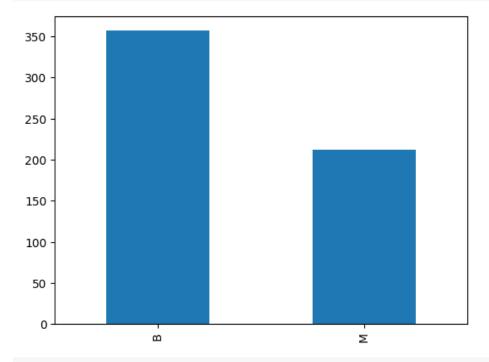
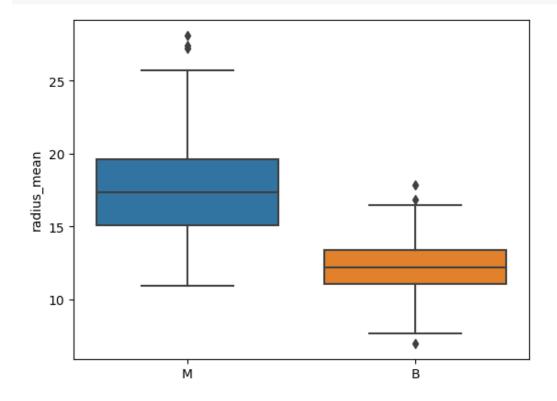
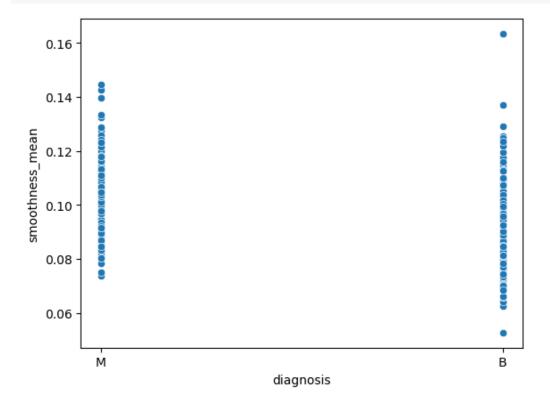
```
#targets distribution
data['diagnosis'].value_counts().plot.bar()
plt.show()
```



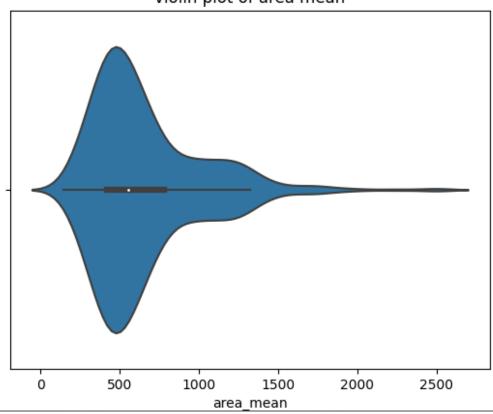


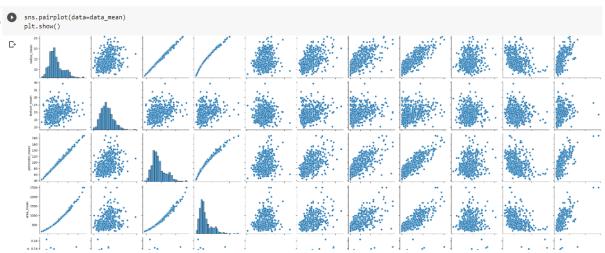
```
sns.scatterplot(y='smoothness\_mean', x='diagnosis', data=data)\\ plt.show()
```

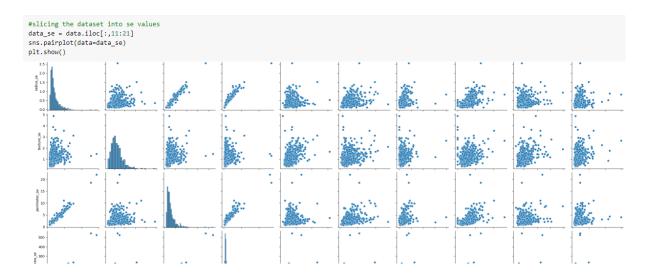


```
sns.violinplot(x='area_mean',data=data)
plt.title('violin plot of area mean')
plt.show()
```

## violin plot of area mean







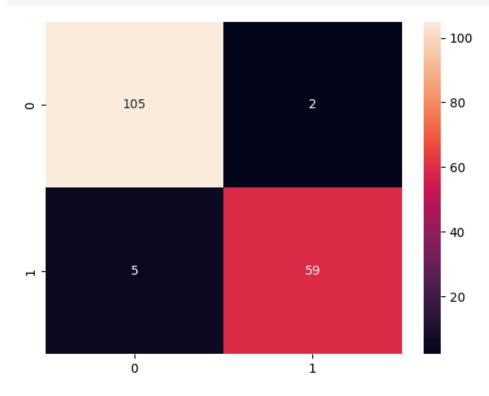
#correlation plot
plt.figure(figsize=(20,20))
sns.heatmap(data.corr(),annot=True)
plt.show()

<ipython-input-57-a714657ce2a8>:3: FutureWarning: The default value of numeric\_only in DataFrame.corr is deprecated. In a future version, it will default to Fai sns.heatmap(data.corr(),annot=True)

radius_mean -	- 1	0.32	1	0.99	0.17	0.51	0.68	0.82	0.15	-0.31	0.68	-0.097	0.67	0.74	-0.22	0.21	0.19	0.38	-0.1	-0.043	0.97	0.3	0.97	0.94	0.12	0.41	0.53	0.74	0.16	0.0071
texture_mean -	0.32	1	0.33	0.32	-0.023			0.29	0.071	-0.076				0.26	0.0066				0.0091	10.054	0.35	0.91	0.36	0.34	0.078			0.3		0.12
perimeter_mean -	1	0.33	1	0.99	0.21	0.56	0.72	0.85	0.18	-0.26		-0.087	0.69	0.74	-0.2				-0.082	0.005	0.97	0.3	0.97	0.94	0.15			0.77	0.19	0.051
area_mean -	0.99	0.32	0.99	1	0.18			0.82	0.15	-0.28	0.73	0.066	0.73	0.8	-0.17				-0.072	-0.02	0.96	0.29	0.96	0.96	0.12			0.72	0.14	0.0037
smoothness_mean -	0.17	-0.023			1	0.66						0.068										0.036			0.81					0.5
compactness_mean	0.51	0.24				1	0.88	0.83				0.046			0.14	0.74										0.87	0.82	0.82	0.51	0.69
concavity_mean	0.68	0.3	0.72			0.88	1	0.92	0.5			0.076	0.66		0.099				0.18				0.73			0.75	0.88	0.86	0.41	0.51
concave points_mean	0.82	0.29	0.85	0.82	0.55	0.83	0.92	1	0.46			0.021	0.71		0.028				0.095		0.83	0.29	0.86	0.81			0.75	0.91	0.38	0.37
symmetry_mean -	0.15	0.071	0.18	0.15	0.56		0.5	0.46	1	0.48	0.3		0.31	0.22							0.19	0.091	0.22	0.18				0.43		0.44
fractal_dimension_mean -	-0.31	-0.076	-0.26	-0.28	0.58	0.57	0.34	0.17	0.48	1	.0001	10.16	0.04	-0.09	0.4	0.56	0.45	0.34	0.35	0.69	-0.25	-0.051	-0.21	-0.23	0.5	0.46	0.35	0.18	0.33	0.77

- 0.8

```
#confusion matrix
sns.heatmap(confusion_matrix(y_test,preds),annot=True,fmt='g')
plt.show()
```



```
#confusion matrix
sns.heatmap(confusion_matrix_qda,annot=True,fmt='g')
plt.show()
```

