

```
1 # https://freecodecam-boilerplate-gr127opof4f.ws-eu117.gitpod.io/
2 # https://www.freecodecamp.org/learn/data-analysis-with-python/data-analysis-with-python-project
```

```
1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 import seaborn as sns
```

```
1 from google.colab import drive
2 drive.mount('content/')
```

⇨ Mounted at content/

```
1 adult_data = pd.read_csv('/content/content/MyDrive/CodeCamp/Adult Data.csv')
2
3 # вывожу перечень атрибутов и их типы, а также количество ненулевых значений
4 adult_data.info()
```

⇨ <class 'pandas.core.frame.DataFrame'>  
RangeIndex: 32561 entries, 0 to 32560  
Data columns (total 15 columns):

#	Column	Non-Null Count	Dtype
0	age	32561 non-null	int64
1	workclass	32561 non-null	object
2	fnlwgt	32561 non-null	int64
3	education	32561 non-null	object
4	education-num	32561 non-null	int64
5	marital-status	32561 non-null	object
6	occupation	32561 non-null	object
7	relationship	32561 non-null	object
8	race	32561 non-null	object
9	sex	32561 non-null	object
10	capital-gain	32561 non-null	int64
11	capital-loss	32561 non-null	int64
12	hours-per-week	32561 non-null	int64
13	native-country	32561 non-null	object
14	salary	32561 non-null	object

dtypes: int64(6), object(9)  
memory usage: 3.7+ MB

```
1 # вывожу несколько записей датасета
2 adult_data.head(3)
```



	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relat
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	No
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	No

Next steps:



[View recommended plots](#)

[New interactive sheet](#)

## Рассмотрим атрибуты датасета

### Категориальные данные

- age
- workclass

'State-gov' – правительство штата  
'Federal-gov' – федеральное управление  
'Local-gov' – местное управление  
'Self-emp-not-inc' – не зарегистрированный самозанятый  
'Self-emp-inc' – Self employee inc/ зарегистрированный самозанятый  
'Without-pay' – волонтерство/ работа в НКО  
'Never-worked' – никогда не работал  
'Private'  
'?' – не указано

- education

'Bachelors', 'HS-grad', '11th', 'Masters', '9th', 'Some-college',  
'Assoc-acdm', 'Assoc-voc', '7th-8th', 'Doctorate', 'Prof-school',  
'5th-6th', '10th', '1st-4th', 'Preschool', '12th'

- education-num
- marital-status: семейное положение

'Never-married': никогда не был(а) в браке  
'Married-civ-spouse': в зарегистрированном браке (незарегистрированный)  
'Divorced': в разводе  
'Married-spouse-absent': в браке (супруг отсутствует)  
'Separated': в браке с раздельным проживанием  
'Married-AF-spouse': в браке с военнослужащим  
'Widowed': овдовевший

- occupation: род занятий
- relationship

'Not-in-family', 'Husband', 'Wife',  
'Own-child', 'Unmarried', 'Other-relative'

- race

'White', 'Black', 'Asian-Pac-Islander', 'Amer-Indian-Eskimo',  
'Other'

- sex
- native-country

'United-States', 'Cuba', 'Jamaica', 'India', '?', 'Mexico',  
'South', 'Puerto-Rico', 'Honduras', 'England', 'Canada', 'Germany',  
'Iran', 'Philippines', 'Italy', 'Poland', 'Columbia', 'Cambodia',  
'Thailand', 'Ecuador', 'Laos', 'Taiwan', 'Haiti', 'Portugal',  
'Dominican-Republic', 'El-Salvador', 'France', 'Guatemala',  
'China', 'Japan', 'Yugoslavia', 'Peru', 'Outlying-US(Guam-USVI-etc)', 'S  
'Vietnam', 'Hong', 'Ireland', 'Hungary',  
'Holand-Netherlands'

## Количественные данные

---


- fnlwgt /final weight -- взвешенный критерий для корректировки выборки относительно генеральной совокупности.
- education-num -- уровень образования
- capital-gain -- прирост капитала / доход от капитала

- capital-loss -- потеря капитала
- hours-per-week -- кол-во рабочих часов в неделю
- salary

```

1 # каков средний возраст, количество классов образования, среднее рабочее время в неделю
2
3 workclass_describe = adult_data.groupby(by='workclass').agg({'workclass':'count',
4
5                                     'age':'mean',
6                                     'education-num':'mean',
7                                     'hours-per-week':'mean'}).rename(columns={
8                                     'workclass':'number_ci
9                                     'age': 'avg_age',
10                                    'education-num':'avg_e
11                                    'hours-per-week':'avg_
12
13 total_number_citizens = workclass_describe['number_citizens'].sum(axis=0)
14 workclass_describe['part_of_citizens, %'] = round(workclass_describe['number_citizens'] / total_
15
16 workclass_describe

```



	number_citizens	avg_age	avg_education_num	avg_hpw	part_of_c
<b>workclass</b>					
Private	22696	36.797585	9.879714	40.267096	
Self-emp-not-inc	2541	44.969697	10.226289	44.421881	
Local-gov	2093	41.751075	11.042045	40.982800	
?	1836	40.960240	9.260349	31.919390	
State-gov	1298	39.436055	11.375963	39.031587	
Self-emp-inc	1116	46.017025	11.137097	48.818100	
Federal-gov	960	42.590625	10.973958	41.379167	

Next steps:

 [View recommended plots](#)

[New interactive sheet](#)

## наблюдения

- **у никогда не работавших граждан** зафиксировано среднее рабочее время в неделю 28 часов. Возможно, это время отражает время учебной практики: в среднем ~ 4 часа в день при пяти дневной рабочей неделе.
- **работающие без оплаты** в среднем работают в неделю больше, чем у граждан с другими типами занятости при этом данные граждане не имеют завершенного школьного образования
- **~ 70% граждан** ответили, что относятся к Private workclass при этом в среднем они имеют 9 классов образования и в среднем отработывают восьми часовой рабочий день

```
1 # рассмотрим роды деятельности в разрезе рабочих классов
2 # выведем кол-во граждан, занятых в каждом роде деятельности
3
4 adult_data.groupby(by=['workclass', 'occupation']).agg({'occupation': 'count'})
```



		occupation
workclass	occupation	
?	?	1836
Federal-gov	Adm-clerical	317
	Armed-Forces	9
	Craft-repair	64
	Exec-managerial	180
...	...	...
Without-pay	Farming-fishing	6
	Handlers-cleaners	1
	Machine-op-inspct	1
	Other-service	1
	Transport-moving	1



83 rows × 1 columns

```
1 # визуализирую распределение занятости граждан в разрезе подгрупп workclass
2 import matplotlib.pyplot as plt
3
4 pie_colors = plt.cm.Paired.colors
5 plt.figure(figsize=(8,8))
6
7 wedges, texts, autotexts = plt.pie(workclass_describe['part_of_citizens', %'],
8                                     labels=workclass_describe.index, autopct='%1.1f%%',
```

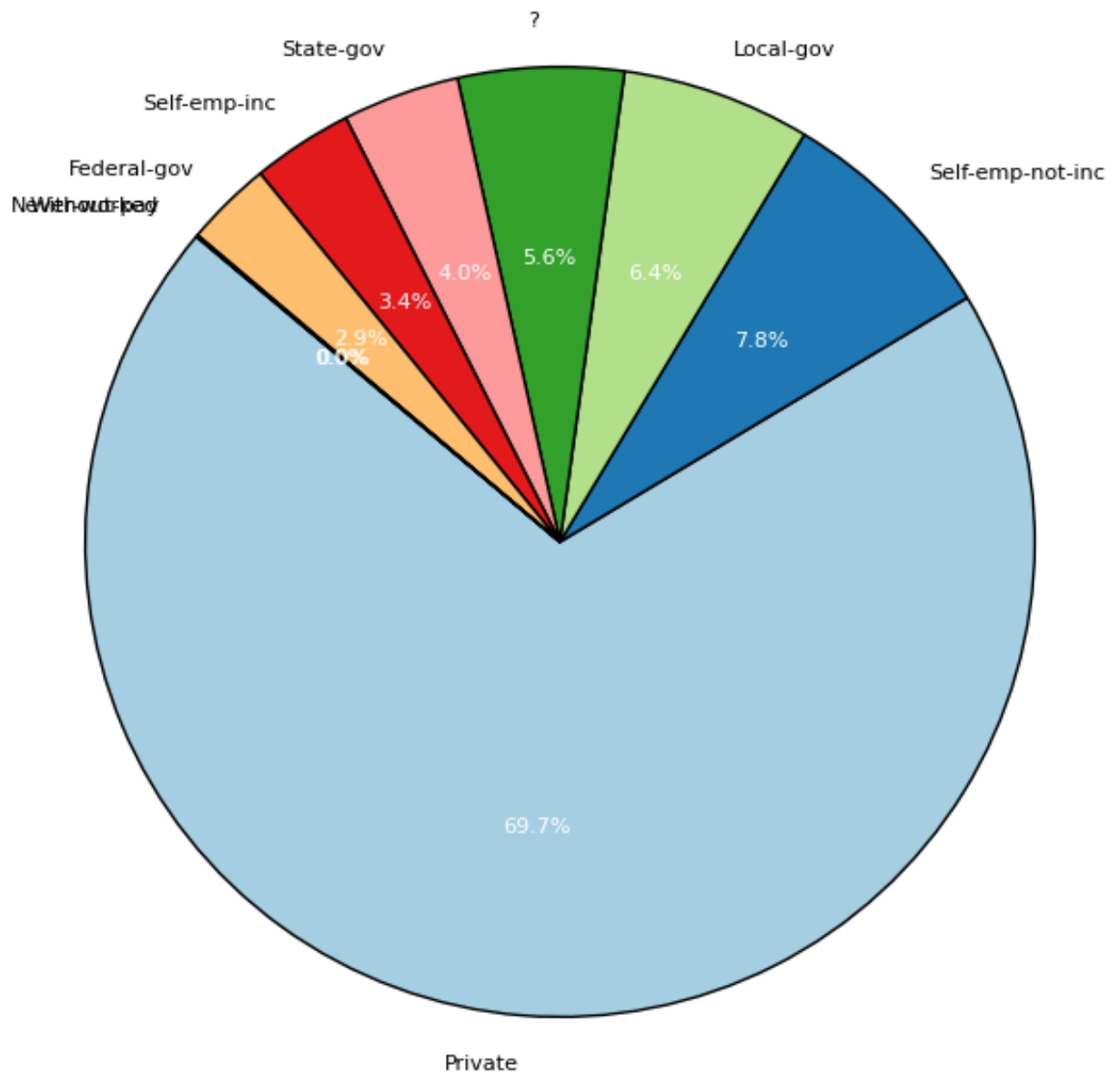
```

9         colors=pie_colors,
10        startangle=140,
11        wedgeprops={'edgecolor':'black'},
12        textprops={'fontsize':8})
13
14
15 for text in texts:
16     text.set_fontsize(8)
17 for autotext in autotexts:
18     autotext.set_fontsize(8)
19     autotext.set_color('white')
20
21 plt.title('Distribution of workclass', fontsize=14, fontweight='bold')
22 plt.show()

```



## Distribution of workclass



```

1 # рассмотрим статистические характеристики количественных данных
2
3 adult_data[['age', 'fnlwgt', 'education-num', 'capital-gain', 'capital-loss', 'hours-per-week']]

```



	age	fnlwgt	education-num	capital-gain	capital-loss	hours-per-week
count	32561.000000	3.256100e+04	32561.000000	32561.000000	32561.000000	32561.000000
mean	38.581647	1.897784e+05	10.080679	1077.648844	87.303830	40.425571
std	13.640433	1.055500e+05	2.572720	7385.292085	402.960219	12.354684
min	17.000000	1.228500e+04	1.000000	0.000000	0.000000	1.000000
25%	28.000000	1.178270e+05	9.000000	0.000000	0.000000	40.000000
50%	37.000000	1.783560e+05	10.000000	0.000000	0.000000	40.000000
75%	48.000000	2.370510e+05	12.000000	0.000000	0.000000	45.000000
max	90.000000	1.484705e+06	16.000000	99999.000000	4356.000000	99.000000

```

1 # строю сводную таблицу в разрезе пола. Рассмотрим кол-во граждан в разрезе полов, медианный возраст
2
3 demographic_data = adult_data.groupby(by='sex').agg({'sex': 'count',
4
5             'age': 'median',
6             'education-num': 'median'
7         })
8 demographic_data['sex'] = np.round(demographic_data['sex'] / demographic_data['sex'].sum(axis=0))
9 demographic_data.rename(columns={'sex': 'part_respondents, %',
10                                'age': 'median_age',
11                                'education-num': 'median education level'}, inplace=True)
12 demographic_data

```



	part_respondents, %	median_age	median education level
sex			
Female	33.08	35.0	10.0
Male	66.92	38.0	10.0



Next steps:

[View recommended plots](#)

[New interactive sheet](#)

## ✓ Происхождение респондентов

для нанесения на карту стран происхождения респондентов переписи, устанавливаю библиотеку geopandas

```
1 pip install geopandas
```

```
⇒ Requirement already satisfied: geopandas in /usr/local/lib/python3.11/dist-  
Requirement already satisfied: numpy>=1.22 in /usr/local/lib/python3.11/dis  
Requirement already satisfied: pyogrio>=0.7.2 in /usr/local/lib/python3.11/  
Requirement already satisfied: packaging in /usr/local/lib/python3.11/dist-  
Requirement already satisfied: pandas>=1.4.0 in /usr/local/lib/python3.11/d  
Requirement already satisfied: pyproj>=3.3.0 in /usr/local/lib/python3.11/d  
Requirement already satisfied: shapely>=2.0.0 in /usr/local/lib/python3.11/  
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/pyt  
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/di  
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/  
Requirement already satisfied: certifi in /usr/local/lib/python3.11/dist-pa  
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-p
```



```

1 import geopandas as gpd
2
3 # Загружаю карту мира из naturalearth
4 world = gpd.read_file('https://naturalearth.s3.amazonaws.com/110m_cultural/ne_110m_admin_0_count
5
6 # Список стран, которые нужно отметить на карте
7 native_countries = adult_data['native-country'].unique()
8 world['select'] = world['ADMIN'].isin(native_countries)
9
10 # Строю карту
11 fig, ax = plt.subplots(figsize=(10, 10))
12 world.plot(ax=ax, color='lightgray', edgecolor='grey')
13 world[world['select']].plot(ax=ax, color='lightblue')
14
15 # Удаляю координатные оси
16 ax.set_xticks([]), ax.set_yticks([]), ax.set_frame_on(False)
17
18 plt.title('Native countries of respondents', fontsize=12, fontweight='medium')
19 plt.show()

```



Native countries of respondents



✓ Детальный анализ количественных атрибутов датасета

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3 import pandas as pd
4 import seaborn as sns
```

## ▼ Age

```
1 ##### age analysis #####
2
3 adult_data.age.describe()
```



age

**count** 32561.000000

**mean** 38.581647

**std** 13.640433

**min** 17.000000

**25%** 28.000000

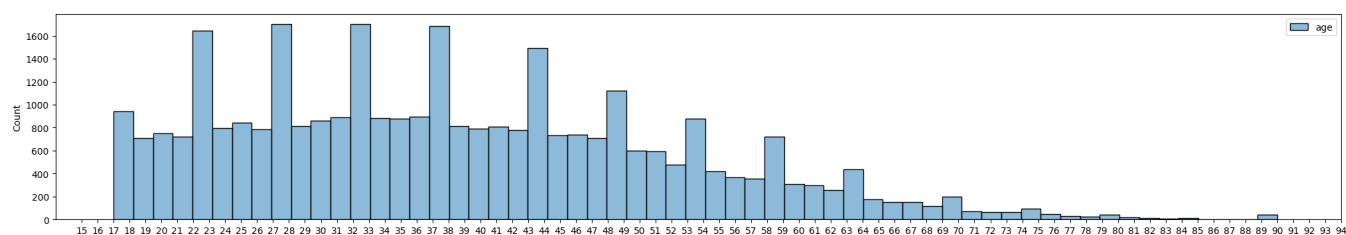
**50%** 37.000000

**75%** 48.000000

**max** 90.000000

**dtype:** float64

```
1 plt.figure(figsize=(25, 4))
2 sns.histplot(adult_data[['age']])
3 plt.xticks(np.arange(15, 95, 1));
```



## анализ графика

- длинный затухающий хвост справа что может свидетельствовать о малой доле пожилого населения в переписи
- 50% данных приходится на диапазон от 28 до 48 лет включительно
- видим мультимодальность. Она проявляется на пиках: [22.5, 27.5, 32.5, 37.5, 43.5, 48.5, 53.5, 58.5, 63.5, 69.5, 74.5, 79.5, 89.5] лет
  - на основе мод выделяю подгруппы:
    - учащиеся  $17 \leq \text{age} < 27.5$
    - молодые специалисты  $27.5 \leq \text{age} < 32.5$
    - профессионалы  $32.5 \leq \text{age} < 58.5$
    - пенсионеры  $58.5 \leq \text{age} < 89.5$

### > Final weight

[ ] ↳ 6 cells hidden

### > Capital gain

[ ] ↳ 5 cells hidden

### ✓ Capital loss

```
1 ##### Capital loss analysis #####
2
3 adult_data['capital-loss'].describe()
```



#### capital-loss

count	32561.000000
mean	87.303830
std	402.960219
min	0.000000
25%	0.000000
50%	0.000000
75%	0.000000
max	4356.000000

**dtype:** float64

```
1 # Рассчитываю отношение стандартного отклонения к среднему
2
3 std_by_mean_capital_loss = round(adult_data['capital-loss'].std()/ adult_data['capital-loss'].me
4 f'Стандартное отклонение превосходит среднее в: {std_by_mean_capital_loss} раз'
```



'Стандартное отклонение превосходит среднее в: 4.6 раз'

Визуализирую распределение потерь капитала

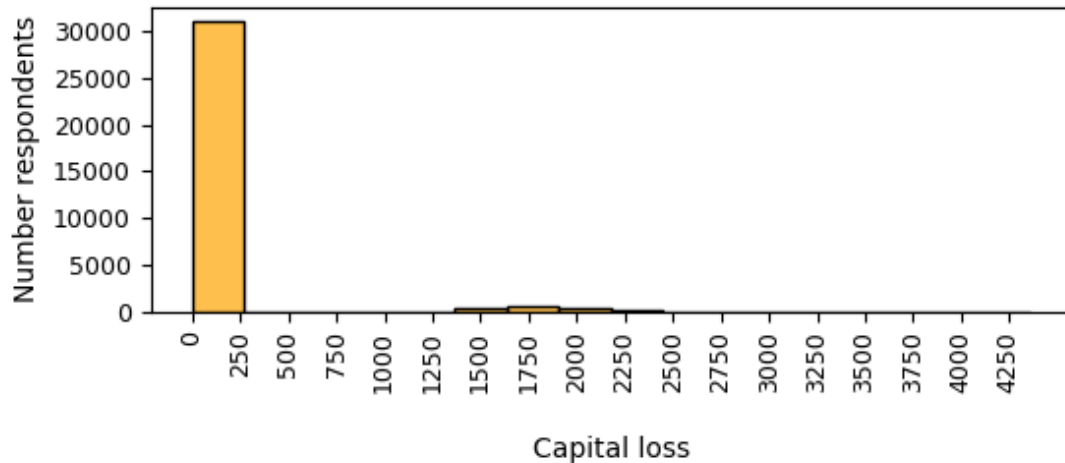
```

1 plt.figure(figsize=(6, 3))
2 sns.histplot(adult_data['capital-loss'], color='orange', alpha=.7)
3
4 plt.title('Гистограмма "Распределение суммарной потери капитала среди респондентов"\n', fontsize
5 plt.xlabel('\nCapital loss'), plt.ylabel('\n\nNumber respondents')
6
7 plt.xticks(np.arange(0, 4500, 250), rotation=90, fontsize=9)
8 plt.yticks(np.arange(0, 35000, 5000), fontsize=9)
9 plt.tight_layout();

```



Гистограмма "Распределение суммарной потери капитала среди респондентов"



### анализ графика

- в среднем потери капитала составляют 87 единиц и при этом стандартное отклонение ~ в 5 раз больше среднего.
- большая часть населения не имеет потерь. Отмечу, что в данных есть записи, в которых ненулевые потери капитала могут быть при нулевом приросте капитала.

```

1 ##### education-num #####
2
3 adult_data['education-num'].describe()

```



education-num	
count	32561.000000
mean	10.080679
std	2.572720
min	1.000000
25%	9.000000
50%	10.000000
75%	12.000000
max	16.000000

**dtype:** float64

## Education num

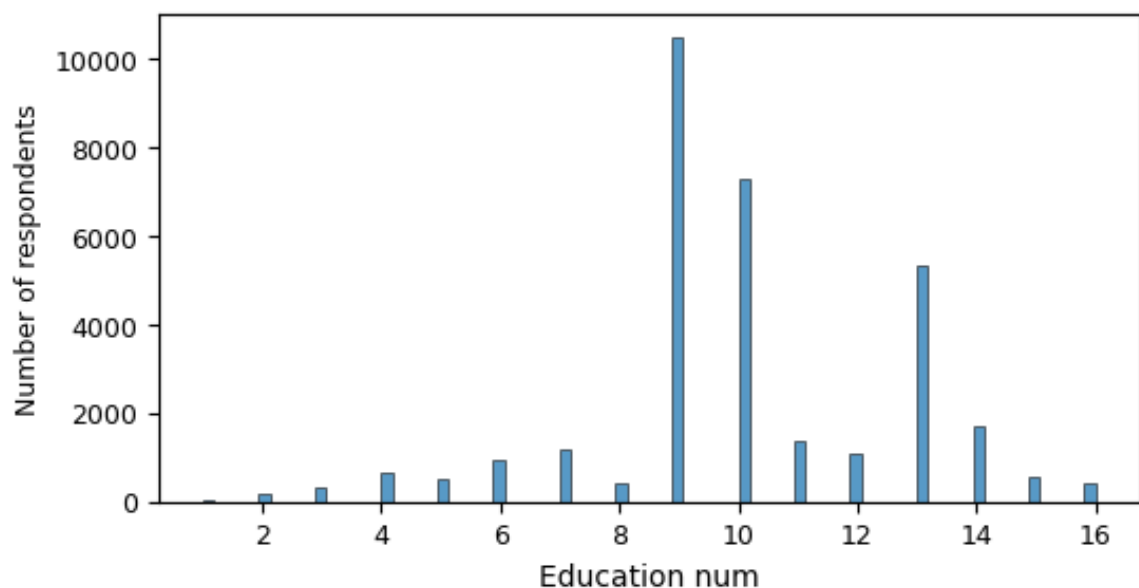
```

1 plt.figure(figsize=(6, 3))
2
3 sns.histplot(adult_data['education-num'])
4
5 plt.title('Распределение количества классов образования среди респондентов\n')
6 plt.xlabel('Education num'), plt.ylabel('Number of respondents', fontsize=9)
7 plt.xticks(fontsize=9), plt.yticks(np.arange(0, 12000, 2000), fontsize=9);

```



## Распределение количества классов образования среди респондентов



```

1 size_dataset = len(adult_data)
2
3 # количество граждан с полным школьным образованием: пройдено 10 классов
4 has_school_education = len(adult_data[adult_data['education-num'] == 10])
5 has_school_education_percent = round(has_school_education / size_dataset , 2) * 100
6
7 hasnt_school_education = len(adult_data[adult_data['education-num'] < 10])
8 hasnt_school_education_percent = round(hasnt_school_education / size_dataset , 2) * 100
9
10 # количество граждан с бакалаврским образованием: 11–14 классы
11 has_bachelor_grade = len(adult_data[adult_data['education-num'] == 14])
12 hasnt_bachelor_grade = len(adult_data[(adult_data['education-num'] > 10) & (adult_data['educatio
13
14 has_bachelors_grade_percent = round(has_bachelor_grade / size_dataset , 2) * 100
15 hasnt_bachelors_grade_percent = round(hasnt_bachelor_grade / size_dataset , 2) * 100
16
17 # количество граждан с магистерским образованием: 15–16 классы
18 has_master_grade = len(adult_data[adult_data['education-num'] == 16])
19 hasnt_master_grade = len(adult_data[(adult_data['education-num'] > 14) & (adult_data['education-
20
21 has_master_grade_percent = round(has_master_grade / size_dataset , 2) * 100
22 hasnt_master_grade_percent = round(hasnt_master_grade / size_dataset , 2) * 100
23
24
25 print(f'количество граждан с полным школьным образованием: {has_school_education}({has_school_ed
26 print(f'\nколичество граждан с бакалаврским образованием: {has_bachelor_grade}({has_bachelors_gr
27 print(f'\nколичество граждан с магистерским образованием: {has_master_grade}({has_master_grade_p

```

➡ количество граждан с полным школьным образованием: 7291(22.0%) человек  
не имеют школьного образования: 14754(45.0%)

количество граждан с бакалаврским образованием: 1723(5.0%) человек  
не имеют бакалаврского образования: 7804(24.0%)

количество граждан с магистерским образованием: 413(1.0%) человек  
не имеют магистерского образования: 576(2.0%)

## анализ графика

- количество граждан с полным школьным образованием меньше количества граждан без полного школьного образования в больше чем 2 раза
- 45% граждан не имеют школьного образования
- 22% граждан имеют 10 классов образования
- 24% граждан имеют школьное образование и не имеют степени бакалавра
- 5% имеют степень бакалавра
- 2% граждан имеют бакалаврское и не имеют магистерского образования
- 1% граждан имеют магистерское образование

```

1 # рассмотрим финансовые показатели каждой группы населения в разрезе уровня образования
2 adult_data.pivot_table(index='education-num', values=['age', 'capital-gain', 'capital-loss'],
3                        aggfunc={'age': 'mean',
4                                'capital-gain': 'mean',
5                                'capital-loss': 'mean'}).sort_values('capital-gain',
6                                                                ascending=False)

```



**age capital-gain capital-loss**



**education-num**



<b>15</b>	44.746528	10414.416667	231.203125
<b>16</b>	47.702179	4770.145278	262.845036
<b>14</b>	44.049913	2562.563552	166.719675
<b>13</b>	38.904949	1756.299533	118.350327
<b>1</b>	42.764706	898.392157	66.490196
<b>11</b>	38.553546	715.051375	72.754703
<b>12</b>	37.381443	640.399250	93.418932
<b>10</b>	35.756275	598.824167	71.637087
<b>9</b>	38.974479	576.800114	70.466622
<b>6</b>	37.429796	404.574491	56.845659
<b>5</b>	41.060311	342.089494	28.998054
<b>8</b>	32.000000	284.087760	32.337182
<b>4</b>	48.445820	233.939628	65.668731
<b>7</b>	32.355745	215.097872	50.079149
<b>3</b>	42.885886	176.021021	68.252252
<b>2</b>	46.142857	125.875000	48.327381

▼ Hours-per-week



```
1 ##### hours-per-week analysis #####
2 adult_data['hours-per-week'].describe()
```



### hours-per-week

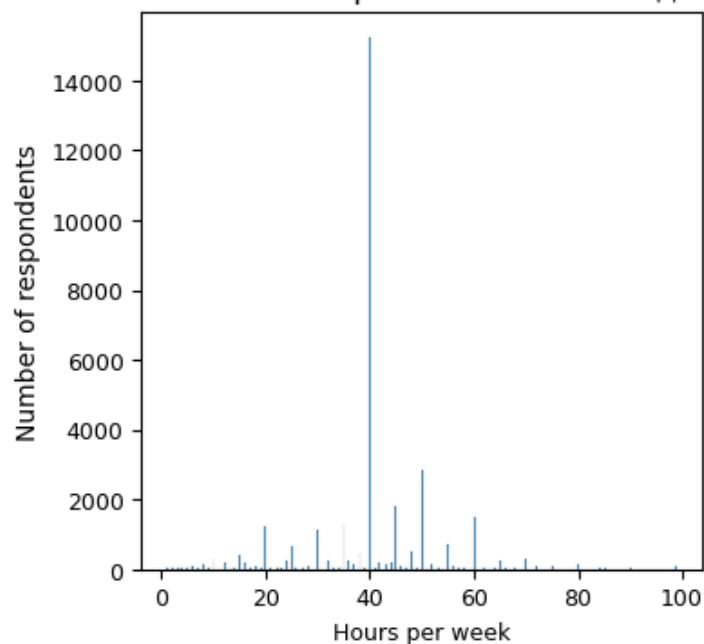
count	32561.000000
mean	40.437456
std	12.347429
min	1.000000
25%	40.000000
50%	40.000000
75%	45.000000
max	99.000000

**dtype:** float64

```
1 plt.figure(figsize=(4, 4))
2
3 plt.title('Распределение количества занятых работой часов в неделю среди респондентов')
4
5 plt.xlabel('Hours per week', fontsize=9), plt.ylabel('Number of respondents')
6 plt.xticks(fontsize=9), plt.yticks(fontsize=9)
7 sns.histplot(adult_data['hours-per-week']);
```



### Распределение количества занятых работой часов в неделю среди респондентов



```
1 # вывожу сводную таблицу количества рабочих часов в неделю в разрезе возраста респондентов
2 adult_data.pivot_table(index='age', values=['hours-per-week'], aggfunc={'hours-per-week':'mean'})
```



**hours-per-week**



**age**



**17** 21.367089

**18** 25.912727

**19** 30.678371

**20** 32.280212

**21** 34.034722

... ...

**85** 29.333333

**86** 40.000000

**87** 2.000000

**88** 40.000000

**90** 36.813953

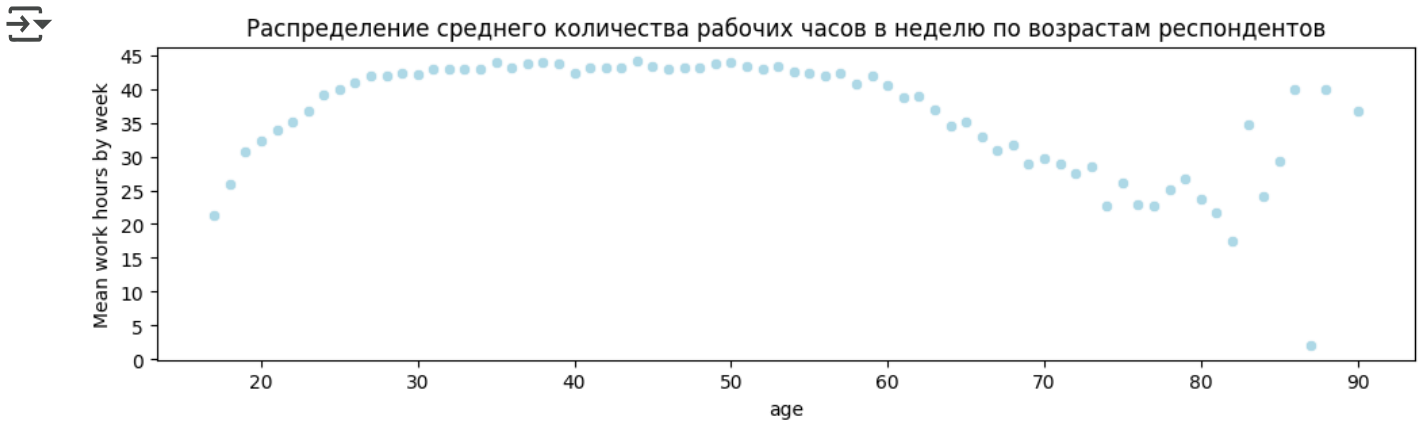
73 rows x 1 columns

```
1 # среднее количество рабочих часов в по возрастным группам
2 work_hours_by_week_for_age = adult_data.pivot_table(index='age', values=['hours-per-week'], aggf
3 work_hours_by_week_for_age['mean_work_hours_by_week'] = round(work_hours_by_week_for_age['mean_w
```

```

1 # визуализация распределение среднего количества рабочих часов в неделю вдоль возрастной шкалы
2
3 from matplotlib import pyplot as plt
4 import seaborn as sns
5
6 plt.figure(figsize=(12, 3))
7 plt.title('Распределение среднего количества рабочих часов в неделю по возрастам респондентов')
8
9 sns.scatterplot(x='age', y='mean_work_hours_by_week', data=work_hours_by_week_for_age, color='li
10
11 plt.yticks(np.arange(0, 50, 5))
12 plt.ylabel('Mean work hours by week');

```



## Анализ графика

- Минимальное количество рабочих часов в неделю в среднем составляет 1 час.
- начиная с 17 лет среднее количество рабочих часов в месяц более 20 часов (может быть подработка в школьные или студенческие годы)
- граждане пенсионного возраста (стандартный возраст выхода на пенсию – от 55 лет)
- от 60 до ~ 75 лет видим плавное снижение средней продолжительности рабочего времени в неделю
- от 80 до ~ 83.5 лет происходит резкое снижение ( $< 10$  часов) средней продолжительности рабочего времени в неделю
- наблюдаем резкий рост (до 40 часов) средней продолжительности рабочего времени в неделю
- максимальная продолжительность рабочей недели равна 99 часов.  
Потенциально, это либо говорит об ошибке в данных либо о наличии граждан, работающие по 19.5 часов в день при пяти дневном рабочем графике или по 14 часов при семи дневном рабочем графике. Оба варианта говорят о высоком уровне нагрузки на сотрудника.

## ➤ Ответы на вопросы / Answers to questions

[ ] ↪ 8 cells hidden