Author: **Andrei Ermishin**

In [1]:
```
import numpy as np
import pandas as pd
%matplotlib inline
import seaborn as sns
```

In [2]:
```
data = pd.read_excel('Задача.xlsx')
# data.to_json('table.json', orient='records')
# data = pd.read_json('table.json')
print(data.shape)
data.head()
```

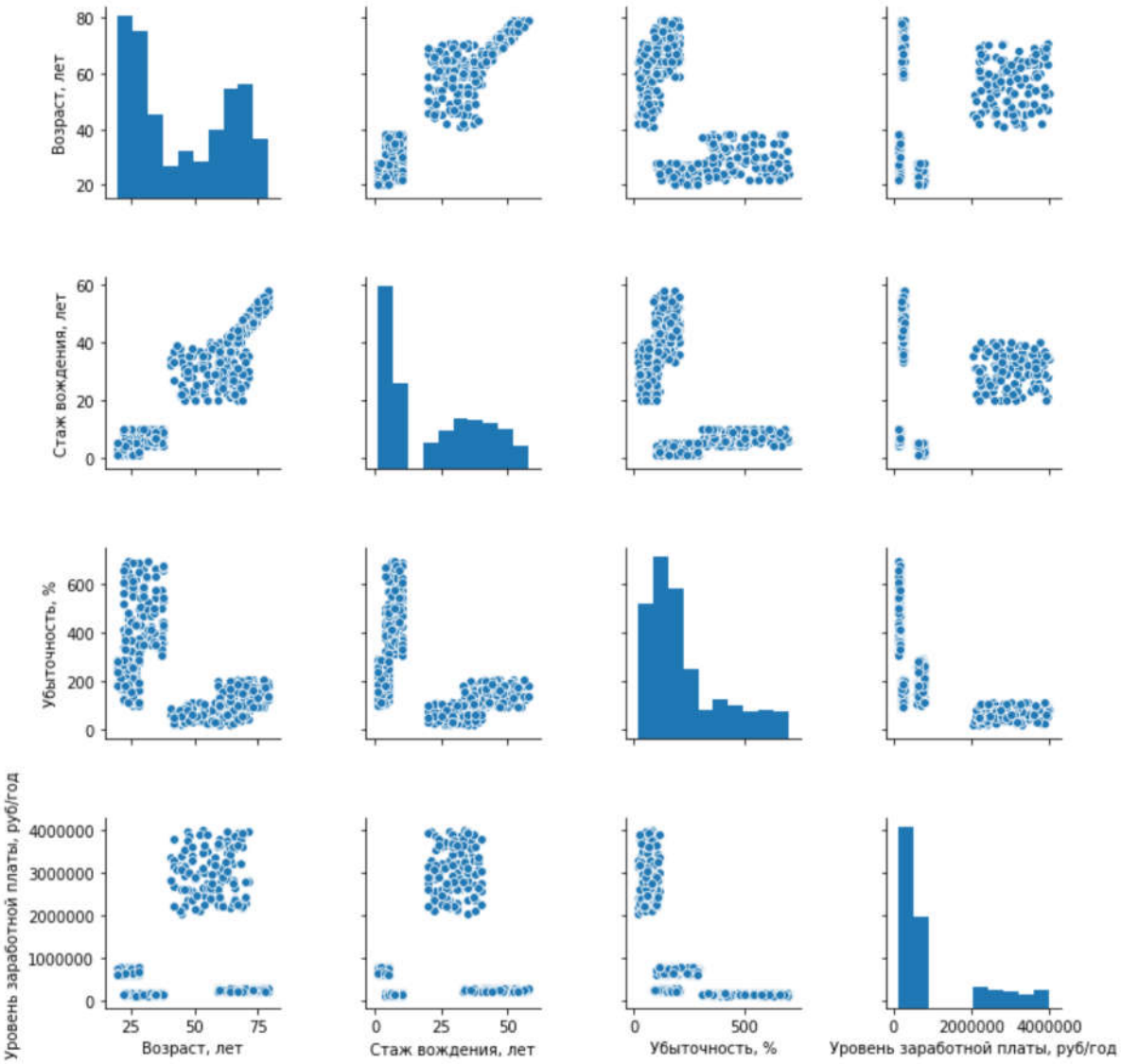
```
(484, 5)
```

Out[2]:

	Персона	Возраст, лет	Стаж вождения, лет	Убыточность, %	Уровень заработной платы, руб/год
0	6-LLJEH	20	1	263	716693
1	2-GLHFG	74	51	107	274393
2	6-FJFKL	27	1	165	723841
3	4-KJEJL	24	6	348	139419
4	5-JFFGH	26	3	286	650003

In [3]:
```
sns.pairplot(data)
```

Out[3]:
```
<seaborn.axisgrid.PairGrid at 0x2064b84cb00>
```



Удалим первый столбец с символами персоны, который не будет использоваться для кластеризации и по-видимому уже размечен на 9 кластеров. 9 кластеров могут использоваться исходя из дополнительных знаний о данных или потребностью компании в более точном разбиении.

In [4]:
```
X = data.drop('Персона', axis='columns')
# X = data.drop(data.columns[0], axis='columns')
X.head(2)
```

Out[4]:

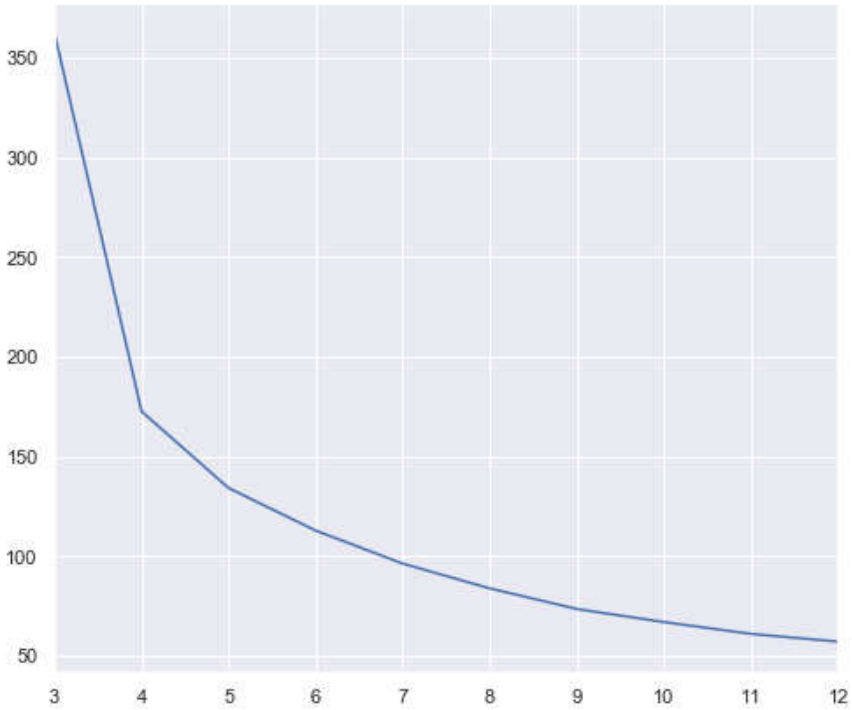
	Возраст, лет	Стаж вождения, лет	Убыточность, %	Уровень заработной платы, руб/год
0	20	1	263	716693
1	74	51	107	274393

K-Means

В качестве меры будем смотреть на сумму квадратов расстояний персон до центра кластера. Если применять Elbow method, то начиная с 4, 5 кластеров наблюдаем более пологое снижение. Возьмем для дальнейшего обучения модели количество кластеров = 5.

In [5]:
```
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

min_c, max_c = 3, 13
inertia_lst = []
for num_clusters in range(min_c, max_c):
    kmeans = KMeans(n_clusters=num_clusters, n_jobs=-1, random_state=11)
    kmeans.fit(X_scaled)
    inertia_lst.append(kmeans.inertia_)

sns.set(rc={'figure.figsize':(8, 7)})
pd.Series(data=inertia_lst, index=list(range(min_c, max_c))).plot()
```

Out[5]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x2064fa81ef0>
```



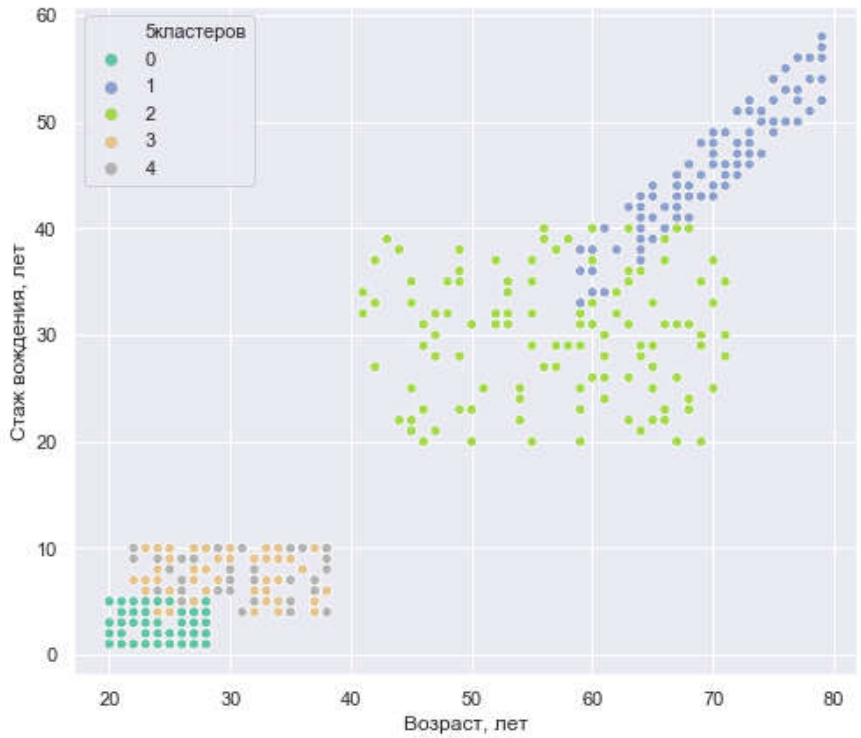
In [6]:
```
new_num_clusters = 5
kmeans_new = KMeans(n_clusters=new_num_clusters, n_jobs=-1, random_state=11)
kmeans_new.fit(X_scaled)

data[f'{new_num_clusters}кластеров'] = kmeans_new.labels_
data.head()
```

Out[6]:

	Персона	Возраст, лет	Стаж вождения, лет	Убыточность, %	Уровень заработной платы, руб/год	5кластеров
0	6-LLJEH	20	1	263	716693	0
1	2-GLHFG	74	51	107	274393	1
2	6-FJFKL	27	1	165	723841	0
3	4-KJEJL	24	6	348	139419	3
4	5-JFFGH	26	3	286	650003	0

In [7]:
```
sns.scatterplot(x='Возраст, лет', y='Стаж вождения, лет',
                hue='5кластеров', palette='Set2', data=data, legend='full')
```

Out[7]: <matplotlib.axes._subplots.AxesSubplot at 0x2064fa7df98>

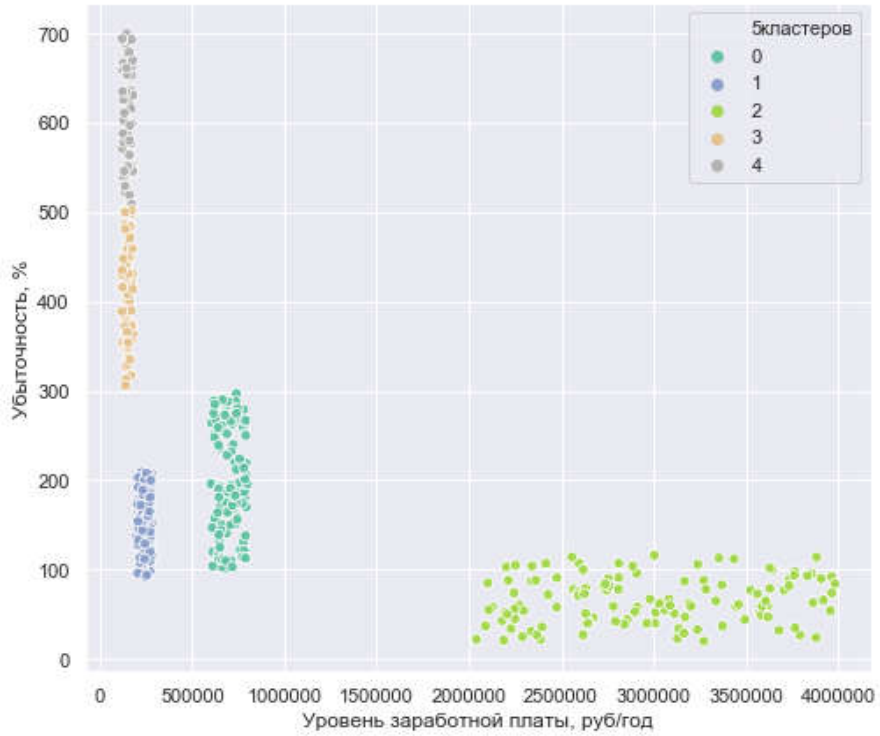


Вообще, зависимость тут близка к линейной, т.к. чем больше лет, тем больше стаж. Разница только в том, когда человек начал водить. На графике можно отметить 3 большие группы:

- возраст до 40 лет + стаж до 10 лет,
- возраст от 40 до 70 лет + стаж от 20 до 40 лет,
- возраст от 60 до 80 лет + стаж от 35 до 60 лет. В первой группе люди молодого возраста и поэтому имеют малый стаж вождения, группа посередине имеет большой разброс, но стаж в пределах [20, 40]. Третья группа - это пожилые люди с очень большим стажем.

In [8]:
```
sns.scatterplot(x='Уровень заработной платы, руб/год', y='Убыточность, %',
                hue='5кластеров', palette='Set2', data=data, legend='full')
```

Out[8]: <matplotlib.axes._subplots.AxesSubplot at 0x2064fb479b0>



Мы видим, что люди среднего возраста (правый нижний угол) имеют высокую зарплату и низкую убыточность. Вообще, как видно из графика, чем больше зарплата, тем меньше убыточность (выбиваются немного пожилые люди, которые не спешат и у них большой стаж). Большая убыточность свойственна людям до 40 лет с низкой зарплатой (2 левых верхних кластера).

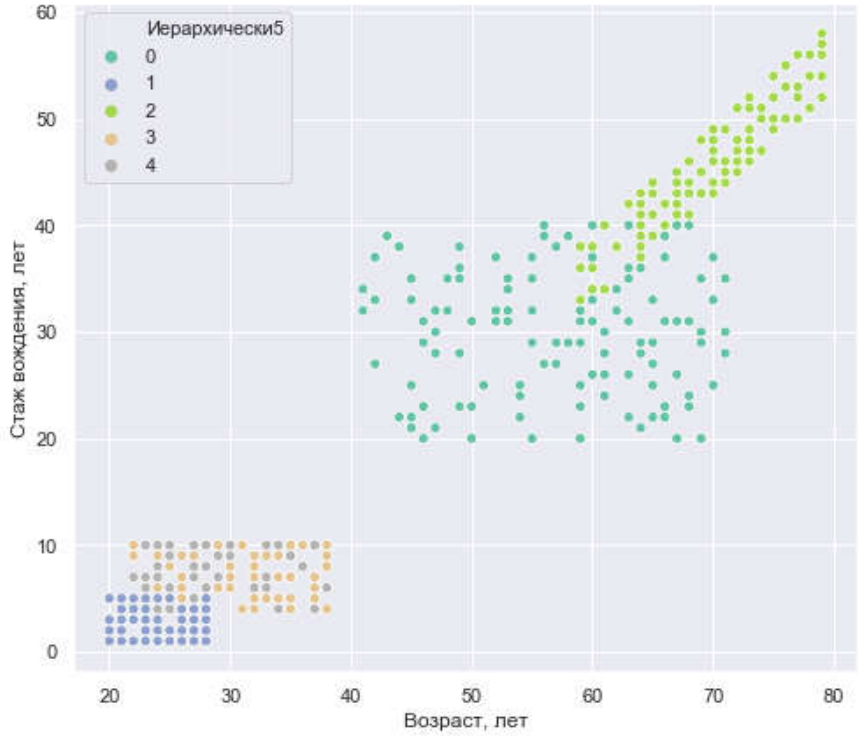
Hierarchical clustering

В иерархической кластеризации мы используем прирост суммы квадратов расстояний персон до центра кластера.

In [9]:
```
from sklearn.cluster import AgglomerativeClustering

num_clusters = 5
agglo = AgglomerativeClustering(n_clusters=num_clusters, linkage='ward')
agglo.fit(X_scaled)

pred = pd.Series(data=agglo.labels_, index=data.index, name=f'Иерархически{num_clusters}')
```

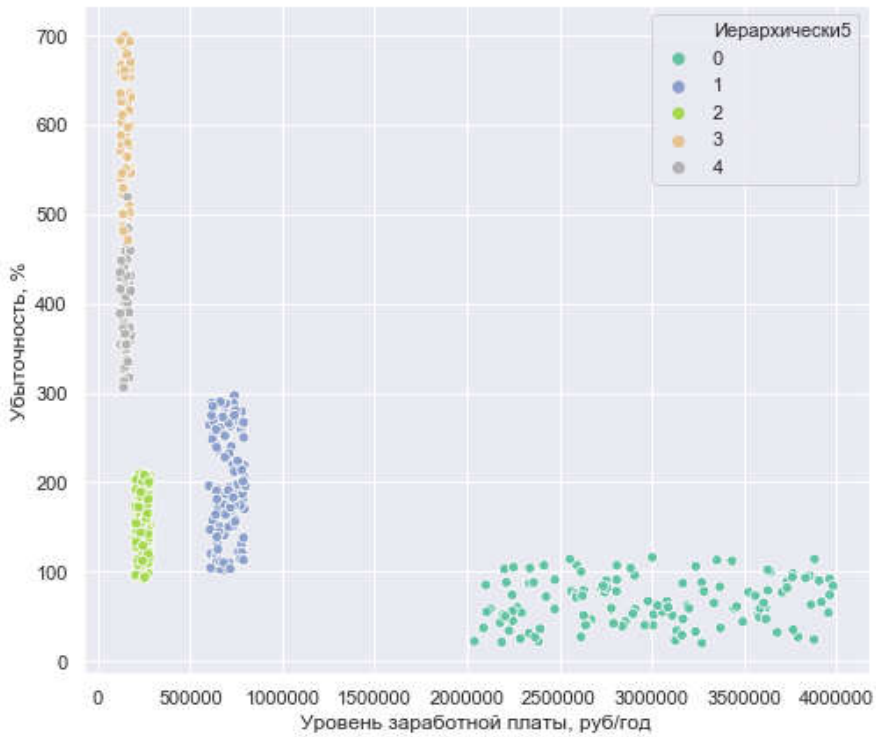
In [10]:
```
sns.scatterplot(x='Возраст, лет', y='Стаж вождения, лет',
                hue=pred, palette='Set2', data=data, legend='full')
```

Out[10]: <matplotlib.axes._subplots.AxesSubplot at 0x2064fbc4400>



In [11]:
```
sns.scatterplot(x='Уровень заработной платы, руб/год', y='Убыточность, %',
                hue=pred, palette='Set2', data=data, legend='full')
```

Out[11]: <matplotlib.axes._subplots.AxesSubplot at 0x2064fc87cf8>



In [ ]:

Посмотрим, правильно ли мы записали результат работы сервиса в файл.

In [12]:
```
# Result of Flask app:
pd.read_json('result.json').head()
```

Out[12]:

	Персона	Возраст, лет	Стаж вождения, лет	Убыточность, %	Уровень заработной платы, руб/год	5кластеров
0	6-LLJEH	20	1	263	716693	1
1	2-GLHFG	74	51	107	274393	2
2	6-FJFKL	27	1	165	723841	1
3	4-KJEJL	24	6	348	139419	4
4	5-JFFGH	26	3	286	650003	1

In [ ]: