



```
In [4]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
from sklearn.preprocessing import StandardScaler
```

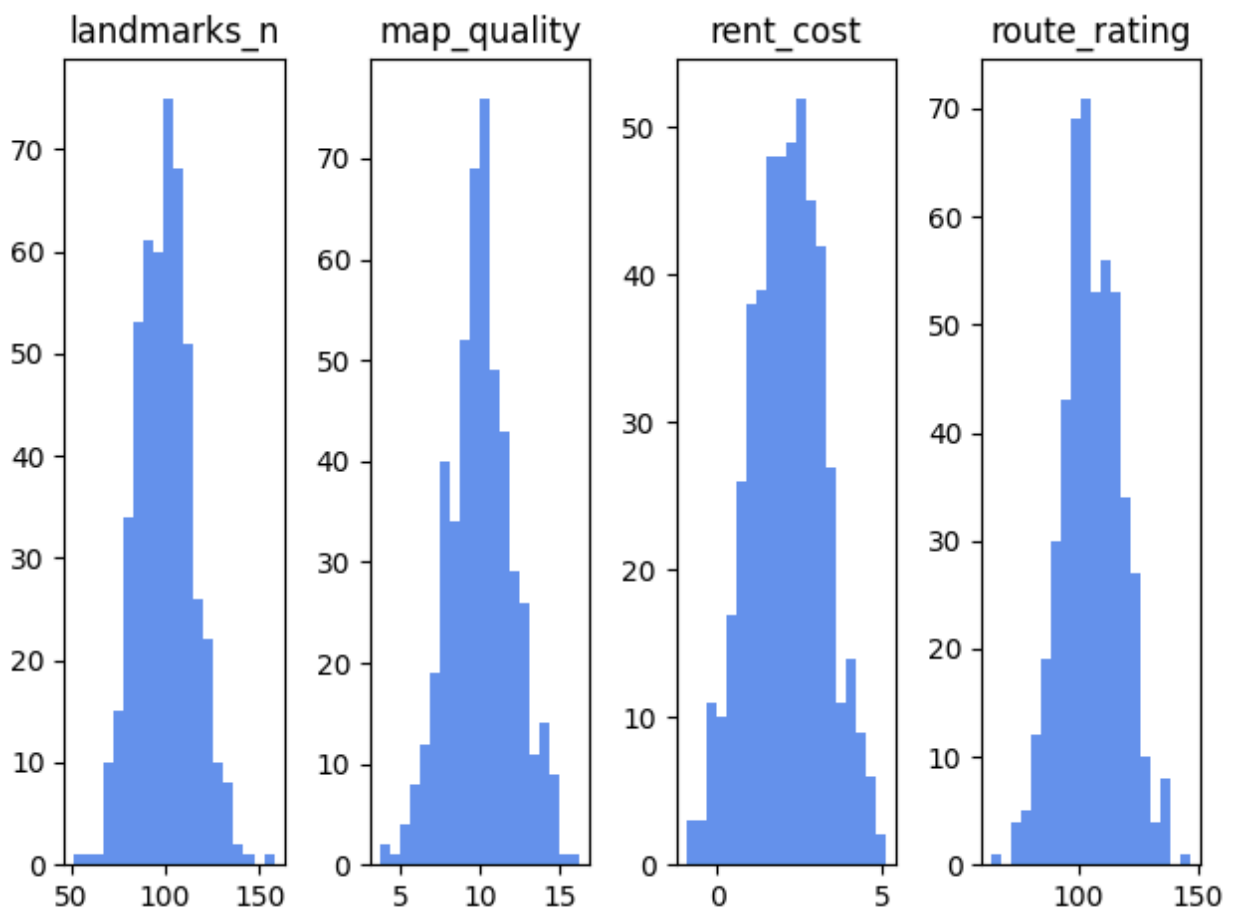
```
In [9]: data_path = "dataset_2task_final.csv"
data = pd.read_csv(data_path)
data.head()
```

```
Out[9]:
```

	landmarks_n	map_quality	rent_cost	route_rating
0	107.450712	12.224891	3.175834	116.521789
1	97.926035	13.715135	2.986853	104.642491
2	109.715328	7.688631	1.768171	106.745418
3	122.845448	12.268211	0.667700	124.682481
4	96.487699	8.523100	2.803592	100.959460

```
In [11]: fig, axes = plt.subplots(1, 4)
for i, col in enumerate(["landmarks_n", "map_quality", "rent_cost", "route_rating"]):
    axes[i].hist(data[col], bins = 20, color = "cornflowerblue")
    axes[i].set_title(col)

plt.tight_layout()
plt.show()
data.corr()["route_rating"].sort_values(ascending=False)
```



```
Out[11]: route_rating    1.000000
landmarks_n    0.909559
map_quality    0.387550
rent_cost     -0.437450
Name: route_rating, dtype: float64
```

Исходя из значений, можно понять, что `landmarks_n` оказывает наибольшее влияние на `route_rating`, из этого следует, что количество уникальных природных и искусственных достопримечательностей на маршруте наиболее влияет на рейтинг туристического маршрута.

```
In [15]: from statsmodels.api import OLS, add_constant

X1 = data[["map_quality"]]
X2 = data[["map_quality", "landmarks_n"]]
X3 = data[["map_quality", "landmarks_n", "rent_cost"]]

Y = data[["route_rating"]]
models = []

for X in [X1, X2, X3]:
    Xc = add_constant(X)
    model = OLS(Y, Xc).fit()
    models.append(model)
```

```
for i, m in enumerate(models):  
    print(f"Модель {i + 1}")  
    print(m.summary())
```

Модель 1

OLS Regression Results

```

=====
Dep. Variable:          route_rating    R-squared:                0.150
Model:                  OLS             Adj. R-squared:           0.148
Method:                 Least Squares   F-statistic:              88.02
Date:                   Sun, 23 Nov 2025 Prob (F-statistic):       2.29e-19
Time:                   11:19:03        Log-Likelihood:          -1932.4
No. Observations:       500            AIC:                     3869.
Df Residuals:           498            BIC:                     3877.
Df Model:                1
Covariance Type:        nonrobust
=====

```

	coef	std err	t	P> t	[0.025	0.975]
const	82.2426	2.610	31.506	0.000	77.114	87.371
map_quality	2.3840	0.254	9.382	0.000	1.885	2.883

```

=====
Omnibus:                0.511    Durbin-Watson:           1.979
Prob(Omnibus):           0.775    Jarque-Bera (JB):        0.621
Skew:                   -0.033    Prob(JB):                0.733
Kurtosis:                2.840    Cond. No.                 52.3
=====

```

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Модель 2

OLS Regression Results

```

=====
Dep. Variable:          route_rating    R-squared:                0.844
Model:                  OLS             Adj. R-squared:           0.844
Method:                 Least Squares   F-statistic:              1348.
Date:                   Sun, 23 Nov 2025 Prob (F-statistic):       1.77e-201
Time:                   11:19:03        Log-Likelihood:          -1508.0
No. Observations:       500            AIC:                     3022.
Df Residuals:           497            BIC:                     3035.
Df Model:                2
Covariance Type:        nonrobust
=====

```

	coef	std err	t	P> t	[0.025	0.975]
const	23.6367	1.673	14.125	0.000	20.349	26.924
map_quality	0.8391	0.114	7.380	0.000	0.616	1.062
landmarks_n	0.7409	0.016	47.080	0.000	0.710	0.772

```

=====
Omnibus:                1.740    Durbin-Watson:           2.016
Prob(Omnibus):           0.419    Jarque-Bera (JB):        1.616
Skew:                   -0.047    Prob(JB):                0.446
Kurtosis:                3.262    Cond. No.                 768.
=====

```

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

y specified.

Модель 3

OLS Regression Results

Dep. Variable:	route_rating	R-squared:	0.847
Model:	OLS	Adj. R-squared:	0.846
Method:	Least Squares	F-statistic:	912.1
Date:	Sun, 23 Nov 2025	Prob (F-statistic):	2.15e-201
Time:	11:19:03	Log-Likelihood:	-1504.5
No. Observations:	500	AIC:	3017.
Df Residuals:	496	BIC:	3034.
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P> t	[0.025	0.975]
const	26.7211	2.026	13.190	0.000	22.741	30.702
map_quality	0.8608	0.113	7.598	0.000	0.638	1.083
landmarks_n	0.7201	0.017	41.238	0.000	0.686	0.754
rent_cost	-0.5842	0.219	-2.666	0.008	-1.015	-0.154

Omnibus:	1.985	Durbin-Watson:	2.017
Prob(Omnibus):	0.371	Jarque-Bera (JB):	1.843
Skew:	-0.076	Prob(JB):	0.398
Kurtosis:	3.256	Cond. No.	938.

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Для модели 1 свободный член 82.2426, t - 31.506, p-value = 0 Оценка при map_quality - 2.3840, t - 9.382, p-value = 0 Для модели 2 свободный член 23.6367, t - 14.125, p-value = 0 Оценка при map_quality - 0.8391, t - 7.380, p-value = 0 Оценка при landmarks_n - 0.7409, t - 47.080, p-value = 0 Для модели 3 свободный член 26.7211, t - 13.190, p-value = 0 Оценка при map_quality - 0.8391, t - 7.380, p-value = 0 Оценка при landmarks_n - 0.7201, t - 41.238, p-value = 0 Оценка при rent_cost - -0.5842, t - -2.666, p-value = 0.008 Для каждой модели коэффициенты значимы потому что p-value < 0.05

```
In [16]: for i, m in enumerate(models):
          print(f"Модель {i + 1}: R^2 = {m.rsquared:.3f}")
          print(m.params)
```

```

Модель 1: R^2 = 0.150
const          82.242584
map_quality    2.384038
dtype: float64
Модель 2: R^2 = 0.844
const          23.636690
map_quality    0.839073
landmarks_n    0.740858
dtype: float64
Модель 3: R^2 = 0.847
const          26.721100
map_quality    0.860822
landmarks_n    0.720146
rent_cost      -0.584217
dtype: float64

```

Значения коэффициентов b_0 и b_1 изменяется при добавлении новых переменных в модель. В первой модели коэффициент получается максимальный = 2.384038 Во второй рейтинг меньше, так как параметров больше В третьей модели коэффициенты немного корректируется и `map_quality` становится 0.860822.

```

In [18]: scaler = StandardScaler()
Xs = scaler.fit_transform(data[["map_quality", "landmarks_n", "rent_cost"]])
Xstd1 = Xs[:,[0]]
Xstd2 = Xs[:,[0, 1]]
Xstd3 = Xs
Y = data[["route_rating"]]
models_std = []

for X in [Xstd1, Xstd2, Xstd3]:
    Xc = add_constant(X)
    model = OLS(Y, Xc).fit()
    models_std.append(model)

for i, m in enumerate(models_std):
    print(f"Модель {i + 1}: ")
    print(m.summary())

```

Модель 1:

OLS Regression Results

```
=====
Dep. Variable:          route_rating    R-squared:                0.150
Model:                  OLS             Adj. R-squared:           0.148
Method:                 Least Squares   F-statistic:              88.02
Date:                  Sun, 23 Nov 2025 Prob (F-statistic):       2.29e-19
Time:                  11:38:11         Log-Likelihood:          -1932.4
No. Observations:      500             AIC:                     3869.
Df Residuals:          498             BIC:                     3877.
Df Model:               1
Covariance Type:       nonrobust
=====
```

```
=====
              coef    std err          t      P>|t|      [0.025    0.975]
-----
const      106.2469     0.517    205.454     0.000    105.231    107.263
x1          4.8516     0.517     9.382     0.000     3.836     5.868
=====
```

```
=====
Omnibus:                 0.511    Durbin-Watson:           1.979
Prob(Omnibus):           0.775    Jarque-Bera (JB):        0.621
Skew:                   -0.033    Prob(JB):                0.733
Kurtosis:                2.840    Cond. No.                 1.00
=====
```

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Модель 2:

OLS Regression Results

```
=====
Dep. Variable:          route_rating    R-squared:                0.844
Model:                  OLS             Adj. R-squared:           0.844
Method:                 Least Squares   F-statistic:              1348.
Date:                  Sun, 23 Nov 2025 Prob (F-statistic):       1.77e-201
Time:                  11:38:11         Log-Likelihood:          -1508.0
No. Observations:      500             AIC:                     3022.
Df Residuals:          497             BIC:                     3035.
Df Model:               2
Covariance Type:       nonrobust
=====
```

```
=====
              coef    std err          t      P>|t|      [0.025    0.975]
-----
const      106.2469     0.222    479.585     0.000    105.812    106.682
x1          1.7075     0.231     7.380     0.000     1.253     2.162
x2         10.8936     0.231    47.080     0.000    10.439    11.348
=====
```

```
=====
Omnibus:                 1.740    Durbin-Watson:           2.016
Prob(Omnibus):           0.419    Jarque-Bera (JB):        1.616
Skew:                   -0.047    Prob(JB):                0.446
Kurtosis:                3.262    Cond. No.                 1.35
=====
```

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

y specified.

Модель 3:

OLS Regression Results

Dep. Variable:	route_rating	R-squared:	0.847
Model:	OLS	Adj. R-squared:	0.846
Method:	Least Squares	F-statistic:	912.1
Date:	Sun, 23 Nov 2025	Prob (F-statistic):	2.15e-201
Time:	11:38:11	Log-Likelihood:	-1504.5
No. Observations:	500	AIC:	3017.
Df Residuals:	496	BIC:	3034.
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P> t	[0.025	0.975]
const	106.2469	0.220	482.524	0.000	105.814	106.680
x1	1.7518	0.231	7.598	0.000	1.299	2.205
x2	10.5891	0.257	41.238	0.000	10.085	11.094
x3	-0.6570	0.246	-2.666	0.008	-1.141	-0.173

Omnibus:	1.985	Durbin-Watson:	2.017
Prob(Omnibus):	0.371	Jarque-Bera (JB):	1.843
Skew:	-0.076	Prob(JB):	0.398
Kurtosis:	3.256	Cond. No.	1.77

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

В первой модели $R^2 = 0.150$ Это низкое значение, так как в модели мы учитываем лишь 1 признак. Во второй модели $R^2 = 0.844$ Такой резкий рост объясняется тем, что у нас стало в два раза больше параметров, при этом новый параметр имеет сильную связь с рейтингом. В третьей модели $R^2 = 0.847$ В третьей модели квадрат почти не отличается от второго, причина этому в том, что `rent_cost` не сильно влияет на рейтинг.

```
In [ ]: for i, m in enumerate(models_std):
        print(f"Модель {i + 1}: R^2 = {m.rsquared}")
        print(m.summary())
```

Ответ на пункт 5 см выше. (по аналогии с пунктом 2)

```
In [20]: np.random.seed(42)
        for i in range(100):
            data[f"rand_{i}"] = np.random.normal(size=len(data))

        X_all = data.drop(columns=["route_rating"])
        Y = data["route_rating"]
        Xc = add_constant(X_all)
```

```
model_all = OLS(Y, Xs).fit()
print(model_all.summary())
```

OLS Regression Results

=====						
Dep. Variable:	route_rating	R-squared (uncentered):				
0.012						
Model:	OLS	Adj. R-squared (uncentered):				
0.006						
Method:	Least Squares	F-statistic:				
1.943						
Date:	Sun, 23 Nov 2025	Prob (F-statistic):				
0.122						
Time:	11:48:49	Log-Likelihood:				
-3042.9						
No. Observations:	500	AIC:				
6092.						
Df Residuals:	497	BIC:				
6104.						
Df Model:	3					
Covariance Type:	nonrobust					
=====						
	coef	std err	t	P> t	[0.025	0.975]

x1	1.7518	4.996	0.351	0.726	-8.064	11.568
x2	10.5891	5.564	1.903	0.058	-0.342	21.520
x3	-0.6570	5.339	-0.123	0.902	-11.146	9.832
=====						
Omnibus:	1.985	Durbin-Watson:	0.004			
Prob(Omnibus):	0.371	Jarque-Bera (JB):	1.843			
Skew:	-0.076	Prob(JB):	0.398			
Kurtosis:	3.256	Cond. No.	1.77			
=====						

Notes:

[1] R² is computed without centering (uncentered) since the model does not contain a constant.

[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.

C:\Users\sie22\AppData\Local\Temp\ipykernel_16032\1229722133.py:3: PerformanceWarning: DataFrame is highly fragmented. This is usually the result of calling `frame.insert` many times, which has poor performance. Consider joining all columns at once using `pd.concat(axis=1)` instead. To get a de-fragmented frame, use `newframe = frame.copy()`

```
data[f"rand_{i}"] = np.random.normal(size=len(data))
```

Ответ на 9 выше, после добавления случайных столбцов видим изменение R² и отсутствие зависимости коэффициентов Метрики модели ухудшаются.

Ответ на 6: При стандартизации все признаки приводятся к среднему 0 и стандартному 1 отклонению, коэффициенты теперь показывают насколько

изменится рейтинг при изменении признака на 1 квадратичную единицу

Ответ на 7: Значение R^2 для моделей на исходных и стандартизированных данных совпадают. Это из-за того, что меняется по сути только масштаб интерпретации коэффициентов, но не качество модели.

Ответ на 8: При добавлении переменной `landmarks_n`, влияние `map_quality` на рейтинг понизилось, связано это с тем, что теперь у нас появилась еще одна переменная, влияющая на рейтинг, и рейтинг теперь еще зависит от `landmarks_n`.