 125., 142., 128., 170., 155., 104., 180., 138.,  
108., 134., 122., 115., 118., 100., 200., 94., 165., 102., 152.,  
101., 126., 174., 148., 178., 158., 192., 129., 144., 123., 136.,  
146., 106., 156., 154., 114., 164.])
```

In [44]:

```
data['chol'].unique()
```

Out [44]:

```
array([233., 286., 229., 250., 204., 236., 268., 354., 254., 203., 192.,  
294., 256., 263., 199., 168., 239., 275., 266., 211., 283., 284.,  
224., 206., 219., 340., 226., 247., 167., 230., 335., 234., 177.,  
221., 201., 213., 341., 227., 248., 168., 231., 336., 235., 178.,  
222., 202., 214., 342., 228., 249., 169., 232., 337., 236., 179.,  
223., 203., 215., 343., 229., 250., 205., 237., 269., 355., 255., 204.,  
193., 295., 257., 264., 198., 170., 231., 276., 267., 212., 284., 285.,  
225., 207., 216., 344., 230., 251., 206., 238., 270., 356., 256., 205.,  
194., 296., 258., 265., 199., 171., 232., 277., 268., 213., 285., 286.,  
226., 208., 217., 345., 231., 252., 207., 239., 271., 357., 257., 206.,  
195., 297., 259., 266., 199., 172., 233., 278., 269., 214., 286., 287.,  
227., 209., 218., 346., 232., 253., 208., 240., 272., 358., 258., 207.,  
196., 298., 260., 267., 199., 173., 234., 279., 270., 215., 287., 288.,  
228., 210., 219., 347., 233., 254., 209., 241., 273., 359., 259., 208.,  
197., 299., 261., 268., 199., 174., 235., 280., 271., 216., 288., 289.,  
229., 211., 220., 348., 234., 255., 210., 242., 274., 360., 260., 209.,  
198., 300., 262., 269., 199., 175., 236., 281., 272., 217., 289., 290.,  
230., 212., 221., 349., 235., 256., 211., 243., 275., 361., 261., 210.,  
199., 301., 263., 270., 199., 176., 237., 282., 273., 218., 290., 291.,  
231., 213., 222., 350., 236., 257., 212., 244., 276., 362., 262., 211.,  
200., 302., 264., 271., 199., 177., 238., 283., 274., 219., 291., 292.,  
232., 214., 223., 351., 237., 258., 213., 245., 277., 363., 263., 212.,  
201., 303., 265., 272., 199., 178., 239., 284., 275., 220., 292., 293.,  
233., 215., 224., 352., 238., 259., 214., 246., 278., 364., 264., 213.,  
202., 304., 266., 273., 199., 179., 240., 285., 276., 221., 293., 294.,  
234., 216., 225., 353., 239., 260., 215., 247., 279., 365., 265., 214.,  
203., 305., 267., 274., 199., 180., 241., 286., 277., 222., 294., 295.,  
235., 217., 226., 354., 240., 261., 216., 248., 280., 366., 266., 215.,  
204., 306., 268., 275., 199., 181., 242., 287., 278., 223., 296., 297.,  
236., 218., 227., 355., 241., 262., 217., 249., 281., 367., 267., 216.,  
205., 307., 269., 276., 199., 182., 243., 288., 279., 224., 297., 298.,  
237., 219., 228., 356., 242., 263., 218., 250., 282., 368., 268., 217.,  
206., 308., 270., 277., 199., 183., 244., 289., 280., 225., 298., 299.,  
238., 220., 229., 357., 243., 264., 219., 251., 283., 369., 269., 218.,  
207., 309., 271., 278., 199., 184., 245., 290., 281., 226., 299., 300.,  
239., 221., 230., 358., 244., 265., 220., 252., 284., 370., 270., 219.,  
208., 310., 272., 279., 199., 185., 246., 291., 282., 227., 301., 301.,  
240., 222., 231., 359., 245., 266., 221., 253., 285., 371., 271., 220.,  
209., 311., 273., 280., 199., 186., 247., 292., 283., 228., 302., 302.,  
241., 223., 232., 360., 246., 267., 222., 254., 286., 372., 272., 221.,  
210., 312., 274., 281., 199., 187., 248., 293., 284., 229., 303., 303.,  
242., 224., 233., 361., 247., 268., 223., 255., 287., 373., 273., 222.,  
211., 313., 275., 282., 199., 188., 249., 294., 285., 230., 304., 304.,  
243., 225., 234., 362., 248., 269., 224., 256., 288., 374., 274., 223.,  
212., 314., 276., 283., 199., 189., 250., 295., 286., 231., 305., 305.,  
244., 226., 235., 363., 251., 270., 225., 257., 289., 375., 275., 224.,  
213., 315., 277., 284., 199., 190., 251., 296., 287., 232., 306., 306.,  
245., 227., 236., 364., 252., 271., 226., 258., 290., 376., 276., 225.,  
214., 316., 278., 285., 199., 191., 252., 297., 288., 233., 307., 307.,  
246., 228., 237., 365., 253., 272., 227., 259., 291., 377., 277., 226.,  
215., 317., 279., 286., 199., 192., 253., 298., 289., 234., 308., 308.,  
247., 229., 238., 366., 254., 273., 228., 260., 292., 378., 278., 227.,  
216., 318., 280., 287., 199., 193., 254., 299., 290., 235., 309., 309.,  
248., 230., 239., 367., 255., 274., 229., 261., 293., 379., 279., 228.,  
217., 319., 281., 288., 199., 194., 255., 300., 291., 236., 310., 310.,  
249., 231., 240., 368., 256., 275., 230., 262., 294., 380., 280., 229.,  
218., 320., 282., 291., 199., 195., 256., 301., 292., 237., 311., 311.,  
250., 232., 241., 369., 257., 276., 231., 263., 295., 381., 281., 230.,  
219., 321., 283., 292., 199., 196., 257., 302., 293., 238., 312., 312.,  
251., 233., 242., 370., 258., 277., 232., 264., 296., 382., 282., 231.,  
220., 322., 284., 293., 199., 197., 258., 303., 294., 239., 313., 313.,  
252., 234., 243., 371., 259., 278., 233., 265., 297., 383., 283., 232.,  
221., 323., 285., 294., 199., 198., 259., 304., 295., 240., 314., 314.,  
253., 235., 244., 372., 260., 279., 234., 266., 298., 384., 284., 233.,  
222., 324., 286., 295., 199., 199., 260., 305., 296., 241., 315., 315.,  
254., 236., 245., 373., 261., 280., 235., 267., 300., 385., 285., 234.,  
223., 325., 287., 296., 199., 200., 261., 306., 297., 242., 316., 316.,  
255., 237., 246., 374., 262., 281., 236., 268., 301., 386., 286., 235.,  
224., 326., 288., 297., 199., 201., 262., 307., 298., 243., 317., 317.,  
256., 238., 247., 375., 263., 282., 237., 269., 302., 387., 287., 236.,  
225., 327., 289., 298., 199., 202., 263., 308., 299., 244., 318., 318.,  
257., 239., 248., 376., 264., 283., 238., 270., 303., 388., 288., 237.,  
226., 328., 290., 299., 199., 203., 264., 309., 294., 245., 319., 319.,  
258., 240., 249., 377., 265., 284., 239., 271., 304., 389., 289., 238.,  
227., 329., 291., 299., 199., 204., 265., 310., 295., 246., 320., 320.,  
259., 241., 250., 378., 266., 285., 240., 272., 305., 390., 290., 239.,  
228., 330., 292., 299., 199., 205., 266., 311., 296., 247., 321., 321.,  
260., 242., 251., 379., 267., 286., 241., 273., 306., 391., 291., 240.,  
229., 331., 293., 299., 199., 206., 267., 312., 297., 248., 322., 322.,  
261., 243., 252., 380., 268., 287., 242., 274., 307., 392., 292., 241.,  
230., 332., 294., 299., 199., 207., 268., 313., 298., 249., 323., 323.,  
262., 244., 253., 381., 269., 288., 243., 275., 308., 393., 293., 242.,  
231., 333., 295., 299., 199., 208., 269., 314., 299., 250., 324., 324.,  
263., 245., 254., 382., 270., 289., 244., 276., 309., 394., 294., 243.,  
232., 334., 296., 299., 199., 209., 270., 315., 291., 251., 325., 325.,  
264., 246., 255., 383., 271., 290., 245., 277., 310., 395., 295., 244.,  
233., 335., 297., 299., 199., 210., 271., 316., 292., 252., 326., 326.,  
265., 247., 256., 384., 272., 291., 246., 278., 311., 396., 296., 245.,  
234., 336., 298., 299., 199., 211., 272., 317., 293., 253., 327., 327.,  
266., 248., 257., 385., 273., 292., 247., 279., 312., 397., 297., 246.,  
235., 337., 299., 299., 199., 212., 273., 318., 294., 254., 328., 328.,  
267., 249., 258., 386., 274., 293., 248., 280., 313., 398., 298., 247.,  
236., 338., 300., 299., 199., 213., 274., 319., 295., 255., 329., 329.,  
268., 250., 259., 387., 275., 294., 249., 281., 314., 399., 299., 248.,  
237., 339., 301., 299., 199., 214., 275., 320., 296., 256., 330., 330.,  
269., 251., 260., 388., 276., 295., 250., 282., 315., 400., 300., 249.,  
238., 340., 302., 299., 199., 215., 276., 321., 297., 257., 331., 331.,  
270., 252., 261., 389., 277., 296., 251., 283., 316., 401., 301., 250.,  
239., 341., 303., 299., 199., 216., 277., 322., 298., 258., 332., 332.,  
271., 253., 262., 390., 278., 297., 252., 284., 317., 402., 302., 251.,  
240., 342., 304., 299., 199., 217., 278., 323., 299., 259., 333., 333.,  
272., 254., 263., 391., 279., 298., 253., 285., 318., 403., 303., 252.,  
241., 343., 305., 299., 199., 218., 279., 324., 291., 260., 334., 334.,  
273., 255., 264., 392., 280., 299., 254., 286., 319., 404., 304., 253.,  
242., 344., 306., 299., 199., 219., 280., 325., 292., 261., 335., 335.,  
274., 256., 265., 393., 281., 299., 255., 287., 320., 405., 305., 254.,  
243., 345., 307., 299., 199., 220., 281., 326., 293., 262., 336., 336.,  
275., 257., 266., 394., 282., 299., 256., 288., 321., 406., 306., 255.,  
244., 346., 308., 299., 199., 221., 282., 327., 294., 263., 337., 337.,  
276., 258., 267., 395., 283., 299., 257., 289., 322., 407., 307., 256.,  
245., 347., 309., 299., 199., 222., 283., 328., 295., 264., 338., 338.,  
277., 259., 268., 396., 284., 299., 258., 290., 323., 408., 308., 257.,  
246., 348., 310., 299., 199., 223., 284., 329., 296., 265., 339., 339.,  
278., 260., 269., 397., 285., 299., 259., 291., 324., 409., 309., 258.,  
247., 349., 311., 299., 199., 224., 285., 330., 297., 266., 340., 340.,  
279., 261., 270., 398., 286., 299., 260., 292., 325., 410., 310., 259.,  
248., 350., 312., 299., 199., 225., 286., 331., 298., 267., 341., 341.,  
280., 262., 271., 399., 287., 299., 261., 293., 326., 411., 311., 260.,  
249., 351., 313., 299., 199., 226., 287., 332., 299., 268., 342., 342.,  
281., 263., 272., 400., 288., 299., 262., 294., 327., 412., 312., 261.,  
250., 352., 314., 299., 199., 227., 288., 333., 291., 269., 343., 343.,  
282., 264., 273., 401., 289., 299., 263., 295., 328., 413., 313., 262.,  
251., 353., 315., 299., 199., 228., 289., 334., 292., 270., 344., 344.,  
283., 265., 274., 402., 290., 299., 264., 296., 329., 414., 314., 263.,  
252., 354., 316., 299., 199., 229., 290., 335., 293., 271., 345., 345.,  
284., 266., 275., 403., 291., 299., 265., 297., 330., 415., 315., 264.,  
253., 355., 317., 299., 199., 230., 291., 336., 294., 272., 346., 346.,  
285., 267., 276., 404., 292., 299., 266., 298., 331., 416., 316., 265.,  
254., 356., 318., 299., 199., 231., 292., 337., 295., 273., 347., 347.,  
286., 268., 277., 405., 293., 299., 267., 299., 332., 417., 317., 266.,  
255., 357., 319., 299., 199., 232., 293., 338., 296., 274., 348., 348.,  
287., 269., 278., 406., 294., 299., 268., 291., 333., 418., 318., 267.,  
256., 358., 320., 299., 199., 233., 294., 339., 297., 275., 349., 349.,  
288., 270., 279., 407., 295., 299., 269., 292., 334., 419., 319., 268.,  
257., 359., 321., 299., 199., 234., 295., 340., 298., 276., 350., 350.,  
289., 271., 280., 408., 296., 299., 270., 293., 335., 420., 320., 269.,  
258., 360., 322., 299., 199., 235., 296., 341., 299., 277., 351., 351.,  
290., 272., 281., 409., 297., 299., 271., 294., 336., 421., 321., 270.,  
259., 361., 323., 299., 199., 236., 297., 342., 291., 278., 352., 352.,  
291., 273., 282., 410., 298., 299., 272., 295., 337., 422., 322., 271.,  
260., 362., 324., 299., 199., 237., 298., 343., 292., 279., 353., 353.,  
292., 274., 283., 411., 299., 299., 273., 296., 338., 423., 323., 272.,  
261., 363., 325., 299., 199., 238., 299., 344., 293., 280., 354., 354.,  
293., 275., 284., 412., 300., 299., 274., 297., 339., 424., 324., 273.,  
262., 364., 326., 299., 199., 239., 291., 345., 294., 281., 355., 355.,  
294., 276., 285., 413., 301., 299., 275., 298., 340., 425., 325., 274.,  
263., 365., 327., 299., 199., 240., 292., 346., 295., 282., 356., 356.,  
295., 277., 286., 414., 302., 299., 276., 299., 341., 426., 326., 275.,  
264., 366., 328., 299., 199., 241., 293., 347., 296., 283., 357., 357.,  
296., 278., 287., 415., 303., 299., 277., 291., 342., 427., 327., 276.,  
265., 367., 329., 299., 199., 242., 294., 348., 297., 284., 358., 358.,  
297., 279., 288., 416., 304., 299., 278., 292., 343., 428., 328., 277.,  
266., 368., 330., 299., 199., 243., 295., 349., 298., 285., 359., 359.,  
298., 280., 289., 417., 305., 299., 279., 293., 344., 429., 329., 278.,  
267., 369., 331., 299., 199., 244., 296., 350., 299., 286., 360., 360.,  
299., 281., 290., 418., 306., 299., 280., 294., 345., 430., 330., 279.,  
268., 370., 332., 299., 199., 245., 297., 351., 291., 287., 361., 361.,  
300., 282., 291., 419., 307., 299., 281., 
```

```
276., 353., 243., 225., 302., 212., 330., 175., 411., 191., 198.,
290., 253., 172., 273., 213., 305., 216., 304., 188., 282., 185.,
232., 326., 231., 269., 267., 248., 360., 258., 308., 245., 270.,
208., 264., 321., 274., 325., 235., 257., 164., 141., 252., 255.,
201., 222., 260., 182., 303., 265., 309., 307., 249., 186., 341.,
183., 407., 217., 288., 220., 209., 227., 261., 174., 281., 221.,
205., 240., 289., 318., 298., 564., 246., 322., 299., 300., 293.,
277., 214., 207., 160., 394., 184., 315., 409., 244., 195., 196.,
126., 313., 259., 200., 262., 215., 228., 193., 271., 210., 327.,
149., 295., 306., 178., 237., 218., 223., 242., 319., 166., 180.,
311., 278., 342., 169., 187., 157., 176., 241., 131., 100.])
```

In [45]:

```
data['fbs'].unique()
```

Out[45]:

```
array([True, False], dtype=object)
```

In [46]:

```
data['restecg'].unique()
```

Out[46]:

```
array(['lv hypertrophy', 'normal', 'st-t abnormality'], dtype=object)
```

In [48]:

```
data['exang'].unique()
```

Out[48]:

```
array([False, True], dtype=object)
```

In [49]:

```
data['oldpeak'].unique()
```

Out[49]:

```
array([2.3, 1.5, 2.6, 3.5, 1.4, 0.8, 3.6, 0.6, 3.1, 0.4, 1.3, 0., 0.5,
       1.6, 1., 1.2, 0.2, 1.8, 3.2, 2.4, 2., 2.5, 2.2, 2.8, 3., 3.4,
       6.2, 4., 5.6, 2.9, 0.1, 2.1, 1.9, 4.2, 0.9, 1.1, 3.8, 0.7, 0.3,
       4.4])
```

In [50]:

```
data['slope'].unique()
```

Out[50]:

```
array(['downsloping', 'flat', 'upsloping'], dtype=object)
```

In [51]:

```
data['ca'].unique()
```

Out[51]:

```
array([0., 3., 2., 1.])
```

In [52]:

```
data['thal'].unique()
```

Out[52]:

```
array(['fixed defect', 'normal', 'reversable defect'], dtype=object)
```

In [56]:

```
# 0 = No disease
```

```
# 1 = Stage 1 disease  
# 2 = Stage 2 disease  
# 3 = Stage 3 disease  
# 4 = Stage 4 disease
```

```
data['num'].unique()
```

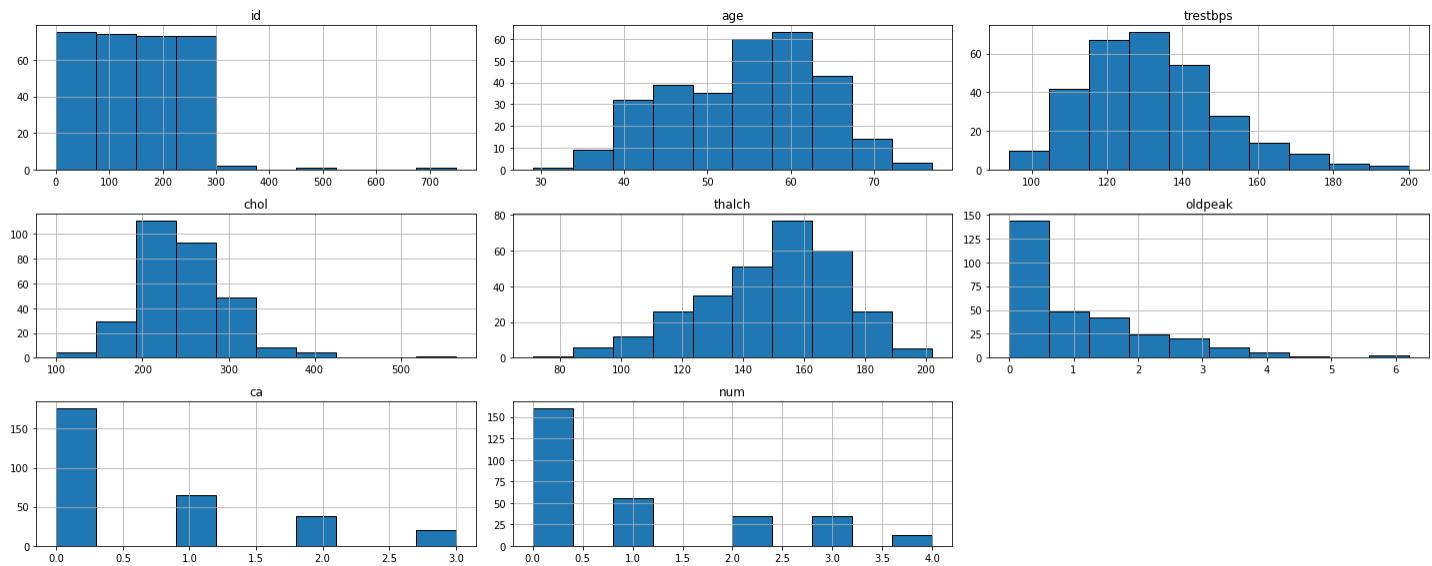
Out [56]:

```
array([0, 2, 1, 3, 4], dtype=int64)
```

Distributions & Visualizations

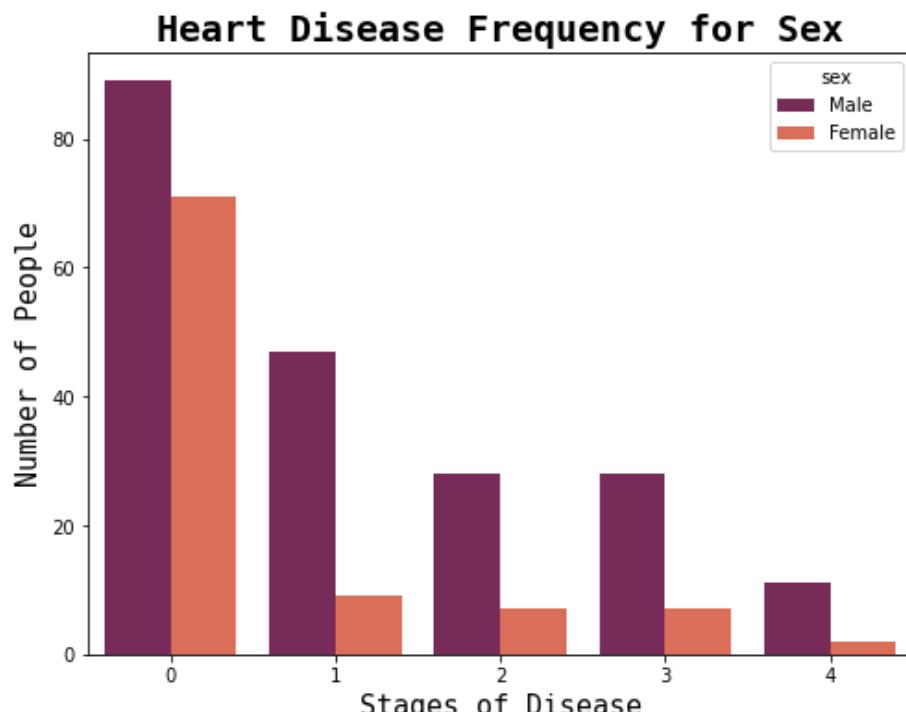
In [54]:

```
data.hist(bins = 10, figsize=[20,8], ec = 'black')  
plt.tight_layout()  
plt.show()
```



In [63]:

```
plt.figure(figsize=[8,6])  
sb.countplot(data = data ,x = 'num', hue = 'sex', palette = 'rocket')  
plt.title('Heart Disease Frequency for Sex', fontdict={'fontname': 'Monospace', 'fontsize': 20, 'fontweight': 'bold'})  
plt.xlabel('Stages of Disease', fontdict={'fontname': 'Monospace', 'fontsize': 15,})  
plt.ylabel('Number of People', fontdict={'fontname': 'Monospace', 'fontsize': 15,})  
plt.show()
```

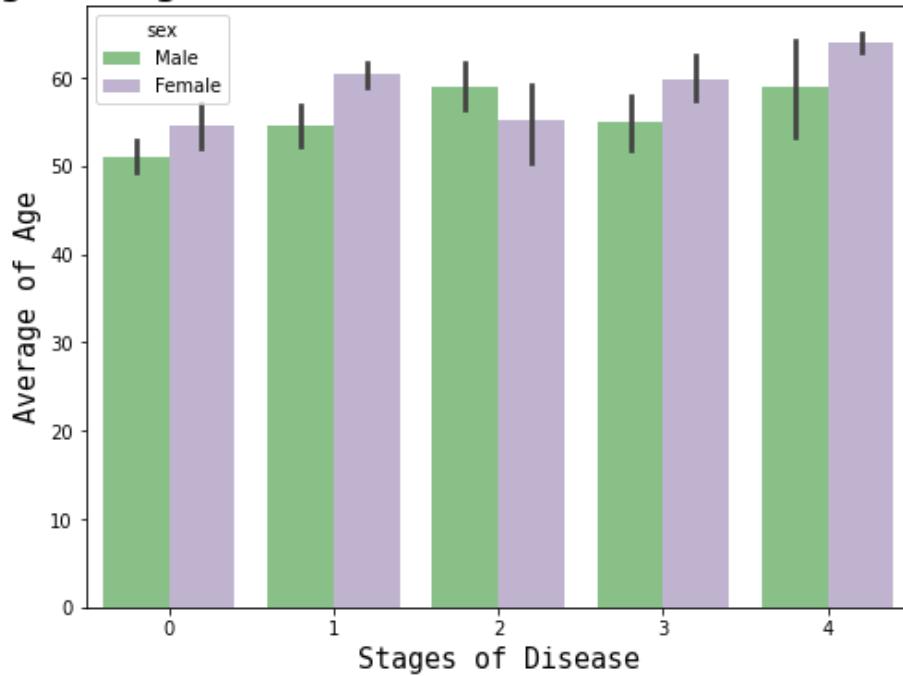


In [62]:

```
# Clustered Bar Chart
```

```
plt.figure(figsize=[8,6])
sb.barplot(data = data ,x = 'num', y = 'age', hue = 'sex', palette = 'Accent')
plt.title('Average of Age for Sex relative to Heart Disease Stages', fontdict={'fontname': 'Monospace', 'fontsize': 20, 'fontweight': 'bold'})
plt.xlabel('Stages of Disease', fontdict={'fontname':'Monospace', 'fontsize': 15,})
plt.ylabel('Average of Age', fontdict={'fontname':'Monospace', 'fontsize': 15,})
plt.show()
```

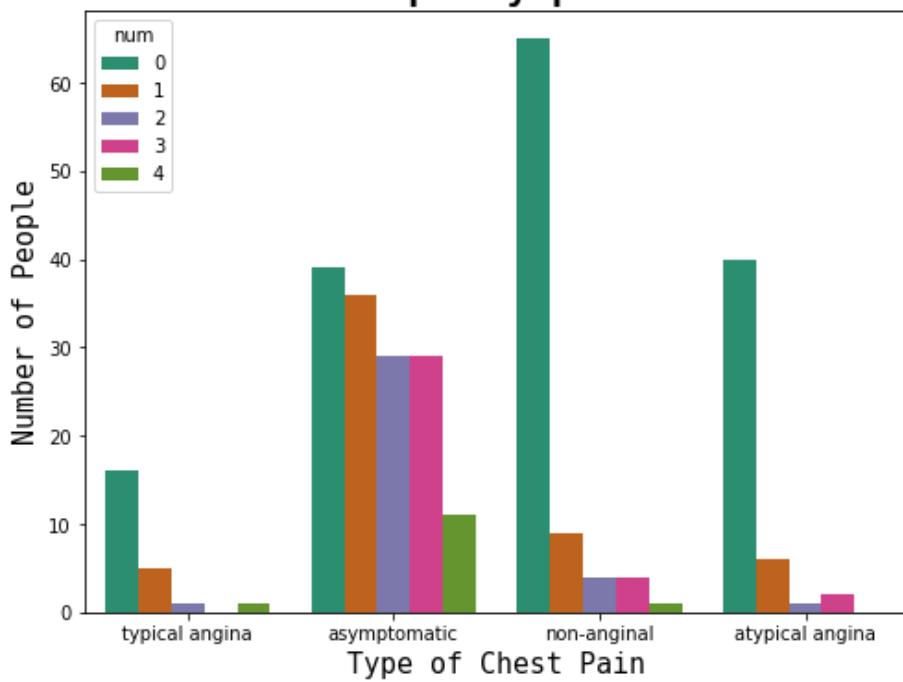
Average of Age for Sex relative to Heart Disease Stages



In [68]:

```
plt.figure(figsize=[8,6])
sb.countplot(data = data ,x = 'cp', hue = 'num',palette = 'Dark2')
plt.title('Heart Disease Frequency per Chest Pain Type', fontdict={'fontname': 'Monospace', 'fontsize': 20, 'fontweight': 'bold'})
plt.xlabel('Type of Chest Pain', fontdict={'fontname':'Monospace', 'fontsize': 15,})
plt.ylabel('Number of People', fontdict={'fontname':'Monospace', 'fontsize': 15,})
plt.show()
```

Heart Disease Frequency per Chest Pain Type

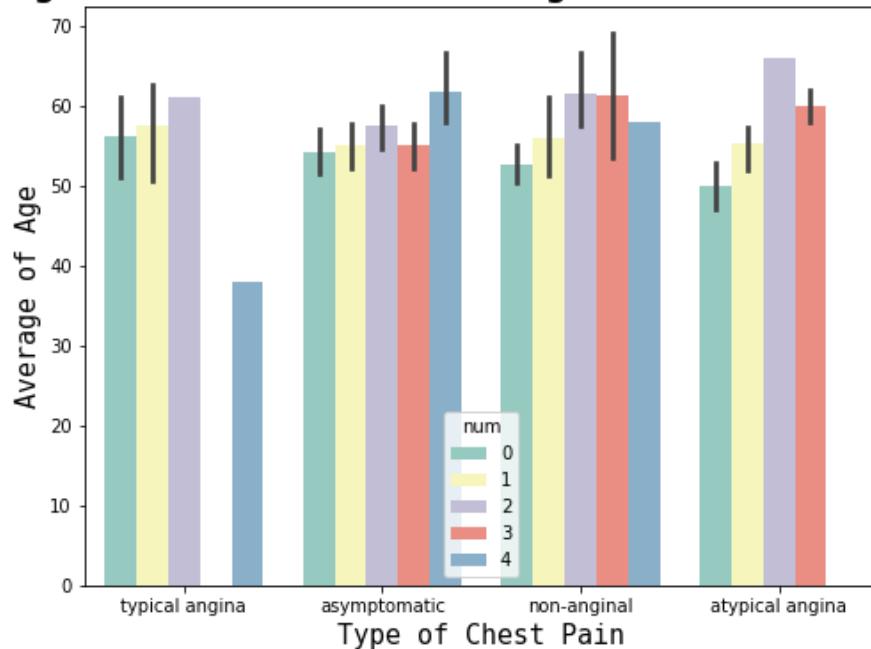


In [79]:

```
# Clustered Bar Chart
```

```
plt.figure(figsize=[8,6])
sb.barplot(data = data ,x = 'cp', y = 'age', hue = 'num', palette = 'Set3')
plt.title('Average of Age for Heart Disease Stage relative to Chest Pain Type', fontdict= {'fontname': 'Monospace', 'fontsize': 20, 'fontweight': 'bold'})
plt.xlabel('Type of Chest Pain', fontdict={'fontname':'Monospace', 'fontsize': 15,})
plt.ylabel('Average of Age', fontdict={'fontname':'Monospace', 'fontsize': 15,})
# plt.legend(['No disease', 'Stage 1', 'Stage 2', 'Stage 3', 'Stage 4'])
plt.show()
```

Average of Age for Heart Disease Stage relative to Chest Pain Type

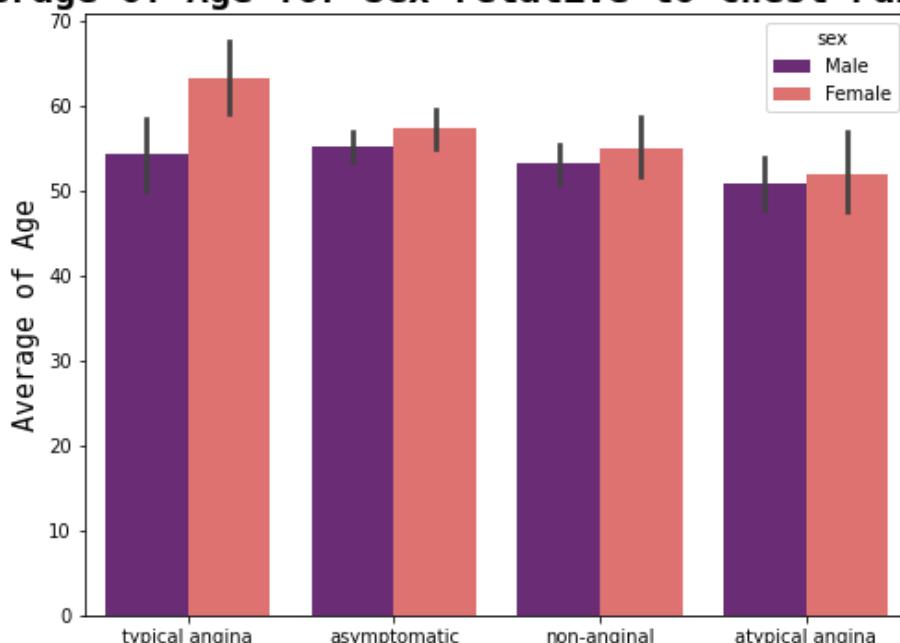


In [90]:

```
# Clustered Bar Chart
```

```
plt.figure(figsize=[8,6])
sb.barplot(data = data ,x = 'cp', y = 'age', hue = 'sex', palette = 'magma')
plt.title('Average of Age for Sex relative to Chest Pain Type', fontdict= {'fontname': 'Monospace', 'fontsize': 20, 'fontweight': 'bold'})
plt.xlabel('Type of Chest Pain', fontdict={'fontname':'Monospace', 'fontsize': 15,})
plt.ylabel('Average of Age', fontdict={'fontname':'Monospace', 'fontsize': 15,})
plt.show()
```

Average of Age for Sex relative to Chest Pain Type

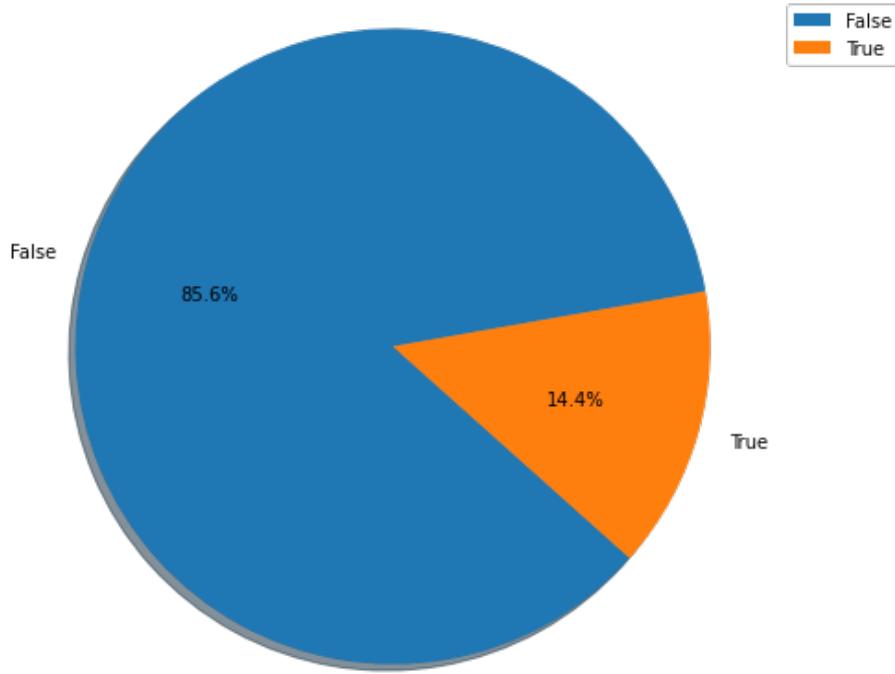


Type of Chest Pain

In [93]:

```
fbs = data['fbs'].value_counts()
plt.figure(figsize=(10, 7))
plt.pie(fbs.values, labels=fbs.index, startangle=10, shadow=True, autopct='%.1f%%')
plt.title('Fasting Blood Sugar Distribution', fontdict={'fontname': 'Monospace', 'fontsize': 20, 'fontweight': 'bold'})
plt.legend()
plt.axis('equal')
plt.show()
```

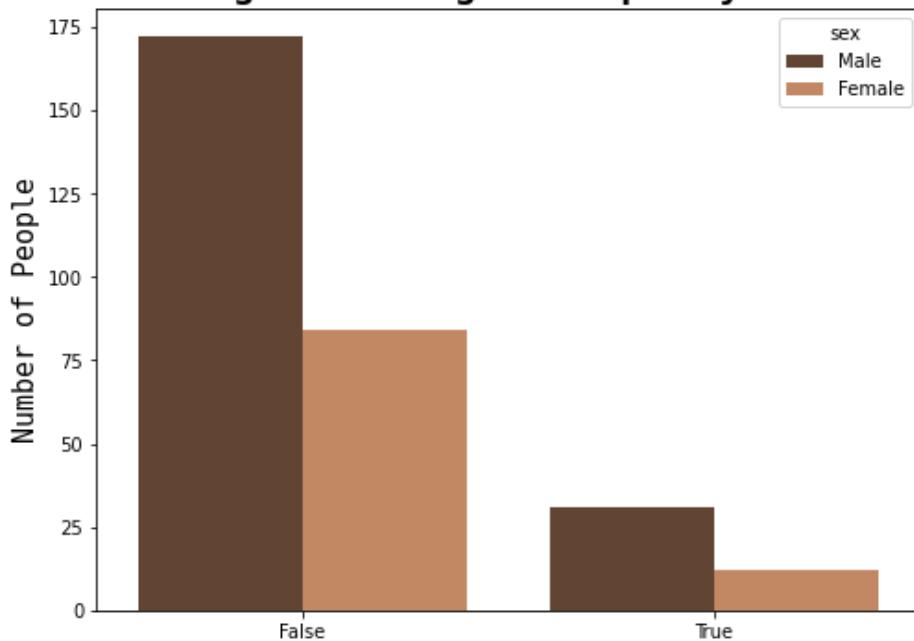
Fasting Blood Sugar Distribution



In [91]:

```
plt.figure(figsize=[8, 6])
sb.countplot(data = data , x = 'fbs', hue = 'sex', palette = 'copper')
plt.title('Fasting Blood Sugar Frequency for Sex', fontdict={'fontname': 'Monospace', 'fontsize': 20, 'fontweight': 'bold'})
plt.xlabel('Fasting Blood Sugar', fontdict={'fontname': 'Monospace', 'fontsize': 15,})
plt.ylabel('Number of People', fontdict={'fontname': 'Monospace', 'fontsize': 15,})
plt.show()
```

Fasting Blood Sugar Frequency for Sex



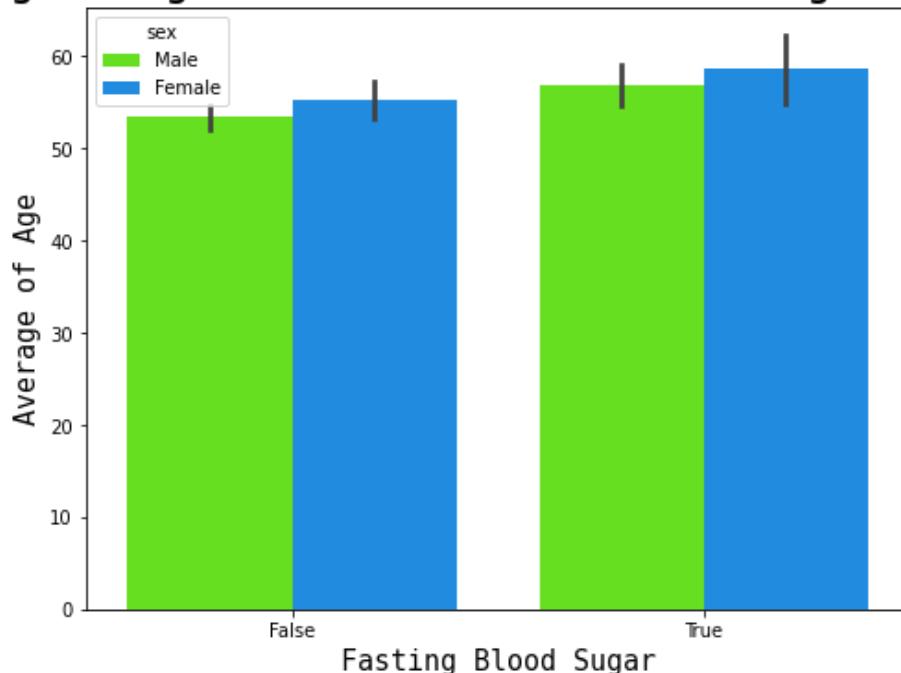
Fasting Blood Sugar

In [92]:

```
# Clustered Bar Chart
```

```
plt.figure(figsize=[8,6])
sb.barplot(data = data ,x = 'fbs', y = 'age', hue = 'sex', palette = 'gist_rainbow')
plt.title('Average of Age for Sex relative to Fasting Blood Sugar', fontdict={'fontname': 'Monospace', 'fontsize': 20, 'fontweight': 'bold'})
plt.xlabel('Fasting Blood Sugar', fontdict={'fontname': 'Monospace', 'fontsize': 15,})
plt.ylabel('Average of Age', fontdict={'fontname': 'Monospace', 'fontsize': 15,})
plt.show()
```

Average of Age for Sex relative to Fasting Blood Sugar



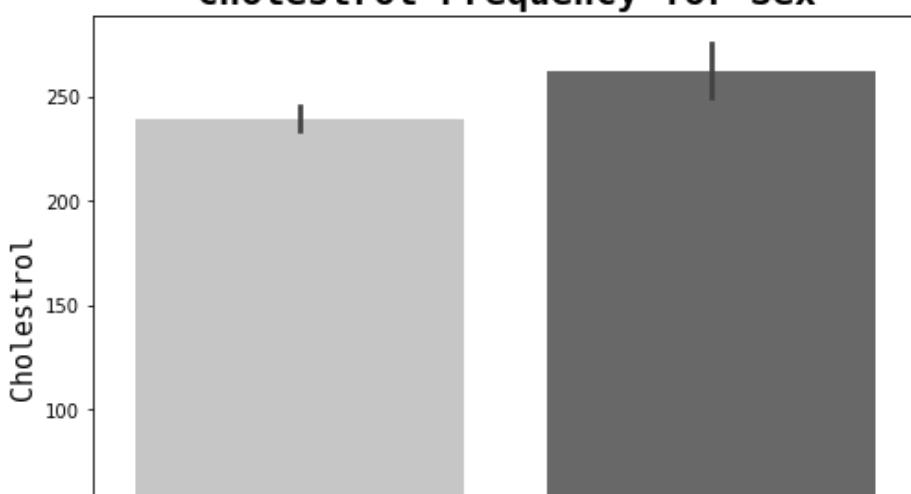
In [110]:

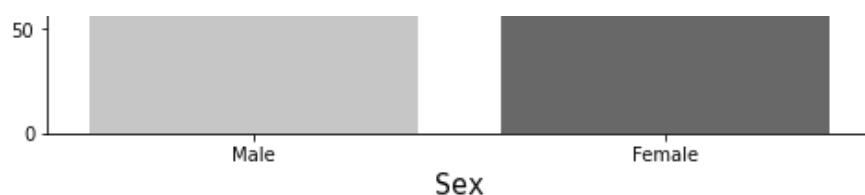
```
plt.figure(figsize=[8,6])
sb.barplot(data['sex'], data['chol'], palette = 'Greys')
plt.title('Cholestrol Frequency for Sex', fontdict={'fontname': 'Monospace', 'fontsize': 20, 'fontweight': 'bold'})
plt.xlabel('Sex', fontdict={'fontname': 'Monospace', 'fontsize': 15,})
plt.ylabel('Cholestrol', fontdict={'fontname': 'Monospace', 'fontsize': 15,})
plt.show()
```

c:\Users\Mannahil Miftah\anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

```
warnings.warn(
```

Cholestrol Frequency for Sex





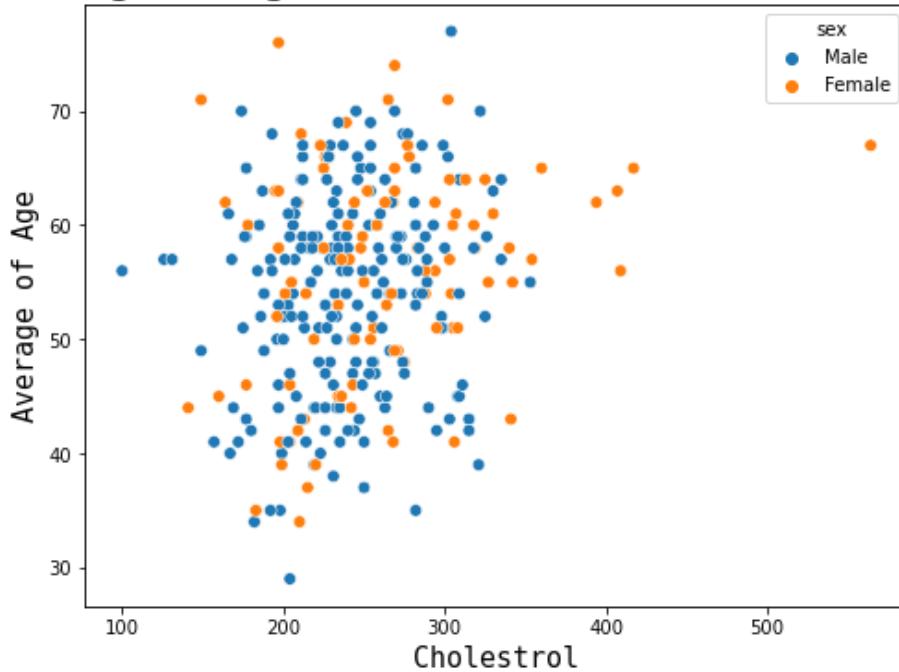
In [159]:

```
plt.figure(figsize=[8, 6])
sb.scatterplot(data['chol'], data['age'], hue = data['sex'], s = 45)
plt.title('Average of Age for Sex relative to Cholestrol', fontdict={'fontname': 'Monospace', 'fontsize': 20, 'fontweight': 'bold'})
plt.xlabel('Cholestrol', fontdict={'fontname': 'Monospace', 'fontsize': 15,})
plt.ylabel('Average of Age', fontdict={'fontname': 'Monospace', 'fontsize': 15,})
plt.show()
```

c:\Users\Mannahil Miftah\anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

```
warnings.warn(
```

Average of Age for Sex relative to Cholestrol



In [158]:

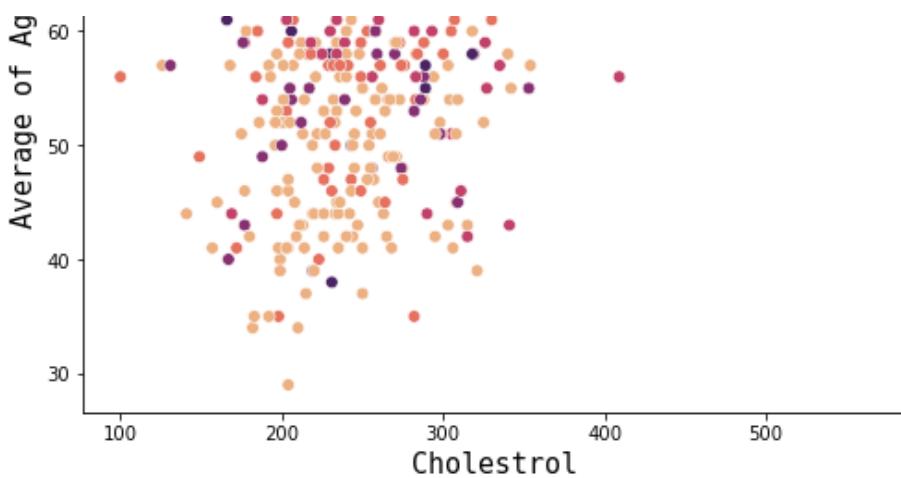
```
plt.figure(figsize=[8, 6])
sb.scatterplot(data['chol'], data['age'], hue = data['num'], palette = 'flare', s = 45)
plt.title('Average of Age for Heart Disease Stage relative to Cholestrol', fontdict={'fontname': 'Monospace', 'fontsize': 20, 'fontweight': 'bold'})
plt.xlabel('Cholestrol', fontdict={'fontname': 'Monospace', 'fontsize': 15,})
plt.ylabel('Average of Age', fontdict={'fontname': 'Monospace', 'fontsize': 15,})
plt.show()
```

c:\Users\Mannahil Miftah\anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

```
warnings.warn(
```

Average of Age for Heart Disease Stage relative to Cholestrol





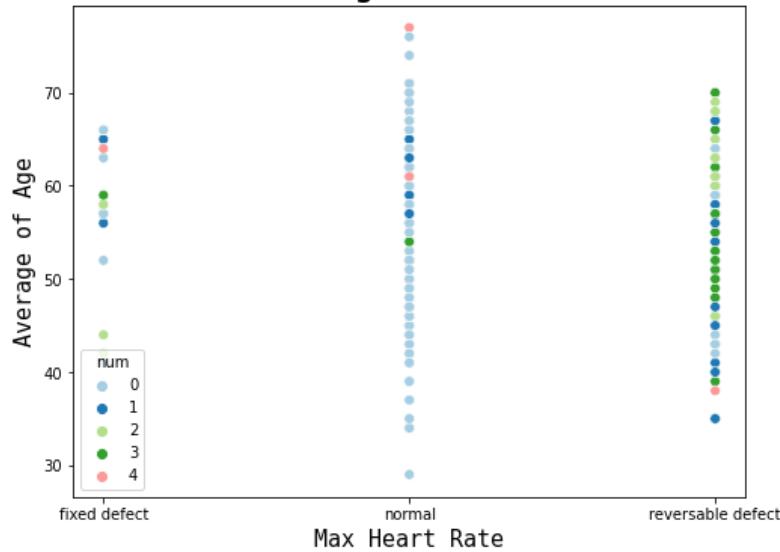
In [155]:

```
plt.figure(figsize=[8,6])
sb.scatterplot(data['thal'], data['age'], hue = data['num'], palette = 'Paired', s = 45)
plt.title('Average of Age for Heart Disease Stage relative to Maximum Heart Rate Achieved', fontdict={'fontname': 'Monospace', 'fontsize': 20, 'fontweight': 'bold'})
plt.xlabel('Max Heart Rate', fontdict={'fontname': 'Monospace', 'fontsize': 15,})
plt.ylabel('Average of Age', fontdict={'fontname': 'Monospace', 'fontsize': 15,})
plt.show()
```

c:\Users\Mannahil Miftah\anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

```
warnings.warn(
```

Average of Age for Heart Disease Stage relative to Maximum Heart Rate Achieved



In [157]:

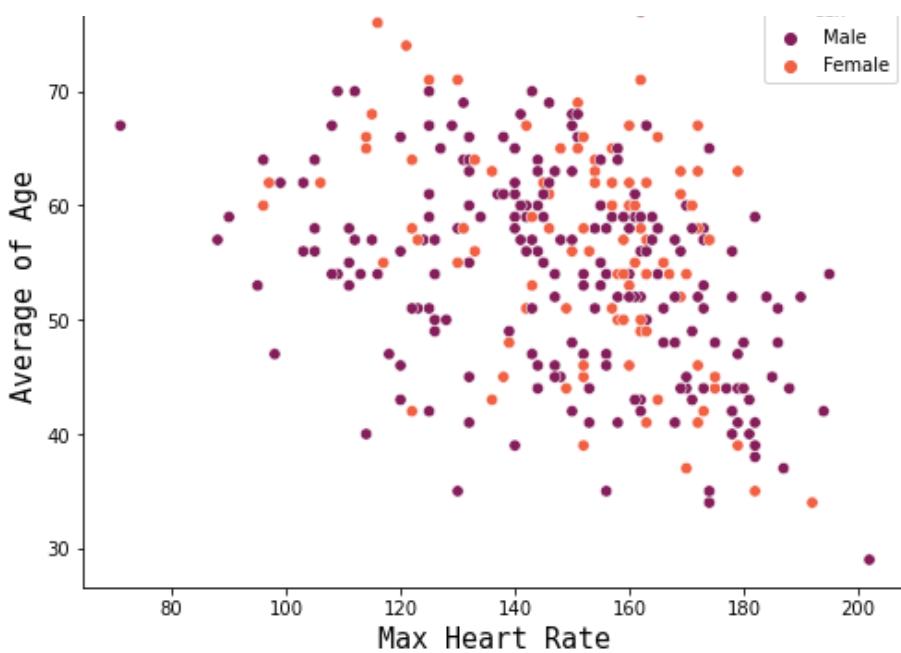
```
plt.figure(figsize=[8,6])
sb.scatterplot(data['thalch'], data['age'], hue = data['sex'], palette = 'rocket', s = 40)
plt.title('Average of Age for Sex relative to Maximum Heart Rate Achieved', fontdict={'fontname': 'Monospace', 'fontsize': 20, 'fontweight': 'bold'})
plt.xlabel('Max Heart Rate', fontdict={'fontname': 'Monospace', 'fontsize': 15,})
plt.ylabel('Average of Age', fontdict={'fontname': 'Monospace', 'fontsize': 15,})
plt.show()
```

c:\Users\Mannahil Miftah\anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

```
warnings.warn(
```

Average of Age for Sex relative to Maximum Heart Rate Achieved





Correlation

In [111]:

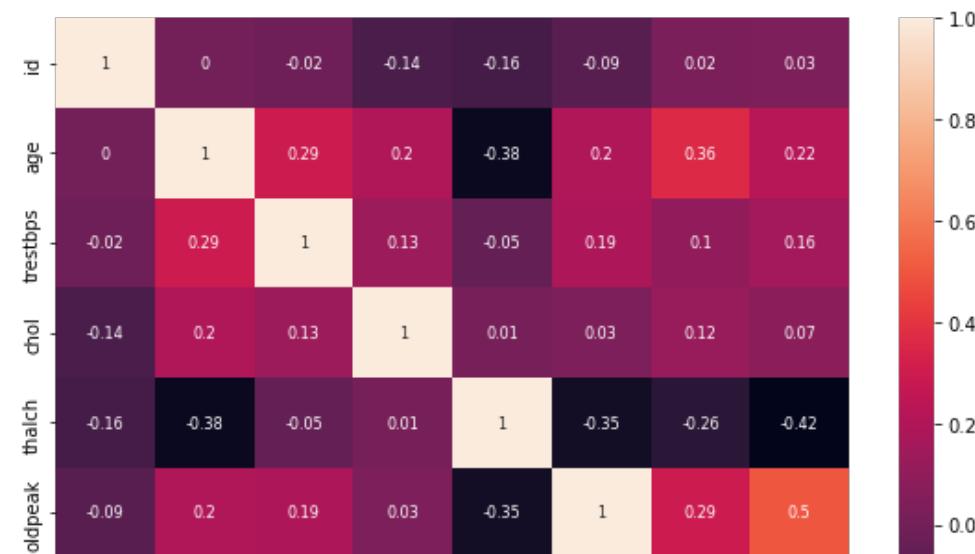
```
data.corr()
```

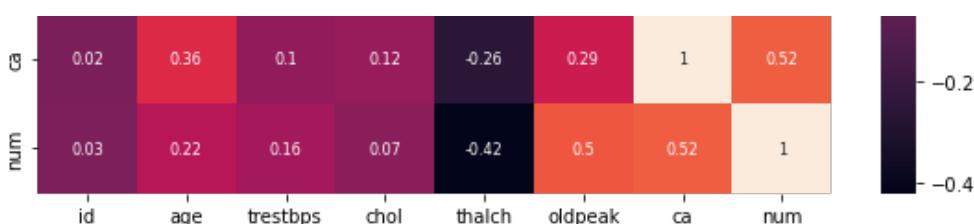
Out[111]:

	id	age	trestbps	chol	thalch	oldpeak	ca	num
id	1.000000	0.001379	-0.021051	-0.138639	-0.159716	-0.091294	0.020103	0.031397
age	0.001379	1.000000	0.286149	0.199258	-0.384176	0.195929	0.362764	0.221787
trestbps	-0.021051	0.286149	1.000000	0.134240	-0.053320	0.191144	0.096641	0.159272
chol	-0.138639	0.199258	0.134240	1.000000	0.014894	0.033964	0.121907	0.065081
thalch	-0.159716	-0.384176	-0.053320	0.014894	1.000000	-0.348089	-0.256831	-0.416480
oldpeak	-0.091294	0.195929	0.191144	0.033964	-0.348089	1.000000	0.291958	0.501325
ca	0.020103	0.362764	0.096641	0.121907	-0.256831	0.291958	1.000000	0.520058
num	0.031397	0.221787	0.159272	0.065081	-0.416480	0.501325	0.520058	1.000000

In [113]:

```
plt.figure(figsize=[8,6])
sb.heatmap(data = data.corr().round(2), annot = True, annot_kws = {"size":8})
plt.tight_layout()
plt.show()
```





Finding significant difference in cholesterol levels between patients with and without heart disease

In [114]:

```
disease = data[data['num']>0]['chol']
```

In [115]:

```
no_disease = data[data['num']==0]['chol']
```

In [116]:

```
t_stats, p_value = stats.ttest_ind(disease, no_disease)
print("T-Statistic:", t_stats)
print("P-Value:", p_value)
```

T-Statistic: 1.1633312359897947

P-Value: 0.24562891817755445

In [117]:

```
def confidence_interval(data):
    mean = np.mean(data)
    std = np.std(data)
    num = len(data)
    std_error = std/np.sqrt(num)
    interval = stats.t.interval(0.95, num-1, loc = mean, scale = std_error)
    return interval
```

In [118]:

```
interval_disease = confidence_interval(disease)
print("Confidence interval for cholesterol levels with heart disease", interval_disease)
```

Confidence interval for cholesterol levels with heart disease (242.0503566005043, 259.10072253618637)

In [119]:

```
interval_nodisease = confidence_interval(no_disease)
print("Confidence interval for cholesterol levels without heart disease", interval_nodisease)
```

Confidence interval for cholesterol levels without heart disease (235.12646978204384, 251.86103021795617)

In [140]:

```
# plotting the results

plt.figure(figsize=[8,6])
labels = ['With Heart Disease', 'Without Heart Disease']
values = [disease.mean(), no_disease.mean()]
plt.bar(labels, values, color = 'yellow')
plt.title('Comparison of Mean Cholesterol Level', fontdict={'fontname': 'Monospace', 'font size': 20, 'fontweight': 'bold'})
plt.xlabel('Detection of Disease', fontdict={'fontname': 'Monospace', 'fontsize': 15,})
plt.ylabel('Mean Cholesterol Level', fontdict={'fontname': 'Monospace', 'fontsize': 15,})
plt.show()
```

Comparison of Mean Cholesterol Level



From the above analysis and findings we can conclude that there is no significant difference in cholesterol levels

between patients with and without heart disease.

In []: