

**Московский государственный технический университет им.
Н.Э. Баумана
Факультет «Информатика и системы управления»
Кафедра «Автоматизированные системы обработки информации и
управления»**



**Отчет по лабораторной работе
№ 1**

**«Разведочный анализ данных. Исследование и
визуализация данных»**

**По курсу
“ Методы машинного обучения ”**

**Выполнил:
Али Диб А.Ж.
Студент группы ИУ5-22М**

Москва, 2020

1) Текстовое описание набора данных

В качестве набора данных мы будем использовать набор данных для выявления рака молочной железы. Ссылка: [https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+\(Diagnostic\)](https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)).

Датасет содержит следующие колонки:

- **radius** (среднее расстояние от центра до точек по периметру)
- **texture** (стандартное отклонение значений оттенков серого)
- **perimeter**
- **area**
- **smoothness** (локальное изменение длины радиуса)
- **compactness** ($\frac{\text{периметр}^2}{\text{площадь}-1,0}$)
- 7. **concavity** (выраженность вогнутых участков контура)
- 8. **concave points** (количество вогнутых частей контура)
- 9. **symmetry**
- 10. **fractal dimension** («приближение береговой линии» - 1)

Среднее значение, стандартная ошибка и «наихудшее» или наибольшее (среднее из трех) были рассчитаны для каждого изображения, что дало 30 признаков. Например, поле 3 - среднее наихудший радиус.

▼ Импорт библиотек

Импортируем библиотеки с помощью команды `import`. Как правило, все команды `import` па:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.datasets import *
```

▼ Загрузка данных

Загрузим файлы датасета в помощью `sklearn` датасет.

```
cancer = load_breast_cancer()
```

```
for x in cancer:
```

```
print(x)
```

```
data
target
target_names
DESCR
feature_names
filename
```

```
cancer['feature_names']
```

```
array(['mean radius', 'mean texture', 'mean perimeter', 'mean area',
      'mean smoothness', 'mean compactness', 'mean concavity',
      'mean concave points', 'mean symmetry', 'mean fractal dimension',
      'radius error', 'texture error', 'perimeter error', 'area error',
      'smoothness error', 'compactness error', 'concavity error',
      'concave points error', 'symmetry error',
      'fractal dimension error', 'worst radius', 'worst texture',
      'worst perimeter', 'worst area', 'worst smoothness',
      'worst compactness', 'worst concavity', 'worst concave points',
      'worst symmetry', 'worst fractal dimension'], dtype='<U23')
```

```
cancer['data'].shape
```

```
(569, 30)
```

```
data1=pd.DataFrame(data=np.c_[cancer['data'],cancer['target']],
                   columns = list(cancer['feature_names']) + ['target'])
```

```
data1
```

▼ 2) Основные характеристики датасета

```
# Первые 5 строк датасета
data1.head()
```



	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points
0	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.1471
1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.0701
2	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.1279
3	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.1052
4	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.1043

5 rows × 31 columns

569 rows × 31 columns

```
# Размер датасета – 8143 строк, 7 колонок
data1.shape
```



(569, 31)

```
total_count = data1.shape[0]
print('Всего строк: {}'.format(total_count))
```



Всего строк: 569

```
# Список колонок
data1.columns
```



```
Index(['mean radius', 'mean texture', 'mean perimeter', 'mean area',
      'mean smoothness', 'mean compactness', 'mean concavity',
      'mean concave points', 'mean symmetry', 'mean fractal dimension',
      'radius error', 'texture error', 'perimeter error', 'area error',
      'smoothness error', 'compactness error', 'concavity error',
      'concave points error', 'symmetry error', 'fractal dimension error',
      'worst radius', 'worst texture', 'worst perimeter', 'worst area',
      'worst smoothness', 'worst compactness', 'worst concavity',
      'worst concave points', 'worst symmetry', 'worst fractal dimension',
      'target'],
      dtype='object')
```

```
# Список колонок с типами данных
data1.dtypes
```



```

mean radius          float64
mean texture         float64
mean perimeter       float64
mean area            float64
mean smoothness      float64
mean compactness     float64
mean concavity       float64
mean concave points  float64
mean symmetry        float64
mean fractal dimension float64
radius error         float64
texture error        float64
perimeter error      float64
area error           float64
smoothness error     float64
compactness error    float64
concavity error      float64
concave points error float64
symmetry error       float64
fractal dimension error float64
worst radius         float64
worst texture        float64
worst perimeter      float64
worst area           float64
worst smoothness     float64
worst compactness    float64
worst concavity      float64
worst concave points float64
worst symmetry       float64
worst fractal dimension float64
target              float64
dtype: object

```

```

# Проверим наличие пустых значений
# Цикл по колонкам датасета
for col in data1.columns:
    # Количество пустых значений - все значения заполнены
    temp_null_count = data1[data1[col].isnull()].shape[0]
    print('{} - {}'.format(col, temp_null_count))

```



```

mean radius - 0
mean texture - 0
mean perimeter - 0
mean area - 0
mean smoothness - 0
mean compactness - 0
mean concavity - 0
mean concave points - 0
mean symmetry - 0
mean fractal dimension - 0
radius error - 0
texture error - 0
perimeter error - 0
area error - 0
smoothness error - 0
compactness error - 0
concavity error - 0
concave points error - 0
symmetry error - 0
fractal dimension error - 0
worst radius - 0
worst texture - 0
worst perimeter - 0
worst area - 0
worst smoothness - 0
worst compactness - 0
worst concavity - 0
worst concave points - 0
worst symmetry - 0
worst fractal dimension - 0
target - 0

```

```

# Основные статистические характеристики набора данных
data1.describe()

```



	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	conc
count	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.
mean	14.127292	19.289649	91.969033	654.889104	0.096360	0.104341	0.
std	3.524049	4.301036	24.298981	351.914129	0.014064	0.052813	0.
min	6.981000	9.710000	43.790000	143.500000	0.052630	0.019380	0.
25%	11.700000	16.170000	75.170000	420.300000	0.086370	0.064920	0.
50%	13.370000	18.840000	86.240000	551.100000	0.095870	0.092630	0.
75%	15.780000	21.800000	104.100000	782.700000	0.105300	0.130400	0.
max	28.110000	39.280000	188.500000	2501.000000	0.163400	0.345400	0.

8 rows x 31 columns

```

#Определим уникальные значения для целевого признака

```

```
data1['target'].unique()
```



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

Целевой признак является бинарным и содержит только значения 0 и 1.

3) Визуальное исследование датасета

Для визуального исследования могут быть использованы различные виды диаграмм, буде диаграмм, которые используются достаточно часто.

Будет использовано две библиотеки:

- **Matplotlib**
- **Seaborn**

▼ Диаграмма рассеяния

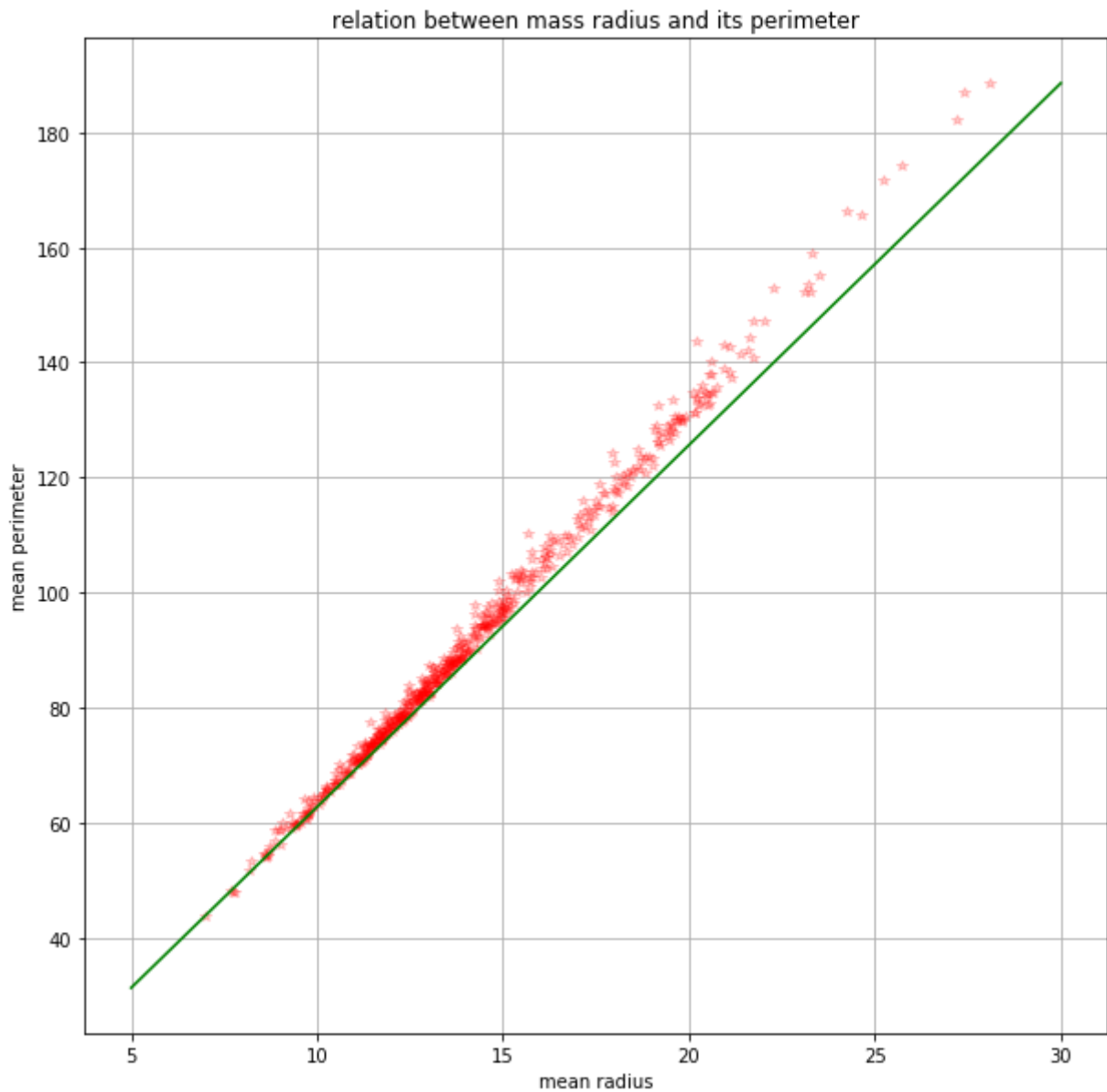
Позволяет построить распределение двух колонок данных и визуально обнаружить наличие значения упорядочены (например, по времени).

Matplotlib

```
fig=plt.figure(figsize=(10,10))
ax=fig.gca()
ax.grid()
plt.plot(data1['mean radius'],data1['mean perimeter'],'*r',alpha=0.2)
ax.set_xlabel('mean radius')
ax.set_ylabel('mean perimeter')
ax.set_title('relation between mass radius and its perimeter')
xxx=np.linspace(5,30)
yyy=2*np.pi*xxx
plt.plot(xxx,yyy,'g')
```



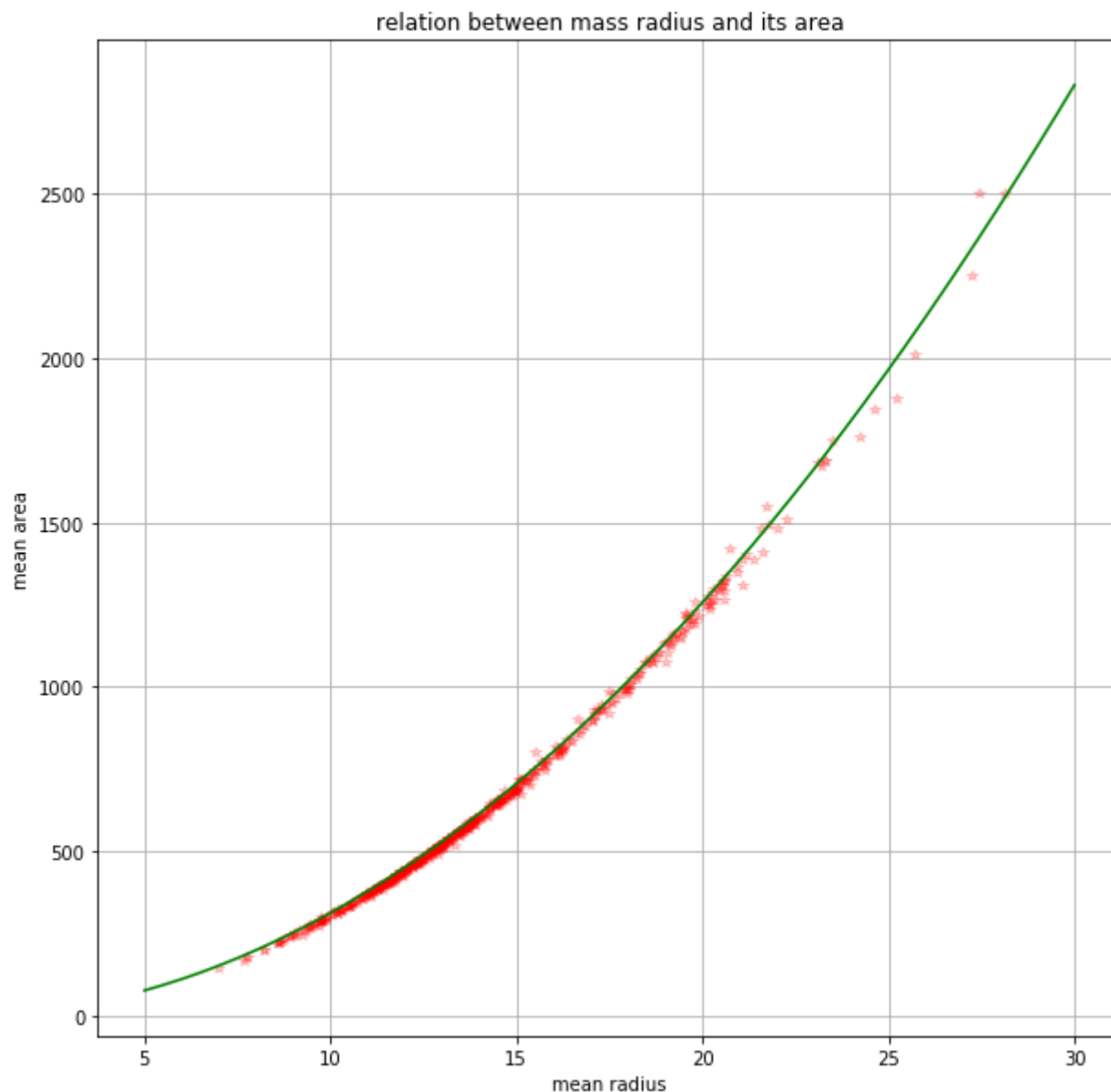
[<matplotlib.lines.Line2D at 0x13209f240>]



```
fig=plt.figure(figsize=(10,10))
ax=fig.gca()
ax.grid()
plt.plot(data1['mean radius'],data1['mean area'],'*r',alpha=0.2)
ax.set_xlabel('mean radius')
ax.set_ylabel('mean area')
ax.set_title('relation between mass radius and its area')
xxx=np.linspace(5,30)
yyy=np.pi*xxx**2
plt.plot(xxx,yyy,'g')
```




```
[<matplotlib.lines.Line2D at 0x1344e7b70>]
```

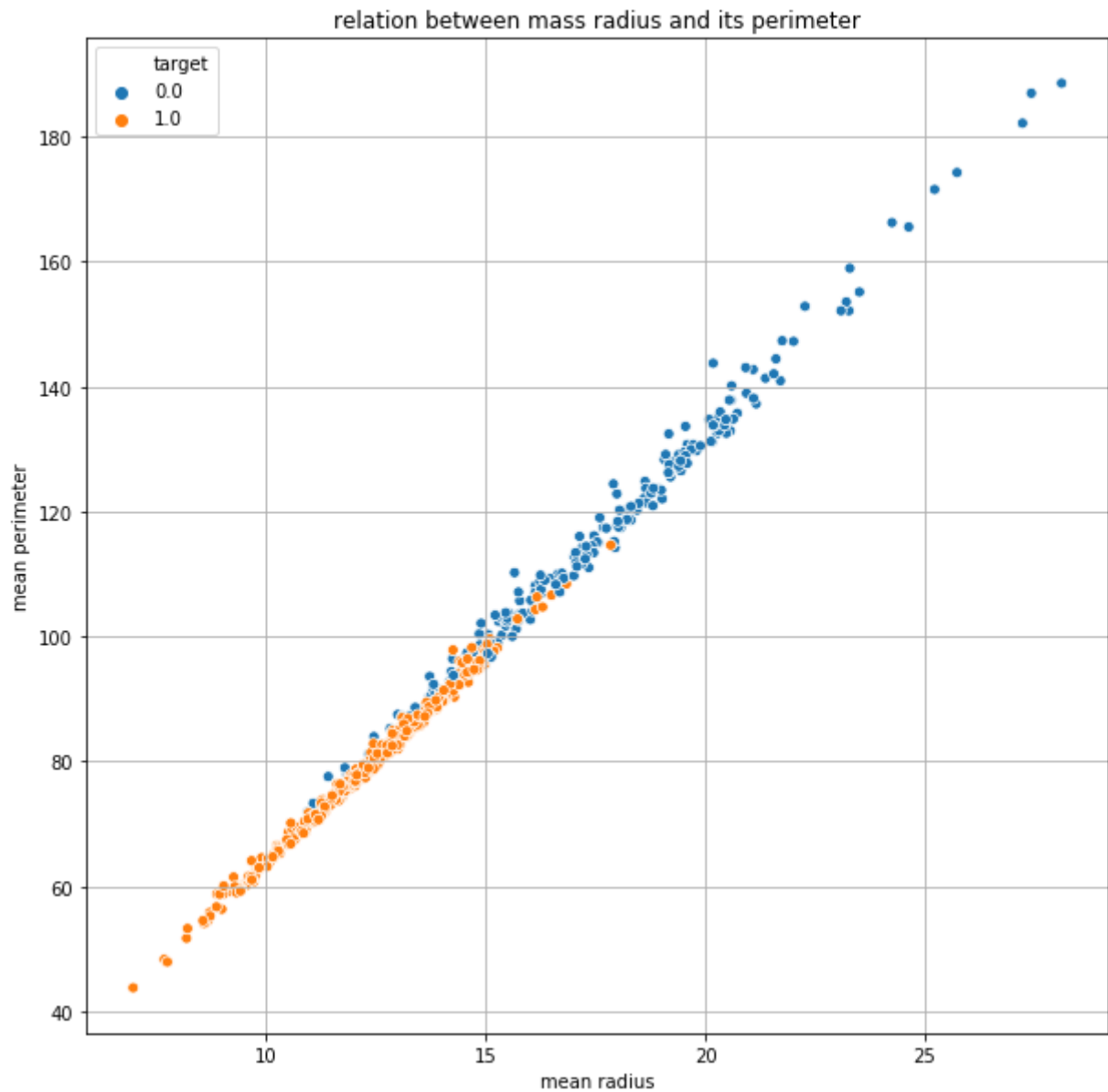


Seaborn

```
fig=plt.figure(figsize=(10,10))
ax=fig.gca()
ax.grid()
sns.scatterplot(data1['mean radius'],data1['mean perimeter'],color='r',hue=data1['t
ax.set_title('relation between mass radius and its perimeter')
```



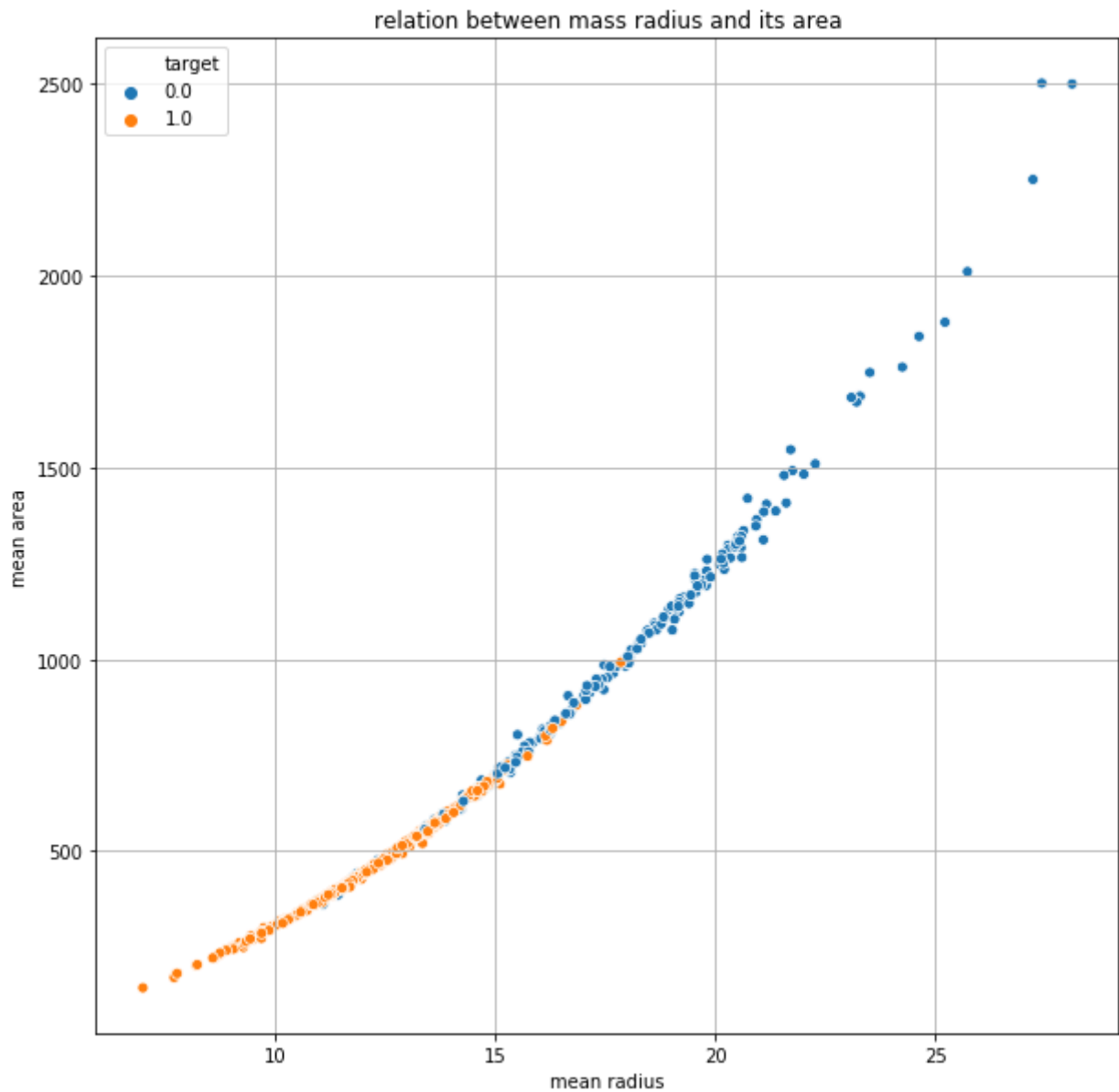
```
Text(0.5, 1.0, 'relation between mass radius and its perimeter')
```



```
fig=plt.figure(figsize=(10,10))
ax=fig.gca()
ax.grid()
sns.scatterplot(data1['mean radius'],data1['mean area'],color='r',hue=data1['target'])
ax.set_title('relation between mass radius and its area')
```



```
Text(0.5, 1.0, 'relation between mass radius and its area')
```



Можно видеть что между полями mean radius и mean perimeter присутствует почти линейн
Можно видеть что между полями mean radius и mean area присутствует почти параболичес

▼ Гистограмма

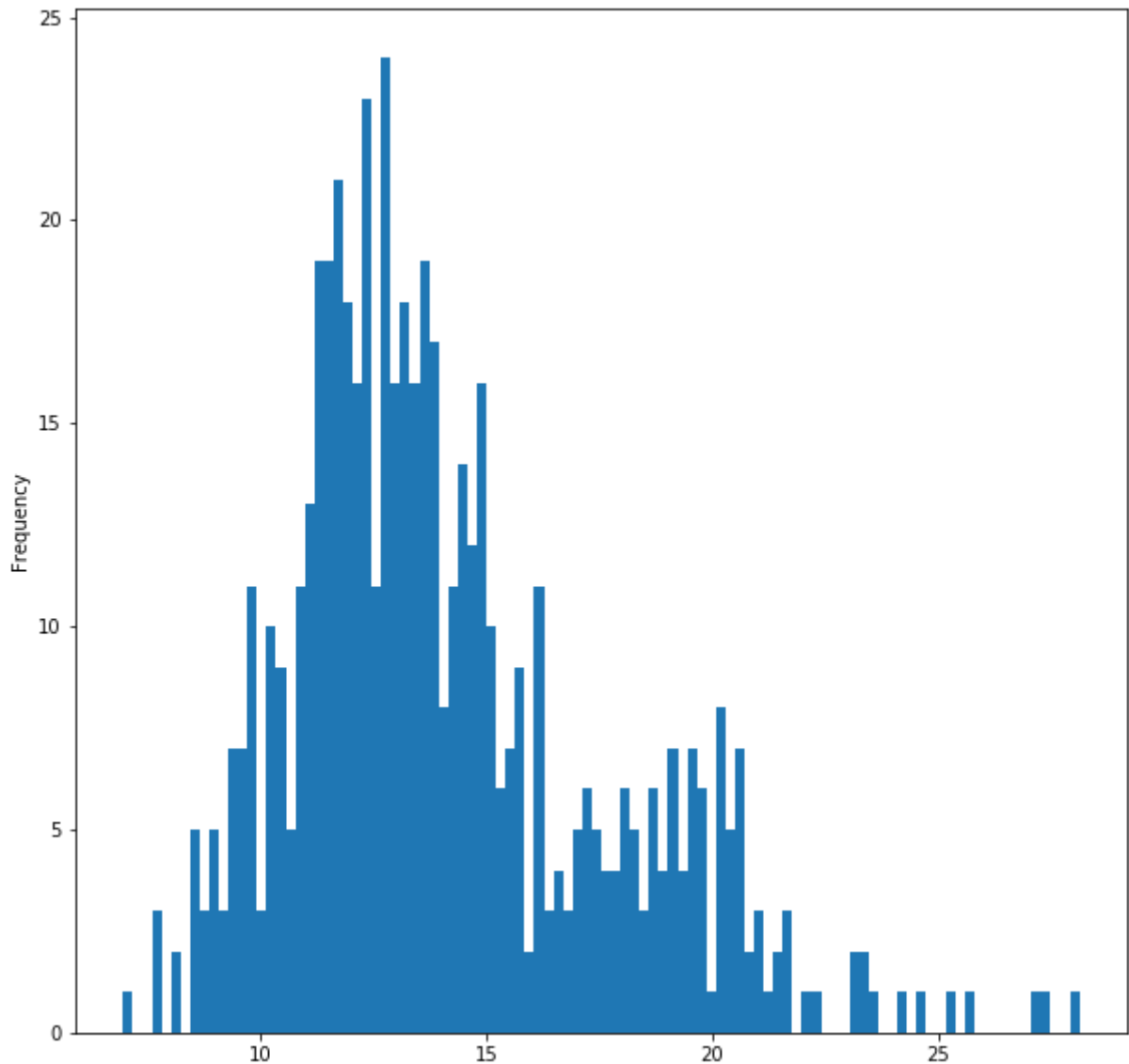
Позволяет оценить плотность вероятности распределения данных.

Matplotlib

```
fig=plt.figure(figsize=(10,10))
ax=fig.gca()
data1['mean radius'].plot.hist(bins=100)
```



<matplotlib.axes._subplots.AxesSubplot at 0x1347d9278>

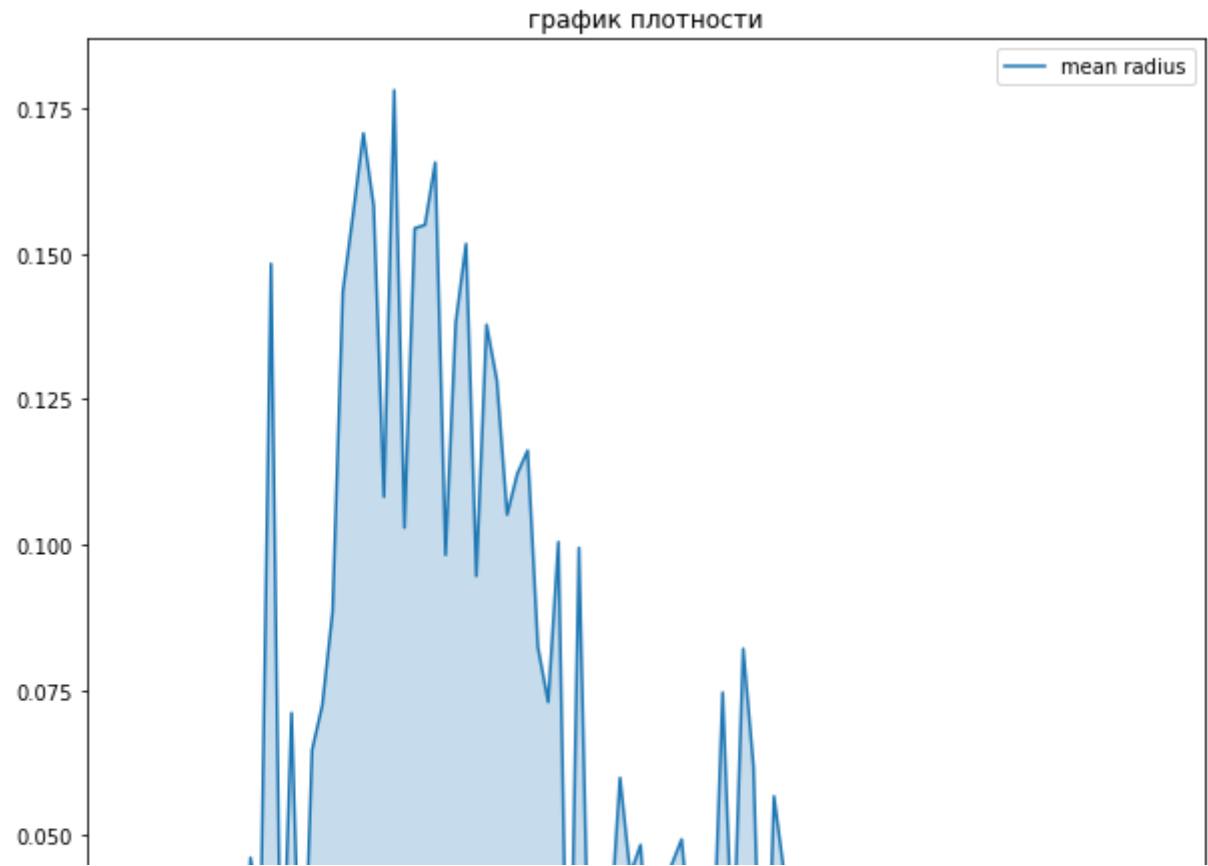
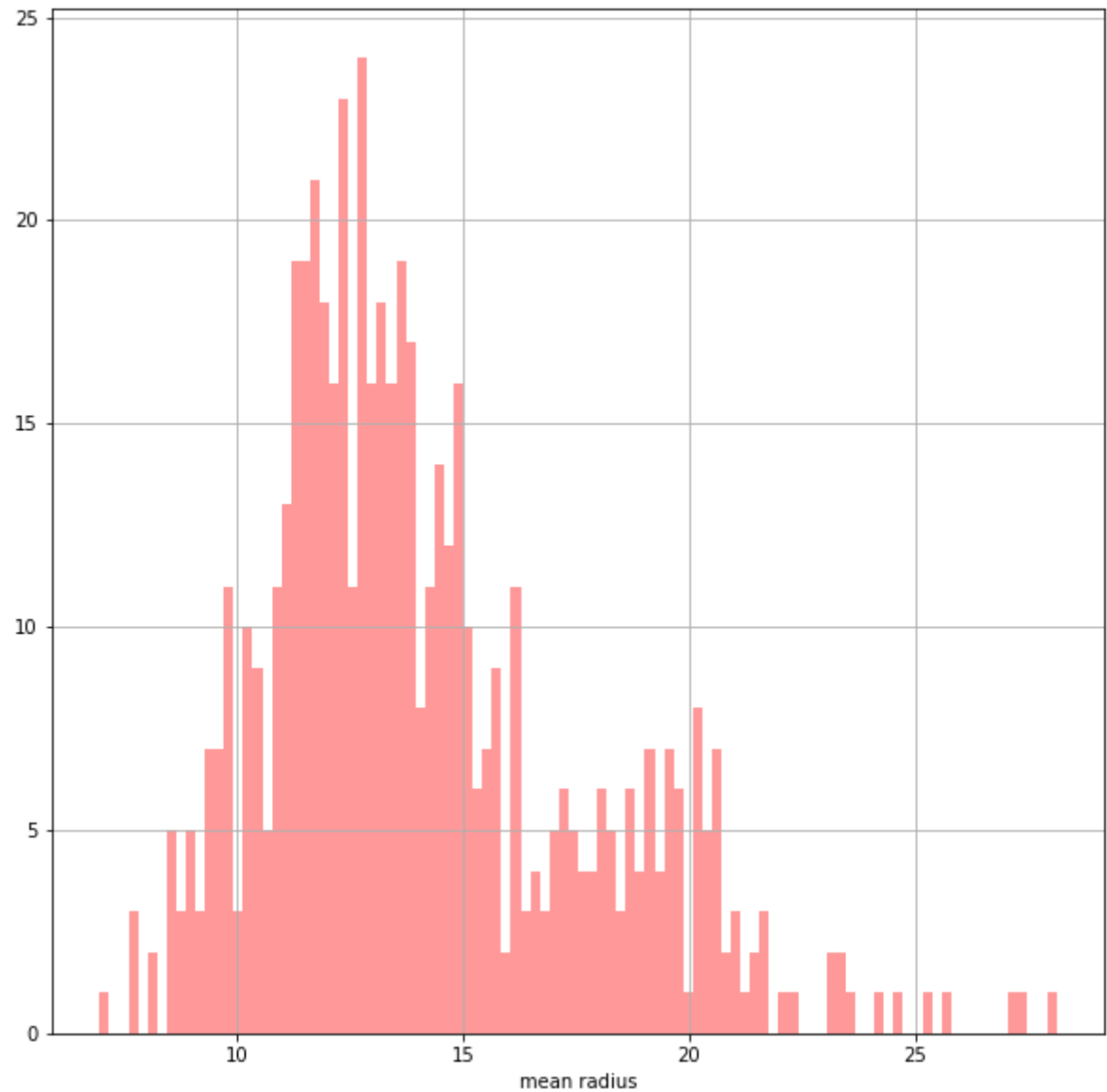


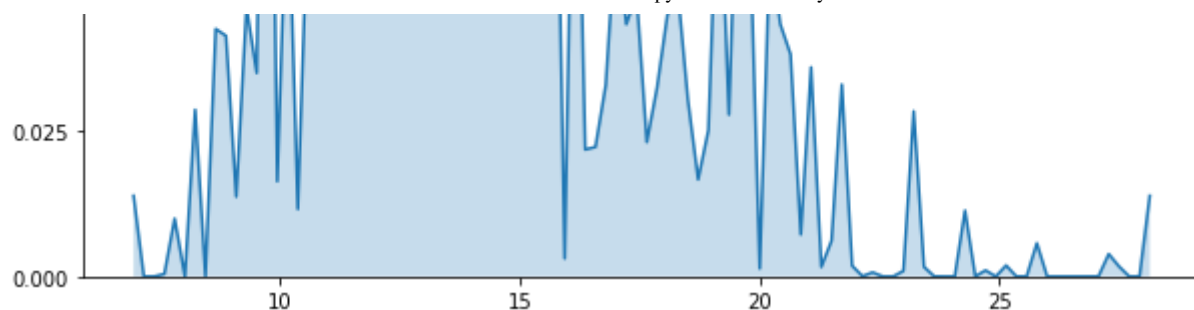
Seaborn

```
fig=plt.figure(1,figsize=(10,10))
ax=fig.gca()
ax.grid()
sns.distplot(data1['mean radius'],rug=False,kde=False,color='red',bins=100)
fig=plt.figure(figsize=(10,10))
ax=fig.gca()
sns.kdeplot(data1['mean radius'], shade=True,bw=0.01);
ax.set_title('график плотности')
```




Text(0.5, 1.0, 'график плотности')

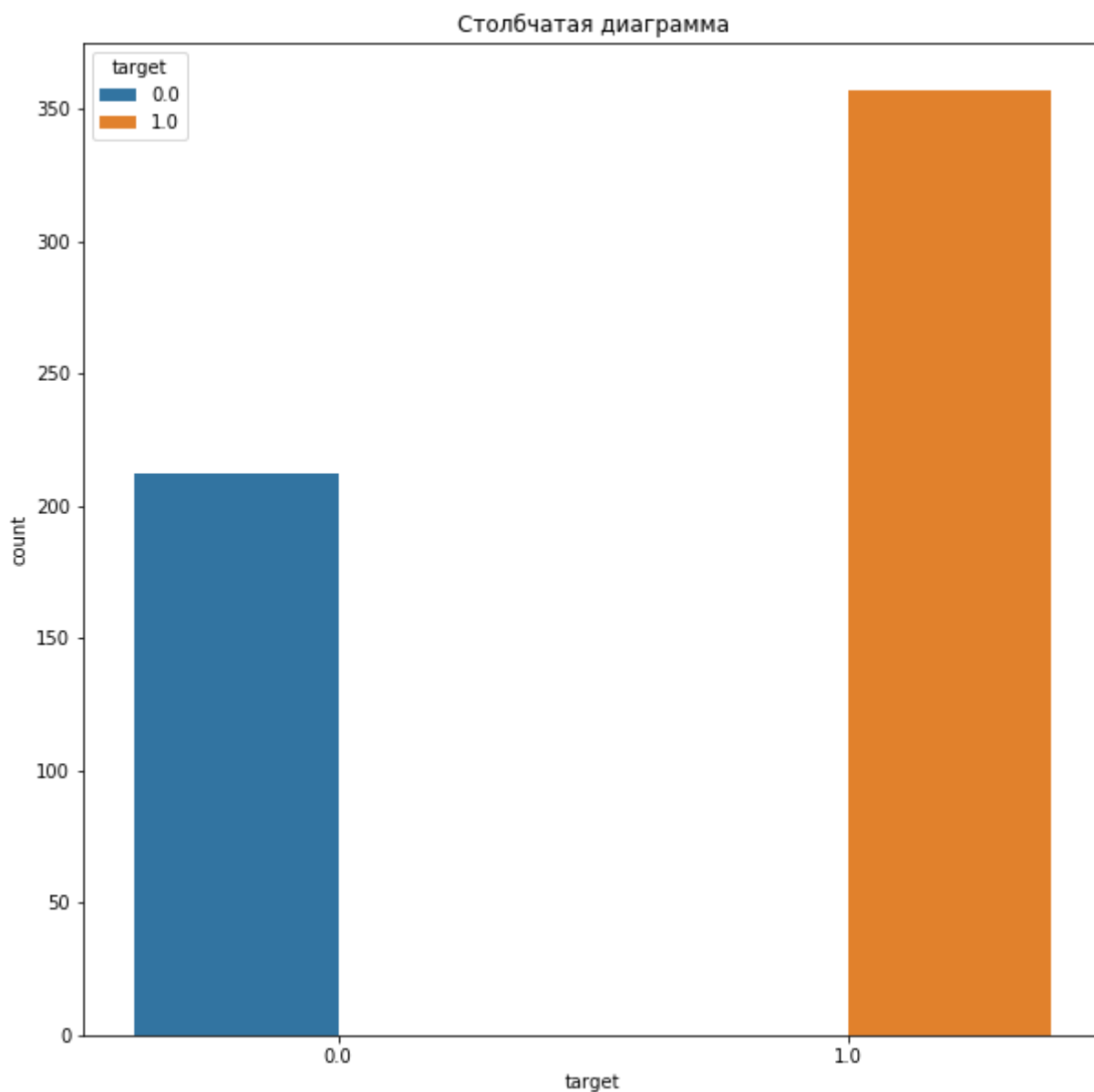




▼ Столбчатая диаграмма

```
fig=plt.figure(figsize=(10,10))
ax=fig.gca()
sns.countplot(x=data1['target'],hue=data1['target'])
ax.set_title('Столбчатая диаграмма')
```

 Text(0.5, 1.0, 'Столбчатая диаграмма')



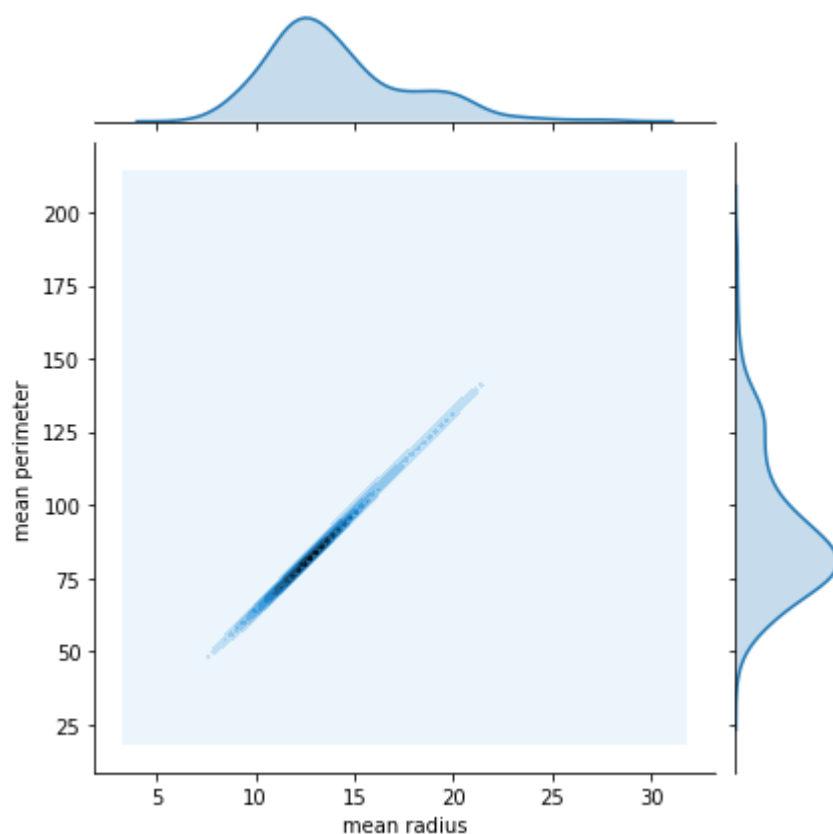
Этот график показывает, когда у человека есть опухоль, какой процент является злокачественным независимо от характеристик опухоли.

▼ Joinplot

Комбинация гистограмм и диаграмм рассеивания.

```
sns.jointplot(data1['mean radius'],data1['mean perimeter'],kind='kde')
```

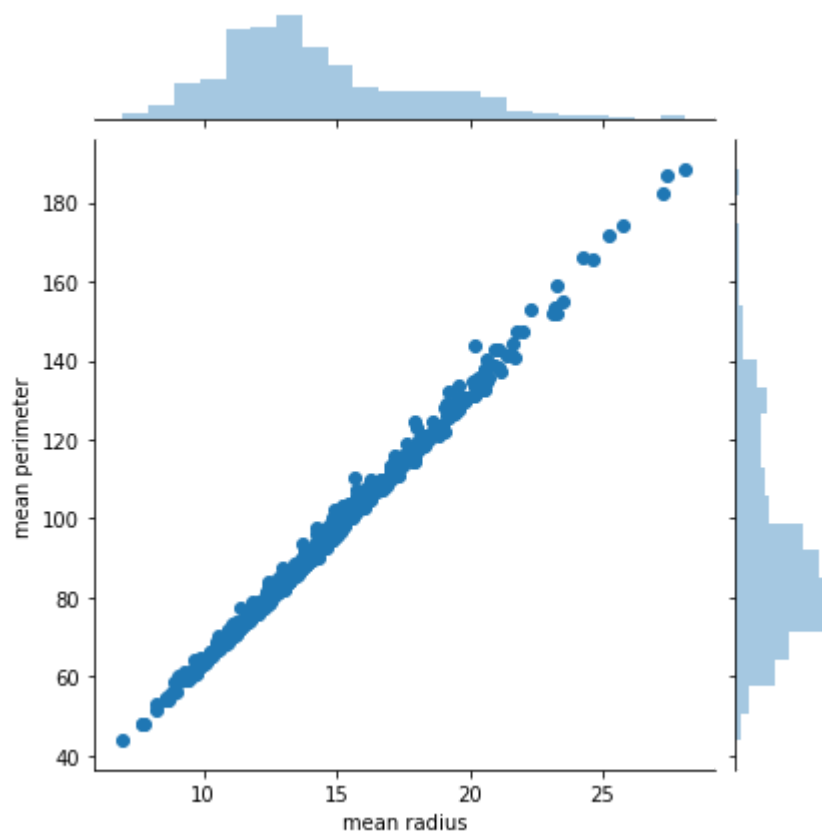
 <seaborn.axisgrid.JointGrid at 0x132052898>



```
sns.jointplot(data1['mean radius'],data1['mean perimeter'],kind='scatter')
```



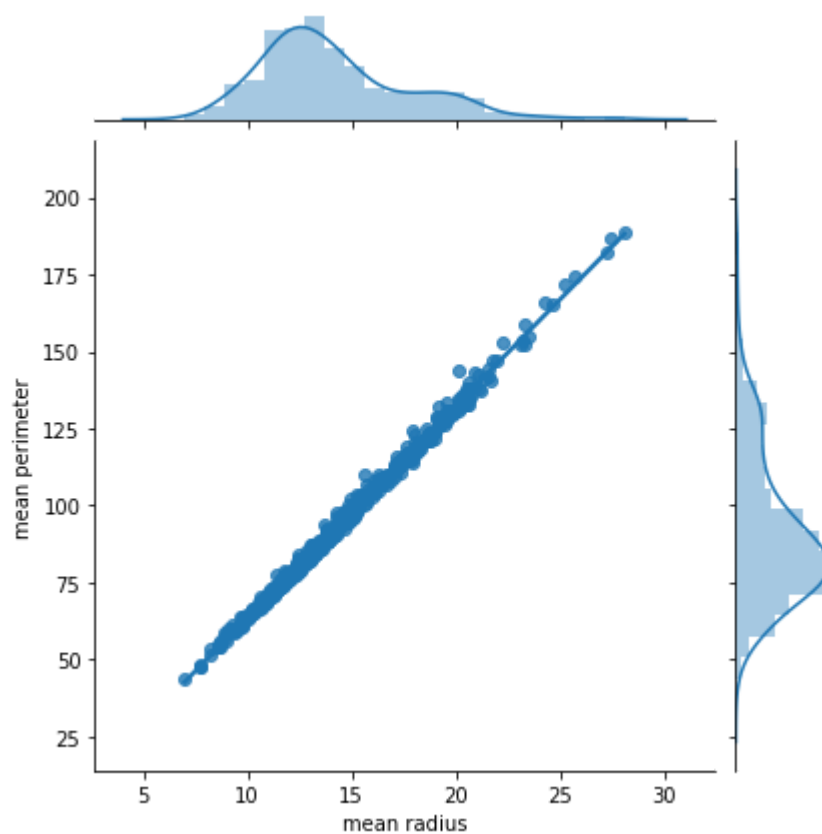
<seaborn.axisgrid.JointGrid at 0x1381bcb38>



```
sns.jointplot(data1['mean radius'],data1['mean perimeter'],kind='reg')
```



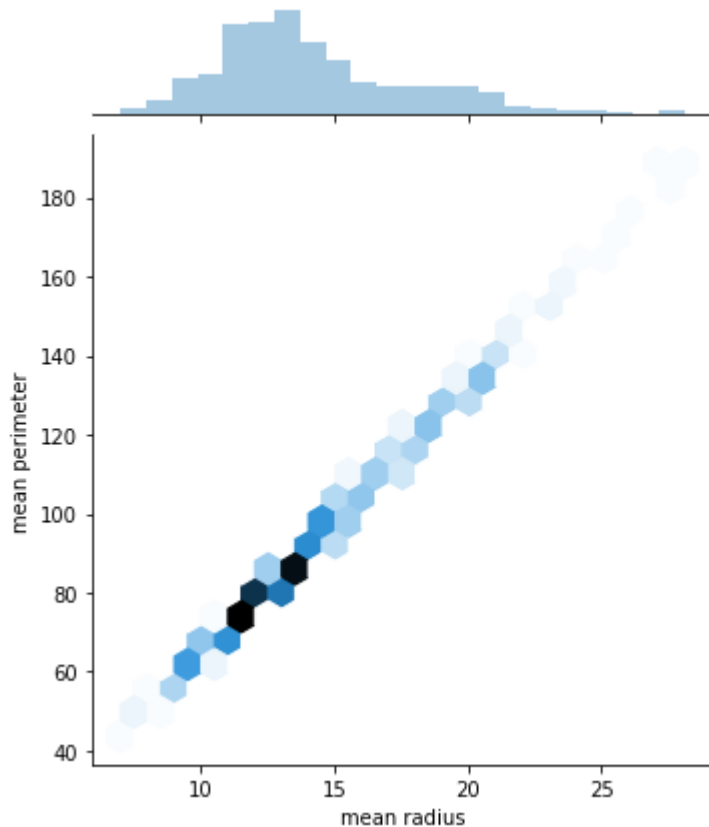
<seaborn.axisgrid.JointGrid at 0x1385934e0>



```
sns.jointplot(data1['mean radius'],data1['mean perimeter'],kind='hex')
```




<seaborn.axisgrid.JointGrid at 0x13909a128>



▼ Парные диаграммы

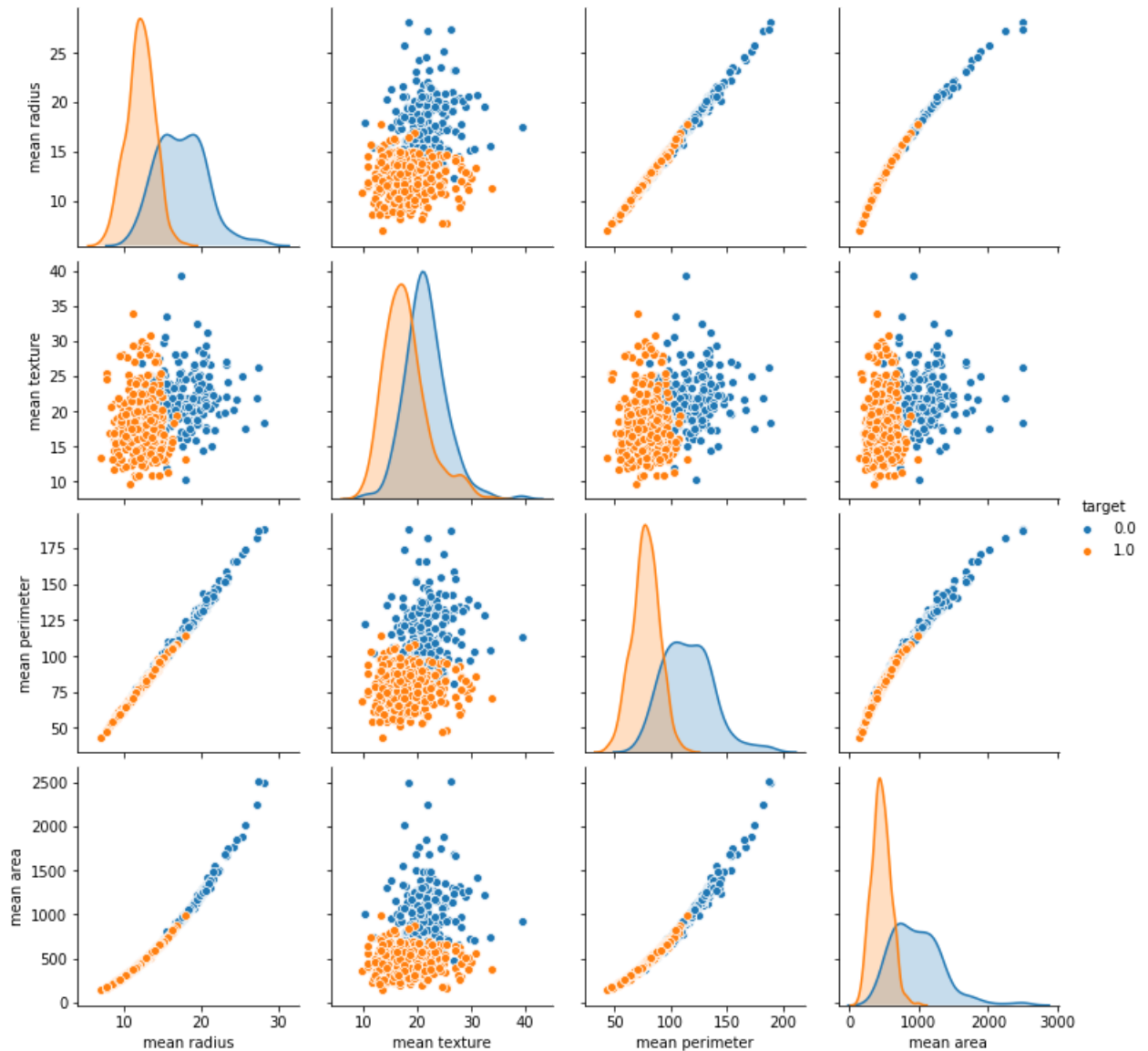
Комбинация гистограмм и диаграмм рассеивания для всего набора данных.

Выводится матрица графиков. На пересечении строки и столбца, которые соответствуют двум переменным, строится диаграмма рассеивания. В главной диагонали матрицы строятся гистограммы распределения соответствующих переменных.

```
sns.pairplot(data1[['mean radius', 'mean texture', 'mean perimeter', 'mean area', 'target']])
```



```
<seaborn.axisgrid.PairGrid at 0x135956550>
```



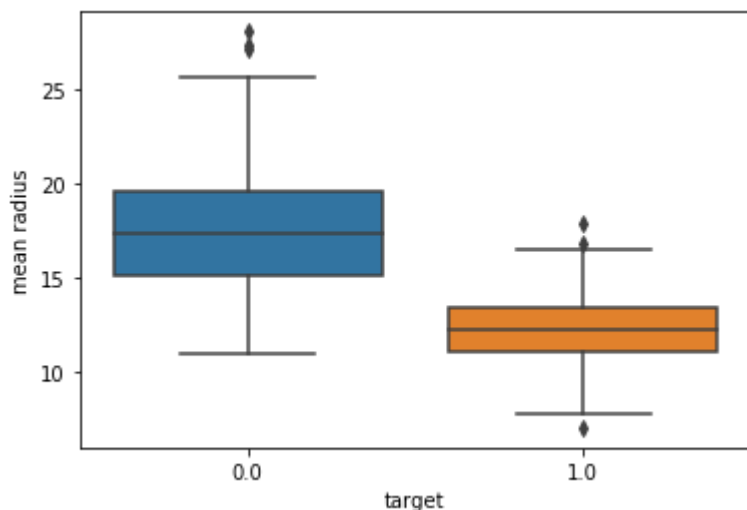
▼ Ящик с усами

Отображает одномерное распределение вероятности.

```
sns.boxplot(x='target', y='mean radius', data=data1)
```



<matplotlib.axes._subplots.AxesSubplot at 0x139c00cf8>



мы можем видеть, что медиана "mean radius" доброкачественной опухоли меньше, чем мед

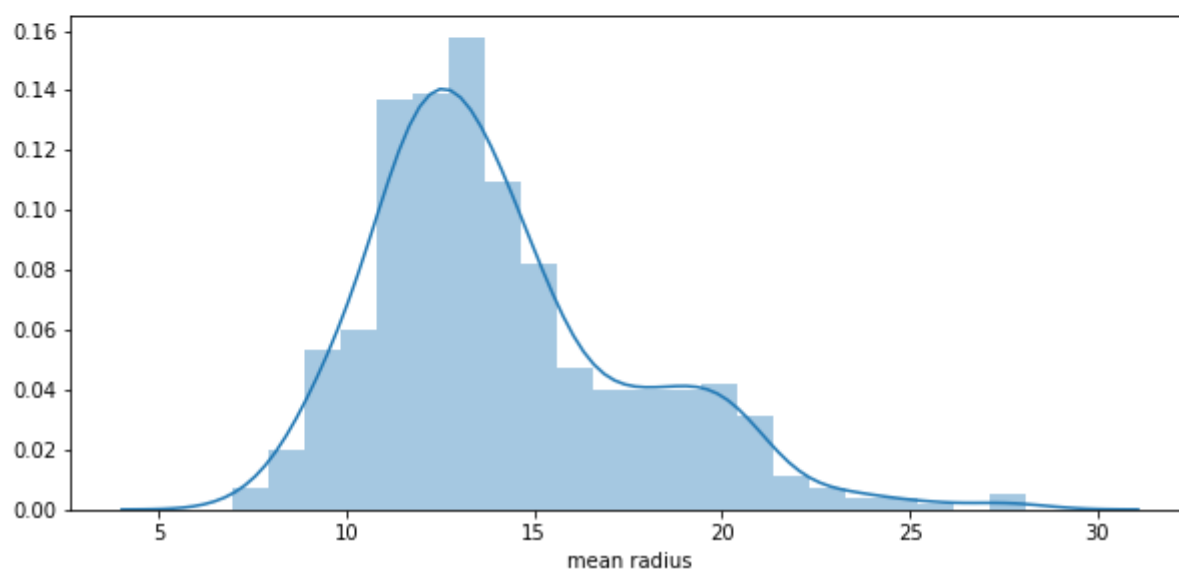
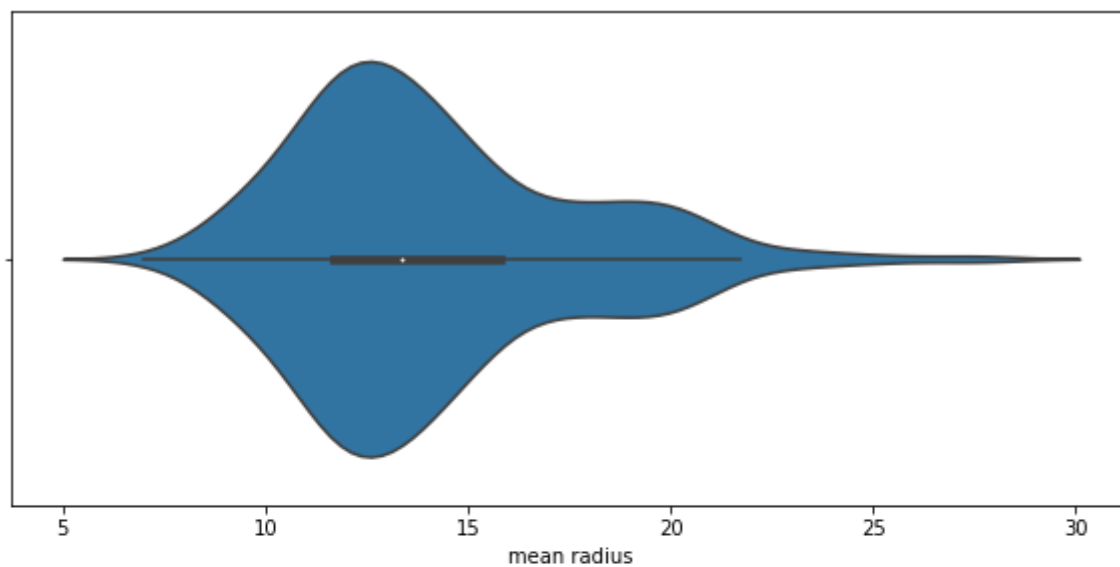
▼ Violin plot

Похоже на предыдущую диаграмму, но по краям отображаются распределения плотности.

```
fig, ax = plt.subplots(2, 1, figsize=(10,10))
sns.violinplot(ax=ax[0], x=data1['mean radius'])
sns.distplot(data1['mean radius'], ax=ax[1])
```



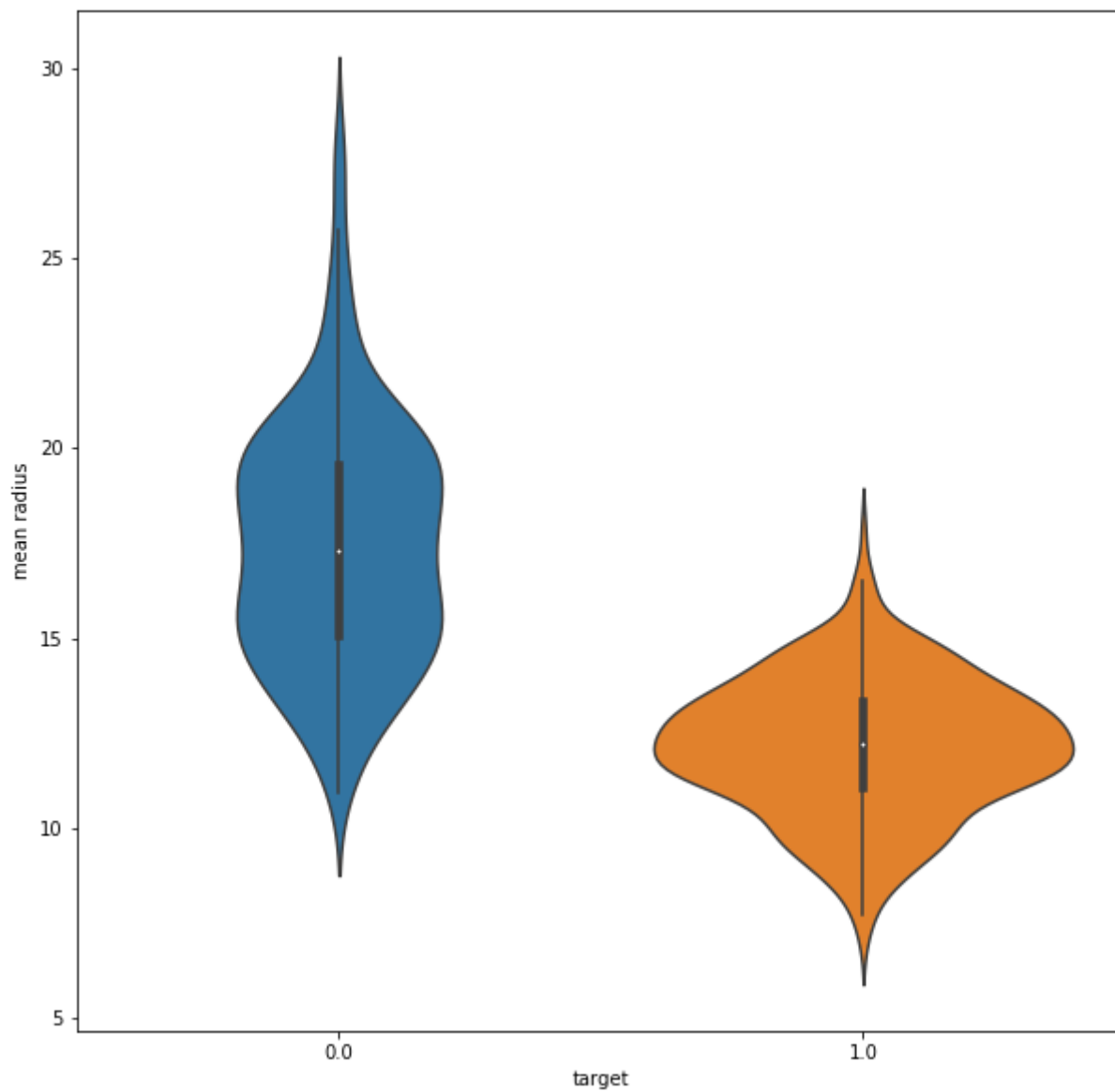
<matplotlib.axes._subplots.AxesSubplot at 0x139fa9198>



```
fig=plt.figure(figsize=(10,10))
ax=fig.gca()
sns.violinplot(x=data1['target'], y=data1['mean radius'], data=data1)
```



<matplotlib.axes._subplots.AxesSubplot at 0x13a0a2390>



4) Информация о корреляции признаков

```
data1.corr()
```



	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	co
mean radius	1.000000	0.323782	0.997855	0.987357	0.170581	0.506124	
mean texture	0.323782	1.000000	0.329533	0.321086	-0.023389	0.236702	
mean perimeter	0.997855	0.329533	1.000000	0.986507	0.207278	0.556936	
mean area	0.987357	0.321086	0.986507	1.000000	0.177028	0.498502	
mean smoothness	0.170581	-0.023389	0.207278	0.177028	1.000000	0.659123	
mean compactness	0.506124	0.236702	0.556936	0.498502	0.659123	1.000000	
mean concavity	0.676764	0.302418	0.716136	0.685983	0.521984	0.883121	
mean concave points	0.822529	0.293464	0.850977	0.823269	0.553695	0.831135	
mean symmetry	0.147741	0.071401	0.183027	0.151293	0.557775	0.602641	
mean fractal dimension	-0.311631	-0.076437	-0.261477	-0.283110	0.584792	0.565369	
radius error	0.679090	0.275869	0.691765	0.732562	0.301467	0.497473	
texture error	-0.097317	0.386358	-0.086761	-0.066280	0.068406	0.046205	
perimeter error	0.674172	0.281673	0.693135	0.726628	0.296092	0.548905	
area error	0.735864	0.259845	0.744983	0.800086	0.246552	0.455653	
smoothness error	-0.222600	0.006614	-0.202694	-0.166777	0.332375	0.135299	
compactness error	0.206000	0.191975	0.250744	0.212583	0.318943	0.738722	
concavity error	0.194204	0.143293	0.228082	0.207660	0.248396	0.570517	
concave points error	0.376169	0.163851	0.407217	0.372320	0.380676	0.642262	
symmetry error	-0.104321	0.009127	-0.081629	-0.072497	0.200774	0.229977	
fractal dimension error	-0.042641	0.054458	-0.005523	-0.019887	0.283607	0.507318	
worst radius	0.969539	0.352573	0.969476	0.962746	0.213120	0.535315	
worst texture	0.297008	0.912045	0.303038	0.287489	0.036072	0.248133	
worst	0.969539	0.912045	0.969476	0.962746	0.213120	0.535315	

perimeter	0.965137	0.358040	0.970387	0.959120	0.238853	0.590210	
worst area	0.941082	0.343546	0.941550	0.959213	0.206718	0.509604	
worst smoothness	0.119616	0.077503	0.150549	0.123523	0.805324	0.565541	
worst compactness	0.413463	0.277830	0.455774	0.390410	0.472468	0.865809	
worst concavity	0.526911	0.301025	0.563879	0.512606	0.434926	0.816275	
worst concave points	0.744214	0.295316	0.771241	0.722017	0.503053	0.815573	
worst symmetry	0.163953	0.105008	0.189115	0.143570	0.394309	0.510223	
worst fractal dimension	0.007066	0.119205	0.051019	0.003738	0.499316	0.687382	
target	-0.730029	-0.415185	-0.742636	-0.708984	-0.358560	-0.596534	-

31 rows x 31 columns

Корреляционная матрица содержит коэффициенты корреляции между всеми парами признаков. Корреляционная матрица симметрична относительно главной диагонали. На главной диагонали значения равны 1 (каждый признак сам с собой).

На основе корреляционной матрицы можно сделать следующие выводы:

- Целевой признак наиболее сильно коррелирует с "worst concave points" (-0.794) и "worst compactness" (-0.597). Эти признаки обязательно следует оставить в модели.
- Целевой признак отчасти коррелирует с "mean compactness" (-0.597), "radius error" (-0.561), "texture" (-0.547), "worst smoothness" (-0.421), "worst symmetry" (-0.416). Этот признак стоит рассмотреть.
- Целевой признак слабо коррелирует с "mean smoothness" (-0.359), "mean symmetry" (-0.359), "texture error" (0.008), "smoothness error" (0.067), "Compactness error" (-0.293), "concavity error" (-0.293), "fractal dimension error" (-0.078) и "worst fractal dimension" (-0.324). Скорее всего эти признаки возможно они только ухудшат качество модели.
- "mean radius" и "mean perimeter" очень сильно коррелируют между собой (0.998). Это не независимые признаки, так как "mean perimeter" — величина производная от "mean radius". Поэтому из этих признаков в модели можно оставить только один.
- "mean texture" и "worst texture" очень сильно коррелируют между собой (0.912). Это не независимые признаки, так как "worst texture" — производная от "mean texture". Поэтому из этих признаков в модели можно оставлять только один.
- "radius error" и "perimeter error" очень сильно коррелируют между собой (0.973). Это не независимые признаки, так как "perimeter error" — производная от "radius error". Поэтому из этих признаков в модели можно оставлять только один.
- "radius error" и "area error" очень сильно коррелируют между собой (0.952). Это не независимые признаки, так как "area error" — производная от "radius error". Поэтому из этих признаков в модели можно оставлять только один.

- "mean concavity" и "worst concave points" очень сильно коррелируют между собой (0.86 concavity величина производная от worst concave points. Поэтому из этих признаков в и то же самое между "worst concave points" и ("mean concave points" (0.910), "worst conc
 - "mean compactness" и "worst compactness" очень сильно коррелируют между собой (0. compactness величина производная от worst compactness. Поэтому из этих признаков
 - Также можно сделать вывод, что выбирая из признаков ("mean radius","mean perimeter "worst area") лучше выбрать "worst perimeter", потому что он сильнее коррелирован с ц
- зависимые признаки сильно коррелированы с целевым, то оставляют именно тот при

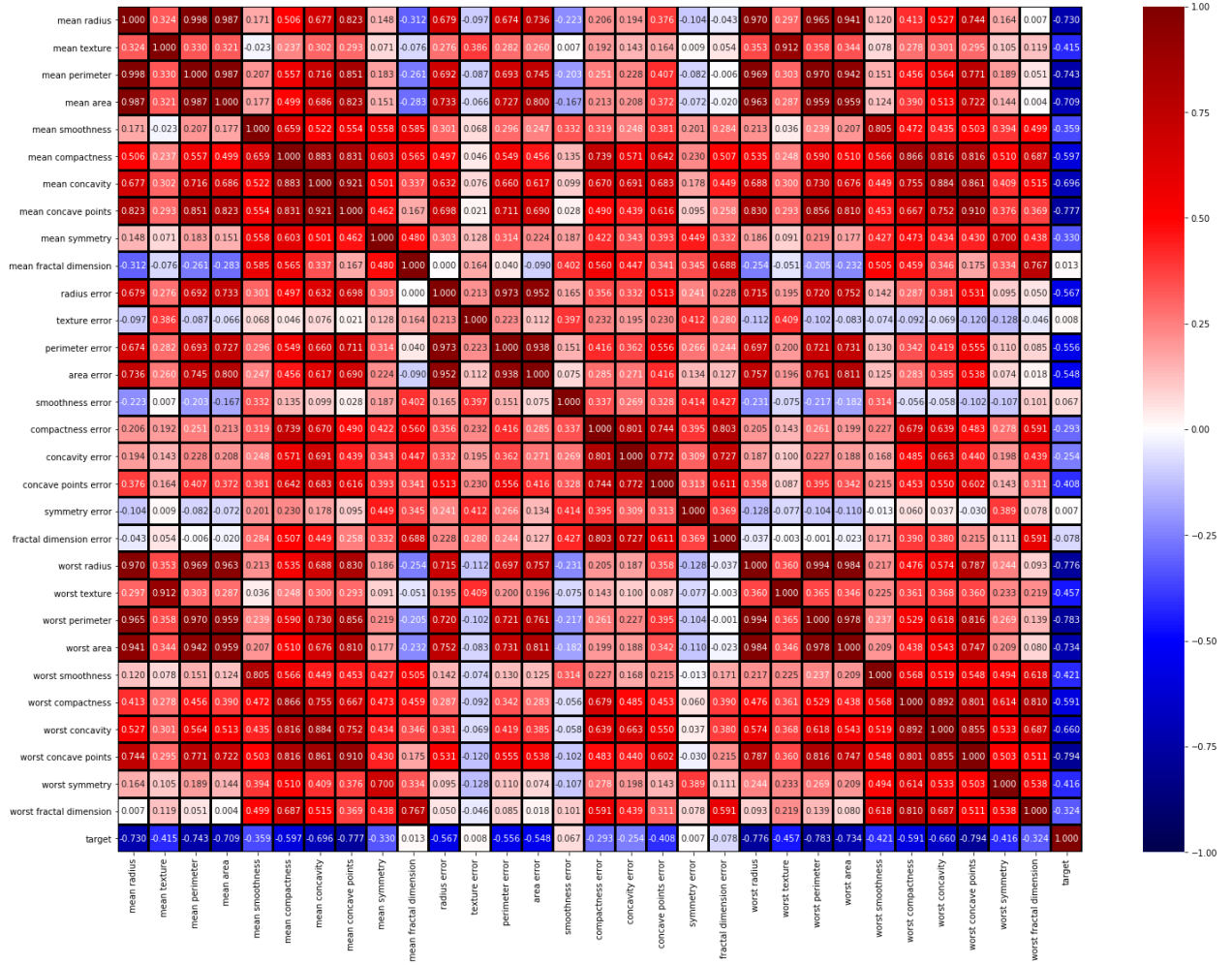
В нашем случае набор данных имеет большое количество признаков, поэтому анализ числ становится неудобным.

Чтобы визуализировать матрицу корреляции, мы будем использовать тепловую карту тепл корреляции в разных цветах.

```
fig=plt.figure(figsize=(25,18))
ax=fig.gca()
sns.heatmap(data1.corr(), annot = True, vmin=-1, vmax=1, center= 0, cmap= 'seismic')
```



<matplotlib.axes._subplots.AxesSubplot at 0x13a26d940>



```
fig=plt.figure(figsize=(15,10))
ax=fig.gca()
sns.heatmap(data1[['mean compactness','radius error','concave points error','worst
```



<matplotlib.axes._subplots.AxesSubplot at 0x13adf1470>

