Напишите программный код или сгенерируйте его с помощью искусственного интеллекта.

Предварительный анализ данных и введение

Цель

1. Изучить наиболее высокооплачиваемые должности и навыки в индустрии обработки данных.
2. Использовать Python для изучения реальных данных о вакансиях.
3. Для соискателей: использовать эти знания, чтобы помочь найти лучшие вакансии.

Предварительный анализ данных для всех ролей, связанных с данными

Роли для изучения

```
!pip install datasets
# Импортируем библиотеки
import ast
import pandas as pd
import seaborn as sns
from datasets import load_dataset
import matplotlib.pyplot as plt
```

```
Collecting datasets
  Downloading datasets-3.2.0-py3-none-any.whl.metadata (20 kB)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from datasets) (3.16.1)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.10/dist-packages (from datasets) (1.26.4)
Requirement already satisfied: pyarrow>=15.0.0 in /usr/local/lib/python3.10/dist-packages (from datasets) (17.0.0)
Collecting dill<0.3.9,>=0.3.0 (from datasets)
  Downloading dill-0.3.8-py3-none-any.whl.metadata (10 kB)
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (from datasets) (2.2.2)
Requirement already satisfied: requests>=2.32.2 in /usr/local/lib/python3.10/dist-packages (from datasets) (2.32.3)
Requirement already satisfied: tqdm>=4.66.3 in /usr/local/lib/python3.10/dist-packages (from datasets) (4.67.1)
Collecting xxhash (from datasets)
  Downloading xxhash-3.5.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (12 kB)
Collecting multiprocess<0.70.17 (from datasets)
  Downloading multiprocess-0.70.16-py310-none-any.whl.metadata (7.2 kB)
Collecting fsspec<=2024.9.0,>=2023.1.0 (from fsspec[http]<=2024.9.0,>=2023.1.0->datasets)
  Downloading fsspec-2024.9.0-py3-none-any.whl.metadata (11 kB)
Requirement already satisfied: aiohttp in /usr/local/lib/python3.10/dist-packages (from datasets) (3.11.10)
Requirement already satisfied: huggingface-hub>=0.23.0 in /usr/local/lib/python3.10/dist-packages (from datasets) (0.27.0)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from datasets) (24.2)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-packages (from datasets) (6.0.2)
Requirement already satisfied: aiohappyeyeballs>=2.3.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (2.4.4)
Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (1.3.2)
Requirement already satisfied: async-timeout<6.0,>=4.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (4.0.3)
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (24.3.0)
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (1.5.0)
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (6.1.0)
Requirement already satisfied: propcache>=0.2.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (0.2.1)
Requirement already satisfied: yarl<2.0,>=1.17.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (1.18.3)
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.23.0-
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests>=2.32.2->datasets
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests>=2.32.2->datasets) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests>=2.32.2->datasets) (2.2
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests>=2.32.2->datasets) (202
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas->datasets) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas->datasets) (2024.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.10/dist-packages (from pandas->datasets) (2024.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2->pandas->datasets)
Downloading datasets-3.2.0-py3-none-any.whl (480 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 480.6/480.6 kB 9.5 MB/s eta 0:00:00
Downloading dill-0.3.8-py3-none-any.whl (116 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 116.3/116.3 kB 9.0 MB/s eta 0:00:00
Downloading fsspec-2024.9.0-py3-none-any.whl (179 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 179.3/179.3 kB 13.9 MB/s eta 0:00:00
Downloading multiprocess-0.70.16-py310-none-any.whl (134 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 134.8/134.8 kB 11.2 MB/s eta 0:00:00
Downloading xxhash-3.5.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (194 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 194.1/194.1 kB 14.6 MB/s eta 0:00:00
Installing collected packages: xxhash, fsspec, dill, multiprocess, datasets
  Attempting uninstall: fsspec
    Found existing installation: fsspec 2024.10.0
    Uninstalling fsspec-2024.10.0:
      Successfully uninstalled fsspec-2024.10.0
```

```
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the sou
gcsfs 2024.10.0 requires fsspec==2024.10.0, but you have fsspec 2024.9.0 which is incompatible.
Successfully installed datasets-3.2.0 dill-0.3.8 fsspec-2024.9.0 multiprocess-0.70.16 xxhash-3.5.0
```

```
# Загрузка данных
dataset = load_dataset('lukebarousse/data_jobs')
df = dataset['train'].to_pandas()
```

```
# Очистка данных
df['job_posted_date'] = pd.to_datetime(df['job_posted_date'])
df['job_skills'] = df['job_skills'].apply(lambda x: ast.literal_eval(x) if pd.notna(x) else x)
```

```
/usr/local/lib/python3.10/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set it as s
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
README.md: 100%  28.0/28.0 [00:00<00:00, 1.37kB/s]
data_jobs.csv: 100%  231M/231M [00:05<00:00, 42.1MB/s]
Generating train split: 100%  785741/785741 [00:08<00:00, 120232.21 examples/s]
```

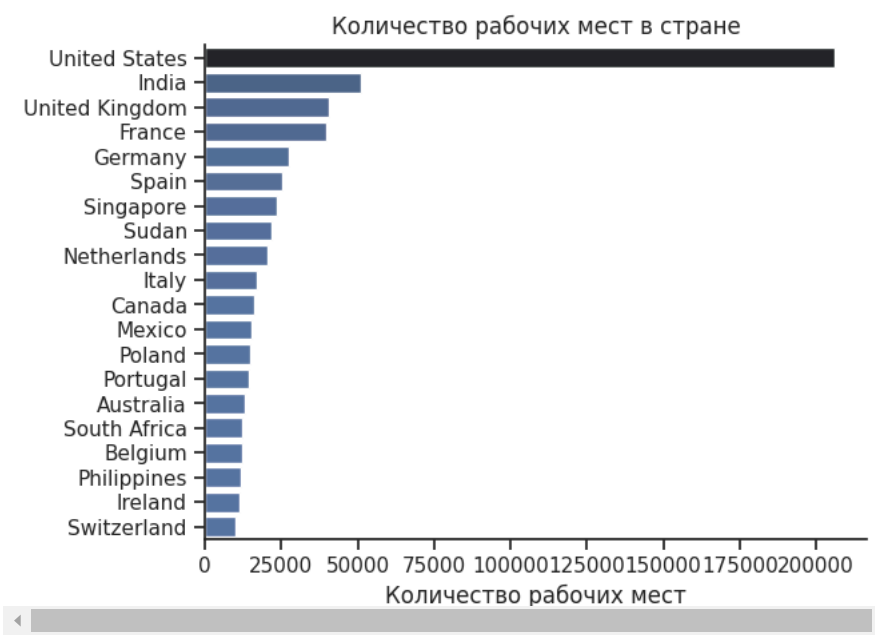
```
df_plot = df['job_title_short'].value_counts().to_frame()
```

```
sns.set_theme(style='ticks')
sns.barplot(data=df_plot, x='count', y='job_title_short', hue='count', palette='dark:b_r', legend=False)
sns.despine()
plt.title('Количество рабочих мест на одну должность')
plt.xlabel('Количество рабочих мест')
plt.ylabel('')
plt.show()
```

Страны для изучения

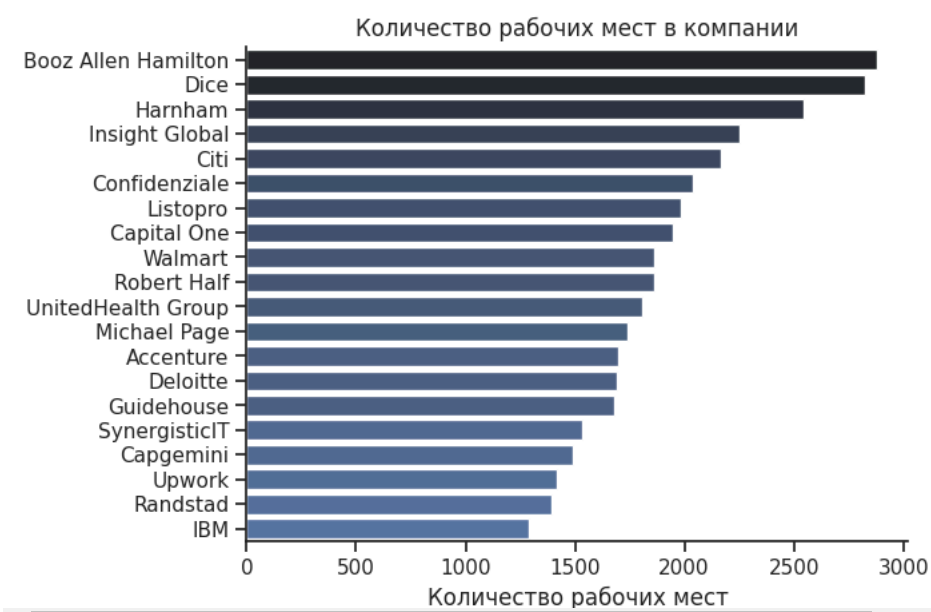
```
df_plot = df['job_country'].value_counts().to_frame().head(20)
```

```
sns.set_theme(style='ticks')
sns.barplot(data=df_plot, x='count', y='job_country', hue='count', palette='dark:b_r', legend=False)
sns.despine()
plt.title('Количество рабочих мест в стране')
plt.xlabel('Количество рабочих мест')
plt.ylabel('')
plt.show()
```



Компании для изучения

```
df_plot = df['company_name'].value_counts().to_frame()[1:].head(20)


sns.set_theme(style='ticks')
sns.barplot(data=df_plot, x='count', y='company_name', hue='count', palette='dark:b_r', legend=False)
sns.despine()
plt.title('Количество рабочих мест в компании')
plt.xlabel('Количество рабочих мест')
plt.ylabel('')
plt.show()
```



Возможности трудоустройства

```
dict_column = {
    'job_work_from_home': 'Предл. работа на дому',
    'job_no_degree_mention': 'Треб. по высш. образов.',
    'job_health_insurance': 'Медицинская страховка'
}


fig, ax = plt.subplots(1, 3, figsize=(11, 3.5))


for i, (column, title) in enumerate(dict_column.items()):
```

```
    ax[i].pie(df[column].value_counts(), labels=['False', 'True'], autopct='%1.1f%%', startangle=90)
    ax[i].set_title(title)


plt.show()
```


```
dict_column = {
    'job_work_from_home': 'Предл. работа на дому',
    'job_no_degree_mention': 'Треб. по высш. образов.',
    'job_health_insurance': 'Медицинская страховка'
}


fig, ax = plt.subplots(1, 3, figsize=(11, 3.5))


for i, (column, title) in enumerate(dict_column.items()):
    ax[i].pie(df[column].value_counts(), labels=['Нет', 'Да'], autopct='%1.1f%%', startangle=90)
    ax[i].set_title(title)


plt.show()
```

Исследовательский анализ данных для аналитиков данных в США

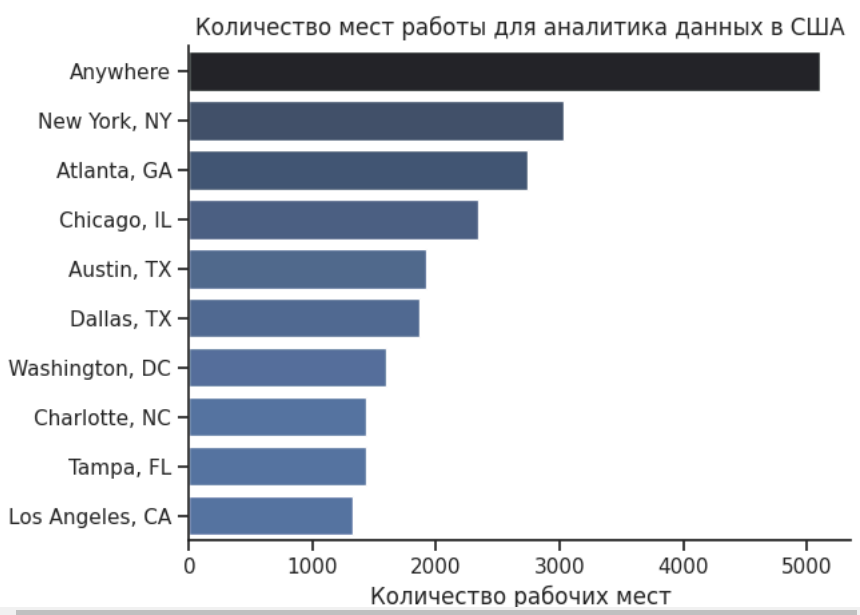
```
# Импорт библиотек
import ast
import pandas as pd
import seaborn as sns
from datasets import load_dataset
import matplotlib.pyplot as plt


# Загрузка данных
dataset = load_dataset('lukebarousse/data_jobs')
df = dataset['train'].to_pandas()


# Очистка данных
df['job_posted_date'] = pd.to_datetime(df['job_posted_date'])
df['job_skills'] = df['job_skills'].apply(lambda x: ast.literal_eval(x) if pd.notna(x) else x)
```

Фильтр для ролей аналитика данных в США

```
df_DA_US = df[(df['job_country'] == 'United States') & (df['job_title_short'] == 'Data Analyst')]
```

Локации для исследования:

```
df_plot = df_DA_US['job_location'].value_counts().head(10).to_frame()


sns.set_theme(style='ticks')
sns.barplot(data=df_plot, x='count', y='job_location', hue='count', palette='dark:b_r', legend=False)
sns.despine()
plt.title('Количество мест работы для аналитика данных в США')
plt.xlabel('Количество рабочих мест')
plt.ylabel('')
plt.show()
```



Возможности работы

```
# rewrite the above with a for loop
dict_column = {
    'job_work_from_home': 'Предл. работа на дому',
    'job_no_degree_mention': 'Треб. по высш. образов.',
    'job_health_insurance': 'Медицинская страховка'
}


fig, ax = plt.subplots(1, 3)
fig.set_size_inches((12, 5))


for i, (column, title) in enumerate(dict_column.items()):
    ax[i].pie(df_DA_US[column].value_counts(), labels=['Нет', 'Да'], autopct='%1.1f%%', startangle=90)
    ax[i].set_title(title)


# plt.suptitle('Benefit Analysis of Data Jobs', fontsize=16)
plt.show()
```

Компании для изучения:

```
df_plot = df_DA_US['company_name'].value_counts().head(10).to_frame()
```

```
sns.set_theme(style='ticks')
sns.barplot(data=df_plot, x='count', y='company_name', hue='count', palette='dark:b_r', legend=False)
sns.despine()
plt.title('Количество компаний для аналитика данных в США')
plt.xlabel('Количество рабочих мест')
plt.ylabel('')
```

```
Text(0, 0.5, '')
```

Каковы наиболее востребованные навыки для 3 самых популярных профессий в области данных?

Методология

1. Очистка колонки навыков
2. Рассчитать количество навыков на основе `job_title_short`.
3. Вычислите процентное соотношение навыков
4. Постройте график итоговых результатов

Оригинальное исследование

[13_Matplotlib_Format_Charts.ipynb]

```
# Импорт библиотек
import ast
import pandas as pd
```

```
import seaborn as sns
from datasets import load_dataset
import matplotlib.pyplot as plt


# Загрузка данных
dataset = load_dataset('lukebarousse/data_jobs')
df = dataset['train'].to_pandas()


# Очистка данных
df['job_posted_date'] = pd.to_datetime(df['job_posted_date'])
df['job_skills'] = df['job_skills'].apply(lambda x: ast.literal_eval(x) if pd.notna(x) else x)
```

Фильтр данных для Соединенных Штатов

Отфильтруйте данные по рынку Соединенных Штатов.

```
df_US = df[df['job_country'] == 'United States']
```

Исследование навыков

Чтобы изменить содержимое ячейки, дважды нажмите на нее (или выберите "Ввод")

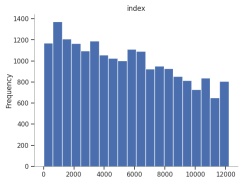
```
df_skills = df_US.explode('job_skills')


df_skills[['job_title', 'job_skills']]
```

	job_title	job_skills
0	Senior Clinical Data Engineer / Principal Clin...	None
3	LEAD ENGINEER - PRINCIPAL ANALYST - PRINCIPAL ...	python
3	LEAD ENGINEER - PRINCIPAL ANALYST - PRINCIPAL ...	c++
3	LEAD ENGINEER - PRINCIPAL ANALYST - PRINCIPAL ...	java
3	LEAD ENGINEER - PRINCIPAL ANALYST - PRINCIPAL ...	matlab
...	...	...
785692	Data Scientist- Hybrid Work Location	r
785703	Data Analyst - CRYPTOGRAPHY - Full-time	None
785705	Expert Business Data Analyst - Now Hiring	sql
785705	Expert Business Data Analyst - Now Hiring	python
785705	Expert Business Data Analyst - Now Hiring	tableau

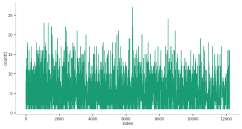
1073565 rows × 2 columns

```
No charts were generated by quickchart
Warning: total number of rows (1073565) exceeds max_rows (20000). Limiting to first (20000) rows.
```

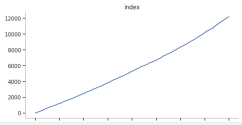
Distributions



Time series



Values



Подсчет навыков для должностей

Группирует DataFrame по `job_skills` и `job_title_short`, подсчитывая количество вхождений каждого навыка в каждое название должности. Затем сбрасывает индекс серии, чтобы превратить ее обратно в DataFrame, и переименовывает серию, содержащую подсчет, в 'count'. Итоговый DataFrame, `df_skills_count`, показывает частоту встречаемости каждого навыка в каждом названии должности.

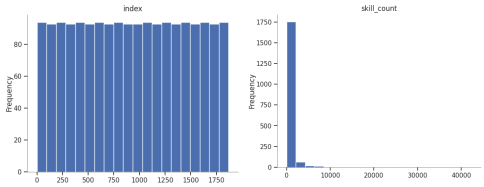
```
# Группировка по job_skills и job_title_short и подсчет количества появлений
df_skills_count = df_skills.groupby(['job_skills', 'job_title_short']).size()


# Назовите столбец подсчета как count
df_skills_count = df_skills_count.reset_index(name='skill_count')


# Отсортируйте значения по счету_навыков в порядке убывания
df_skills_count.sort_values(by='skill_count', ascending=False, inplace=True)
```

```
df_skills_count
```

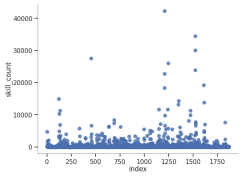
	job_skills	job_title_short	skill_count
1209	python	Data Scientist	42379
1521	sql	Data Analyst	34452
1523	sql	Data Scientist	30034
455	excel	Data Analyst	27519
1243	r	Data Scientist	26022
...	...	...	...
245	clojure	Software Engineer	1
1738	vb.net	Senior Data Scientist	1
530	fortran	Machine Learning Engineer	1
1116	planner	Cloud Engineer	1
960	nltk	Senior Data Engineer	1

1870 rows × 3 columns

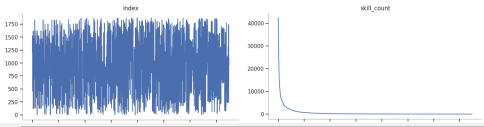
Distributions



2-d distributions



Values



Создайте список из 3 лучших ролей

Фокус: Аналитики данных, инженеры данных и ученые по данным

Отфильтруйте названия должностей по наиболее популярным.

```
job_titles = df_skills_count['job_title_short'].unique().tolist()
```

```
job_titles = sorted(job_titles[:3])
```

```
job_titles
```

```
['Data Analyst', 'Data Engineer', 'Data Scientist']
```

Начертить графики умений

Создает сложенную горизонтальную гистограмму для 5 лучших навыков для топ-3 ролей, отображающую частоту использования каждого навыка.

```
fig, ax = plt.subplots(len(job_titles), 1)


sns.set_theme(style='ticks')


for i, job_title in enumerate(job_titles):
    df_plot = df_skills_count[df_skills_count['job_title_short'] == job_title].head(5)[::-1]
    sns.barplot(data=df_plot, x='skill_count', y='job_skills', ax=ax[i], hue='skill_count', palette='dark:b_r')
    ax[i].set_title(job_title)
    ax[i].invert_yaxis()
    ax[i].set_ylabel('')
    ax[i].set_xlabel('')
    ax[i].get_legend().remove()
    ax[i].set_xlim(0, 45000) # сделать шкалы одинаковыми


fig.suptitle('Количество навыков, требуемых в объявлениях о вакансиях в США', fontsize=15)
fig.tight_layout(h_pad=0.5) # исправить перекрытие


plt.show()
```

Преобразование числа в проценты

Фокус: Подсчеты не показывают, в какой части вакансий требуются эти навыки.

Поэтому мы преобразуем подсчеты в проценты, что поможет нам понять, как каждое название должности выглядит по отношению ко всему набору данных.

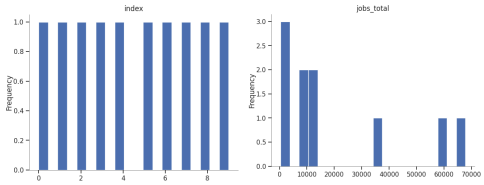
Прежде чем вычислять процентное соотношение, нам нужно получить общее количество вакансий, размещенных по названию должности. Вычислите частоту каждого названия вакансии, используя метод `value_counts()` для столбца `job_title_short`. Затем сбросьте индекс, чтобы преобразовать серию в DataFrame, и переименуйте столбцы в `job_title_short` и `total`. Теперь DataFrame `df_job_title_count` содержит список названий должностей вместе с их общим количеством.

```
# Используйте исходный df для получения количества названий должностей
df_job_title_count = df_US['job_title_short'].value_counts().reset_index(name='jobs_total')
```

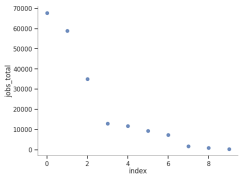
```
df_job_title_count
```

	job_title_short	jobs_total
0	Data Analyst	67816
1	Data Scientist	58830
2	Data Engineer	35080
3	Senior Data Scientist	12946
4	Senior Data Analyst	11791
5	Senior Data Engineer	9289
6	Business Analyst	7382
7	Software Engineer	1814
8	Machine Learning Engineer	921
9	Cloud Engineer	423

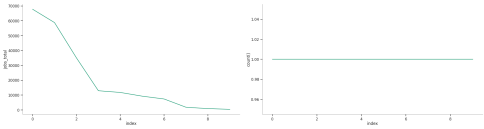
Distributions



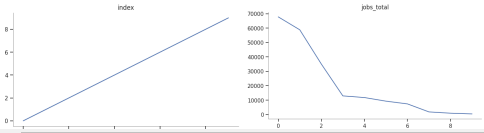
2-d distributions



Time series



Values



Затем мы вычисляем процентное соотношение. Сначала объедините `df_skills_count` и `df_job_title_count`, основываясь на столбце 'job_title_short', обеспечивая, чтобы количество навыков было связано с общим количеством объявлений о работе для этого названия. Затем рассчитайте процентное соотношение каждого навыка в названии должности, разделив количество навыков на общее количество вакансий и умножив на 100, и добавьте эти новые данные в столбец «процент».

```
df_skills_perc = pd.merge(df_skills_count, df_job_title_count, on='job_title_short', how='left')
```

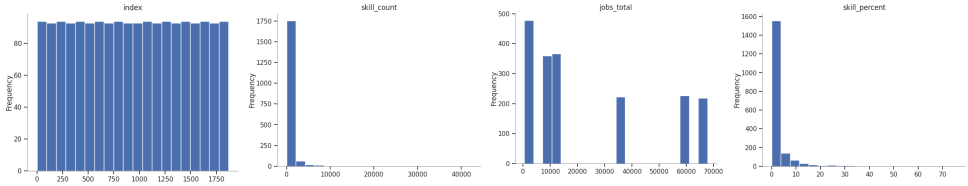
```
df_skills_perc['skill_percent'] = (df_skills_perc['skill_count'] / df_skills_perc['jobs_total']) * 100
```

```
df_skills_perc
```

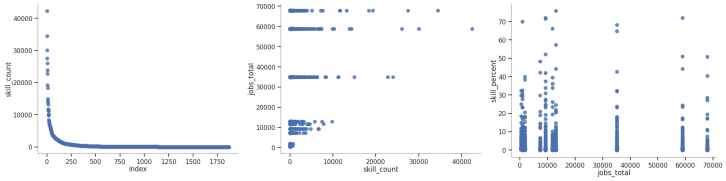
	job_skills	job_title_short	skill_count	jobs_total	skill_percent
0	python	Data Scientist	42379	58830	72.036376
1	sql	Data Analyst	34452	67816	50.802171
2	sql	Data Scientist	30034	58830	51.052184
3	excel	Data Analyst	27519	67816	40.578919
4	r	Data Scientist	26022	58830	44.232534
...	...	...	...	...	...
1865	clojure	Software Engineer	1	1814	0.055127
1866	vb.net	Senior Data Scientist	1	12946	0.007724
1867	fortran	Machine Learning Engineer	1	921	0.108578
1868	planner	Cloud Engineer	1	423	0.236407
1869	nltk	Senior Data Engineer	1	9289	0.010765

1870 rows × 5 columns

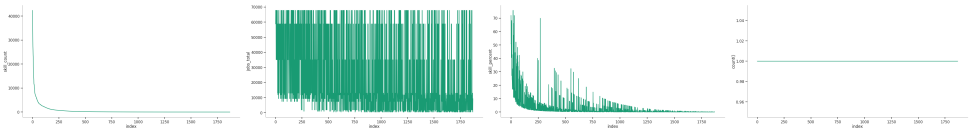
Distributions



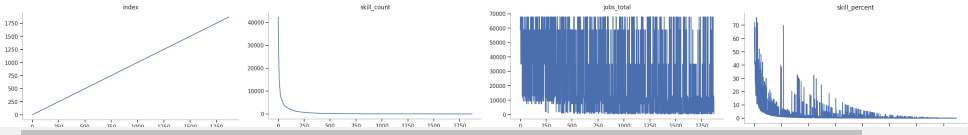
2-d distributions



Time series



Values

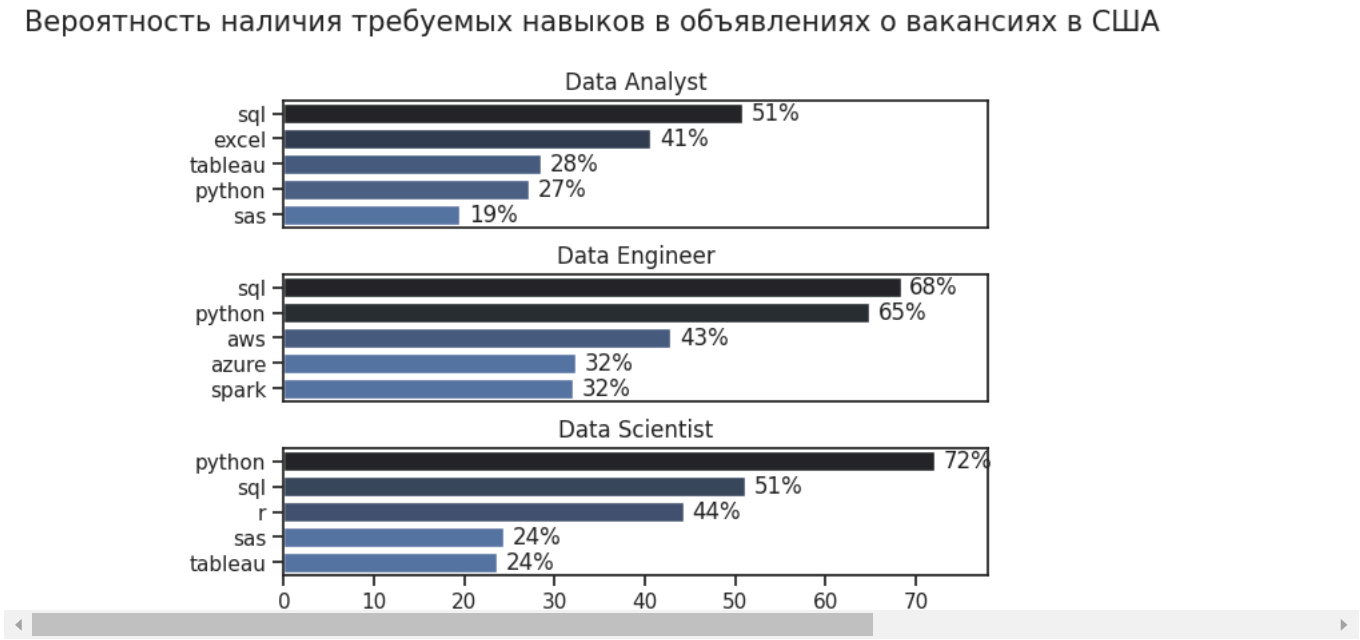


Начертить счетчик процентов

Фильтрует и сортирует DataFrame, чтобы получить 5 лучших навыков в процентах для этих трех лучших ролей. После сортировки навыков по убыванию процента измените порядок этих 5 лучших записей, чтобы использовать их в горизонтальной гистограмме, которая по умолчанию начинает строиться снизу.

```
fig, ax = plt.subplots(len(job_titles), 1)
```

```
for i, job_title in enumerate(job_titles):
    df_plot = df_skills_perc[df_skills_perc['job_title_short'] == job_title].head(5)
    sns.barplot(data=df_plot, x='skill_percent', y='job_skills', ax=ax[i], hue='skill_count', palette='dark:b_r')
    ax[i].set_title(job_title)
    ax[i].set_ylabel('')
    ax[i].set_xlabel('')
    ax[i].get_legend().remove()
    ax[i].set_xlim(0, 78)
    # удалите метки на оси x для лучшей читаемости
    if i != len(job_titles) - 1:
        ax[i].set_xticks([])


    # обозначьте процент на столбиках
    for n, v in enumerate(df_plot['skill_percent']):
        ax[i].text(v + 1, n, f'{v:.0f}%', va='center')


fig.suptitle('Вероятность наличия требуемых навыков в объявлениях о вакансиях в США', fontsize=15)
fig.tight_layout(h_pad=.8)
plt.show()
```



Насколько хорошо оплачивается работа и навыки аналитиков данных?

Методология

1. Оцените медианную зарплату для 6 лучших профессий, связанных с данными.
2. Найдите медианную зарплату по каждому навыку для аналитиков данных
3. Визуализация наиболее высокооплачиваемых и наиболее востребованных навыков

```
# Импорт библиотек
import ast
import pandas as pd
import seaborn as sns
from datasets import load_dataset
import matplotlib.pyplot as plt


# Загрузка данных
dataset = load_dataset('lukebarousse/data_jobs')
df = dataset['train'].to_pandas()


# Очистка данных
df['job_posted_date'] = pd.to_datetime(df['job_posted_date'])
df['job_skills'] = df['job_skills'].apply(lambda x: ast.literal_eval(x) if pd.notna(x) else x)
```

Распределение зарплаты по должностям

Отфильтруйте наши данные, чтобы включить только значения зарплат из Соединенных Штатов.

```
# filter for the job titles and country
df_US = df[(df['job_country'] == 'United States')].dropna(subset=['salary_year_avg'])
```

Создайте список основных названий должностей в нашем наборе данных и отфильтруйте наш кадр данных, чтобы он содержал только эти названия должностей.

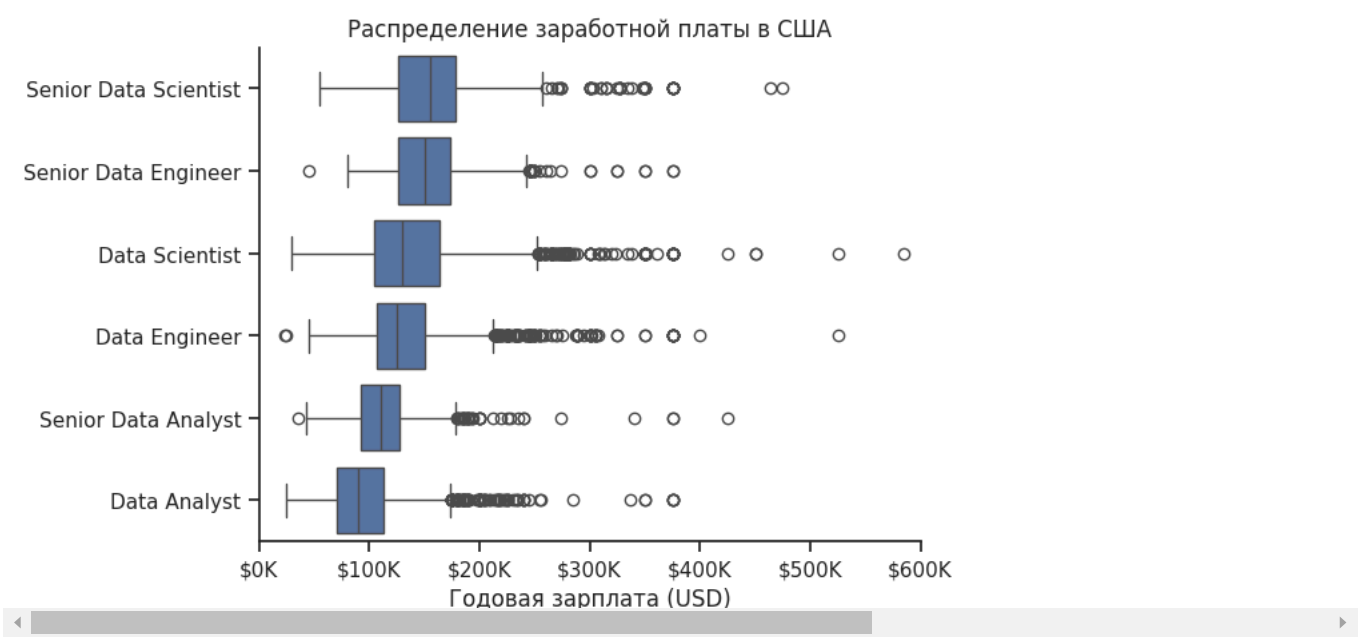
```
job_titles = df_US['job_title_short'].value_counts().index[:6].tolist()


# отфильтровать df для 6 лучших названий должностей
df_US_top6 = df_US[df_US['job_title_short'].isin(job_titles)]


# упорядочить названия должностей по медианной зарплате
job_order = df_US_top6.groupby('job_title_short')['salary_year_avg'].median().sort_values(ascending=False).index


job_titles
```

```
['Data Scientist',
 'Data Analyst',
 'Data Engineer',
 'Senior Data Scientist',
 'Senior Data Engineer',
 'Senior Data Analyst']
```

График распределения заработной платы

Постройте распределение зарплат 6 лучших наименований должностей с помощью бокс-диаграммы.

```
sns.boxplot(data=df_US_top6, x='salary_year_avg', y='job_title_short', order=job_order)
sns.set_theme(style='ticks')
sns.despine()


plt.title('Распределение заработной платы в США')
plt.xlabel('Годовая зарплата (USD)')
plt.ylabel('')
plt.xlim(0, 600000)
ticks_x = plt.FuncFormatter(lambda y, pos: f'${int(y/1000)}K')
plt.gca().xaxis.set_major_formatter(ticks_x)
plt.show()
```



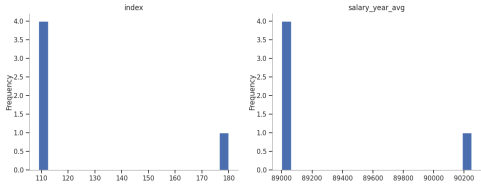
Исследуйте среднюю зарплату аналитиков данных в зависимости от их квалификации

Отфильтруйте исходный набор данных, чтобы получить только строки, в которых название должности - 'Data Analyst', а страна - 'United States', и создайте новый фрейм данных df_DA_US. Удалите NaN-значения из столбца 'salary_year_avg'. Затем он использует метод explode для столбца job_skills, чтобы создать новую строку в DataFrame для каждого навыка, связанного с работой. Наконец, он отображает первые пять записей в столбцах salary_year_avg и job_skills.

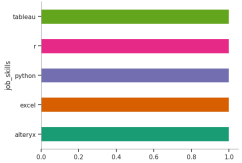
```
# Работа аналитика данных только по США
df_DA_US = df[(df['job_title_short'] == 'Data Analyst') & (df['job_country'] == 'United States')].copy()


# Удалите NaN-значения из столбца 'salary_year_avg' для точной визуализации
df_DA_US = df_DA_US.dropna(subset=['salary_year_avg'])


df_DA_US = df_DA_US.explode('job_skills')


df_DA_US[['salary_year_avg', 'job_skills']].head(5)
```

	salary_year_avg	job_skills
109	89000.0	python
109	89000.0	r
109	89000.0	alteryx
109	89000.0	tableau
180	90250.0	excel

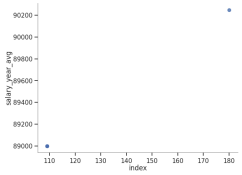
Distributions



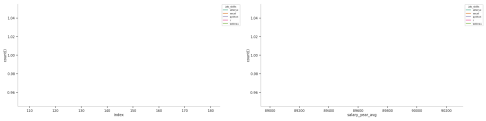
Categorical distributions



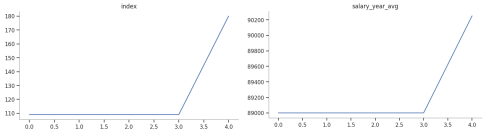
2-d distributions



Time series

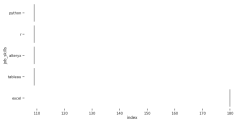


Values



Faceted distributions

```
<string>:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `le
```



```
<string>:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `le
```

Определите самые высокооплачиваемые и самые востребованные навыки

Получает десять самых высокооплачиваемых навыков для аналитиков данных, вычисляя медианную зарплату для каждого навыка, указанного в `df_DA_US`. Он группирует данные по навыкам работы, вычисляет медианную зарплату, сортирует эти значения в порядке убывания по медиане, а затем выбирает 10 лучших. Затем эти данные форматируются в новый DataFrame (`df_DA_top_pay`) со сброшенным индексом и переименованным столбцом зарплаты с меткой 'median_salary'.

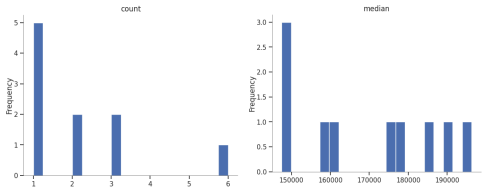
```
df_DA_top_pay = df_DA_US.groupby('job_skills')['salary_year_avg'].agg(['count', 'median']).sort_values(by='median', ascending=False)
```

```
df_DA_top_pay = df_DA_top_pay.head(10)
```

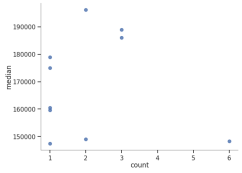
```
df_DA_top_pay
```

job_skills	count	median
dplyr	2	196250.0
bitbucket	3	189000.0
gitlab	3	186000.0
solidity	1	179000.0
hugging face	1	175000.0
couchbase	1	160515.0
ansible	1	159640.0
mxnet	2	149000.0
cassandra	6	148250.0
vmware	1	147500.0

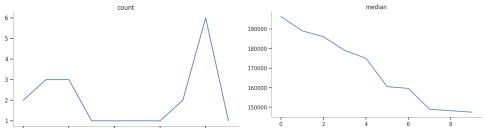
Distributions



2-d distributions



Values



Вычисляет количество и медианную зарплату для каждого навыка в `df_DA_US`. Он группирует данные по `job_skills`, агрегирует их, чтобы найти количество и медианную зарплату для каждого навыка, а затем сортирует результаты по количеству в порядке убывания по количеству. Это подмножество повторно сортируется по медианной зарплате в порядке убывания.

```
df_DA_skills = df_DA_US.groupby('job_skills')['salary_year_avg'].agg(['count', 'median']).sort_values(by='count', ascending=False)
```