

ГУАП

КАФЕДРА № 41

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

ассистент

должность, уч. степень, звание

подпись, дата

В.В. Боженко

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ

Предварительный анализ данных

по курсу: Введение в анализ данных

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

4917

подпись, дата

Е.А. Ясиновский

инициалы, фамилия

Санкт-Петербург 2022

Цель работы: осуществить предварительную обработку данных csv файла, выявить и устранить проблемы в данных, построить сводные таблицы по предоставленным данным

Вариант 2: Файл salary.csv, в котором предоставлены данные о заработной плате разработчиков и инженеров IT сферы разделенные по годам, опыту работы, типу занятости (полная\неполная), валюте, размеру компании и стране компании.

```
import pandas as pd
import numpy as np
df = pd.read_csv("2salary.csv")
```

Рисунок 1 — Загрузил данные с помощью pandas

df.head(20)

✓ 0.7s

Python

Unnamed: 0	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ratio	company_location	company_size	
0	0	2020	MI	FT	Data Scientist	70000.0	EUR	79833	DE	0	DE	L
1	1	2020	SE	FT	Machine Learning Scientist	260000.0	USD	260000	JP	0	JP	S
2	2	2020	SE	FT	Big Data Engineer	85000.0	GBP	109024	GB	50	GB	M
3	3	2020	MI	FT	Product Data Analyst	20000.0	USD	20000	HN	0	HN	S
4	4	2020	SE	FT	Machine Learning Engineer	150000.0	USD	150000	US	50	US	L
5	5	2020	EN	FT	Data Analyst	72000.0	USD	72000	US	100	US	L
6	6	2020	SE	FT	Lead Data Scientist	190000.0	USD	190000	US	100	US	S
7	7	2020	MI	FT	Data Scientist	11000000.0	HUF	35735	HU	50	HU	L
8	8	2020	MI	FT	Business Data Analyst	135000.0	USD	135000	US	100	US	L
9	9	2020	SE	FT	Lead Data Engineer	125000.0	USD	125000	NZ	50	NZ	S
10	10	2020	EN	FT	Data Scientist	45000.0	EUR	51321	FR	0	FR	S
11	11	2020	MI	FT	Data Scientist	3000000.0	INR	40481	IN	0	IN	L
12	12	2020	EN	FT	Data Scientist	35000.0	EUR	39916	FR	0	FR	M
13	13	2020	MI	FT	Lead Data Analyst	87000.0	USD	87000	US	100	US	L
14	14	2020	MI	FT	Data Analyst	85000.0	USD	85000	US	100	US	L
15	15	2020	MI	FT	Data Analyst	8000.0	USD	8000	PK	50	PK	L
16	16	2020	EN	FT	Data Engineer	4450000.0	JPY	41689	JP	100	JP	S
17	17	2020	SE	FT	Big Data Engineer	100000.0	EUR	114047	PL	100	GB	S
18	18	2020	EN	FT	Data Science Consultant	423000.0	INR	5707	IN	50	IN	M
19	19	2020	MI	FT	Lead Data Engineer	56000.0	USD	56000	PT	100	US	M

Рисунок 2 — Вывел первые 20 строк



Рисунок 3 — Выполнил обзор данных с помощью MD

```
df.info()
✓ 0.1s

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 610 entries, 0 to 609
Data columns (total 12 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Unnamed: 0            610 non-null   int64
1   work_year             610 non-null   int64
2   experience_level      610 non-null   object
3   employment_type       610 non-null   object
4   job_title             610 non-null   object
5   salary                607 non-null   float64
6   salary_currency       609 non-null   object
7   salary_in_usd         610 non-null   object
8   employee_residence    610 non-null   object
9   remote_ratio          610 non-null   int64
10  company_location      610 non-null   object
11  company_size          610 non-null   object
dtypes: float64(1), int64(3), object(8)
memory usage: 57.3+ KB
```

Рисунок 4 — Выполнение оценки данных с помощью метода info()

Исходя из полученного выше вывода можно увидеть, что всего у нас есть 610 строк данных, но сами данные корректны только в 607 строках, так как это минимальное число строк, где данные не равны null. В добавок можно заметить, что строковые значения здесь привелись к типу object, а у первого столбца (номера строки) отсутствует имя

```
df.columns
```

✓ 0.9s

```
Index(['Unnamed: 0', 'work_year', 'experience_level', 'employment_type',  
      'job_title', 'salary', 'salary_currency', 'salary_in_usd',  
      'employee_residence', 'remote_ratio', 'company_location',  
      'company_size'],  
      dtype='object')
```

Исправим название у первого столбца

```
df = df.rename(columns={df.columns[0]: "row_number"})  
df.columns
```

✓ 0.9s

```
Index(['row_number', 'work_year', 'experience_level', 'employment_type',  
      'job_title', 'salary', 'salary_currency', 'salary_in_usd',  
      'employee_residence', 'remote_ratio', 'company_location',  
      'company_size'],  
      dtype='object')
```

Рисунок 5 — Выявил проблемный столбец и исправил ошибку

Пропуски данных

Выше мы уже выяснили, что у нас отсутствуют 3 значения в столбце salary и одного значение в столбце salary_currency. Теперь мы можем более точно проверить пропуски данных с помощью метода `isna()`.

```
print(df.isna().sum())
```

✓ 0.7s

Python

```
row_number      0
work_year       0
experience_level 0
employment_type 0
job_title       0
salary          3
salary_currency  1
salary_in_usd   0
employee_residence 0
remote_ratio    0
company_location 0
company_size    0
dtype: int64
```

Данные о пропусках совпадают, теперь устраним пропуски с помощью `dropna()`.

```
df = df.dropna()
```

✓ 0.8s

Python

Рисунок 6 — Выявил пропуски в данных и устранил их

Поиск полных дубликатов в файле

Поиск дубликатов выполняется с помощью метода `duplicated()`

```
duplicates = df.duplicated().where(lambda x: x == True).dropna()
duplicates
```

✓ 0.6s

Python

```
607    True
608    True
609    True
dtype: object
```

Из кода выше видно, что дублируются строки 607,608,609. Последние три строки, проверим их

```
df[-3:]
```

✓ 0.6s

Python

	row_number	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary
607	606	2022	MI	FT	AI Scientist	200000.0	USD	
608	606	2022	MI	FT	AI Scientist	200000.0	USD	
609	606	2022	MI	FT	AI Scientist	200000.0	USD	

Действительно, строки полностью идентичны, удалим дубликаты с помощью `drop_duplicates` и обновим индексацию

```
df = df.drop_duplicates().reset_index()
```

✓ 0.6s

Python

Рисунок 7 — Нашел и устранил явные дубликаты в данных

```
for i in df.columns[2:]:
    print(df[i].unique())
```

✓ 0.4s Python

Output exceeds the [size limit](#). Open the full output data [in a text editor](#)

```
[2020 2021 2022]
['MI' 'SE' 'EN' 'EX']
['FT' 'CT' 'PT' 'FL']
['Data Scientist' 'Machine Learning Scientist' 'Big Data Engineer'
 'Product Data Analyst' 'Machine Learning Engineer' 'Data Analyst'
 'Lead Data Scientist' 'Business Data Analyst' 'Lead Data Engineer'
 'Lead Data Analyst' 'Data Engineer' 'Data Science Consultant'
 'BI Data Analyst' 'Director of Data Science' 'Research Scientist'
 'Machine Learning Manager' 'Data Engineering Manager'
 'Machine Learning Infrastructure Engineer' 'ML Engineer' 'AI Scientist'
 'Computer Vision Engineer' 'Principal Data Scientist'
 'Data Science Manager' 'Head of Data' '3D Computer Vision Researcher'
 'Data Analytics Engineer' 'Applied Data Scientist'
 'Marketing Data Analyst' 'Cloud Data Engineer' 'Financial Data Analyst'
 'Computer Vision Software Engineer' 'Director of Data Engineering'
 'Data Science Engineer' 'Principal Data Engineer'
 'Machine Learning Developer' 'Applied Machine Learning Scientist'
 'Data Analytics Manager' 'Head of Data Science' 'Data Specialist'
 'Data Architect' 'Finance Data Analyst' 'Principal Data Analyst'
 'Big Data Architect' 'Staff Data Scientist' 'Analytics Engineer'
 'ETL Developer' 'Head of Machine Learning' 'NLP Engineer'
 'Lead Machine Learning Engineer' 'Data Analytics Lead' 'DataScientist'
 'Data AnalyticsManager']
[ 70000 260000  85000  20000 150000  72000 190000 11000000
 135000 125000  45000 3000000  35000  87000   8000 4450000
 ...
 'MX' 'CA' 'AT' 'NG' 'ES' 'PT' 'DK' 'IT' 'HR' 'LU' 'PL' 'SG' 'RO' 'IQ'
 'BR' 'BE' 'UA' 'IL' 'RU' 'MT' 'CL' 'IR' 'CO' 'MD' 'KE' 'SI' 'CH' 'VN'
 'AS' 'TR' 'CZ' 'DZ' 'EE' 'MY' 'AU' 'IE']
['L' 'S' 'M']
```

На первый взгляд из полученного анализа видно, что неявные дубликаты могут быть только в названиях должностей. Например, Lead Data Scientics и Lead Data Analyst выглядят как одна и та же должность, так же как и Data Science Engineer и Big Data Engineer, однако нельзя сказать, точно, что это одна и та же дожность, поэтому эти неявные дубликаты мы оставим нетронутыми.

Рисунок 8 — Проверил данные на неявные дубликаты

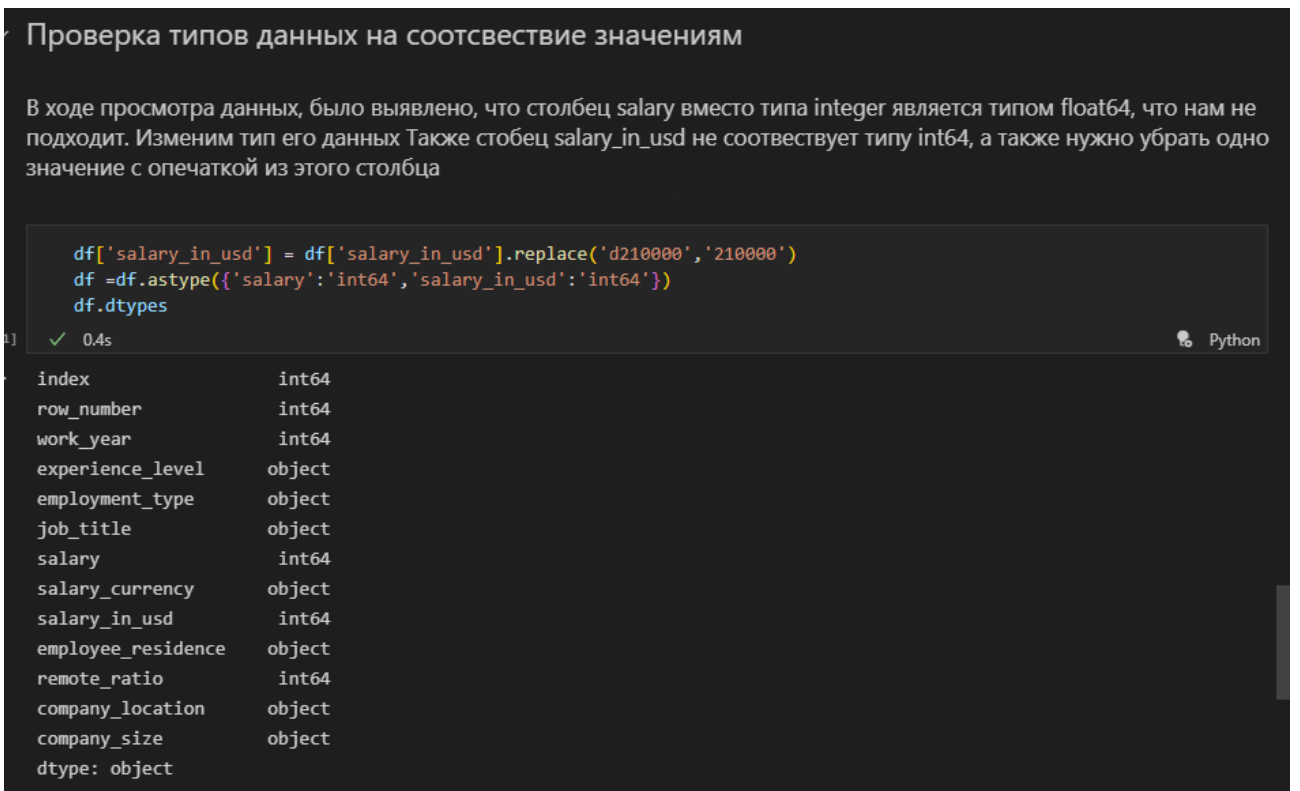


Рисунок 9 — Проверил типы данных на соответствие

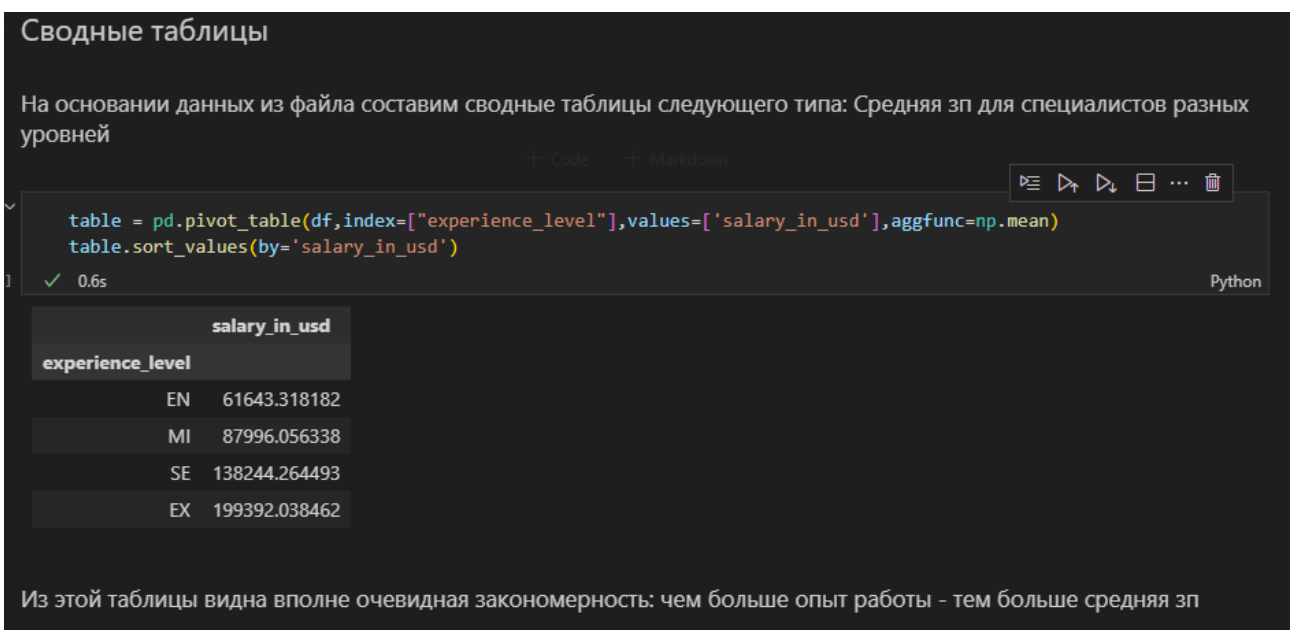


Рисунок 10 — Построил первую сводную таблицу

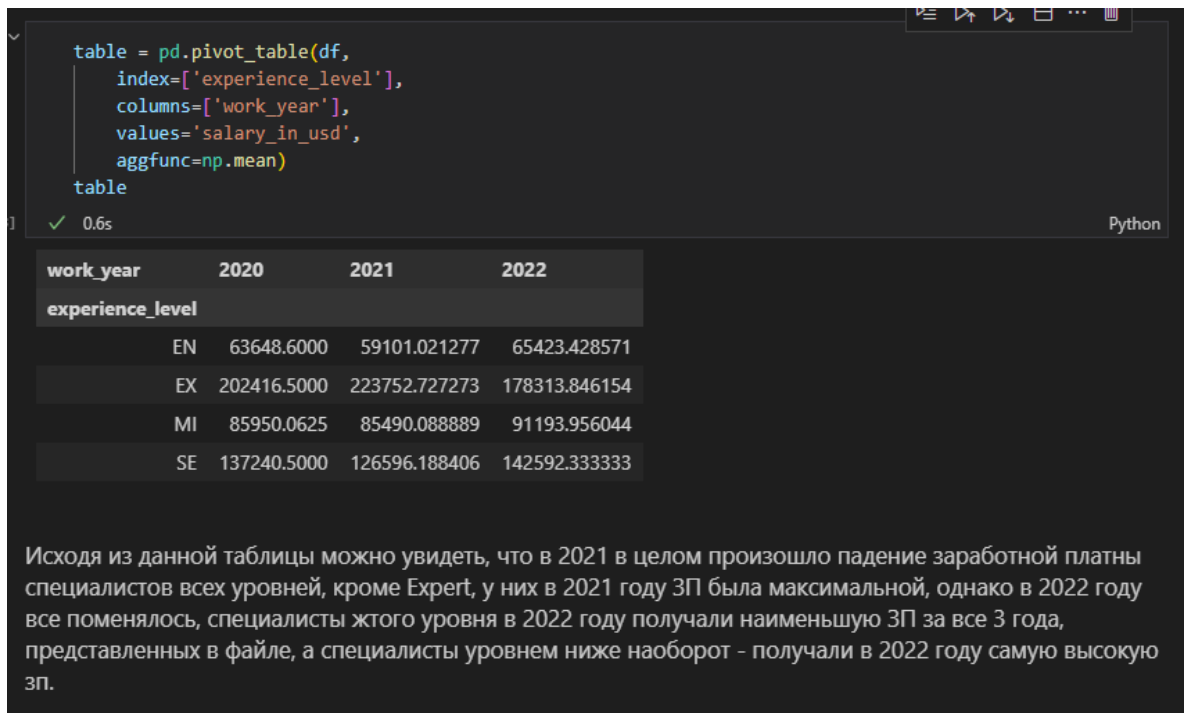


Рисунок 11 — Построил вторую сводную таблицу

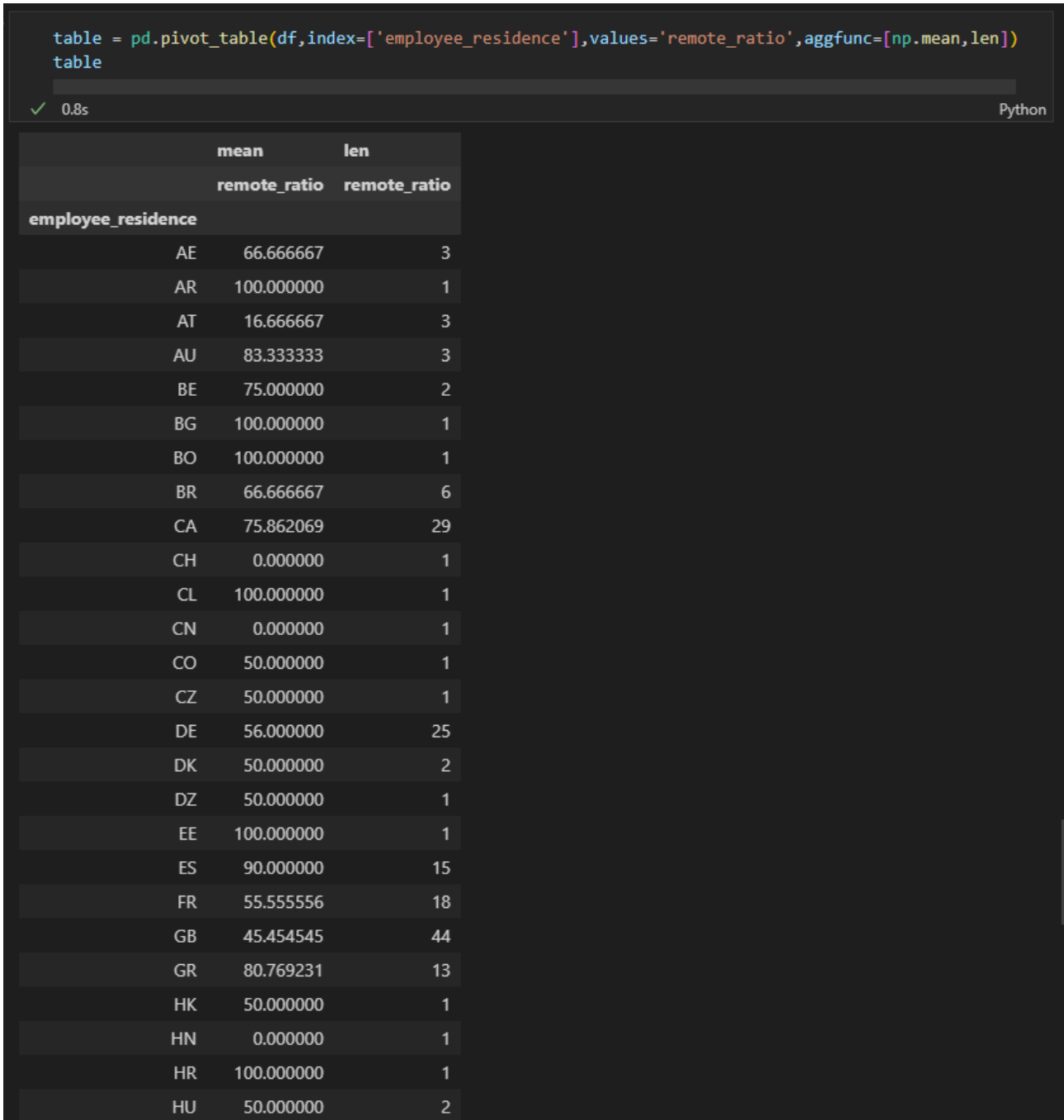


Рисунок 12 — Первая часть третьей сводной таблицы

IE	100.000000	1
IN	63.333333	30
IQ	50.000000	1
IR	100.000000	1
IT	50.000000	4
JE	0.000000	1
JP	50.000000	7
KE	100.000000	1
LU	100.000000	1
MD	0.000000	1
MT	50.000000	1
MX	0.000000	2
MY	100.000000	1
NG	100.000000	2
NL	90.000000	5
NZ	50.000000	1
PH	100.000000	1
PK	66.666667	6
PL	100.000000	4
PR	50.000000	1
PT	66.666667	6
RO	25.000000	2
RS	100.000000	1
RU	62.500000	4
SG	75.000000	2
SI	75.000000	2
TN	100.000000	1
TR	50.000000	3
UA	100.000000	1
US	76.981707	328
VN	66.666667	3

Рисунок 13 — Вторая часть третьей сводной таблицы

Так же был произведен анализ третьей таблицы, в ходе которого выяснилось количество работников в каждой стране, и то, как они предпочитают работать (удаленно, полуудаленно, только в офисе).

Вывод: В ходе выполнения лабораторной работы освоил методы предварительно анализа данных с помощью библиотеки pandas. Научился очищать исходные данные от невалидных строк и дубликатов, анализировать столбцы и типы исходных данных на соответствие действительности. Также научился выполнять простые группировки данных с помощью сводных таблиц.