

Загружу и очищу данные от людей без зарплаты и с кодами затрудняюсь, отказ, нет ответа

```
In [1]: import matplotlib.pyplot as plt
import pandas as pd

i = pd.read_excel("RLMS_ind.xlsx")

d = i.copy()
d = d[~d.wage.isna()]
d = d[d.wage < 99999997]
```

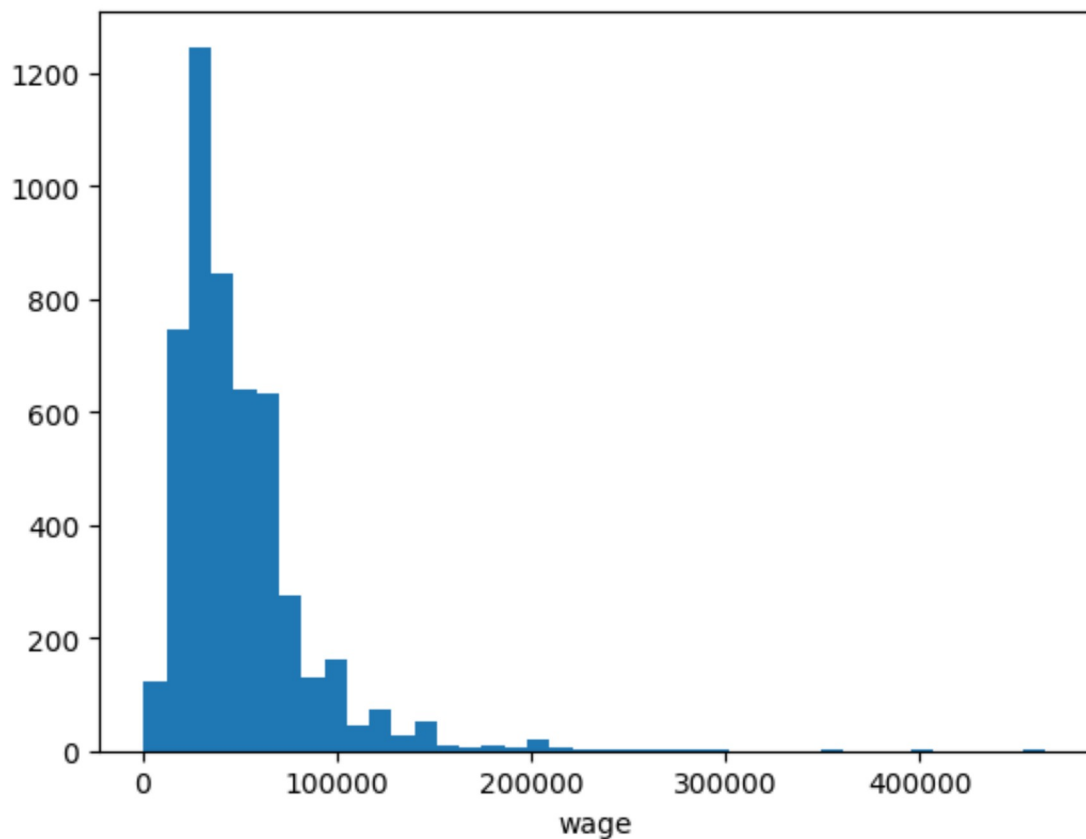
Средняя зп и гистограмма

```
In [2]: m = d.wage.mean()
m
```

```
Out[2]: np.float64(49614.76107128791)
```

```
In [3]: plt.figure()
plt.hist(d.wage, bins=40)
plt.xlabel("wage")
plt
```

```
Out[3]: <module 'matplotlib.pyplot' from 'C:\\Users\\dan\\PycharmProjects\\dano\\.venv\\Lib\\site-packages\\matplotlib\\pyplot.py'>
```



Средняя заработная плата в выборке равна 49615 рублей. Распределение правосторонне с длинным хвостом, есть несколько очень высоких наблюдений

Теперь, посчитаю зарплату по уровню образования. Маппинг взял из pdf-ки

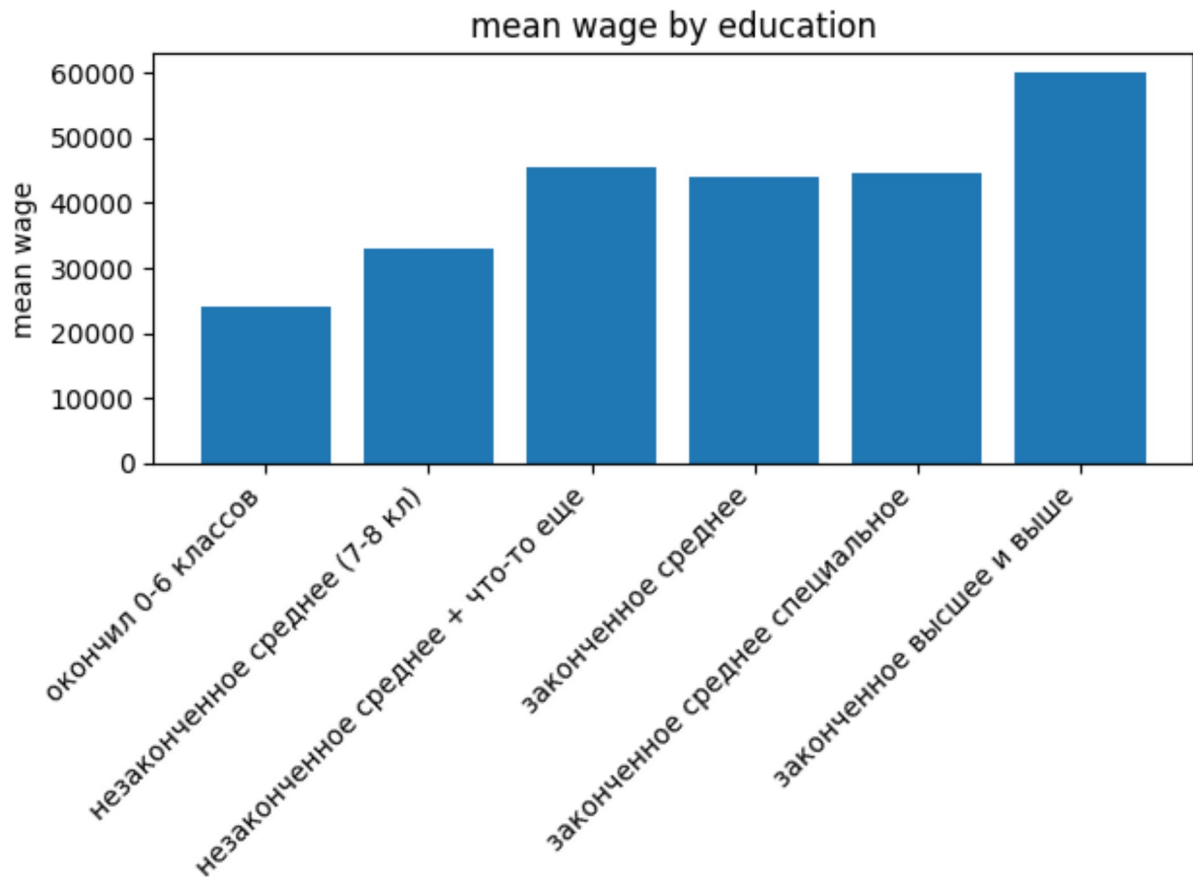
```
In [4]: e = {
        1:"окончил 0-6 классов",
        2:"незаконченное среднее (7-8 кл)",
        3:"незаконченное среднее + что-то еще",
        4:"законченное среднее",
        5:"законченное среднее специальное",
        6:"законченное высшее и выше"
    }
    d["e"] = d.educ.map(e)

    t = d.dropna(subset=["e"]).groupby("e")["wage"].mean().reindex(e.values())
    t
```

```
Out[4]: e
окончил 0-6 классов                24214.285714
незаконченное среднее (7-8 кл)     32908.000000
незаконченное среднее + что-то еще  45497.214689
законченное среднее                43912.074616
законченное среднее специальное    44748.008333
законченное высшее и выше         59949.226568
Name: wage, dtype: float64
```

Визуализирую

```
In [5]: plt.figure()
        plt.bar(t.index, t.values)
        plt.xticks(rotation=45, ha="right")
        plt.ylabel("mean wage")
        plt.title("mean wage by education")
        plt.tight_layout()
        plt.show()
```



Чем выше образование, тем выше средняя заработная плата. График это отлично иллюстрирует

Объединю файл домохозяйств с файлом индивидиуалов

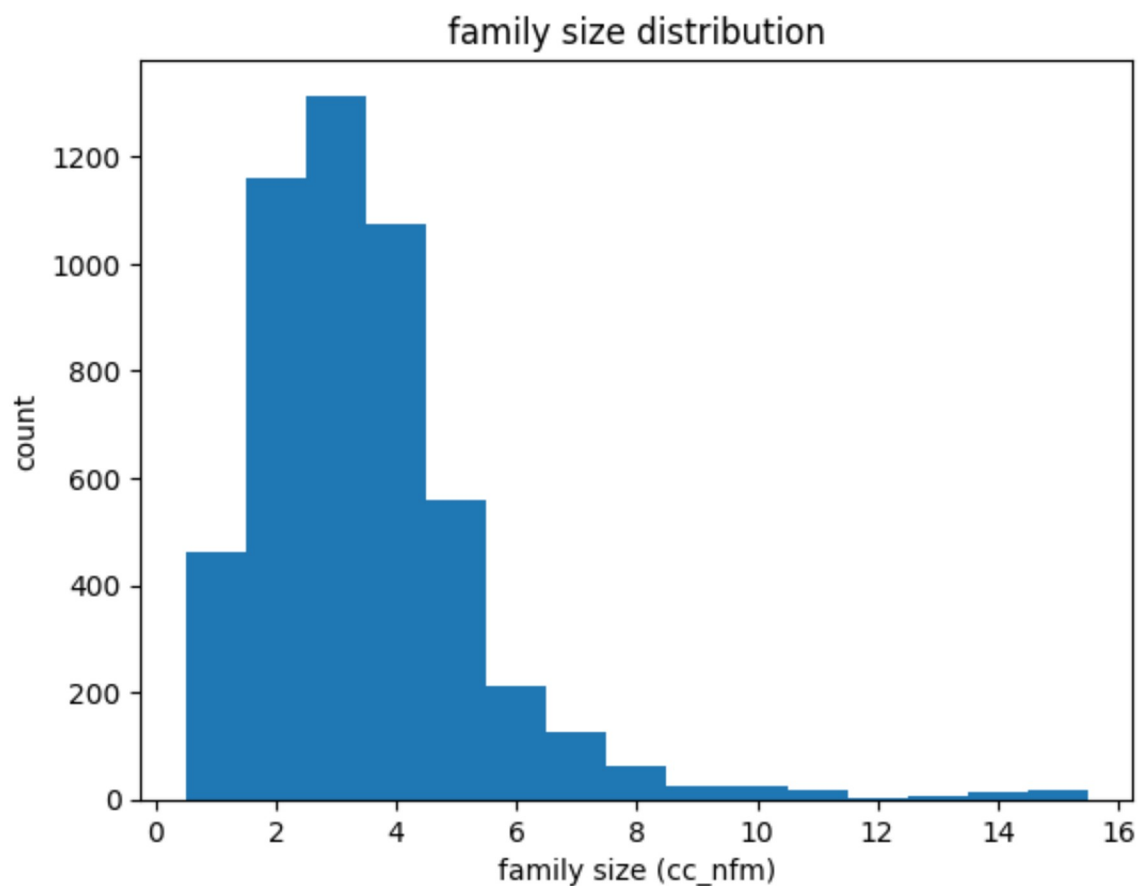
```
In [6]: h = pd.read_excel("RLMS_HH.xlsx")
hs = h[["ccid_h", "cc_nfm"]]
d2 = d.merge(hs, on="ccid_h", how="left")
```

средний размер семьи и распространение

```
In [7]: s = d2.cc_nfm.mean()
s
```

```
Out[7]: np.float64(3.4824734147302085)
```

```
In [8]: plt.figure()
plt.hist(d2.cc_nfm.dropna(), bins=np.arange(1, d2.cc_nfm.max()+2)-0.5)
plt.xlabel("family size (cc_nfm)")
plt.ylabel("count")
plt.title("family size distribution")
plt.show()
```

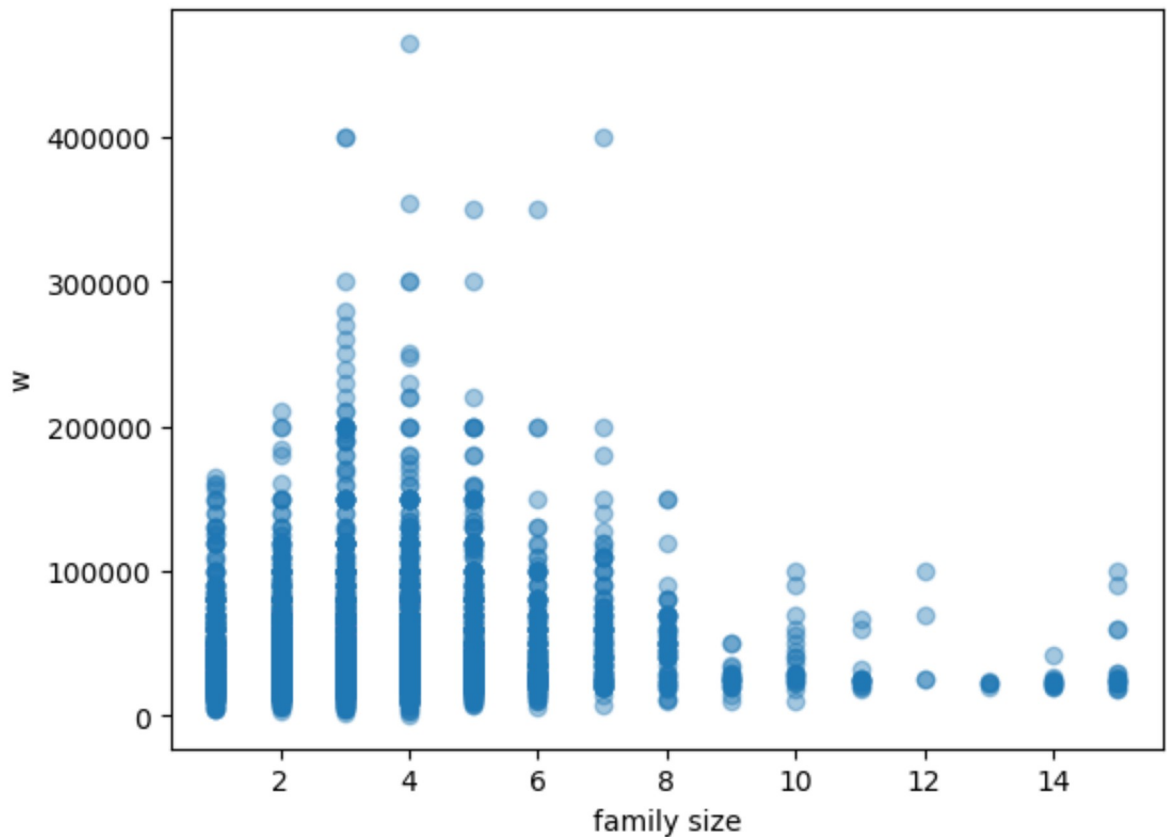


Средний размер семьи в выборке равен 3.48 чел. В основном встречаются небольшие семьи ≤ 4 чел

Связь размера семьи и ЗП, используя корреляцию Пирсона и p value

```
In [9]: x = d2.cc_nfm
y = d2.wage
plt.figure()
plt.scatter(x, y, alpha=0.4)
plt.xlabel("family size")
plt.ylabel("w")
```

```
Out[9]: Text(0, 0.5, 'w')
```



Пик дохода наблюдается на размере семьи в 3 чел. Корреляция есть, она стат значима, но очень слаба

Разделю на мужчин, женщин. Рассмотрю 40+

```
In [10]: for g in [1,2]:
          q = d2[d2.gender==g]
          r, p = st.pearsonr(q.cc_nfm.dropna(), q.wage.loc[q.cc_nfm.dropna().index])
          print(g, r, p)

          q40 = d2[d2.age>=40]
          r40, p40 = st.pearsonr(q40.cc_nfm.dropna(), q40.wage.loc[q40.cc_nfm.dropna().index])
          r40, p40
```

```
1 0.03201804210424221 0.11623926402176096
2 -0.04214518344772215 0.029430203998616036
```

```
Out[10]: (np.float64(0.046257828357301536), np.float64(0.00980841743121213))
```

связь незначима

Самый богатый индивид

```
In [15]: g = d.loc[d.wage == d.wage.max()].iloc[0]
h0 = int(g.ccid_h)
m0 = int(g.cch4)
n = int(h.loc[h.ccid_h == h0, "cc_nfm"].iloc[0])

g.wage, h0, m0, n
```

```
Out[15]: (np.float64(465000.0), 13811087, 2, 4)
```

У самого богатого человека зарплата 465000

Впишу данные из таблицы родственников

```
In [16]: gd = {1: "мужчина", 2: "женщина"}

rd = {
    1: "муж/жена",
    2: "отец/мать",
    3: "отчим/мачеха",
    4: "сын/дочь",
    5: "неродной сын/неродная дочь",
    6: "родной брат/родная сестра",
    7: "сводный брат/сводная сестра",
    8: "дедушка/бабушка",
    9: "внук/внучка",
    10: "племянник/племянница",
    11: "тесть/теща/свекор/свекровь",
    12: "зять/невестка",
    13: "другие родственники",
    14: "другие лица, не являющиеся родственниками",
    15: "дядя/тётя",
    16: "двоюродный брат/двоюродная сестра",
    17: "прадедушка/прабабушка",
    18: "правнук/правнучка",
}

ri = {
    1: 1,
    2: 4,
    3: 5,
    4: 2,
    5: 3,
    6: 6,
    7: 7,
    8: 9,
    9: 8,
    10: 15,
    11: 12,
    12: 11,
    13: 13,
    14: 14,
    15: 10,
    16: 16,
    17: 18,
    18: 17,
```

```
}
```

Теперь, найду всех, кто живет с героем

```
In [17]: year = 2024
r = h.loc[h.ccid_h == h0].iloc[0]

lst = []

for j in range(1, n + 1):
    g_code = r[f"ccb{j}.4"]
    yb = r[f"ccb{j}.5"]
    if not np.isnan(yb):
        a = year - int(yb)
    else:
        a = np.nan

    if j == m0:
        rel = "наш герой"
    else:
        big = max(j, m0)
        small = min(j, m0)
        col = f"ccb{big}.9.{small}"
        if (col in r.index) and (not np.isnan(r[col])):
            c = int(r[col])
            if big == j:
                c2 = c
            else:
                c2 = ri.get(c, c)
            rel = rd.get(c2, f"код {c2}")
        else:
            rel = "нет данных"

    lst.append(
        (
            j,
            gd.get(int(g_code), f"код {g_code}"),
            int(yb) if not np.isnan(yb) else np.nan,
            a,
            rel,
        )
    )

t = pd.DataFrame(lst, columns=["номер", "пол", "год_рожд", "возраст", "кем_приходит
t
```

```
Out[17]:
```

	номер	пол	год_рожд	возраст	кем_приходится
0	1	женщина	1989	35	муж/жена
1	2	мужчина	1989	35	наш герой
2	3	женщина	1956	68	тесть/теща/свекор/свекровь
3	4	женщина	2014	10	сын/дочь

У него есть жена 1989 г.р., дочь 2010 г.р. и теща 1956 г.р.

In []: