In [324…
```
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.patches as mpatches
import seaborn as sns
import ast
```

In [325…
```
df = pd.read_pickle('cleaned_df.pkl')
```

In [326…
```
df.sample(5)
```

Out[326…

	Job title	Medium	Job Schedule Type	Remote or On-site	Search Location	Job Posted Date	Degree required or not	Job Country	S
441779	Data Scientist	Jobgether	Full-time	1	United Kingdom	2023-06-17 16:27:33	0	United Kingdom	
92629	Data Analyst	BeBee	Full-time	0	Texas, United States	2023-10-13 00:01:49	0	United States	
273083	Senior Data Analyst	Trabajo.org	Full-time	0	Texas, United States	2023-05-06 07:01:39	1	United States	
355203	Data Analyst	Sercanto	Full-time	0	France	2023-09-12 23:30:24	0	France	
183090	Software Engineer	BeBee Costa Rica	Full-time	0	Costa Rica	2023-12-27 15:45:00	1	Costa Rica	

Global design

In [327…
```
plt.rcParams.update({
    'axes.titlesize': 20,
    'axes.titlepad':20,
    'axes.titleweight': 'bold',
    'axes.labelsize': 16,
    'axes.labelpad': 20,
    'axes.labelweight': 'bold',
    'xtick.labelsize': 10,
    'ytick.labelsize': 10,
    'figure.figsize': [10,6]
})
```

```
def custom_bar_params():
    return {
            'palette' : 'Blues_r',
            'saturation' : 1,
            'edgecolor' : 'lightgrey',
            'width' : .7,
            'legend' : False
    }

def custom_pie_params():
    return {
            'color' : 'white',
            'fontsize' : 12,
            'fontweight' : 'bold'
    }
```

Dynamic job function

In [328…

```
all_job_titles = df['Job title'].unique().tolist()
all_job_titles.insert(0, 'all data')

# function for dynamic role
def job_title_switcher(job_title):
      if job_title == "all data":
              return df.copy()
      else:
              return df[df['Job title'] == job_title]
```

Job counts by titles

In [329…

```
df_job_title = pd.DataFrame(df['Job title'].value_counts()).reset_index()
df_job_title.columns = ["Job title", "Number of jobs"]
df_job_title
```

Out[329…

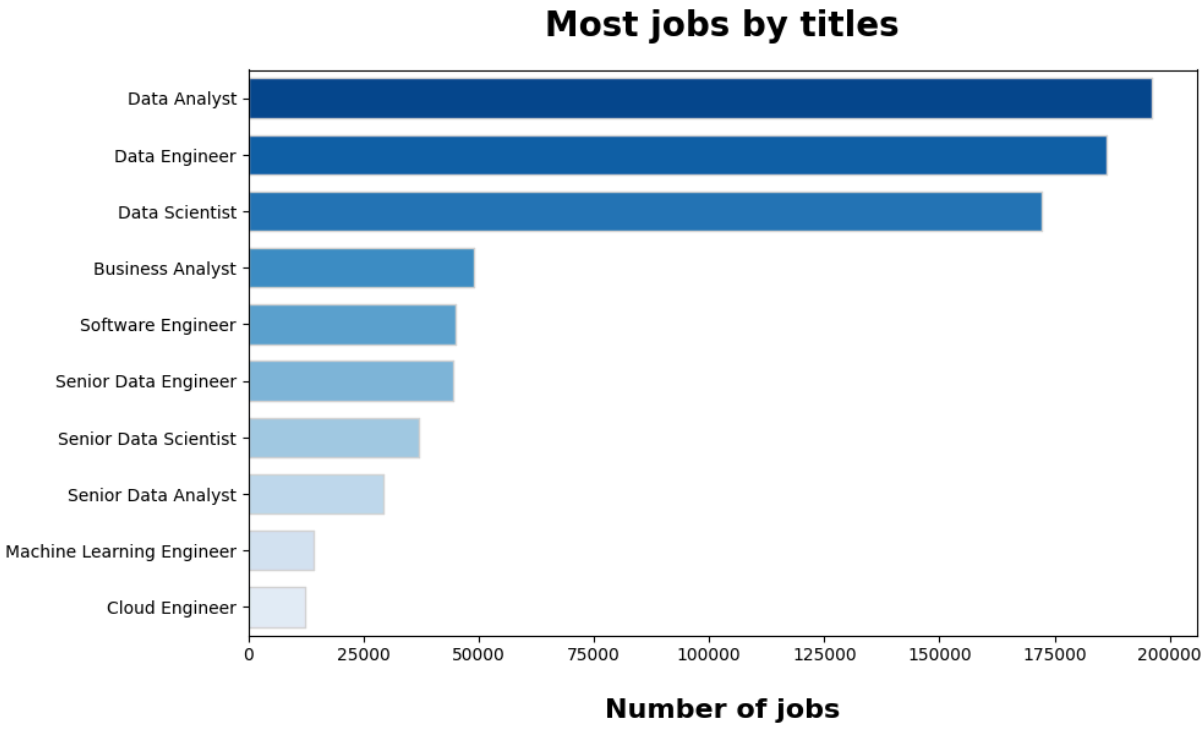
	Job title	Number of jobs
0	Data Analyst	196050
1	Data Engineer	186216
2	Data Scientist	172263
3	Business Analyst	49053
4	Software Engineer	44918
5	Senior Data Engineer	44561
6	Senior Data Scientist	36955
7	Senior Data Analyst	29214
8	Machine Learning Engineer	14079
9	Cloud Engineer	12331

In [330…

```
sns.barplot(
            data=df_job_title,
            x='Number of jobs',
            y='Job title',
            hue='Job title',
            **custom_bar_params()
            )

plt.title('Most jobs by titles')
plt.xlabel('Number of jobs')
plt.ylabel('')

plt.show()
```



Top ten mediums by job counts

In [331…
```
all_job_titles
```

Out[331…
```
['all data',
 'Senior Data Engineer',
 'Data Analyst',
 'Data Engineer',
 'Business Analyst',
 'Data Scientist',
 'Machine Learning Engineer',
 'Senior Data Analyst',
 'Cloud Engineer',
 'Senior Data Scientist',
 'Software Engineer']
```

In [332…
```
medium_job_title = 'all data' # put any title in the place of "all data "
top10_medium_for_role = pd.DataFrame(job_title_switcher(medium_job_title)['Medium']
top10_medium_for_role.columns = ['Job medium', 'Number of jobs']
top10_medium_for_role
```

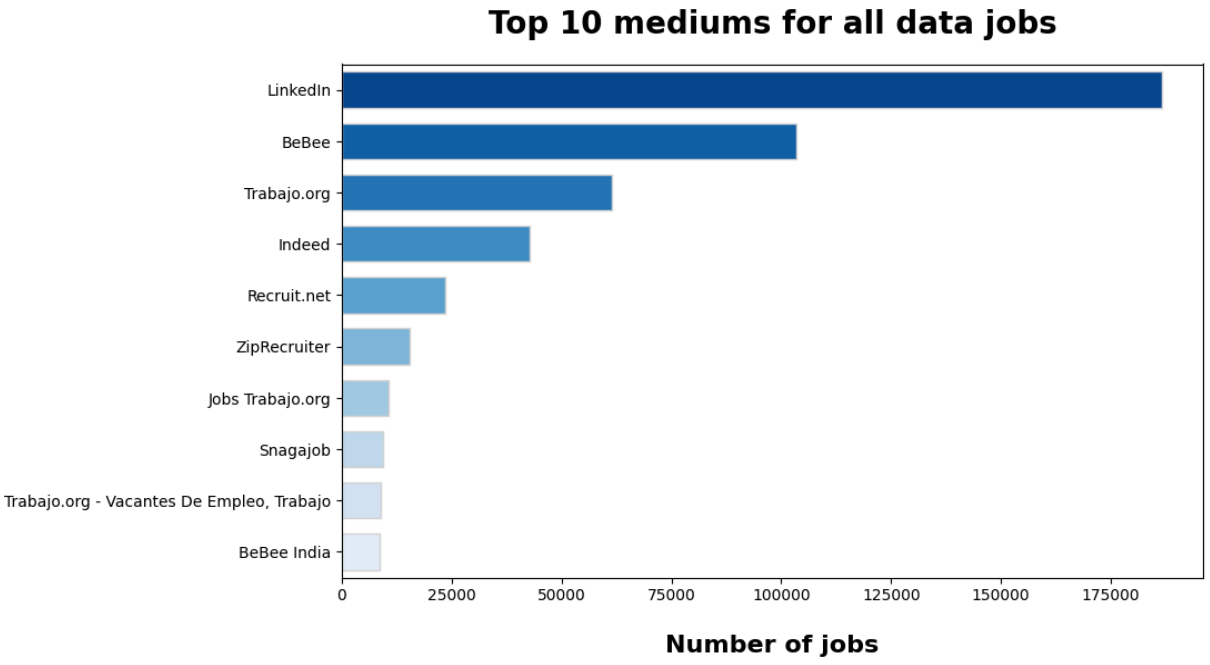
Out[332…

	Job medium	Number of jobs
0	LinkedIn	186658
1	BeBee	103500
2	Trabajo.org	61545
3	Indeed	42748
4	Recruit.net	23646
5	ZipRecruiter	15533
6	Jobs Trabajo.org	10601
7	Snagajob	9355
8	Trabajo.org - Vacantes De Empleo, Trabajo	8912
9	BeBee India	8637

In [333…

```
sns.barplot(
            data=top10_medium_for_role,
            x= 'Number of jobs',
            y= 'Job medium',
            hue= 'Job medium',
            **custom_bar_params()
            )

plt.title(f'Top 10 mediums for {medium_job_title} jobs')
plt.xlabel('Number of jobs')
plt.ylabel('')
plt.show()
```



Remote jobs ratio for data job

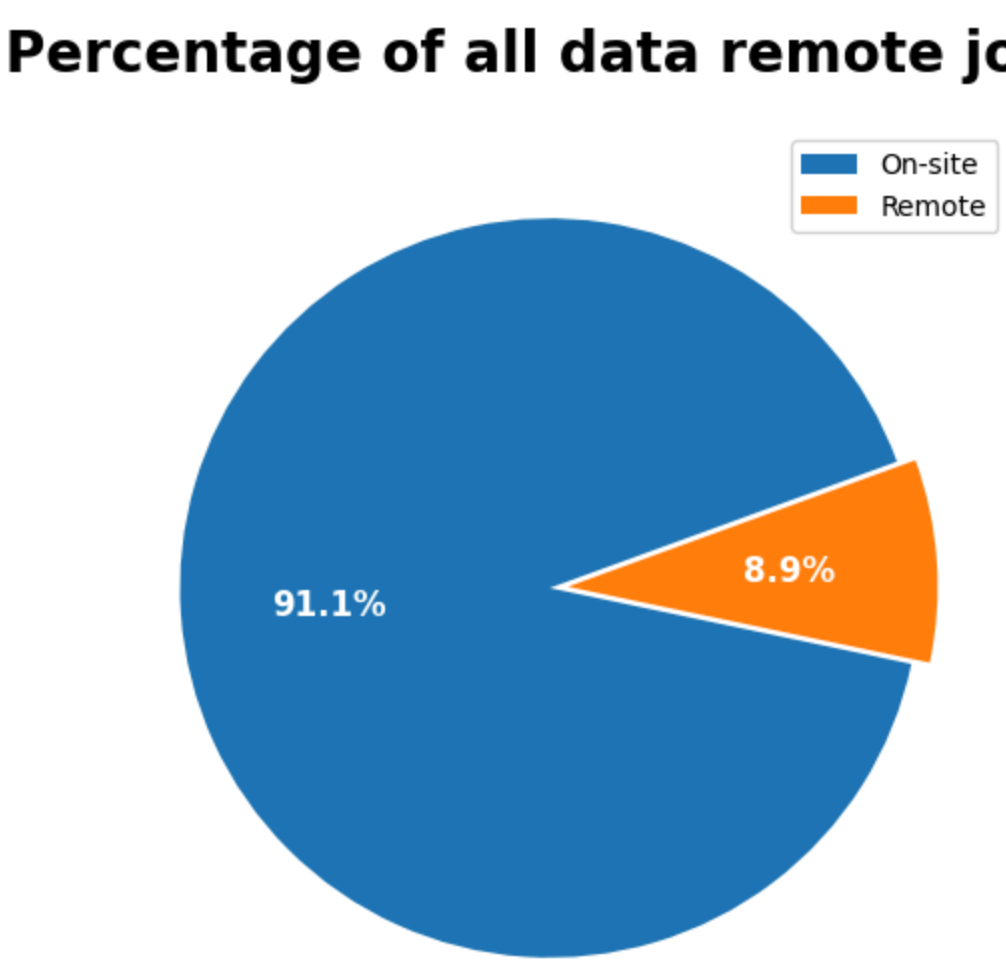
In [334…
```
all_job_titles
```

Out[334…
```
['all data',
 'Senior Data Engineer',
 'Data Analyst',
 'Data Engineer',
 'Business Analyst',
 'Data Scientist',
 'Machine Learning Engineer',
 'Senior Data Analyst',
 'Cloud Engineer',
 'Senior Data Scientist',
 'Software Engineer']
```

In [335…
```
remote_title = 'all data' # put any title in the place of "all data"
df_remote = job_title_switcher(remote_title)['Remote or On-site'].value_counts(norm
df_remote
```

Out[335…
```
Remote or On-site
0    0.91149
1    0.08851
Name: proportion, dtype: float64
```

In [336…
```
plt.pie(
        df_remote,
        labels=['On-site', 'Remote'],
        autopct=lambda p: f'{p:.1f}%',
        startangle=20,
        explode = [0, .05],
        textprops = custom_pie_params()
        )
plt.title(f'Percentage of {remote_title} remote jobs')
plt.legend()
plt.show()
```



Degree relevancy

In [337…
```
all_job_titles
```

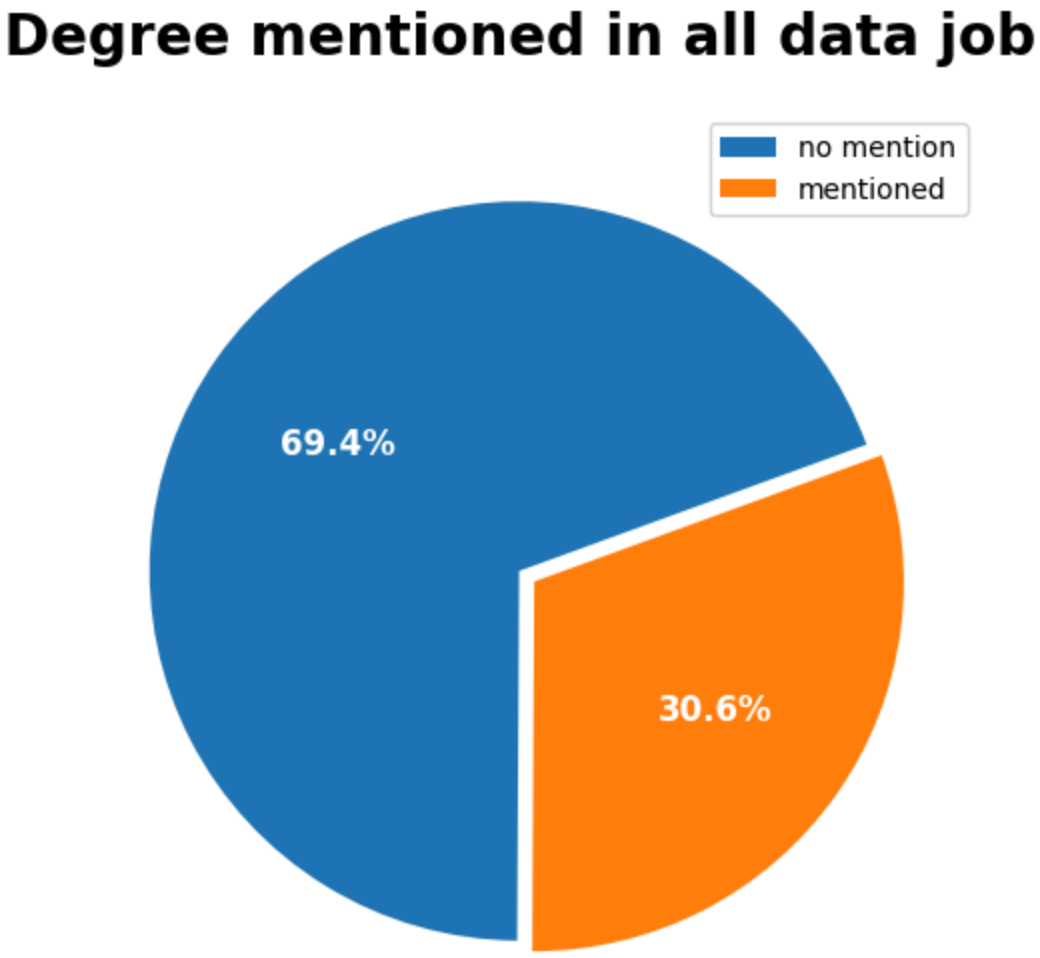
Out[337…
```
['all data',
 'Senior Data Engineer',
 'Data Analyst',
 'Data Engineer',
 'Business Analyst',
 'Data Scientist',
 'Machine Learning Engineer',
 'Senior Data Analyst',
 'Cloud Engineer',
 'Senior Data Scientist',
 'Software Engineer']
```

In [338…
```
degree_title = 'all data' # put any title in the place of "all data"
df_degree = job_title_switcher(degree_title)['Degree required or not'].value_counts
df_degree
```

Out[338…
```
Degree required or not
0    0.693584
1    0.306416
Name: proportion, dtype: float64
```

In [339…
```
plt.pie(
        df_degree,
        labels=['no mention', 'mentioned'],
        autopct=lambda p: f'{p:.1f}%',
        startangle=20,
        explode = [0, .05],
        textprops = custom_pie_params()
        )

plt.title(f'Degree mentioned in {degree_title} job')
plt.legend()
plt.show()
```



Jobs by country

In [340…
```
all_job_titles
```

Out[340…
```
['all data',
 'Senior Data Engineer',
 'Data Analyst',
 'Data Engineer',
 'Business Analyst',
 'Data Scientist',
 'Machine Learning Engineer',
 'Senior Data Analyst',
 'Cloud Engineer',
 'Senior Data Scientist',
 'Software Engineer']
```

In [341…
```
country_title = 'all data' # put any title in the place of "all data"
top10_country_for_role = pd.DataFrame(job_title_switcher(country_title)['Job Countr
top10_country_for_role.columns = ['Country', 'Number of jobs']
top10_country_for_role
```

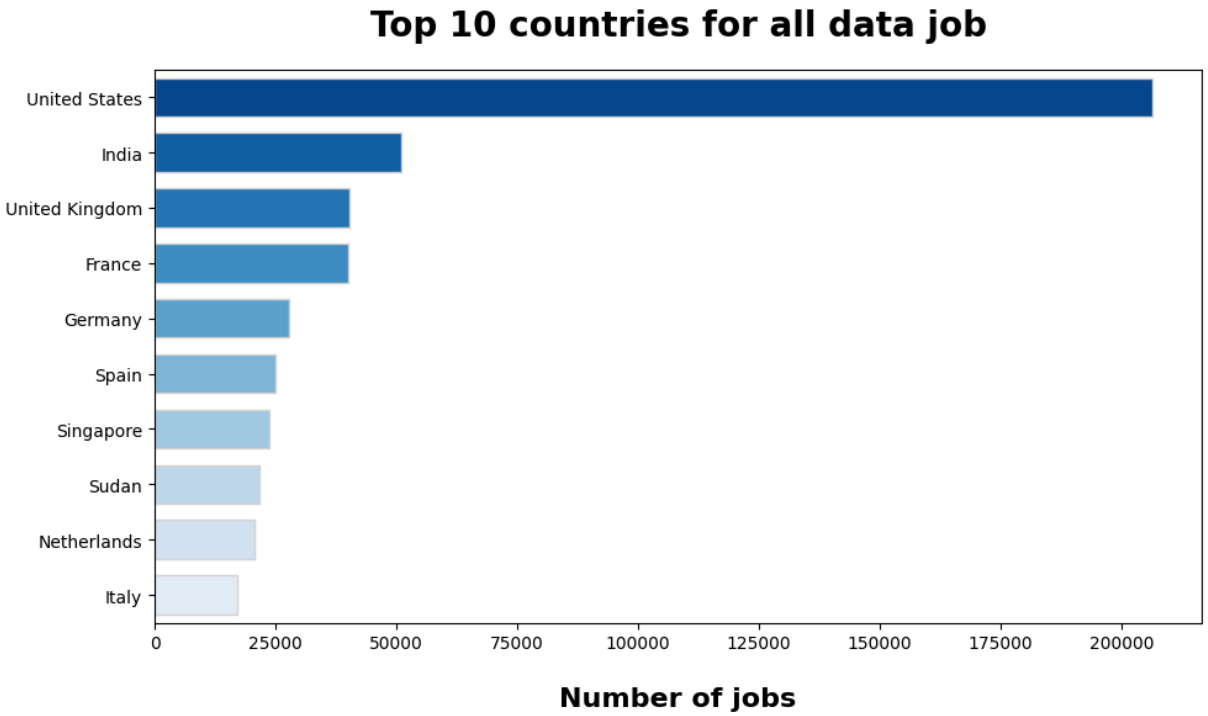
Out[341…

	Country	Number of jobs
0	United States	206287
1	India	51080
2	United Kingdom	40374
3	France	39919
4	Germany	27693
5	Spain	25099
6	Singapore	23693
7	Sudan	21780
8	Netherlands	20629
9	Italy	17012

In [342…
```
sns.barplot(
            data=top10_country_for_role,
            x = 'Number of jobs',
            y = 'Country',
            hue = 'Country',
            **custom_bar_params()
            )

plt.title(f'Top 10 countries for {country_title} job')
plt.xlabel('Number of jobs')
plt.ylabel('')
plt.tight_layout()

plt.show()
```



Monthly job trends

In [343…
```
all_job_titles
```

Out[343…
```
['all data',
 'Senior Data Engineer',
 'Data Analyst',
 'Data Engineer',
 'Business Analyst',
 'Data Scientist',
 'Machine Learning Engineer',
 'Senior Data Analyst',
 'Cloud Engineer',
 'Senior Data Scientist',
 'Software Engineer']
```

In [344…
```
month_title = 'all data' # put any title in the place of "all data"
df_job_trend_month = pd.DataFrame(job_title_switcher(month_title).groupby(by='Job M
df_job_trend_month
```

Out[344...

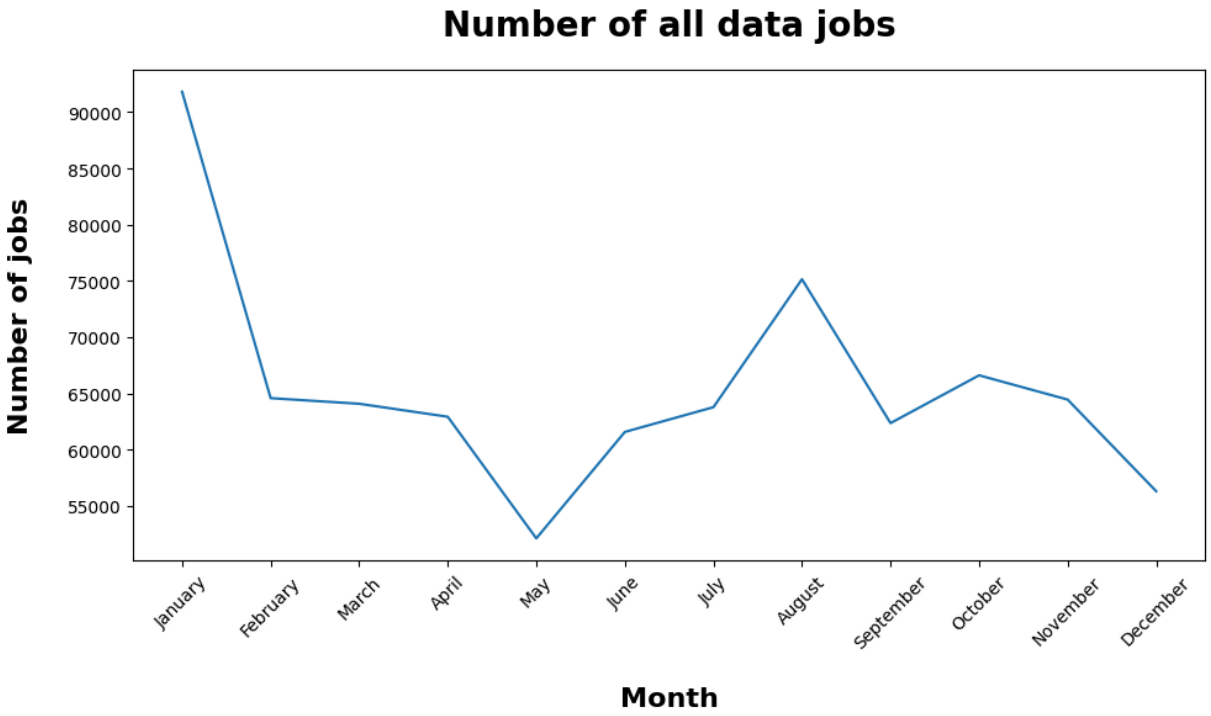
	Job Month	Job title
0	April	62916
1	August	75145
2	December	56292
3	February	64571
4	January	91816
5	July	63767
6	June	61566
7	March	64073
8	May	52101
9	November	64443
10	October	66600
11	September	62350

In [345...

```
months_order = [
    'January', 'February', 'March', 'April', 'May', 'June',
    'July', 'August', 'September', 'October', 'November', 'December'
]
df_job_trend_month['Job Month'] = pd.Categorical(df_job_trend_month['Job Month'], c

sns.lineplot(df_job_trend_month, x='Job Month', y='Job title')
plt.title(f'Number of {month_title} jobs')
plt.xticks(rotation = 45)

plt.xlabel('Month')
plt.ylabel('Number of jobs')
plt.tight_layout()
plt.show()
```



Top 10 most hiring company

In [346…
```
all_job_titles
```

Out[346…
```
['all data',
 'Senior Data Engineer',
 'Data Analyst',
 'Data Engineer',
 'Business Analyst',
 'Data Scientist',
 'Machine Learning Engineer',
 'Senior Data Analyst',
 'Cloud Engineer',
 'Senior Data Scientist',
 'Software Engineer']
```

In [347…
```
company_title = 'all data' # put any title in the place of "all data"
top_10_company_hiring = pd.DataFrame(job_title_switcher(company_title)['Company Nam
                                     .str.strip()
                                     .value_counts()
                                     .head(10)).reset_index()

top_10_company_hiring.columns = ['Company name', 'Number of jobs hired' ]

top_10_company_hiring
```

Out[347…

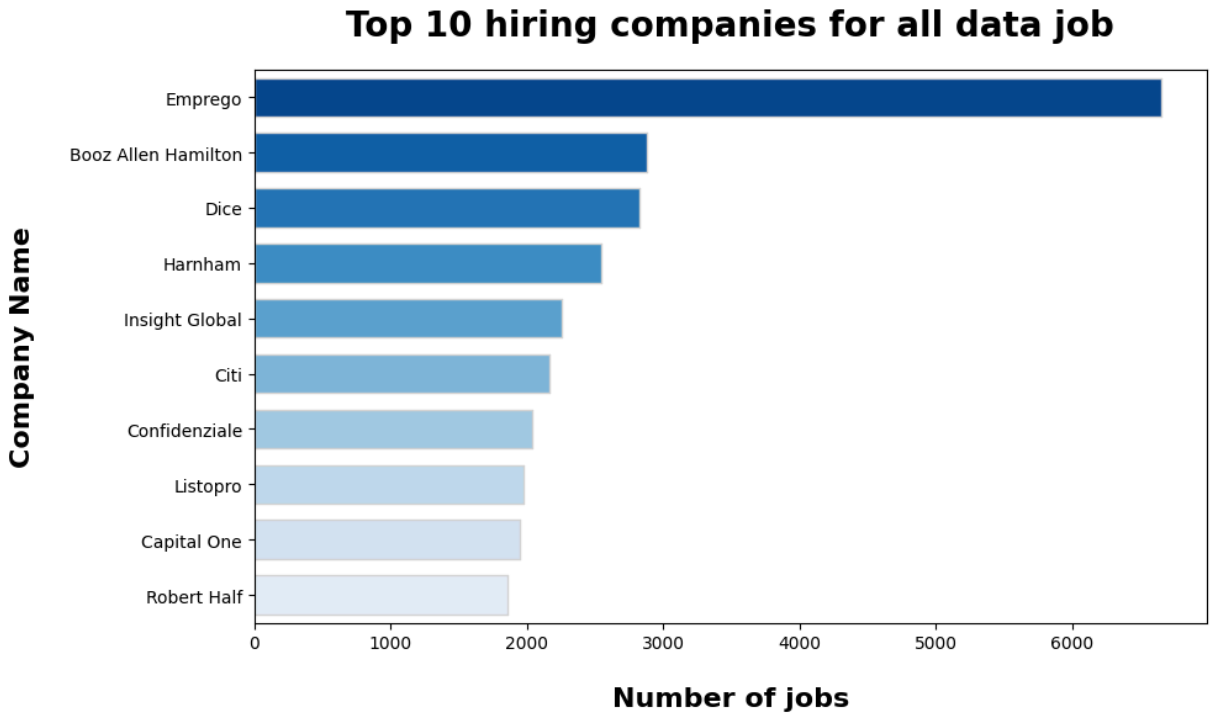
	Company name	Number of jobs hired
0	Emprego	6658
1	Booz Allen Hamilton	2879
2	Dice	2827
3	Harnham	2546
4	Insight Global	2254
5	Citi	2164
6	Confidenziale	2040
7	Listopro	1978
8	Capital One	1946
9	Robert Half	1862

In [348…

```
sns.barplot(
            top_10_company_hiring,
            x='Number of jobs hired',
            y="Company name",
            hue="Company name",
            **custom_bar_params()
            )

plt.title(f'Top 10 hiring companies for {company_title} job')
plt.xlabel('Number of jobs')
plt.ylabel('Company Name')


plt.tight_layout()
plt.show()
```



Top 10 demanding skills in data industry

In [349…
```
all_job_titles
```

Out[349…
```
['all data',
 'Senior Data Engineer',
 'Data Analyst',
 'Data Engineer',
 'Business Analyst',
 'Data Scientist',
 'Machine Learning Engineer',
 'Senior Data Analyst',
 'Cloud Engineer',
 'Senior Data Scientist',
 'Software Engineer']
```

In [350…
```
df.columns
```

Out[350…
```
Index(['Job title', 'Medium', 'Job Schedule Type', 'Remote or On-site',
       'Search Location', 'Job Posted Date', 'Degree required or not',
       'Job Country', 'Salary Year Avg', 'Company Name', 'Job Skills',
       'Job Month'],
      dtype='object')
```

In [351…
```
skill_title = 'all data'
top_10_skills = pd.DataFrame(
                            job_title_switcher(skill_title)['Job Skills'].
                            explode().
                            value_counts().
                            head(10)).reset_index()
```

```
top_10_skills.columns = ['Skill name', 'Number of skill required']
top_10_skills
```

Out[351…

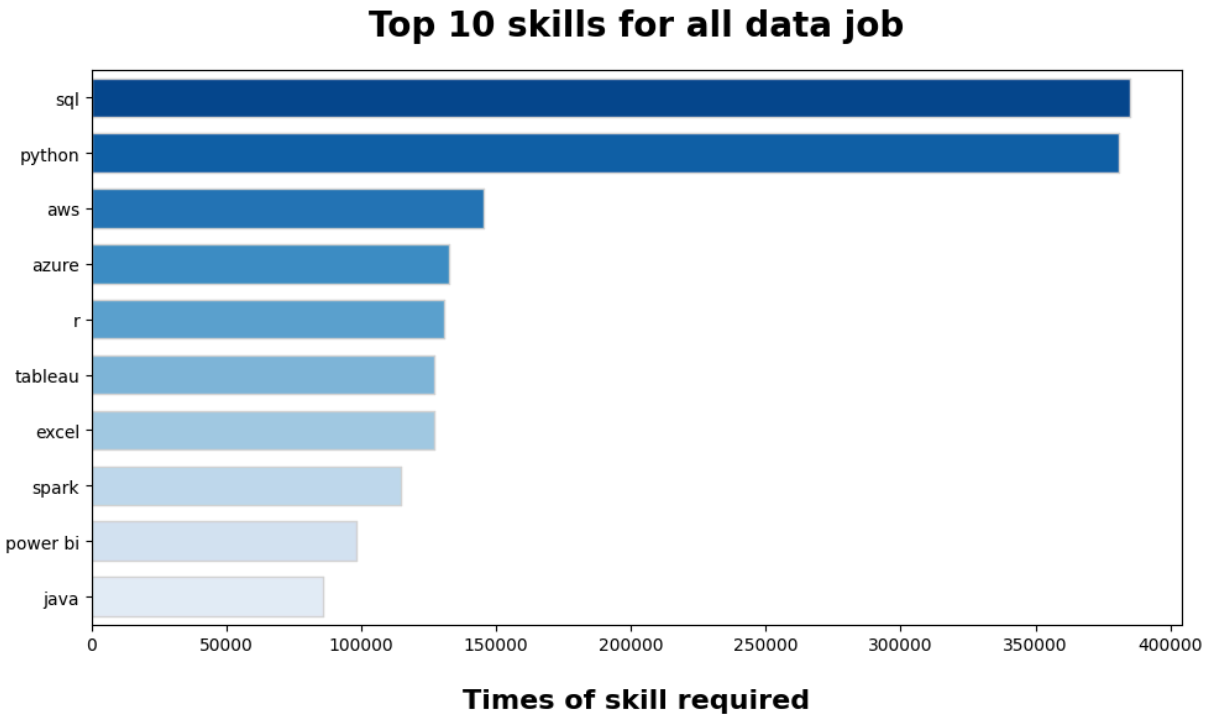
	Skill name	Number of skill required
0	sql	384822
1	python	380883
2	aws	145377
3	azure	132525
4	r	130884
5	tableau	127207
6	excel	127011
7	spark	114604
8	power bi	98141
9	java	85607

In [352…

```
sns.barplot(
            top_10_skills,
            x='Number of skill required',
            y='Skill name',
            hue='Skill name' ,
            **custom_bar_params()
            )

plt.title(f'Top 10 skills for {skill_title} job')
plt.xlabel('Times of skill required')
plt.ylabel('')

plt.tight_layout()
plt.show()
```



Average and median salary by job titles

In [353…

```
mean_salary = (
              df.groupby(by='Job title')['Salary Year Avg']
              .mean()
              .apply(lambda x : int(x))
              .sort_values(ascending = False)
              )

median_salary = (
                df.groupby(by='Job title')['Salary Year Avg']
                .median()
                .apply(lambda x : int(x))
                .sort_values(ascending = False)
                )

salary_df = pd.DataFrame({
    'Job title' : mean_salary.index,
    'Mean salary' : mean_salary.values,
    'Median salary' : median_salary.values
})

salary_df_melted = salary_df.melt(id_vars='Job title', var_name='Type', value_name=

salary_df
```

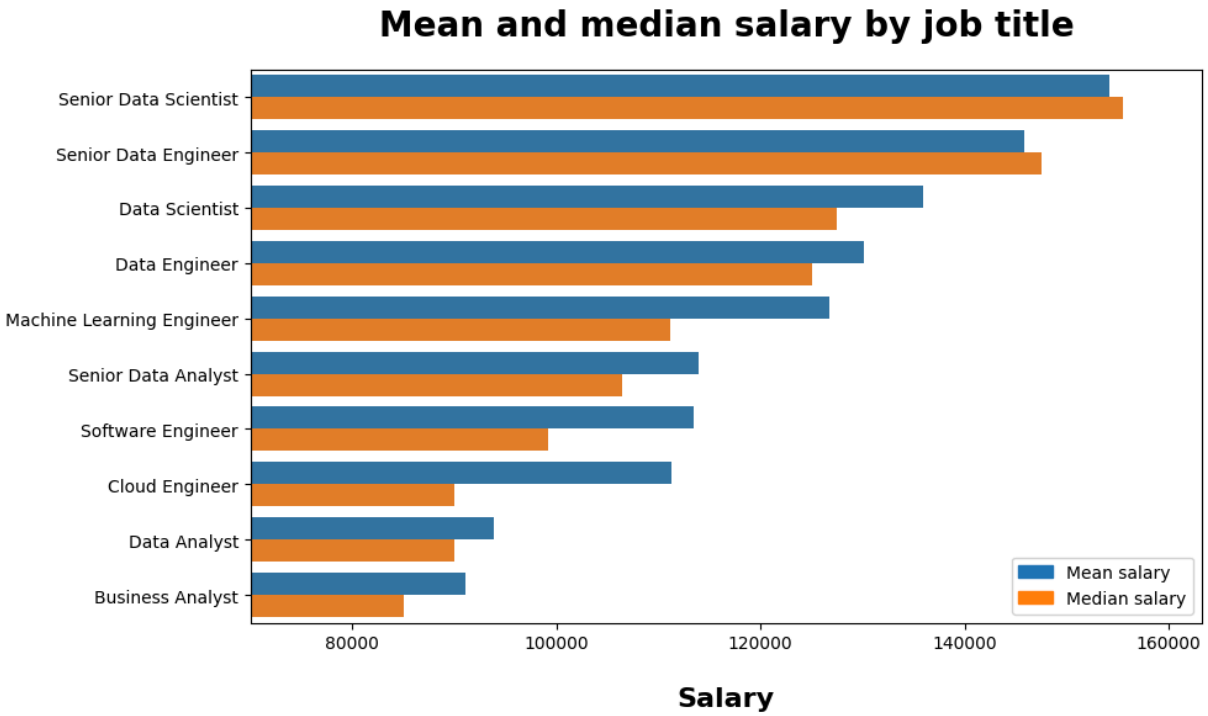
Out[353...

	Job title	Mean salary	Median salary
0	Senior Data Scientist	154206	155500
1	Senior Data Engineer	145840	147500
2	Data Scientist	135988	127500
3	Data Engineer	130125	125000
4	Machine Learning Engineer	126774	111175
5	Senior Data Analyst	113911	106415
6	Software Engineer	113393	99150
7	Cloud Engineer	111268	90000
8	Data Analyst	93842	90000
9	Business Analyst	91082	85000

In [354...

```
sns.barplot(
            salary_df_melted,
            x='Salary',
            y='Job title',
            hue='Type'
            )

plt.title('Mean and median salary by job title')
plt.xlabel('Salary')
plt.ylabel('')
custom_handles = [mpatches.Patch(color=sns.color_palette()[0], label='Mean salary')
                  mpatches.Patch(color=sns.color_palette()[1], label='Median salary')
plt.legend(handles = custom_handles)
plt.xlim(left=70000)
plt.tight_layout()
plt.show()
```



Top country according to salary

In [355…
```
df.columns
```

Out[355…
```
Index(['Job title', 'Medium', 'Job Schedule Type', 'Remote or On-site',
       'Search Location', 'Job Posted Date', 'Degree required or not',
       'Job Country', 'Salary Year Avg', 'Company Name', 'Job Skills',
       'Job Month'],
      dtype='object')
```

In [356…
```
df_salary_country = df[['Job Country', 'Salary Year Avg']].dropna(
    subset='Salary Year Avg')
country_counts = df_salary_country['Job Country'].value_counts()
# at least 50 salary value check
countries_withatleast_50_values = country_counts[country_counts.values > 50].index

filtered_country = df_salary_country[df_salary_country['Job Country'].isin(
    countries_withatleast_50_values)]
df_avg_salary_country = pd.DataFrame(
    filtered_country
    .groupby('Job Country')['Salary Year Avg']
    .mean()
    .sort_values(ascending=False)
    .head(10)
).reset_index()
df_avg_salary_country
```

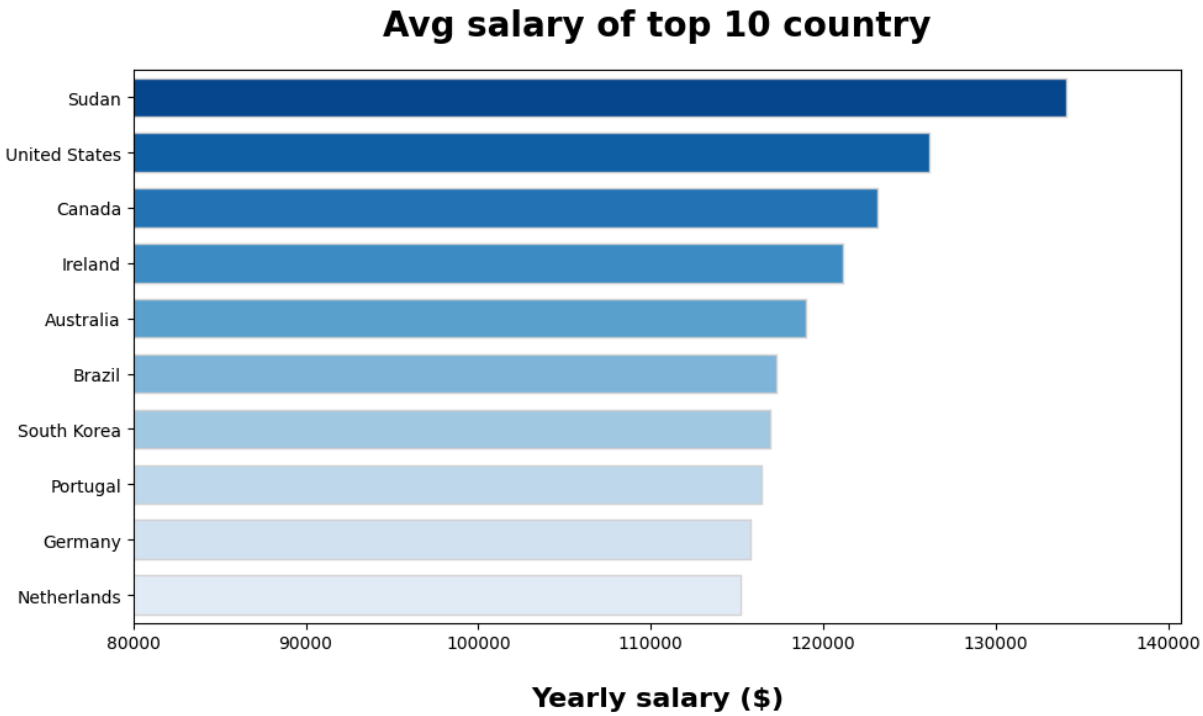
Out[356…

	Job Country	Salary Year Avg
0	Sudan	134051.577942
1	United States	126136.780029
2	Canada	123121.533477
3	Ireland	121133.443396
4	Australia	118987.574324
5	Brazil	117263.352459
6	South Korea	116930.333333
7	Portugal	116437.671756
8	Germany	115800.564202
9	Netherlands	115232.247126

In [357…

```
sns.barplot(
            df_avg_salary_country,
            x='Salary Year Avg',
            y='Job Country',
            hue='Job Country',
            **custom_bar_params()
            )

plt.title('Avg salary of top 10 country')
plt.xlabel('Yearly salary ($)')
plt.ylabel('')
plt.xlim(left = 80000)
plt.tight_layout()
plt.show()
```


Avg salary of top 10 country
Sudan
United States
Canada
Ireland
Australia
Brazil
South Korea
Portugal
Germany
Netherlands
80000
90000
100000
110000
120000
130000
140000
Yearly salary ($)