

Нулевая гипотеза:

Мы предполагаем, что самые популярные основные источники дохода не изменились за десятилетие, но изменилась их популярность среди разных возрастных групп.

Обоснование выбора файлов:

Мы собираемся исследовать изменчивость основного способа заработка населения в России, для этого мы выбрали данные в переписи населения по основному источнику средств к существованию о доходах населения за 2010 и 2020 годы”. Нам было интересно узнать , как изменился способ заработка граждан, выделенных по возрастным категориям в 2020 году от картины в 2010, именно для этого мы выбрали эти файлы.

Обоснование выбора такого вида словарей:

Мы собрали данные из каждой таблицы в 4 словаря: В словарях d_ev_2010, d_ev_2020 за 2010 и за 2020 годы соответственно ключом является название видов дохода виды доходов, а значениями - словари, в которых ключами будут являться категории населения (по возрасту), а значениями - количество человек данной категории, получающие этот доход. Мы выбрали такую структуру словаря, поскольку благодаря ней видно какое количество людей определенных возрастных групп зарабатывает способом, указанном как ключ

Кроме того, мы также использовали словари d_total_2010, d_total_2020 за 2010 и за 2020 годы соответственно, где ключами являлись виды доходов, а значениями- суммарное количество человек, которые используют этот финансовый инструмент для заработка. Мы использовали такую структуру словаря, поскольку в рамках нашего исследования мы хотели узнать, как сильно изменились подходы к заработку среди населения за прошедшие 10 лет вне зависимости от возраста в связи с новыми тенденциями (например, возникло понятие “lifelong education”, распространились идеи пассивного заработка).

В ходе нашего исследования, нами были подсчитаны следующие описательные статистики:

- среднее количество людей с определенным источником дохода
- медианный возраст по каждому источнику дохода (а также нижние и верхние квартили, нижние и верхние границы, нехарактерные значения)
- математическое ожидание возраста по каждому источнику дохода
- корреляцию Спирмена между возрастом и видом доходов
- самый популярный и непопулярный основной источник дохода

Интерпретация среднего значения по каждому источнику дохода:

Интерпретируя результаты по среднему, рассмотрим показатели, которые сильно изменились:

- 1) Среднее количество людей, которые получают стипендию сильно увеличилось (приблизительно в 3.5 раз). Скорее всего это связано с тем, что появился тренд на пожизненное образование. Кейсы, когда люди идут получать высшее образование, будучи старше 20 лет, сильно увеличились. Это связано с тем, что стало модным понятие “саморазвитие”, которое подразумевает обучение чему-то новому на регулярной основе.
- 2) Среднее количество человек, получающие доход от сдачи внаем или в аренду имущества, патентов и авторских прав критично возросло(приблизительно в 3.5 раз). Это можно связать с тем, что современное платежеспособное поколение перестало демонизировать творческие профессии и стартапы, а также стало более образовано в сфере финансовой грамотности, поскольку вклад в имущество является хорошим финансовым инструментом, который помогает сберечь деньги от роста инфляции. Помимо этого законодательством были созданы более благоприятные условия, защищающие обе стороны договора о сдаче в аренду имущества, что сделало его выгоднее, а значит число белых схем увеличилось.

- 3) Среднее количество человек, получающих доход от пенсии, пособия и другие выплаты от организаций, государства возросло (на приблизительно 1 миллион человек), что скорее всего связано с тенденцией старения населения РФ и, соответственно, увеличения числа людей, получающих пенсию и не являющихся трудоспособными.
- 4) Среднее количество человек, получающих доход от сбережения, дивиденды, проценты, увеличилось вдвое, что может быть связано с быстрым ростом инфляции и повышением финансовой грамотности среди населения.

Интерпретация математического ожидания и медианы

Эти показатели мы будем интерпретировать в связке, поскольку медианы проверяют корректность математического ожидания. (стоит отметить, что медиана никак не изменилась, в отличие от математического ожидания.

Однако благодаря другим показателям, таким как квартили и нехарактерные значения, мы понимаем, что математическое ожидание определено корректно)

Согласно изменениям в математическом ожидании, мы можем сделать выводы:

- 1) Что ожидаемый возраст личного подсобного хозяйства увеличился (с 40.74 в 2010 до 49.65 в 2020), по названным ранее причинам
- 2) Ожидаемый возраст людей, которые получают доход со сбережений, дивидендов, процентов вырос (с 39.5 в 2010 до 47.09 в 2020). Это произошло, поскольку в период с 2010 по 2020 год темпы инфляции увеличились, и люди стали больше задумываться о том, как безопасно сберечь деньги от инфляции.

Интерпретация корреляции Спирмена

Мы выбрали ее, поскольку данный показатель корреляции является устойчивым. Говоря про корреляцию Спирмена между возрастом и видом доходов, можем отметить следующее:

- 1) Изменилась корреляция для личного подсобного хозяйства (она была отрицательной, стала положительной). Мы считаем, что это связано с

увеличением продолжительностью жизни (раньше меньший процент людей доживало до пенсии, на которой люди начинают вести такой образ жизни). Кроме того, вероятно, опыт covid-19 сыграл роль в изменении корреляции Спирмена, так как многие пенсионеры именно в тот период переехали на дачи, где смогли открыть для себя новый вид заработка.

- 2) Изменилась корреляция для стипендии (она снизилась с точки зрения обратной пропорциональности, то есть большее количество старших людей начало/продолжило получать высшее образование и получать доход от стипендии). Мы считаем, что это связано с упомянутой выше тенденцией непрерывного образования.
- 3) Изменилась корреляция для сбережений, дивидендов, процентов (она была отрицательной, стала положительной). Мы считаем, что это связано с тем, что старшее поколение, имеющее тенденцию хранить свои средства в золоте/недвижимости начало бороться с инфляцией, создавая активы.

Интерпретация минимального и максимального источника дохода:

Интерпретируя результаты по самым популярным и непопулярным основным источникам дохода, можно сделать следующие выводы:

Самый популярный вид дохода не изменился, в то время как самый непопулярным вместо сдачи внаем или в аренду имущества, патентов и авторских прав стал заработок на чем-то другом, не указанном в таблицах. Это произошло по тем же причинам, что и рост среднего количества человек, получающие доход от сдачи внаем или в аренду имущества, патентов и авторских прав. Отсюда, он перестал быть непопулярным и заменился другим.

Интерпретация рейтинга:

Как мы и предполагали в нулевой гипотезе, самые популярные источники дохода не сильно изменились за десятилетие: лидирующее место все еще

занимает заработная плата, поменялись лишь местами иждивение и пенсии (иждивение спустилось на третье место, а пенсии наоборот поднялись в рейтинге). Данная тенденция объясняется следующим: несмотря на государственные программы, в стране все равно наблюдается низкий уровень рождаемости и “старение” населения. Таким образом, число пожилых людей возросло (как и основного источника их заработка – пенсии), а число детей растет недостаточно быстро и не может выровняться с высоким уровнем “старения”(поэтому иждивение как основной источник дохода несовершеннолетних уступил свое место в рейтинге пенсиям).

```

###
### Для начала откроем файл за 2010 год и считаем его данные в словарь
###

with open ('КОД2010.csv', encoding = 'utf-8') as infol:
    d_total_2010 = {} #d_total_2010 - словарь, в котором ключами будут
    являться виды доходов, а значениями - количество человек, получающие
    его (за 2010 год)
    d_ev_2010 = {} #d_ev_2010 - словарь, в котором ключами будут
    являться виды доходов, а значениями - словари, в которых ключами будут
    являться категории населения (по возрасту), а значениями - количество
    человек данной категории, получающие этот доход (за 2010 год)
    age2010 = [] #age2010 - список, в котором будут содержаться
    различные категории населения по возрасту (age2010 = ['до 15', '15 -
    19', '20 - 29', '30 - 49', '50 - 54', '55 - 59', '60 - 64', '65 и
    более'])
    counter2010 = 0 #создадим переменную счетчик, чтобы на моменте
    создания списка категорий населения перебрать строки только из первой
    строки, так как там содержится информация о них
    for line in infol: #переберем строки файла
        line = line.strip().split(';') #уберем лишние пробелы и
        разделим данные по ';'
        if line[0][0].isalpha(): #поскольку для составления словаря
        d_total_2010 необходимо перебрать данные со второй строки по
        последнюю, составим условие, чтобы проигнорировать первую строку. Так
        как первая строка пустая, под условие 'if line[0][0].isalpha():' она
        не подойдет и будет пропущена
        d_total_2010[line[0]] = int(line[1].replace(' ', ''))
        #составим словарь и уберем пробелы из данных о количестве с помощью
        метода '.replace()', чтобы превратить данные в числовой формат для
        удобства дальнейших вычислений
        while counter2010 != 1: #составим условие, чтобы программа
        запустила цикл только по первой строке с помощью счетчика
        'counter2010'
            for category in line: #переберем все данные в первой
            строке файла, чтобы составить из него список
                if category != '\uffff' and category != '' and category
                != 'Мужчины и женщины': #поскольку в списке нам нужна информация
                только о том, на какие категории населения по возрасту идет
                разделение, нам следует написать условие, чтобы в список не
                добавились пустые ячейки (category != '\uffff' and category != '') и
                ячейка 'Всего (мужчины и женщины)' (category != 'Мужчины и женщины')
                age2010.append(category) #с помощью
                метода .append() добавим строчные перебираемые объекты в список
                counter2010 += 1 #прибавим к переменной - счетчику 1,
                чтобы после перебора первой строки программа остановилась

#далее займемся составлением словаря d_ev_2010

```

#поскольку d_ev_2010 - словарь словарей, то сначала мы определим ключи данного словаря - виды источников дохода

if line[0] != '\uffff': #нулевой объект первой строки не содержит нужную нам для ключей словаря информацию, поэтому составим условие для того, чтобы проигнорировать эту строку

d_ev_2010 [line[0]] = {}

for i in range(len(age2010)): #начнем перебор индексов с помощью списка age2010, в котором объект:'до 15' имеет индекс 0, объект:'15 – 19' имеет индекс 1 и так далее

if line[i+2] == '-': #or line[i+2] == '': #поскольку для анализа данных словарей, которые у нас получаются, нам нужно будет работать с данными числового формата, напомним условие, чтобы программа заменяла данные '-' на 0

d_ev_2010 [line[0]][age2010[i]] = 0

else: #если значения словаря словаря содержат числа в строчном формате, тогда напомним условие, чтобы программа удаляла из них пробелы и конвертировала их в числовой формат

d_ev_2010 [line[0]][age2010[i]] =

int(line[i+2].replace(' ', ''))

print(f'Словарь по общему количеству человек по каждому источнику дохода в 2010 году: {d_ev_2010}')

print(f'\nСловарь по общему количеству человек по возрастам по каждому источнику дохода в 2010 году: {d_total_2010}')

###

Для начала откроем файл за 2020 год и считаем его данные в словарь
###

with open ('К0Д2020.csv', encoding = 'utf-8') as info2:

d_total_2020 = {} #d_total_2020 - словарь, в котором приведены общие суммы по каждому виду доходов за 2020 год

d_ev_2020 = {} #d_ev_2020 - словарь, в котором приведены доходы каждой возрастной категории по каждому виду доходов за 2020 год

age2020 = [] #age2020 - список, в котором будут содержаться различные категории населения по возрасту (age2020 = ['до 15', '15 – 19', '20 – 29', '30 – 49', '50 – 54', '55 – 59', '60 – 64', '65 и более'])

counter2020 = 0 #создадим переменную счетчик, чтобы на моменте создания списка категорий населения перебрать строки только из первой строки, так как там содержится информация о них

for line in info2: #переберем строки файла

line = line.strip().split(';') #уберем лишние пробелы и разделим данные по ';'

if line[0][0].isalpha(): #поскольку для составления словаря d_total_2010 необходимо перебрать данные со второй строки по последнюю, составим условие, чтобы проигнорировать первую строку. Так

```

как первая строка пустая, под условие 'if line[0][0].isalpha():' она
не подойдет будет пропущена
    d_total_2020[line[0]] = int(line[1].replace(' ', ''))
#составим словарь и уберем пробелы из данных о количестве с помощью
метода '.replace()', чтобы превратить данные в числовой формат для
удобства дальнейших вычислений
    while counter2020 != 1: #составим условие, чтобы программа
запустила цикл только по первой строке с помощью счетчика
'counter2020'
        for category in line: #переберем все данные в первой
строке файла, чтобы составить из него список
            if category != '\uffeff' and category != '' and category
!= 'Мужчины и женщины': #поскольку в списке нам нужна информация
только о том, на какие категории населения по возрасту идет
разделение, нам следует написать условие, чтобы в список не
добавились пустые ячейки (category != '\uffeff' and category != '') и
ячейка 'Всего (мужчины и женщины)' (category != 'Мужчины и женщины')
                age2020.append(category) #с помощью
метода .append() добавим строчные перебираемые объекты в список
                counter2020 += 1 #age = ['до 15', '15 – 19', '20 – 29',
'30 – 49', '50 – 54', '55 – 59', '60 – 64', '65 и более']
#далее займемся составлением словаря d_ev_2020
#поскольку d_ev_2020 - словарь словарей, то сначала мы определим ключи
данного словаря - виды источников дохода
            if line[0] != '\uffeff': #нулевой объект первой строки не
содержит нужную нам для ключей словаря информацию, поэтому составим
условие для того, чтобы проигнорировать эту строку
                d_ev_2020 [line[0]] = {}
                for i in range(len(age2020)): #начнем перебор индексов с
помощью списка age2020, в котором объект:'до 15' имеет индекс 0,
объект:'15 – 19' имеет индекс 1 и так далее
                    if line[i+2] == '-': #поскольку для анализа данных
словарей, которые у нас получатся, нам нужно будет работать с данными
числового формата, напишем условие, чтобы программа заменяла данные '-'
на 0
                        d_ev_2020 [line[0]][age2020[i]] = 0
                    else: #если значения словаря словаря содержат числа в
строчном формате, тогда напишем условие, чтобы программа удаляла из
них пробелы и конвертировала их в числовой формат
                        d_ev_2020 [line[0]][age2020[i]] =
int(line[i+2].replace(' ', ''))
                print(f'\nСловарь по общему количеству человек по каждому
источнику дохода в 2020 году: {d_ev_2020}')
                print(f'\nСловарь по общему количеству человек по возрастам по
каждому источнику дохода в 2020 году: {d_total_2020}')

###
### Создадим функцию, которая будет считать среднее количество

```



```
человек, получающих деньги указанным способом
###
```

```
def average(d_ev): #определим функцию, которая, принимая на вход
словарь, ключами в котором являются виды источников дохода,
# а значениями- словари, в которых ключами являются возрастные
диапазоны, а значениями- количество человек, которые входят в этот
возрастной
```

```
#диапазон и зарабатывают указанным способом, будет считать среднее
количество человек, получающих деньги указанным способом
```

```
    lst={} # создадим словарь, в котором ключами будут являться виды
источников дохода, а значениями-среднее количество человек, которые
получают таким образом доход
```

```
    for k,v in d_ev.items(): #переберем значения и ключи исходного
словаря
```

```
        avg=sum(v.values())/len(v) #среднее в нашем случае будет сумма
всех людей всех возрастов, которые имеют определенный вид дохода,
разделенное на количество возрастных групп
```

```
        lst[k]=round(avg) # создаем в нашем словаре ключ, значением
которого будет округленное среднее количество человек по возрастным
группам, которые имеют такой вид заработка
```

```
    return lst #функция возвращает словарь с данными
```

```
###
```

```
### Выведем результаты функции по словарям 2010 и 2020 года
```

```
###
```

```
average_2010 = average(d_ev_2010)
```

```
average_2020 = average(d_ev_2020)
```

```
print(f'\nСреднее значение количества человек по каждому возрасту и
источнику дохода в 2010 году: {average_2010}')
```

```
print(f'\nСреднее значение количества человек по каждому возрасту и
источнику дохода в 2020 году: {average_2020}')
```

```
###
```

```
### Создадим функцию, которая будет считать медиану, нижнюю и верхнюю
квартили, нижнюю и верхнюю границы и нехарактерные значения возраста
по каждому источнику дохода
```

```
###
```

```
def mediana_age(d_age):#создаем функцию, которая будет считать
медиану, нижнюю и верхнюю квартили, нижнюю и верхнюю границы и
нехарактерные значения по каждому источнику дохода
```

```
    d_med_age={} #создаем пустой словарь для будущих данных по каждому
источнику
```

```
    for k in d_age: #перебираем ключи (то есть каждый источник дохода)
в словаре вида ключ - источник дохода, значение - словарь (вида ключ -
```

```

возраст, значение - количество людей с таким источником дохода в виде
основного)
    lst=[] #создаем пустой список для значений словаря в словаре
    d_med_age[k]={} #создаем пустой словарь под ключом конкретного
источника дохода
    for el in d_age[k]: #перебираем все ключи в словаре исходного
словаря под ключом конкретного источника дохода
        ages = list(map(int, el.split(' - '))) #чтобы найти
среднее значение диапазона (для дальнейших подсчетов), создадим
список, разделенный тире, который будет задавать начало и конец
диапазона
        age = sum(range(ages[0], ages[1]+1))/len(range(ages[0],
ages[1]+1)) #посчитаем среднее, используя последовательность по началу
диапазона и концу (на единицу больше, так как в последовательности
конец не включен). Мы суммируем числа последовательности и дел
        lst.append(age) #добавляем все значения (возраста) словаря
в словарь в список
    var_line=sorted(lst) #сортируем список, преобразовывая его в
вариационный ряд
    if len(var_line)%2==0: #проверяем четно ли количество членов
вариационного ряда (для дальнейшего подсчета медианы), чтобы пройти
под условия, элементов должно быть четное количество
        a=int(len(var_line)/2) #создаем переменную, значение
которой - половина количества ряда (нужно для индекса переменных,
находящихся в середине вариационного ряда)
        mediana=(var_line[a-1]+var_line[a])/2 #ищем медиану -
среднее арифметическое элементов в середине ряда (так как индексация
идет с 0, у первого члена отнимаем единицу)
        if a%2==0: #проверяем четно ли половина количества членов
вариационного ряда (для дальнейшего подсчета квартилей), чтобы пройти
под условия, их должно быть четное количество
            b=int(a/2) #создаем переменную, значение которой -
четверть количества ряда (нужно для индекса переменных, находящихся в
середине первой половины вариационного ряда)
            nij_kvartil=(var_line[b-1]+var_line[b])/2 #ищем нижнюю
квартиль - среднее арифметическое элементов в середине первой половины
ряда (так как индексация идет с 0, у первого члена отнимаем единицу)
            verh_kvartil=(var_line[a+b-1]+var_line[a+b])/2 #ищем
верхнюю квартиру - среднее арифметическое элементов в середине второй
половины ряда (так как индексация идет с 0, у первого члена отнимаем
единицу)
            mejkvart_razmah=verh_kvartil-nij_kvartil #находим
межквартильный размах (разница между квартилями)
            nij_gran=nij_kvartil-1.5*mejkvart_razmah #находим
нижнюю границу по формуле
            verh_gran=verh_kvartil+1.5*mejkvart_razmah #находим
верхнюю границу по формуле
        else: #проверяем четно ли половина количества членов
вариационного ряда (для дальнейшего подсчета квартилей), чтобы пройти

```

```

под условия, их должно быть нечетное количество
    b=int(a/2) #создаем переменную, значение которой -
четверть количества ряда (нужно для индекса переменной, находящейся в
середине первой половины вариационного ряда)
    nij_kvartil=var_line[b] #находим нижнюю квартиль -
серединный элемент первой половины ряда
    verh_kvartil=var_line[b] #находим верхнюю квартиль -
серединный элемент второй половины ряда
    mejkvart_razmah=verh_kvartil-nij_kvartil #находим
межквартильный размах (разница между квартилями)
    nij_gran=nij_kvartil-1.5*mejkvart_razmah #находим
нижнюю границу по формуле
    verh_gran=verh_kvartil+1.5*mejkvart_razmah #находим
верхнюю границу по формуле
else: #проверяем четно ли количество членов вариационного ряда
(для дальнейшего подсчета медианы), чтобы пройти под условия,
элементов должно быть нечетное количество
    a=int(len(var_line)//2) #создаем переменную, значение
которой - половина количества ряда (нужно для индекса переменной,
находящейся в середине вариационного ряда)
    mediana=var_line[a] #находим медиану - середину
вариационного ряда
    if a%2==0: #проверяем четно ли половина количества членов
вариационного ряда (для дальнейшего подсчета квартилей), чтобы пройти
под условия, их должно быть четное количество
        b=int(a/2) #создаем переменную, значение которой -
четверть количества ряда (нужно для индекса переменных, находящихся в
середине первой половины вариационного ряда)
        nij_kvartil=(var_line[b-1]+var_line[b])/2 #ищем нижнюю
квартиль - среднее арифметическое элементов в середине первой половины
ряда (так как индексация идет с 0, у первого члена отнимаем единицу)
        verh_kvartil=(var_line[a+b-1]+var_line[a+b])/2 #ищем
верхнюю квартиль - среднее арифметическое элементов в середине второй
половины ряда (так как индексация идет с 0, у первого члена отнимаем
единицу)
        mejkvart_razmah=verh_kvartil-nij_kvartil #находим
межквартильный размах (разница между квартилями)
        nij_gran=nij_kvartil-1.5*mejkvart_razmah #находим
нижнюю границу по формуле
        verh_gran=verh_kvartil+1.5*mejkvart_razmah #находим
верхнюю границу по формуле
    else: #проверяем четно ли половина количества членов
вариационного ряда (для дальнейшего подсчета квартилей), чтобы пройти
под условия, их должно быть нечетное количество
        b=int(a/2) #создаем переменную, значение которой -
четверть количества ряда (нужно для индекса переменной, находящейся в
середине первой половины вариационного ряда)
        nij_kvartil=var_line[b] #находим нижнюю квартиль -
серединный элемент первой половины ряда

```

```

        verh_kvartil=var_line[b] #находим верхнюю квартиль -
серединный элемент второй половины ряда
        mejkvart_razmah=verh_kvartil-nij_kvartil #находим
межквартильный размах (разница между квартилями)
        nij_gran=nij_kvartil-1.5*mejkvart_razmah #находим
нижнюю границу по формуле
        verh_gran=verh_kvartil+1.5*mejkvart_razmah #находим
верхнюю границу по формуле
        neharacter_znachenia=[] #создаем пустой список для
нехарактерных значений
        for vi in d_age[k]: #перебираем ключ словаре исходного словаря
под ключом конкретного источника дохода
            agees = list(map(int, el.split(' - '))) #чтобы найти
среднее значение диапазона (для дальнейших подсчетов), создадим
список, разделенный тире, который будет задавать начало и конец
диапазона
            v = sum(range(ages[0], ages[1]+1))/len(range(ages[0],
ages[1]+1)) #посчитаем среднее, используя последовательность по началу
диапазона и концу (на единицу больше, так как в последовательности
конец не включен). Мы суммируем числа последовательности и дел
            if v<nij_gran or v>verh_gran: #сравниваем значение с
границами, найденными выше (условие проходит если значение в границы
не попало)

                cortej=(ele, v) #создаем кортеж из ключа и значения
                neharacter_znachenia.append(cortej) #добавляем кортеж
в список
            if v==mediana: #сравниваем значение с медианой (проходит,
если равно)
                mediana=(ele, v) #в случае, если равно, чтобы
использовать данные по максимуму перезапишем медиану кортежем с ключом
и значением
            if v==nij_kvartil: #сравниваем значение с нижней квартилью
(проходит, если равно)
                nij_kvartil=(ele, v) #в случае, если равно, чтобы
использовать данные по максимуму перезапишем нижнюю квартиль кортежем
с ключом и значением
            if v==verh_kvartil: #сравниваем значение с верхней
квартилью (проходит, если равно)
                verh_kvartil=(ele, v) #в случае, если равно, чтобы
использовать данные по максимуму перезапишем верхнюю квартиль кортежем
с ключом и значением
            d_med_age[k]['медиана']=mediana #добавим в словарь словарей
под ключом конкретного источника дохода и под ключом "медиана" ее
значение
            d_med_age[k]['нижняя квартиль']=nij_kvartil #добавим в словарь
словарей под ключом конкретного источника дохода и под ключом "нижняя
квартиль" ее значение
            d_med_age[k]['верхняя квартиль']=verh_kvartil #добавим в
словарь словарей под ключом конкретного источника дохода и под ключом

```

```

"верхняя квартиль" ее значение
    d_med_age[k]['нижняя граница']=nij_gran #добавим в словарь
словарей под ключом конкретного источника дохода и под ключом "нижняя
граница" ее значение
    d_med_age[k]['верхняя граница']=verh_gran #добавим в словарь
словарей под ключом конкретного источника дохода и под ключом "верхняя
граница" ее значение
    d_med_age[k]['нехарактерные значения']=neharacter_znachenia
#добавим в словарь словарей под ключом конкретного источника дохода и
под ключом "нехарактерные значения" их список
    return d_med_age #функция возвращает созданный словарь словарей с
данными

```

```

###
### Выведем результаты функции по словарям 2010 и 2020 года
###

```

```

mediana_2010 = mediana_age(d_ev_2010)
mediana_2020 = mediana_age(d_ev_2020)

```

```

print(f'\nМедианный возраст по каждому источнику дохода в 2010 году:
{mediana_2010}')
print(f'\nМедианный возраст по каждому источнику дохода в 2020 году:
{mediana_2020}')

```

```

###
### Создадим функцию, которая будет считать математическое ожидание
возраста по каждому источнику
###

```

```

def matexpect(d_total, d_age): #определим функцию, которая на вход
будет принимать два словаря (с общими данными по основному источнику
дохода и по данным рахделенным по возрасту), а на выходе будет
возвращать математическое ожидание возраста по каждому источнику)
    d_possibilities={} #создадим пустой словарь для вероятностей по
каждому источнику дохода и возрасту (вида: ключ - источник дохода,
значение - словарь (вида: ключ - возраст (средний по диапазону),
значение - вероятность))
    d_matexpect={} #создадим пустой словарь для математических
ожиданий по источникам дохода (вида: ключ - источник дохода, значение
- мат. ожидание)
    for key in d_age: #переберем источники дохода
        d_possibilities[key]={} #сформируем пустой словарь внутри
словаря по каждому источнику дохода
        vsego = int(d_total[key]) #создадим переменную, хранящую
данные об общем количестве людей с перебираемым источником дохода

```

```

        for i in d_age[key]: #переберем в словаре возрастные диапазоны
            podhodyat = int(d_age[key][i]) #создадим переменную,
#хранящую данные о количестве людей заданного возраста с заданным
#источником дохода
            possibility = podhodyat/vsego #создадим переменную,
#хранящую вероятность того, что человек с заданным источником дохода
#будет относиться к заданному диапазону возраста
            ages = list(map(int, i.split(' - '))) #чтобы найти среднее
#значение диапазона (для дальнейших подсчетов), создадим список,
#разделенный тире, который будет задавать начало и конец диапазона
            age = sum(range(ages[0], ages[1]+1))/len(range(ages[0],
ages[1]+1)) #посчитаем среднее, используя последовательность по началу
#диапазона и концу (на единицу больше, так как в последовательности
#конец не включен). Мы суммируем числа последовательности и делим на
#длину (количество членов) последовательности
            d_possibilities[key][age] = possibility #добавляем в
#словарь словарей значение по заданному источнику дохода и среднему
#возрасту диапазона (значение - высчитанная выше вероятность)
        for el in d_possibilities: #перебираем источники дохода в словаре
#вероятностей
            s=0 #создаем переменную, равную нулю, чтобы прибавлять к ней
#последующие значения
            for k, v in d_possibilities[el].items(): #перебираем по
#переменным k, v словарь словаря вероятностей (по ключу и значению),
#где k-средний возраст диапазона, а v - вероятность
                s+=k*v #мы найдем произведение k, v и прибавим к
#переменной s, чтобы найти математическое ожидание возраста
#(вероятность * значение (возраст))
            d_matexpect[el]=round(s,2) #в словаре под ключом источника
#дохода вложим значение математического ожидания возраста человека с
#таким источником дохода
        return d_matexpect #функция вернет словарь таких мат. ожиданий

###
### Выведем результаты функции по словарям 2010 и 2020 года
###

matexp_2010 = matexpect(d_total_2010, d_ev_2010)
matexp_2020 = matexpect(d_total_2020, d_ev_2020)

print(f'\nМатематическое ожидание возраста по каждому источнику дохода
в 2010 году: {matexp_2010}')
print(f'\nМатематическое ожидание возраста по каждому источнику дохода
в 2020 году: {matexp_2020}')

###
### Создадим функцию, которая будет считать корреляцию Спирмена между

```


возрастом и количеством человек по каждому источнику дохода
###

```
def correlation_spearman(d_ages): #определим функцию, которая будет
высчитывать корреляцию Спирмена, так как этот показатель устойчив (на
вход словарь с источниками дохода и возрастами)
    d_final = {} #создаем пустой словарь для финальных результатов,
где ключ - источник дохода, значение - корреляция между возрастом и
количеством человек, у кого этот источник основной
    for k in d_ages: #перебираем ключи в исходном словаре (источники
дохода)
        lst_ages = [] #создаем пустой список, в котором будут
храниться возраста
        lst_values = [] #создаем пустой список, в котором будут
храниться данные о количестве человек с таким источником дохода в
качестве основного
        for el in d_ages[k]: #перебираем ключи в словаре словаря
(возраста)
            a = list(map(int, el.split(' - '))) #так как возраст
представлен в виде диапазона, разделим начало и конец диапазона, чтобы
найти среднее
            ele=(sum(range(a[0], a[1]+1)))/len(a) #ищем среднее
диапазона
            lst_ages.append(ele) #добавляем средний по диапазону
возраст в созданный выше список возрастов
            lst_values.append(d_ages[k][el]) #добавляем количество
человек с таким источником дохода в созданный выше соответствующий
список

            lst_values_var = sorted(list(map(int, lst_values)))
#отсортируем список с количеством людей, чтобы создать вариационный
ряд и считать ранги (по формуле)
            d_spearman = 0 #создадим переменную меры согласования рангов и
приравняем ее к нулю
            for i in range(len(lst_ages)): #переберем последовательность
по длине одного из созданных списков (допустим возрастов)
                d_spearman += (i-lst_values_var.index(lst_values[i]))**2
#по формуле находим сумму квадратов разности рангов двух критерием
(возраста и количества), причем так как возраст идет в порядке
возрастания изначально. оставим перебираемую переменную в виде его
ранга, а для количества людей будем искать соответствующий индекс в
вариационном ряду
            koefficient = 1 - ((6*d_spearman)/((len(lst_ages)**3)-
len(lst_ages))) #посчитаем по формуле коэффициент корреляции Спирмена
            d_final[k]=round(koefficient, 3) #создадим в финальном словаре
под ключом конкретного источника дохода (основного) значение -
корреляцию для этого источника между возрастом и количеством человек
    return d_final #функция вернет финальный словарь
```

```

###
### Выведем результаты функции по словарям 2010 и 2020 года
###

correl_2010 = correlation_spearman(d_ev_2010)
correl_2020 = correlation_spearman(d_ev_2020)

print(f'\nКорреляция возраста и количества человек по каждому
источнику дохода в 2010 году: {correl_2010}')
print(f'\nКорреляция возраста и количества человек по каждому
источнику дохода в 2020 году: {correl_2020}')

###
### Создадим функцию, которая будет считать самый популярный и
непопулярный основной источник дохода
###

def max_min(d_total): #определим функцию, которая будет считать самый
популярный и непопулярный основной источник дохода (по максимальному и
минимальному количеству человек с таким источником)
    maximum = sorted(d_total.items(), key = lambda x: x[1])[-1]
#отсортирует словарь в порядке возрастания по значению и возьмем
последний элемент
    minimum = sorted(d_total.items(), key = lambda x: x[1])[0]
#отсортирует словарь в порядке возрастания по значению и возьмем
первый элемент
    d={} #создадим пустой словарь для полученных далее данных
    d['Самый популярный источник'] = maximum #в созданном словаре под
ключом "максимум" вложим его значение, полученное выше
    d['Самый непопулярный источник'] = minimum #в созданном словаре
под ключом "минимум" вложим его значение, полученное выше
    return d #функция вернет созданный словарь

###
### Выведем результаты функции по словарям 2010 и 2020 года
###

maxmin_2010 = max_min(d_total_2010)
maxmin_2020 = max_min(d_total_2020)

print(f'\nСамый популярный и непопулярный источник дохода в 2010 году:
{maxmin_2010}')
print(f'\nСамый популярный и непопулярный источник дохода в 2020 году:
{maxmin_2020}')

```



```

###
### Сделаем ранжирование и составим рейтинг ТОП-3 самых популярных
основных источников дохода по каждому году и запишем их в отдельный
файл
###

rat2010=sorted(d_total_2010, key = d_total_2010.get, reverse = True)
[:3] #отсортируем общий словарь за 2010 год по значению по убыванию и
возьмем первые три элемента
rat2020=sorted(d_total_2020, key = d_total_2020.get, reverse = True)
[:3] #отсортируем общий словарь за 2020 год по значению по убыванию и
возьмем первые три элемента

with open('rating.csv', mode = 'w', encoding = 'utf-8') as outf:
#создадим новый файл и запишем в него полученный ТОП-3 рейтинг по
каждому году
    print(f'\nРейтинг ТОП-3 самых популярных основных источников
дохода в 2010 году.; Рейтинг ТОП-3 самых популярных основных
источников дохода в 2020 году:', file = outf) #не забываем указывать
путь к новому файлу, в который записываем
    for i in range(len(rat2010)):
        print(f'{rat2010[i]};{rat2020[i]}', file = outf) #не забываем
указывать путь к новому файлу, в который записываем

```

Словарь по общему количеству человек по каждому источнику дохода в 2010 году: {'трудовая деятельность, включая работу по совместительству': {'0 – 14': 855, '15 – 19': 875826, '20 – 29': 15843017, '30 – 49': 31913201, '50 – 54': 8028027, '55 – 59': 4328124, '60 – 64': 1375935, '65 – 100': 483588}, 'личное подсобное хозяйство': {'0 – 14': 1310, '15 – 19': 47161, '20 – 29': 369698, '30 – 49': 1106952, '50 – 54': 357352, '55 – 59': 156271, '60 – 64': 25686, '65 – 100': 34270}, 'пенсии, пособия и другие выплаты от организаций, государства': {'0 – 14': 1119136, '15 – 19': 443632, '20 – 29': 1357230, '30 – 49': 2489461, '50 – 54': 1775900, '55 – 59': 4748314, '60 – 64': 5902529, '65 – 100': 16822882}, 'стипендия': {'0 – 14': 651, '15 – 19': 308840, '20 – 29': 268928, '30 – 49': 4200, '50 – 54': 112, '55 – 59': 29, '60 – 64': 0, '65 – 100': 0}, 'сбережения, дивиденды, проценты': {'0 – 14': 1139, '15 – 19': 4021, '20 – 29': 60081, '30 – 49': 156710, '50 – 54': 41516, '55 – 59': 18258, '60 – 64': 2424, '65 – 100': 3148}, 'сдача внаем или в аренду имущества, доход от патентов, авторских прав': {'0 – 14': 1000, '15 – 19': 1381, '20 – 29': 9507, '30 – 49': 29558, '50 – 54': 8815, '55 – 59': 4956, '60 – 64': 2543, '65 – 100': 4964}, 'иждивение, помощь других лиц, алименты': {'0 – 14': 19727484, '15 – 19': 6332577, '20 – 29': 5148496, '30 – 49': 3586506, '50 – 54': 784400, '55 – 59': 249080, '60 – 64': 33437, '65 – 100': 92730}, 'иной источник средств к существованию': {'0 – 14': 2261, '15 – 19': 2297, '20 – 29': 13331, '30 – 49': 46657, '50 – 54': 13760, '55 – 59': 8537, '60 – 64': 2201, '65 – 100': 1126}}

Словарь по общему количеству человек по возрастам по каждому источнику дохода в 2010 году: {'трудовая деятельность, включая работу по совместительству': 62849588, 'личное подсобное хозяйство': 2098712, 'пенсии, пособия и другие выплаты от организаций, государства': 34659426, 'стипендия': 582905, 'сбережения, дивиденды, проценты': 287306, 'сдача внаем или в аренду имущества, доход от патентов, авторских прав': 62726, 'иждивение, помощь других лиц, алименты': 35955010, 'иной источник средств к существованию': 90223}

Словарь по общему количеству человек по каждому источнику дохода в 2020 году: {'трудовая деятельность, включая работу по совместительству': {'0 – 14': 743, '15 – 19': 643182, '20 – 29': 9382806, '30 – 49': 34479001, '50 – 54': 6555534, '55 – 59': 5359252, '60 – 64': 3156160, '65 – 100': 1892779}, 'личное подсобное хозяйство': {'0 – 14': 20026, '15 – 19': 59166, '20 – 29': 203487, '30 – 49': 755165, '50 – 54': 213432, '55 – 59': 225377, '60 – 64': 215876, '65 – 100': 326877}, 'пенсии, пособия и другие выплаты от организаций, государства': {'0 – 14': 2746856, '15 – 19': 631746, '20 – 29': 1168848, '30 – 49': 3542161, '50 – 54': 1332831, '55 – 59': 3889138, '60 – 64': 7808953, '65 – 100': 21616235}, 'стипендия': {'0 – 14': 397, '15 – 19': 1240984, '20 – 29': 775937, '30 – 49': 38991, '50 – 54': 2414, '55 – 59': 907, '60 – 64': 359, '65 – 100': 224}, 'сбережения, дивиденды, проценты': {'0 – 14': 14396, '15 – 19': 11421, '20 – 29': 64838, '30 – 49': 247292, '50 – 54': 53185, '55 – 59': 50682, '60 – 64': 45672, '65 – 100': 79635}, 'сдача внаем или в аренду имущества, доход от патентов, авторских прав': {'0 – 14': 15798, '15 – 19': 8722, '20 – 29': 40968, '30 – 49': 150943, '50 – 54': 36473, '55 – 59': 35295, '60 – 64': 31083, '65 – 100': 57897}, 'иждивение, помощь других лиц, алименты': {'0 – 14': 20215306, '15 – 19': 5052381, '20 – 29': 2761779, '30 – 49': 3396882, '50 – 54': 480836, '55 – 59': 317284, '60 – 64': 127856, '65 – 100': 121099}, 'иной источник средств к существованию': {'0 – 14': 1929, '15 – 19': 3311, '20 – 29': 6998, '30 – 49': 24364, '50 – 54': 5421, '55 – 59': 4185, '60 – 64': 1683, '65 – 100': 1273}}

Словарь по общему количеству человек по возрастам по каждому источнику дохода в 2020 году: {'трудовая деятельность, включая работу по совместительству': 61469457, 'личное подсобное хозяйство': 2019406, 'пенсии, пособия и другие выплаты от организаций, государства': 42736768, 'стипендия': 2060213, 'сбережения, дивиденды, проценты': 567121, 'сдача внаем или в аренду имущества, доход от патентов, авторских прав': 377179, 'иждивение, помощь других лиц, алименты': 32473423, 'иной источник средств к существованию': 49164}

Среднее значение количества человек по каждому возрасту и источнику дохода в 2010 году: {'трудовая деятельность, включая работу по совместительству': 7856072, 'личное подсобное хозяйство': 262338, 'пенсии, пособия и другие выплаты от организаций, государства': 4332386, 'стипендия': 72845, 'сбережения, дивиденды, проценты': 35912,

'сдача внаем или в аренду имущества, доход от патентов, авторских прав': 7840, 'иждивение, помощь других лиц, алименты': 4494339, 'иной источник средств к существованию': 11271}

Среднее значение количества человек по каждому возрасту и источнику дохода в 2020 году: {'трудовая деятельность, включая работу по совместительству': 7683682, 'личное подсобное хозяйство': 252426, 'пенсии, пособия и другие выплаты от организаций, государства': 5342096, 'стипендия': 257527, 'сбережения, дивиденды, проценты': 70890, 'сдача внаем или в аренду имущества, доход от патентов, авторских прав': 47147, 'иждивение, помощь других лиц, алименты': 4059178, 'иной источник средств к существованию': 6146}

Медианный возраст по каждому источнику дохода в 2010 году: {'трудовая деятельность, включая работу по совместительству': {'медиана': 45.75, 'нижняя квартиль': 20.75, 'верхняя квартиль': 59.5, 'нижняя граница': -37.375, 'верхняя граница': 117.625, 'нехарактерные значения': []}, 'личное подсобное хозяйство': {'медиана': 45.75, 'нижняя квартиль': 20.75, 'верхняя квартиль': 59.5, 'нижняя граница': -37.375, 'верхняя граница': 117.625, 'нехарактерные значения': []}, 'пенсии, пособия и другие выплаты от организаций, государства': {'медиана': 45.75, 'нижняя квартиль': 20.75, 'верхняя квартиль': 59.5, 'нижняя граница': -37.375, 'верхняя граница': 117.625, 'нехарактерные значения': []}, 'стипендия': {'медиана': 45.75, 'нижняя квартиль': 20.75, 'верхняя квартиль': 59.5, 'нижняя граница': -37.375, 'верхняя граница': 117.625, 'нехарактерные значения': []}, 'сбережения, дивиденды, проценты': {'медиана': 45.75, 'нижняя квартиль': 20.75, 'верхняя квартиль': 59.5, 'нижняя граница': -37.375, 'верхняя граница': 117.625, 'нехарактерные значения': []}, 'сдача внаем или в аренду имущества, доход от патентов, авторских прав': {'медиана': 45.75, 'нижняя квартиль': 20.75, 'верхняя квартиль': 59.5, 'нижняя граница': -37.375, 'верхняя граница': 117.625, 'нехарактерные значения': []}, 'иждивение, помощь других лиц, алименты': {'медиана': 45.75, 'нижняя квартиль': 20.75, 'верхняя квартиль': 59.5, 'нижняя граница': -37.375, 'верхняя граница': 117.625, 'нехарактерные значения': []}, 'иной источник средств к существованию': {'медиана': 45.75, 'нижняя квартиль': 20.75, 'верхняя квартиль': 59.5, 'нижняя граница': -37.375, 'верхняя граница': 117.625, 'нехарактерные значения': []}}

Медианный возраст по каждому источнику дохода в 2020 году: {'трудовая деятельность, включая работу по совместительству': {'медиана': 45.75, 'нижняя квартиль': 20.75, 'верхняя квартиль': 59.5, 'нижняя граница': -37.375, 'верхняя граница': 117.625, 'нехарактерные значения': []}, 'личное подсобное хозяйство': {'медиана': 45.75, 'нижняя квартиль': 20.75, 'верхняя квартиль': 59.5, 'нижняя граница': -37.375, 'верхняя граница': 117.625, 'нехарактерные значения': []}, 'пенсии, пособия и другие выплаты от организаций, государства': {'медиана': 45.75, 'нижняя квартиль': 20.75, 'верхняя квартиль': 59.5, 'нижняя граница': -37.375, 'верхняя граница': 117.625, 'нехарактерные значения': []}, 'стипендия': {'медиана': 45.75, 'нижняя квартиль': 20.75, 'верхняя

квартиль': 59.5, 'нижняя граница': -37.375, 'верхняя граница': 117.625, 'нехарактерные значения': []}, 'сбережения, дивиденды, проценты': {'медиана': 45.75, 'нижняя квартиль': 20.75, 'верхняя квартиль': 59.5, 'нижняя граница': -37.375, 'верхняя граница': 117.625, 'нехарактерные значения': []}, 'сдача внаем или в аренду имущества, доход от патентов, авторских прав': {'медиана': 45.75, 'нижняя квартиль': 20.75, 'верхняя квартиль': 59.5, 'нижняя граница': -37.375, 'верхняя граница': 117.625, 'нехарактерные значения': []}, 'иждивение, помощь других лиц, алименты': {'медиана': 45.75, 'нижняя квартиль': 20.75, 'верхняя квартиль': 59.5, 'нижняя граница': -37.375, 'верхняя граница': 117.625, 'нехарактерные значения': []}, 'иной источник средств к существованию': {'медиана': 45.75, 'нижняя квартиль': 20.75, 'верхняя квартиль': 59.5, 'нижняя граница': -37.375, 'верхняя граница': 117.625, 'нехарактерные значения': []}}

Математическое ожидание возраста по каждому источнику дохода в 2010 году: {'трудовая деятельность, включая работу по совместительству': 39.03, 'личное подсобное хозяйство': 40.74, 'пенсии, пособия и другие выплаты от организаций, государства': 65.32, 'стипендия': 20.62, 'сбережения, дивиденды, проценты': 39.5, 'сдача внаем или в аренду имущества, доход от патентов, авторских прав': 43.67, 'иждивение, помощь других лиц, алименты': 16.08, 'иной источник средств к существованию': 40.52}

Математическое ожидание возраста по каждому источнику дохода в 2020 году: {'трудовая деятельность, включая работу по совместительству': 42.31, 'личное подсобное хозяйство': 49.65, 'пенсии, пособия и другие выплаты от организаций, государства': 64.51, 'стипендия': 20.32, 'сбережения, дивиденды, проценты': 47.09, 'сдача внаем или в аренду имущества, доход от патентов, авторских прав': 47.29, 'иждивение, помощь других лиц, алименты': 15.1, 'иной источник средств к существованию': 39.33}

Корреляция возраста и количества человек по каждому источнику дохода в 2010 году: {'трудовая деятельность, включая работу по совместительству': 0.048, 'личное подсобное хозяйство': -0.048, 'пенсии, пособия и другие выплаты от организаций, государства': 0.952, 'стипендия': -0.988, 'сбережения, дивиденды, проценты': -0.048, 'сдача внаем или в аренду имущества, доход от патентов, авторских прав': 0.262, 'иждивение, помощь других лиц, алименты': -0.976, 'иной источник средств к существованию': -0.333}

Корреляция возраста и количества человек по каждому источнику дохода в 2020 году: {'трудовая деятельность, включая работу по совместительству': 0.19, 'личное подсобное хозяйство': 0.738, 'пенсии, пособия и другие выплаты от организаций, государства': 0.81, 'стипендия': -0.643, 'сбережения, дивиденды, проценты': 0.429, 'сдача внаем или в аренду имущества, доход от патентов, авторских прав': 0.429, 'иждивение, помощь других лиц, алименты': -0.976, 'иной источник средств к существованию': -0.381}

Самый популярный и непопулярный источник дохода в 2010 году: {'Самый популярный источник': ('трудовая деятельность, включая работу по совместительству', 62849588), 'Самый непопулярный источник': ('сдача внаем или в аренду имущества, доход от патентов, авторских прав', 62726)}

Самый популярный и непопулярный источник дохода в 2020 году: {'Самый популярный источник': ('трудовая деятельность, включая работу по совместительству', 61469457), 'Самый непопулярный источник': ('иной источник средств к существованию', 49164)}