РК1 Авдеев Ю. В. ИУ5-24М

In [0]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [32]:

```
from sklearn.datasets import load_boston
X, y = load_boston(return_X_y=True)
print(X.shape)
(506, 13)
```

```
(506, 13)
```

Out[32]:

```
(506, 13)
```

Создание Pandas Dataframe

In [0]:

```
def make_dataframe(ds_function):
    ds = ds_function()
    df = pd.DataFrame(data= np.c_[ds['data'], ds['target']],
                      columns= list(ds['feature_names']) + ['target'])
    return df
```

In [34]:

```
data = make_dataframe(load_boston)    #Создание датафрейма
data.head()                           #Вывод первых 5 строк
```

Out[34]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	L
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	296.0	15.3	396.90	
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	242.0	17.8	396.90	
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	242.0	17.8	392.83	
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	222.0	18.7	394.63	
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	222.0	18.7	396.90	

Поиск пустых значений в колонках

In [35]:

```
for col in data.columns:
    # Количество пустых значений
    temp_null_count = data[data[col].isnull()].shape[0]
    print('{} - {}'.format(col, temp_null_count))
    #Пустых значений не обнаружено
```

```
CRIM - 0
ZN - 0
INDUS - 0
CHAS - 0
NOX - 0
RM - 0
AGE - 0
DIS - 0
RAD - 0
TAX - 0
PTRATIO - 0
B - 0
LSTAT - 0
target - 0
```

In [36]:

```
data.describe() #Описательные статистики
```

Out[36]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE
count	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000
mean	3.613524	11.363636	11.136779	0.069170	0.554695	6.284634	68.574901
std	8.601545	23.322453	6.860353	0.253994	0.115878	0.702617	28.148861
min	0.006320	0.000000	0.460000	0.000000	0.385000	3.561000	2.900000
25%	0.082045	0.000000	5.190000	0.000000	0.449000	5.885500	45.025000
50%	0.256510	0.000000	9.690000	0.000000	0.538000	6.208500	77.500000
75%	3.677083	12.500000	18.100000	0.000000	0.624000	6.623500	94.075000
max	88.976200	100.000000	27.740000	1.000000	0.871000	8.780000	100.000000

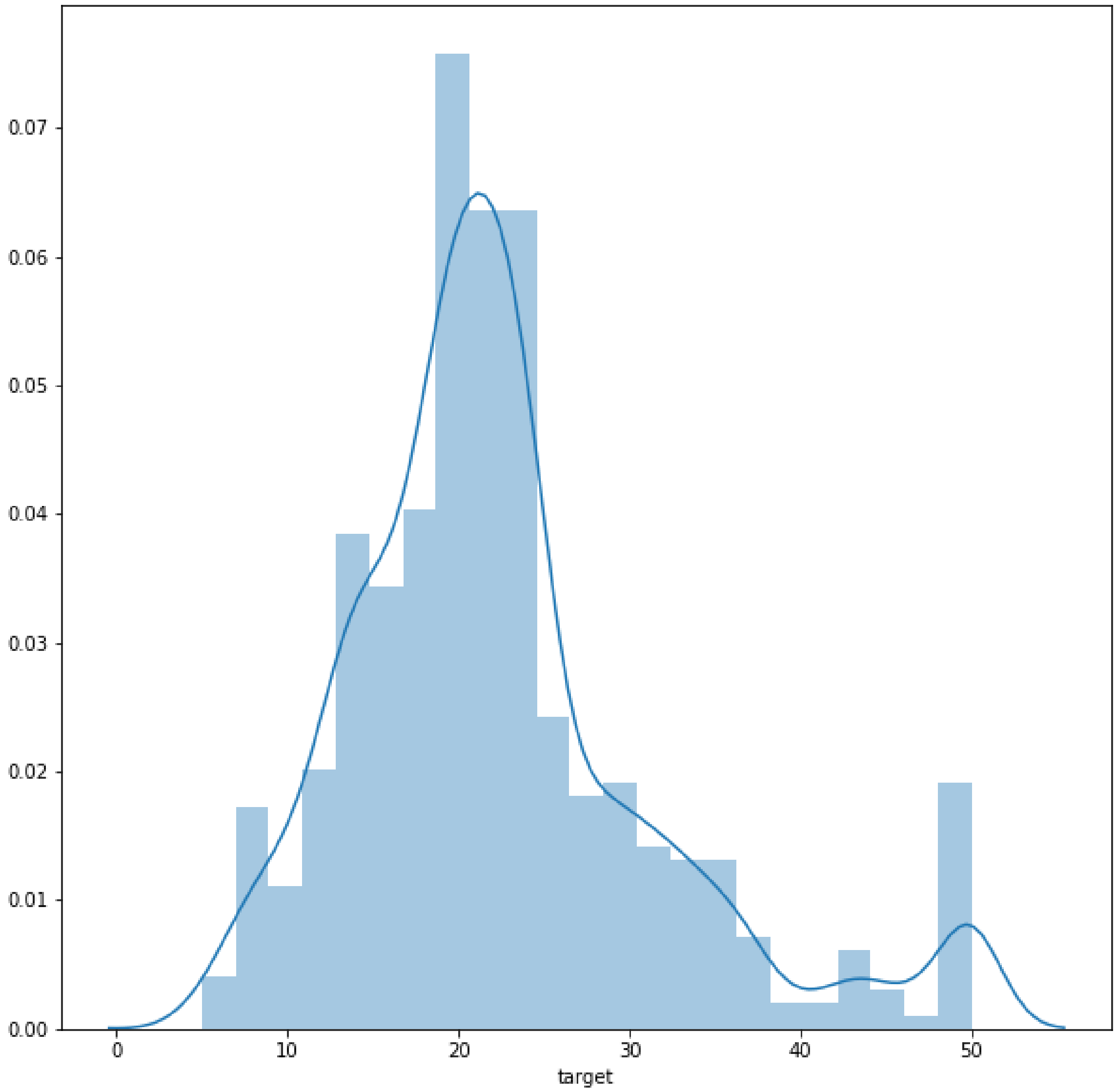
Распределениие значений целевого признака

In [37]:

```
fig, ax = plt.subplots(figsize=(10,10))
sns.distplot(data['target'])
```

Out[37]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fcff7006390>
```



Распределение похоже на нормальное

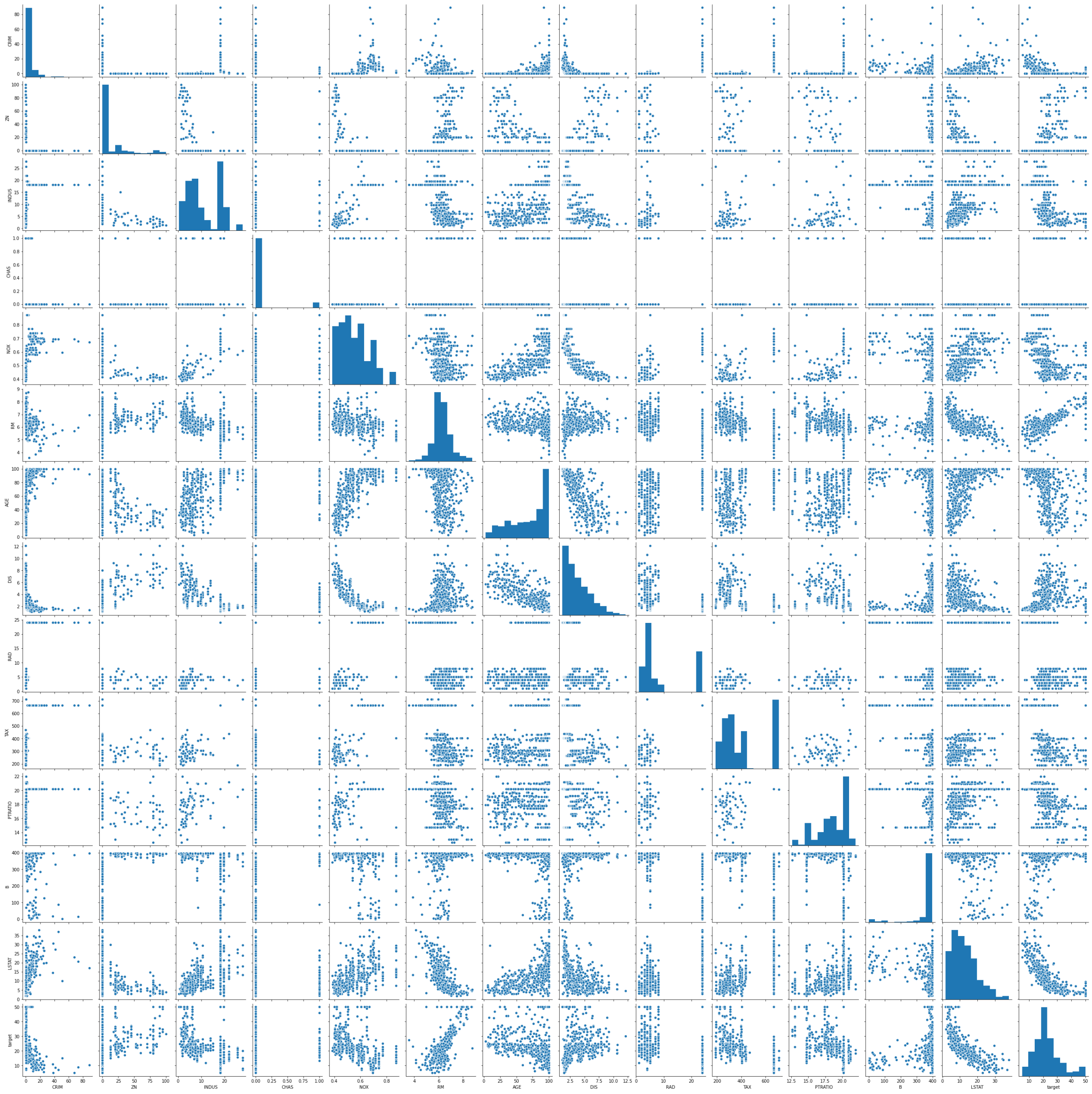
Парные диаграммы для понимания общей картины

In [38]:

```
sns.pairplot(data)
```

Out[38]:

<seaborn.axisgrid.PairGrid at 0x7fcff7849908>



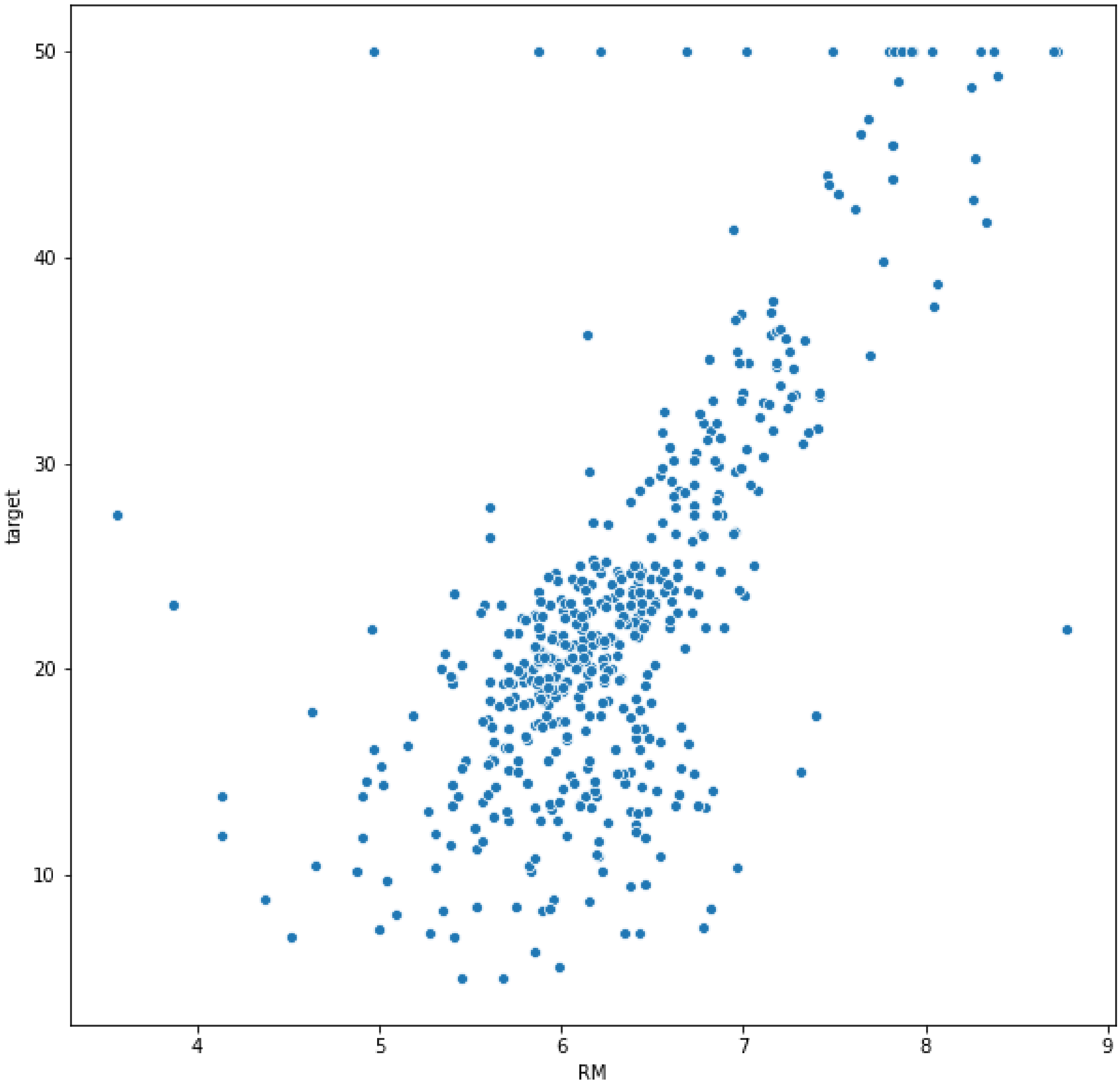
Находим почти линейную зависимость между значениями двух колонок с содержанием "выбросов"

In [39]:

```
fig, ax = plt.subplots(figsize=(10,10))
sns.scatterplot(ax=ax, x='RM', y='target', data=data)
```

Out[39]:

<matplotlib.axes._subplots.AxesSubplot at 0x7fcff2cea668>

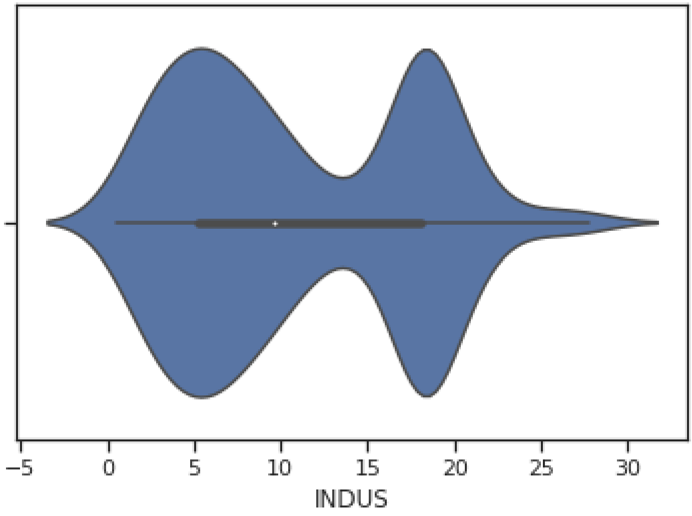


In [0]:

```
sns.violinplot(x=data['INDUS'])
```

Out[0]:

<matplotlib.axes._subplots.AxesSubplot at 0x7f39e74599e8>



По violin plot видим, что распределение бимодальное.

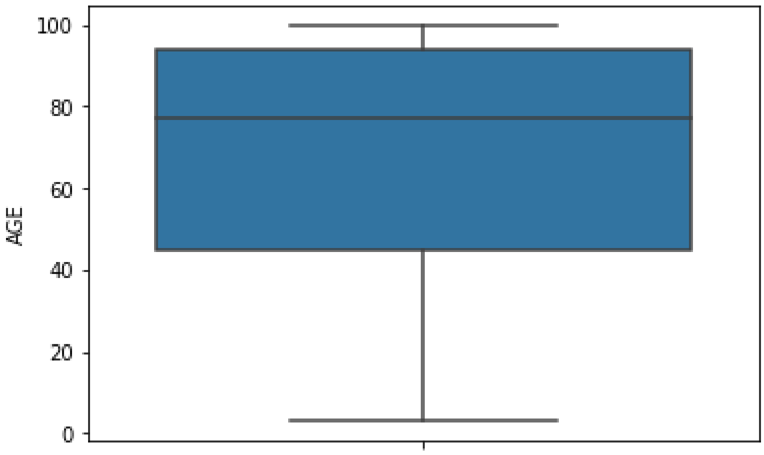
Задание для ИУ5-23М (boxplot для колонки с возрастом)

In [40]:

```
sns.boxplot(y=data['AGE'])
```

Out[40]:

<matplotlib.axes._subplots.AxesSubplot at 0x7fcff2fefc18>



Корреляционный анализ

Построим корреляционную матрицу

In [41]:

```
data.corr()
```

Out[41]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DI
CRIM	1.000000	-0.200469	0.406583	-0.055892	0.420972	-0.219247	0.352734	-0.37967
ZN	-0.200469	1.000000	-0.533828	-0.042697	-0.516604	0.311991	-0.569537	0.66440
INDUS	0.406583	-0.533828	1.000000	0.062938	0.763651	-0.391676	0.644779	-0.70802
CHAS	-0.055892	-0.042697	0.062938	1.000000	0.091203	0.091251	0.086518	-0.09917
NOX	0.420972	-0.516604	0.763651	0.091203	1.000000	-0.302188	0.731470	-0.76923
RM	-0.219247	0.311991	-0.391676	0.091251	-0.302188	1.000000	-0.240265	0.20524
AGE	0.352734	-0.569537	0.644779	0.086518	0.731470	-0.240265	1.000000	-0.74788
DIS	-0.379670	0.664408	-0.708027	-0.099176	-0.769230	0.205246	-0.747881	1.00000
RAD	0.625505	-0.311948	0.595129	-0.007368	0.611441	-0.209847	0.456022	-0.49458
TAX	0.582764	-0.314563	0.720760	-0.035587	0.668023	-0.292048	0.506456	-0.53443
PTRATIO	0.289946	-0.391679	0.383248	-0.121515	0.188933	-0.355501	0.261515	-0.23247
B	-0.385064	0.175520	-0.356977	0.048788	-0.380051	0.128069	-0.273534	0.29151
LSTAT	0.455621	-0.412995	0.603800	-0.053929	0.590879	-0.613808	0.602339	-0.49699
target	-0.388305	0.360445	-0.483725	0.175260	-0.427321	0.695360	-0.376955	0.24992

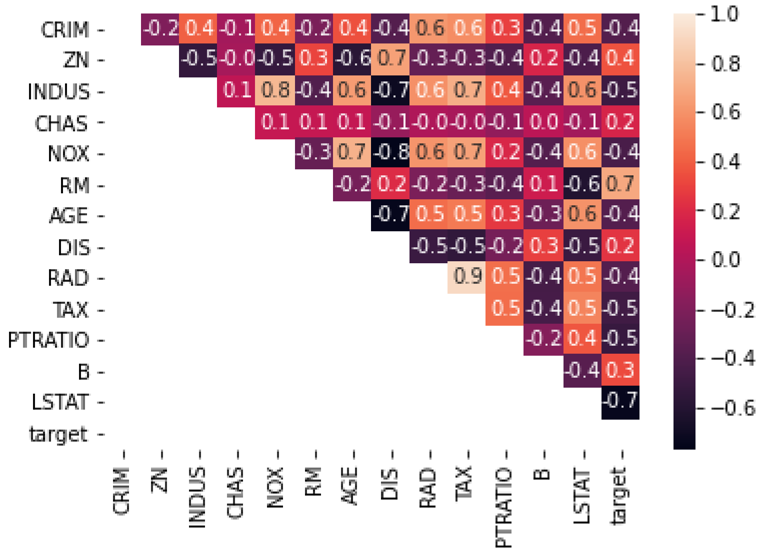
Также построим матрицу корреляций по Пирсону

In [42]:

```
# Треугольный вариант матрицы Пирсона
mask = np.zeros_like(data.corr(), dtype=np.bool)
mask[np.tril_indices_from(mask)] = True
sns.heatmap(data.corr(), mask=mask, annot=True, fmt='.1f')
```

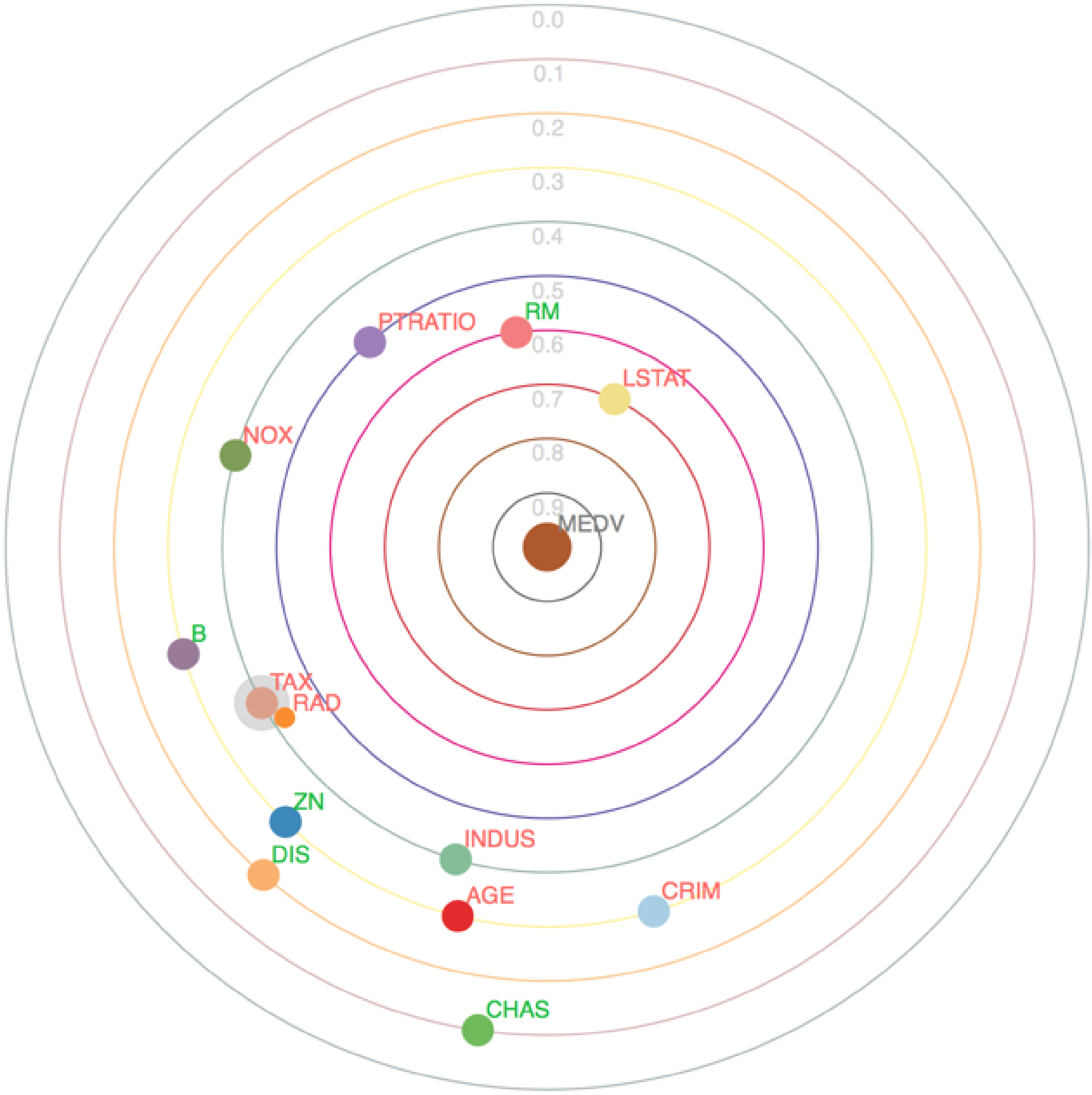
Out[42]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fcff2b41978>
```



Выявлена корреляция между показателями RAD и TAX

Использовав Solar correlation map, получаем ту же зависимость


0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
MEDV
RM
PTRATIO
LSTAT
NOX
B
TAX
RAD
ZN
INDUS
DIS
AGE
CRIM
CHAS