

Домашнее задание №1

1. Подберите скорость обучения (alpha) и количество итераций:

С помощью аналитического метода мы получили $mse = 45.937499999999986$. Это значение и будет для нас ориентиром при подборе скорости обучения (alpha) и количества итераций

In [1]:

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 plt.rcParams.update({'font.size': 14})
```

In [2]:

```
1 X = np.array([[1, 1, 1, 1, 1, 1, 1, 1, 1, 1], # для умножения на intercept
2               [1, 1, 2, 1, 3, 0, 5, 10, 1, 2]]) # стаж
3
4 # средний балл ЕГЭ (целевая переменная)
5 y = [45, 55, 50, 59, 65, 35, 75, 80, 50, 60]
```

In [3]:

```
1 def calc_mse(y, y_pred):
2     err = np.mean((y - y_pred)**2)
3     return err
```

In [4]:

```
1 cnames = [  
2     '#F0F8FF',  
3     '#FAEBD7',  
4     '#00FFFF',  
5     '#7FFFD4',  
6     '#F0FFFF',  
7     '#F5F5DC',  
8     '#FFE4C4',  
9     '#000000',  
10    '#FFEBCD',  
11    '#0000FF',  
12    '#8A2BE2',  
13    '#A52A2A',  
14    '#DEB887',  
15    '#5F9EA0',  
16    '#7FFF00',  
17    '#D2691E',  
18    '#FF7F50',  
19    '#6495ED',  
20    '#FFF8DC',  
21    '#DC143C',  
22    '#00FFFF',  
23    '#00008B',  
24    '#008B8B',  
25    '#B8860B',  
26    '#A9A9A9',  
27    '#006400',  
28    '#BDB76B',  
29    '#8B008B',  
30    '#556B2F',  
31    '#FF8C00',  
32    '#9932CC',  
33    '#8B0000',  
34    '#E9967A',  
35    '#8FBC8F',  
36    '#483D8B',  
37    '#2F4F4F',  
38    '#00CED1',  
39    '#9400D3',  
40    '#FF1493',  
41    '#00BFFF',  
42    '#696969',  
43    '#1E90FF',  
44    '#B22222',  
45    '#FFFAF0',  
46    '#228B22',  
47    '#FF00FF',  
48    '#DCDCDC',  
49    '#F8F8FF',  
50    '#FFD700',  
51    '#DAA520',  
52    '#808080',  
53    '#008000',  
54    '#ADFF2F',  
55    '#F0FFF0',  
56    '#FF69B4',  
57    '#CD5C5C',  
58    '#4B0082',  
59    '#FFFFFF0',
```

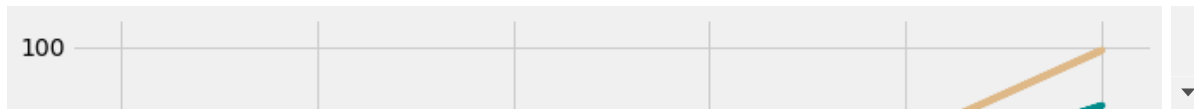
60	'#F0E68C',
61	'#E6E6FA',
62	'#FFF0F5',
63	'#7CFC00',
64	'#FFFACD',
65	'#ADD8E6',
66	'#F08080',
67	'#E0FFFF',
68	'#FAFAD2',
69	'#90EE90',
70	'#D3D3D3',
71	'#FFB6C1',
72	'#FFA07A',
73	'#20B2AA',
74	'#87CEFA',
75	'#778899',
76	'#B0C4DE',
77	'#FFFFFFE0',
78	'#00FF00',
79	'#32CD32',
80	'#FAF0E6',
81	'#FF00FF',
82	'#800000',
83	'#66CDAA',
84	'#0000CD',
85	'#BA55D3',
86	'#9370DB',
87	'#3CB371',
88	'#7B68EE',
89	'#00FA9A',
90	'#48D1CC',
91	'#C71585',
92	'#191970',
93	'#F5FFFA',
94	'#FFE4E1',
95	'#FFE4B5',
96	'#FFDEAD',
97	'#000080',
98	'#FDF5E6',
99	'#808000',
100	'#6B8E23',
101	'#FFA500',
102	'#FF4500',
103	'#DA70D6',
104	'#EEE8AA',
105	'#98FB98',
106	'#AFEEEE',
107	'#DB7093',
108	'#FFEFD5',
109	'#FFDAB9',
110	'#CD853F',
111	'#FFC0CB',
112	'#DDA0DD',
113	'#B0E0E6',
114	'#800080',
115	'#FF0000',
116	'#BC8F8F',
117	'#4169E1',
118	'#8B4513',
119	'#FA8072',
120	'#FAA460',

121	'#2E8B57',
122	'#FFF5EE',
123	'#A0522D',
124	'#C0C0C0',
125	'#87CEEB',
126	'#6A5ACD',
127	'#708090',
128	'#FFFAFA',
129	'#00FF7F',
130	'#4682B4',
131	'#D2B48C',
132	'#008080',
133	'#D8BFD8',
134	'#FF6347',
135	'#40E0D0',
136	'#EE82EE',
137	'#F5DEB3',
138	'#FFFFFF',
139	'#F5F5F5',
140	'#FFF000',
141	'#9ACD32']

In [5]:

```
1 plt.style.use('fivethirtyeight')
2
3 n = X.shape[1]
4 alpha = 1e-3
5 w = np.array([1, 0.5])
6 plt.figure(figsize=(10, 7))
7 for i in range(10001):
8     y_pred = np.dot(w, X)
9     err = calc_mse(y, y_pred)
10    for j in range(w.shape[0]):
11        w[j] -= alpha * (1/n * 2 * np.sum(X[j] * (y_pred - y)))
12    if i % 1000 == 0:
13        # print(w.shape[0])
14        print(f"Номер итерации: {i}, веса модели: {w}, MSE: {err}")
15        plt.scatter(X[1], y)
16        plt.plot(X[1], w[0] + w[1] * X[1], color = cnames[int(i / 1000)+2], label=i)
17        plt.legend()
```

Номер итерации: 0, веса модели: [1.1102 0.84], MSE: 3173.15
Номер итерации: 1000, веса модели: [31.66648131 6.78503276], MSE: 176.468065
66379882
Номер итерации: 2000, веса модели: [41.73215361 4.92632937], MSE: 62.2342399
245102
Номер итерации: 3000, веса модели: [45.28877016 4.26957292], MSE: 47.9721478
2992787
Номер итерации: 4000, веса модели: [46.54546925 4.03751382], MSE: 46.1915257
6288303
Номер итерации: 5000, веса модели: [46.98951279 3.95551778], MSE: 45.9692151
1416331
Номер итерации: 6000, веса модели: [47.14641166 3.92654521], MSE: 45.9414596
3171208
Номер итерации: 7000, веса модели: [47.2018505 3.916308], MSE: 45.937994359
98288
Номер итерации: 8000, веса модели: [47.22143932 3.91269078], MSE: 45.9375617
20839334
Номер итерации: 9000, веса модели: [47.22836086 3.91141266], MSE: 45.9375077
0584623
Номер итерации: 10000, веса модели: [47.23080652 3.91096105], MSE: 45.937500
96207482



Эмпирически было установлено, что целевого MSE модель достигает при скорости обучения (alpha) = 0.001 и количестве итераций = 10000. Визуально график линейной регрессии практически не меняется, начиная с 5000 итераций

*2. В этом коде мы избавляемся от итераций по весам, но тут есть ошибка, исправьте ее:

In [6]:

```
1 w = np.array([1, 0.5])
2 alpha = 1e-3
3 for i in range(10000):
4     y_pred = np.dot(w, X)
5     err = calc_mse(y, y_pred)
6     '''for j in range(w.shape[0]):
7         w[j] -= alpha * (1/n * 2 * np.sum(X[j] * (y_pred - y)))'''
8     w -= (alpha * 1/n * 2 * np.dot(X, (np.dot(w, X) - y)))
9
10    if i % 1000 == 0:
11        print(f"Номер итерации: {i}, веса модели: {w}, MSE: {err}")
```

```
Номер итерации: 0, веса модели: [1.1102 0.84 ], MSE: 3173.15
Номер итерации: 1000, веса модели: [31.66648131 6.78503276], MSE: 176.468065
66379882
Номер итерации: 2000, веса модели: [41.73215361 4.92632937], MSE: 62.2342399
245102
Номер итерации: 3000, веса модели: [45.28877016 4.26957292], MSE: 47.9721478
2992787
Номер итерации: 4000, веса модели: [46.54546925 4.03751382], MSE: 46.1915257
6288303
Номер итерации: 5000, веса модели: [46.98951279 3.95551778], MSE: 45.9692151
1416331
Номер итерации: 6000, веса модели: [47.14641166 3.92654521], MSE: 45.9414596
3171208
Номер итерации: 7000, веса модели: [47.2018505 3.916308 ], MSE: 45.937994359
98288
Номер итерации: 8000, веса модели: [47.22143932 3.91269078], MSE: 45.9375617
20839334
Номер итерации: 9000, веса модели: [47.22836086 3.91141266], MSE: 45.9375077
0584623
```

*3. Вместо того, чтобы задавать количество итераций, задайте условие остановки алгоритма - когда ошибка за итерацию начинает изменяться ниже определенного порога

In [9]:

```
1 w = np.array([1, 0.5])
2 alpha = 1e-3
3 e = float(input("Введите порог ошибки в формате '0.00001': "))
4 count = 0
5 y_pred = np.dot(w, X)
6 err = calc_mse(y, y_pred)
7 err_2 = 1000
8 while err - err_2 > e:
9     y_pred = np.dot(w, X)
10    err = calc_mse(y, y_pred)
11    w -= (alpha * 1/n * 2 * np.dot(X, (np.dot(w, X) - y)))
12    y_pred = np.dot(w, X)
13    err_2 = calc_mse(y, y_pred)
14    count += 1
15    if count % 100 == 0:
16        print(f"Номер итерации: {count}, веса модели: {w}, MSE: {err}")
17
```

Номер итерации: 2600, веса модели: [44.28276646 4.4553392], MSE: 50.62387950613113

Номер итерации: 2700, веса модели: [44.5741756 4.40152827], MSE: 49.74356592648801

Номер итерации: 2800, веса модели: [44.83679246 4.35303406], MSE: 49.02861496792367

Номер итерации: 2900, веса модели: [45.07346181 4.30933125], MSE: 48.447964066958136

Номер итерации: 3000, веса модели: [45.28674737 4.26994644], MSE: 47.97638561146634

Номер итерации: 3100, веса модели: [45.47895954 4.234453], MSE: 47.59339087346774

Номер итерации: 3200, веса модели: [45.65218045 4.20246643], MSE: 47.28233983280531

Номер итерации: 3300, веса модели: [45.80828649 4.17364026], MSE: 47.02971821611491

Номер итерации: 3400, веса модели: [45.94896869 4.14766221], MSE: 46.824550340503926

Номер итерации: 3500, веса модели: [46.07575098 4.12425089], MSE: 46.65792225168797