

ГУАП

ЦИФРОВАЯ КАФЕДРА

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

канд. техн. наук, доцент

должность, уч. степень, звание

подпись, дата

В.В. Боженко

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №1

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)

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	1100000.0	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	300000.0	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

```
### Предметная область
Данные о заработной плате разработчиков и инженеров IT сферы разделенные по годам, опыту работы, типу занятости (полная\неполная), валюте, размеру компании и стране компании

#### Столбцы
1. Номер строки
2. Год выплаты ЗП
3. Опыт работы
4. Тип занятости
5. Наименование должности
6. ЗП выраженная в валюте выплаты
7. Валюта выплаты
8. Зп выраженная в долларах USD
9. Страна проживания сотрудника
10. Объем удаленной работы
11. Страна главного офиса компании
12. Размер компании

#### Оценка данных
Оценка данных производится с помощью метода .info() у объекта датафрейма
```

Рисунок 3 — Выполнил обзор данных в notebook с помощью Markdown

```
df.info()

<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

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

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 его

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())
```

```
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()
```

Рисунок 6 — Выявил пропуски в данных и удалил их

Поиск полных дубликатов в файле

Поиск дубликатов выполняется с помощью метода duplicated()

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

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:]
```

Python

row_number	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
607	606	2022	MI	FT AI Scientist	200000.0	USD	200000	IN	100	US	L
608	606	2022	MI	FT AI Scientist	200000.0	USD	200000	IN	100	US	L
609	606	2022	MI	FT AI Scientist	200000.0	USD	200000	IN	100	US	L

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

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

Python

Рисунок 7 — Выявил явные дубликаты и удалил их

После удаление явных дубликатов, поищем неявные, делая выборку уникальных значений для различных столбцов.

```

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

```

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']

```

Рисунок 8 — Поиск неявных дубликатов

На первый взгляд из полученного анализа видно, что неявные дубликаты могут быть только в названиях должностей. Например, Lead Data Scientists 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 должность, поэтому эти неявные дубликаты мы оставим нетронутыми.

```
df['salary_in_usd'] = df['salary_in_usd'].replace('d210000', '210000')
df = df.astype({'salary': 'int64', 'salary_in_usd': 'int64'})
df.dtypes
```

index	int64
row_number	int64
work_year	int64
experience_level	object
employment_type	object
job_title	object
salary	int64
salary_currency	object
salary_in_usd	int64
employee_residence	object
remote_ratio	int64
company_location	object
company_size	object
dtype:	object

Рисунок 9 — Исправил неверные типы данных в столбцах

После подготовки данных к анализу произвел небольшой анализ данных с помощью сводных таблиц.

```
table = pd.pivot_table(df, index=["experience_level"], values=['salary_in_usd'], aggfunc=np.mean)
table.sort_values(by='salary_in_usd')
```

salary_in_usd	
experience_level	
EN	61643.318182
MI	87996.056338
SE	138244.264493
EX	199392.038462

Рисунок 10 — Средняя ЗП для специалистов разных уровней опыта

Из этой таблицы видна вполне очевидная закономерность: чем больше опыт работы - тем больше средняя зп

```

table = pd.pivot_table(df,
    index=['experience_level'],
    columns=['work_year'],
    values='salary_in_usd',
    aggfunc=np.mean)
table

```

work_year	2020	2021	2022
experience_level			
EN	63648.6000	59101.021277	65423.428571
EX	202416.5000	223752.727273	178313.846154
MI	85950.0625	85490.088889	91193.956044
SE	137240.5000	126596.188406	142592.333333

Рисунок 11 — Средняя ЗП специалистов разного уровня по годам

Исходя из данной таблицы можно увидеть, что в 2021 в целом произошло падение заработной платы специалистов всех уровней, кроме Expert, у них в 2021 году ЗП была максимальной, однако в 2022 году все поменялось, специалисты этого уровня в 2022 году получали наименьшую ЗП за все 3 года, представленных в файле, а специалисты уровнем ниже наоборот - получали в 2022 году самую высокую зп.


```
table = pd.pivot_table(df, index=['employee_residence'], values='remote_ratio', aggfunc=[np.mean, len])
table
```

	mean	len
	remote_ratio	remote_ratio
employee_residence		
AE	66.666667	3
AR	100.000000	1
AT	16.666667	3
AU	83.333333	3
BE	75.000000	2
BG	100.000000	1
BO	100.000000	1
BR	66.666667	6
CA	75.862069	29
CH	0.000000	1
CL	100.000000	1
CN	0.000000	1
CO	50.000000	1
CZ	50.000000	1
DE	56.000000	25
DK	50.000000	2
DZ	50.000000	1
EE	100.000000	1
ES	90.000000	15
FR	55.555556	18
GB	45.454545	44
GR	80.769231	13
HK	50.000000	1

Рисунок 12 — Количество сотрудников в каждой стране и их частота появления в офисе

Исходя из этой таблицы можно узнать сколько работников в каждой стране, и какой способ работы для них наиболее предпочтителен (удаленный, полуудаленный, либо офисный)

Ссылка на Jupyter Notebook проект на GitHub - <https://github.com/EgorYasinovskiy/Data-Analys/tree/master/ЛП1>