Data job market analysis

February 20, 2025

1 Import libraries

[1]:
```
import pandas as pd
import numpy as np
import sqlite3
import matplotlib.pyplot as plt
import seaborn as sns
import warnings

warnings.filterwarnings('ignore')
```

2 Loading Dataset

[2]:
```
conn = sqlite3.connect('jobs.db')

query = "SELECT * FROM jobs_cleaned_table WHERE (job_group='Data Scientist' OR␣
 ↪job_group='Data Analyst' OR job_group='Data Engineer')"
df = pd.read_sql_query(query,conn)

df['date_posted'] = pd.to_datetime(df['date_posted'].str.split().str[0],␣
 ↪errors='coerce').dt.strftime('%Y-%m-%d')

conn.close()
```

3 EDA

[3]:
```
df.head()
```

[3]:
```
                      id    site  \
0  in-9936bd8d30f8a34d  indeed
1  in-65c826860da559a3  indeed
2  in-0123143eb77645f8  indeed
3  in-679b46dfdbe7b4ea  indeed
4  in-506a48e047b0e57c  indeed

                                            job_url  \
0  https://www.indeed.com/viewjob?jk=9936bd8d30f8…
1  https://www.indeed.com/viewjob?jk=65c826860da5…
2  https://www.indeed.com/viewjob?jk=0123143eb776…
3  https://www.indeed.com/viewjob?jk=679b46dfdbe7…
4  https://www.indeed.com/viewjob?jk=506a48e047b0…

                                      job_url_direct  \
0               https://jobs.vccs.edu/postings/79840
1  https://jobs.colgate.com/job/Piscataway-Data-A…
2                        https://grnh.se/0be7ee141us
3                        https://grnh.se/de2a9b121us
4  http://www.indeed.com/job/data-analyst-employe…

                               title                            company  \
0                       Data Analyst  Virginia Community College System
1          Data Analytics Internship                  Colgate-Palmolive
2             Financial Data Analyst                     EquipmentShare
3  Data Analyst, Customer Operations                        Squarespace
4   Data Analyst (Employee Benefits)                       GBS Benefits

  date_posted      level     job_group  remote  … country  \
0  2024-12-11  Mid-Level  Data Analyst  On Site  …      US
1  2024-12-11     Junior  Data Analyst   Hybrid  …      US
2  2024-12-11  Mid-Level  Data Analyst   Remote  …      US
3  2024-12-11  Mid-Level  Data Analyst   Hybrid  …      US
4  2024-12-11  Mid-Level  Data Analyst   Hybrid  …      US

           city_state max_salary min_salary  mean_salary  \
0     Chesterfield,VA    78000.0    61000.0      69500.0
1       Piscataway,NJ    58880.0    42320.0      50600.0
2         Columbia,MO    86876.0    68611.0      77743.5
3         New York,NY   138000.0    85500.0     111750.0
4  South Salt Lake,UT    74386.0    58746.0      66566.0

                                        skills  experience education  \
0                                Bachelor, SQL           2  Bachelor
1                Master, Bachelor, SQL, Python           0    Master
2  Python, Bachelor, SQL, PowerPoint, Excel, R           3  Bachelor
3                       Python, SQL, Looker, R           2      None
4                              Bachelor, Excel           0  Bachelor

   programming_languages languages
0                    SQL
1            SQL, Python
2         Python, SQL, R
3         Python, SQL, R
4
```

```
[5 rows x 22 columns]
```

[4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1679 entries, 0 to 1678
Data columns (total 22 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   id                     1679 non-null   object
 1   site                   1679 non-null   object
 2   job_url                1679 non-null   object
 3   job_url_direct         1679 non-null   object
 4   title                  1679 non-null   object
 5   company                1653 non-null   object
 6   date_posted            1679 non-null   object
 7   level                  1679 non-null   object
 8   job_group              1679 non-null   object
 9   remote                 1679 non-null   object
 10  city                   1449 non-null   object
 11  state                  1516 non-null   object
 12  country                1648 non-null   object
 13  city_state             1449 non-null   object
 14  max_salary             1531 non-null   float64
 15  min_salary             1531 non-null   float64
 16  mean_salary            1531 non-null   float64
 17  skills                 1679 non-null   object
 18  experience             1679 non-null   int64
 19  education              1150 non-null   object
 20  programming_languages  1679 non-null   object
 21  languages              1679 non-null   object
dtypes: float64(3), int64(1), object(18)
memory usage: 288.7+ KB
```

[5]:
```
df.describe(include='all')
```

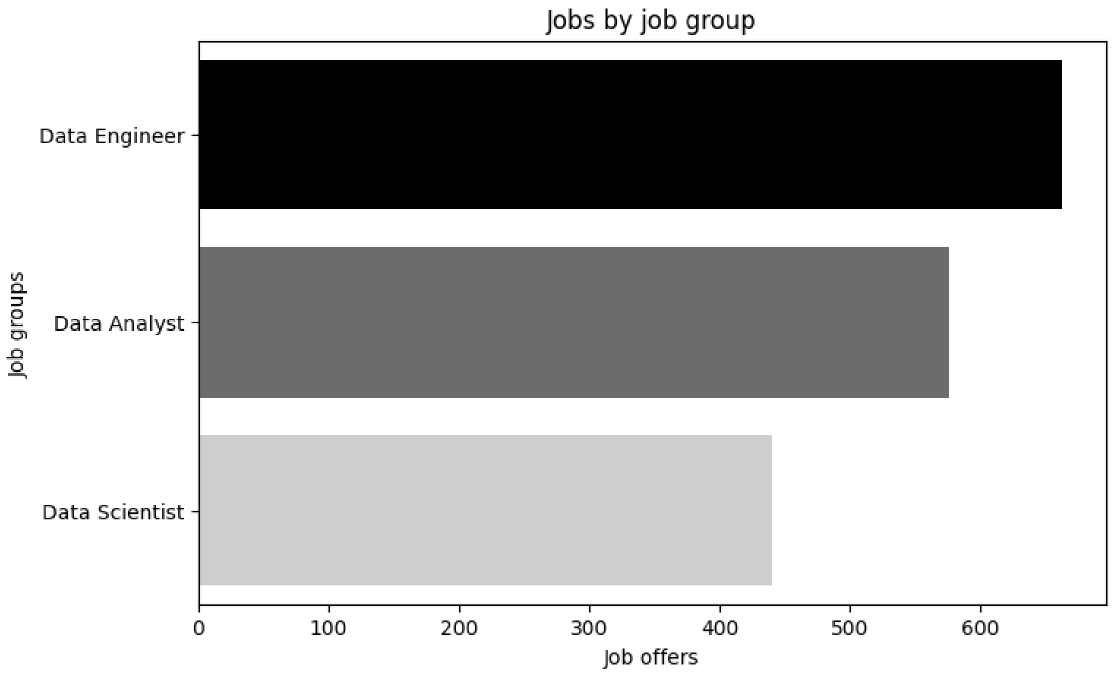
[5]:
```
                       id    site  \
count                1679    1679
unique               1679       1
top     in-9936bd8d30f8a34d  indeed
freq                    1    1679
mean                  NaN     NaN
std                   NaN     NaN
min                   NaN     NaN
25%                   NaN     NaN
50%                   NaN     NaN
75%                   NaN     NaN
```

```
max                     NaN     NaN

                                                  job_url  \
count                                                1679
unique                                               1679
top     https://www.indeed.com/viewjob?jk=9936bd8d30f8…
freq                                                    1
mean                                                  NaN
std                                                   NaN
min                                                   NaN
25%                                                   NaN
50%                                                   NaN
75%                                                   NaN
max                                                   NaN

                                      job_url_direct          title  \
count                                           1679           1679
unique                                          1599           1068
top     https://intonenetworks.com/careers-at-intone  Data Engineer
freq                                              14            135
mean                                             NaN            NaN
std                                              NaN            NaN
min                                              NaN            NaN
25%                                              NaN            NaN
50%                                              NaN            NaN
75%                                              NaN            NaN
max                                              NaN            NaN

           company date_posted      level      job_group   remote  …  \
count         1653        1679       1679           1679     1679  …
unique        1025          71          4              3        3  …
top     Amazon.com  2024-12-10  Mid-Level  Data Engineer  On Site  …
freq            94         195        989            663      908  …
mean           NaN         NaN        NaN            NaN      NaN  …
std            NaN         NaN        NaN            NaN      NaN  …
min            NaN         NaN        NaN            NaN      NaN  …
25%            NaN         NaN        NaN            NaN      NaN  …
50%            NaN         NaN        NaN            NaN      NaN  …
75%            NaN         NaN        NaN            NaN      NaN  …
max            NaN         NaN        NaN            NaN      NaN  …

       country   city_state     max_salary     min_salary    mean_salary  \
count     1648         1449    1531.000000    1531.000000    1531.000000
unique       1          453            NaN            NaN            NaN
top         US  New York,NY            NaN            NaN            NaN
freq      1648           89            NaN            NaN            NaN
mean       NaN          NaN  146602.506858  104676.729589  125639.618223
```

```
std        NaN          NaN   56354.458212   33643.605431   42621.890718
min        NaN          NaN   19159.000000   10388.000000   14773.500000
25%        NaN          NaN  110400.000000   81874.500000   99784.250000
50%        NaN          NaN  139586.000000  102317.000000  122931.500000
75%        NaN          NaN  175900.000000  124404.500000  150150.750000
max        NaN          NaN  720000.000000  260100.000000  445000.000000

          skills   experience education  programming_languages languages
count       1679  1679.000000      1150                   1679      1679
unique      1225          NaN         4                    122        12
top     Bachelor          NaN  Bachelor
freq          53          NaN       633                    404      1610
mean         NaN     3.945801       NaN                    NaN       NaN
std          NaN     3.373463       NaN                    NaN       NaN
min          NaN     0.000000       NaN                    NaN       NaN
25%          NaN     1.000000       NaN                    NaN       NaN
50%          NaN     3.000000       NaN                    NaN       NaN
75%          NaN     5.000000       NaN                    NaN       NaN
max          NaN    20.000000       NaN                    NaN       NaN

[11 rows x 22 columns]
```

[6]:

```python
job_groups_counts = df['job_group'].value_counts()

def␣
 ↪plot_bars(x,y,figsize=(8,5),title=None,xlabel=(None),ylabel=(None),palette=None):
 ↪

    plt.figure(figsize=figsize)
    cmap = plt.cm.get_cmap(palette).reversed()
    palette = [cmap(x) for x in np.linspace(0,0.7,len(job_groups_counts))]

    sns.barplot(x=x,y=y,palette=palette)

    plt.title(title)
    plt.xlabel(xlabel)
    plt.ylabel(ylabel)
    plt.show()

plot_bars(x=job_groups_counts.values,y=job_groups_counts.index,title='Jobs by␣
 ↪job group',xlabel='Job offers',ylabel='Job groups',palette='Greys')
```



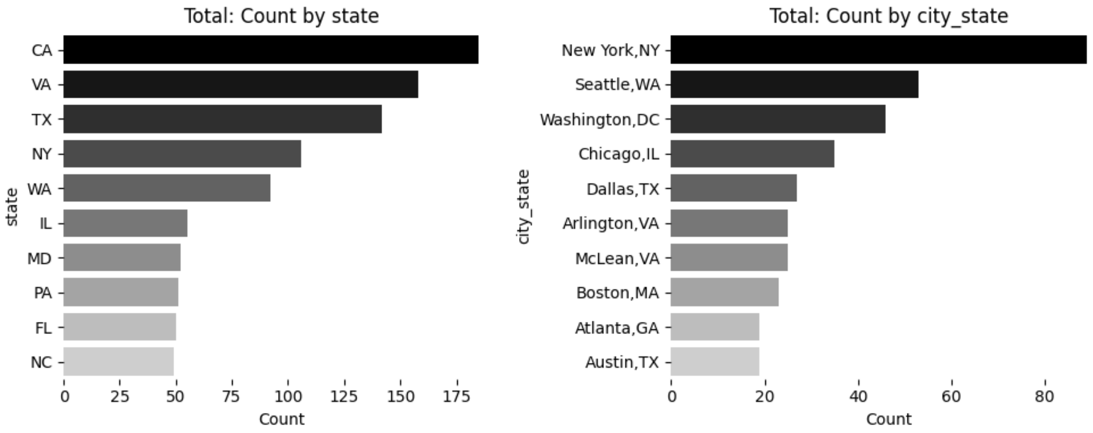
4 Analysis

4.1 Location

```
[36]: def␣
       ↪plot_multiple_bars(columns,category=None,figsize=(10,12),count_percentage=False,x_y=False,t
       ↪

          rows = df[category].unique() if category is not None else ['Total']
          n_rows, n_cols = len(rows), len(columns)

          fig, axes = plt.subplots(n_rows,n_cols,figsize=figsize,squeeze=False)

          category_palettes = {
              'Data Analyst': 'Blues',
              'Data Engineer': 'Reds',
              'Data Scientist': 'Greens'
          }

          for i, row in enumerate(rows):
              group_data = df[df[category] == row] if category is not None else df

              for j, column in enumerate(columns):
```

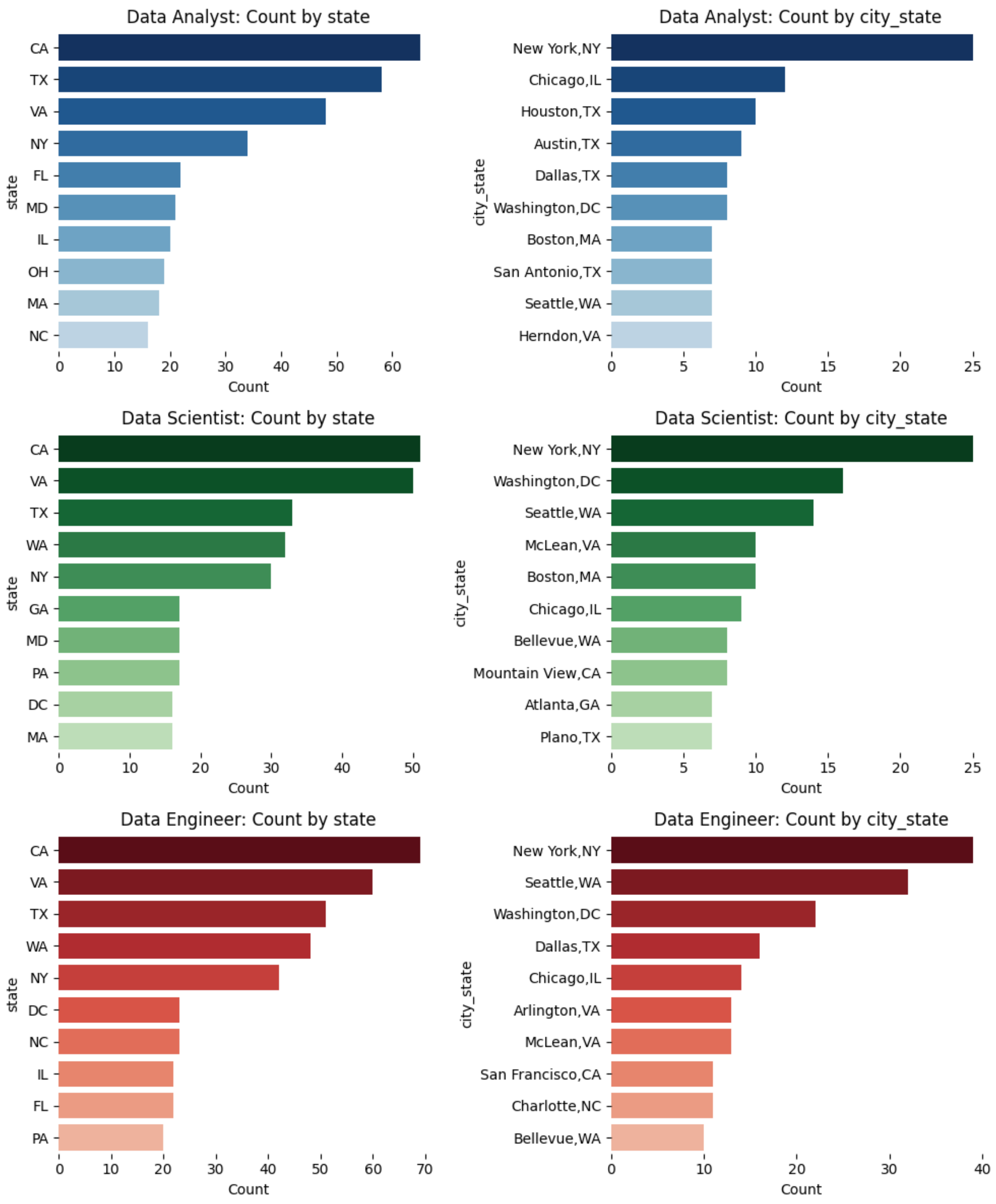
```
            if aggregate_column is None:
                if top_10 and list_values is False:
                    column_top = group_data[column].value_counts().head(10)

                elif list_values is not False and top_10 is None:
                    column_top=(group_data[column].str.split(', ').explode()
                                .value_counts()
                                .dropna()
                                .sort_values(ascending=False))
                    column_top = column_top.drop(labels='',errors='ignore')

                elif list_values is not False and top_10 is not None:
                    column_top=(group_data[column].str.split(', ').explode()
                                .value_counts()
                                .dropna()
                                .sort_values(ascending=False).head(10))
                    column_top = column_top.drop(labels='',errors='ignore')

                else:
                    column_top = group_data[column].value_counts()

            else:
                grouped =(
                    group_data.explode(column)
                    .groupby(column)[aggregate_column]
                    .mean()
                    .sort_values(ascending=False)
                )

                if list_values:
                    group_data = group_data.reset_index(drop=True)
                    grouped = pd.DataFrame({
                        column: group_data[column].str.split(', ').explode(),
                        aggregate_column: group_data[aggregate_column]
                    })
                    grouped = grouped.groupby(column)[aggregate_column].mean().
↪sort_values(ascending=False)

                if '' in grouped.index:
                    grouped = grouped.drop('')

                column_top = grouped.head(10) if top_10 else grouped

            ax = axes[i,j]
            total = group_data[column].count()

            for spine in ax.spines.values():
                spine.set_visible(False)

            if category is not None:
                palette = category_palettes.get(row,'Blues')
                cmap = plt.cm.get_cmap(palette).reversed()
                palette = [cmap(x) for x in np.linspace(0,0.7,len(column_top))]

            else:
                cmap = plt.cm.get_cmap('Greys').reversed()
                palette = [cmap(x) for x in np.linspace(0,0.7,len(column_top))]

            kwargs ={
                'y':column_top.values,
                'x':column_top.index
            } if x_y else {
                'x':column_top.values,
                'y':column_top.index
            }

            sns.barplot(ax=ax,palette=palette, **kwargs)

            aggregate_label = 'Count' if aggregate_column is None else f'Avg{aggregate_column}'
            ax.set_title(f'{row}: {aggregate_label} by {column}')
            ax.set_xlabel(aggregate_label if not x_y else column)
            ax.set_ylabel(column if not x_y else aggregate_label)

            if count_percentage:
                for k, value in enumerate(column_top.values):
                    percentage = value/total*100
                    annotation_kwargs = {
                        'text': f'{value:.1f}({percentage:.1f}%)',
                        'xy':(k, value) if x_y else (value,k),
                        'xytext': (-30,5) if x_y else (5,0),
                        'textcoords': 'offset points',
                        'va':'center',
                        'ha':'left',
                        'fontsize':10,
                        'color':'black'
                    }
                    ax.annotate(**annotation_kwargs)

    plt.tight_layout()
    plt.show()

columns_location = ['state','city_state']
```

```
plot_multiple_bars(columns_location,figsize=(10,4),top_10=True)
```



[8]:

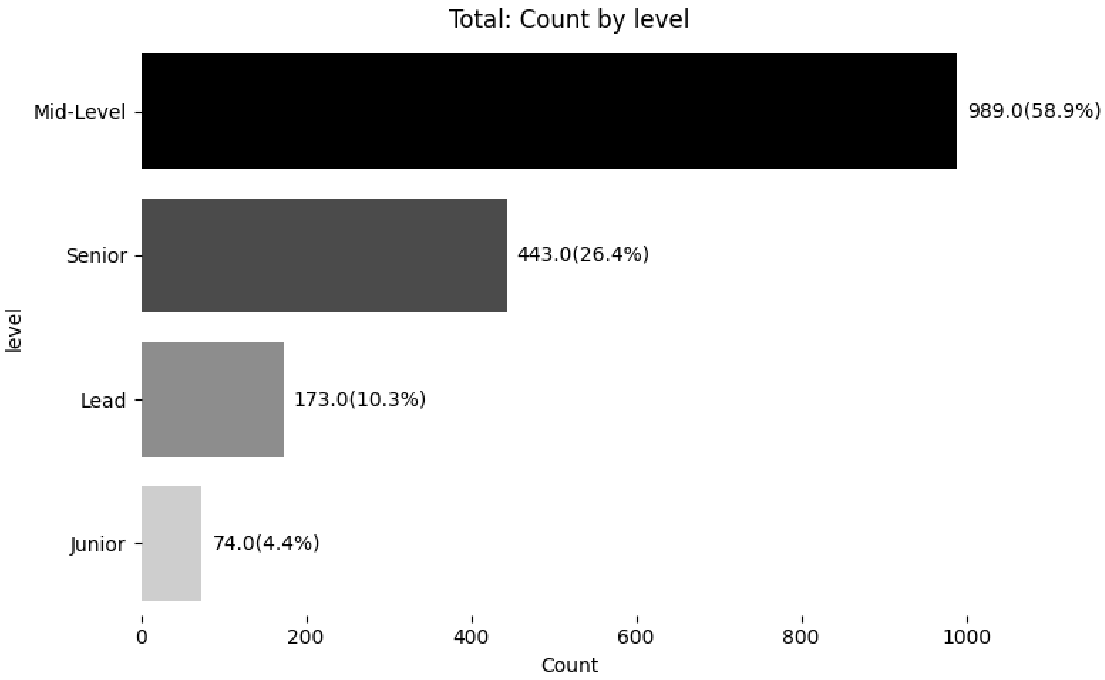
```
plot_multiple_bars(columns_location,category='job_group',figsize=(10,12),top_10=True)
```



4.2 Level

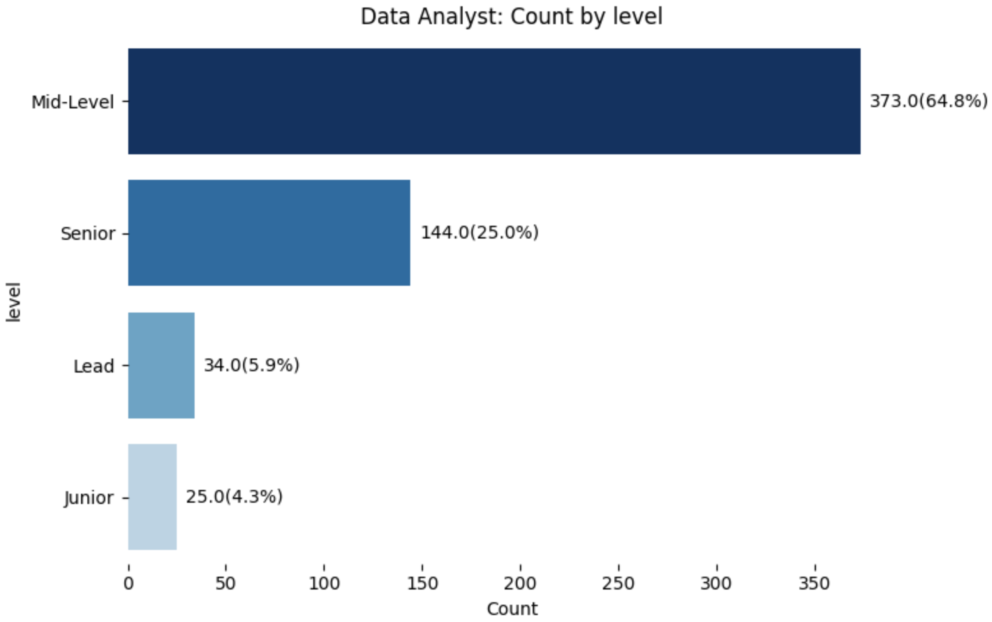
[13]:

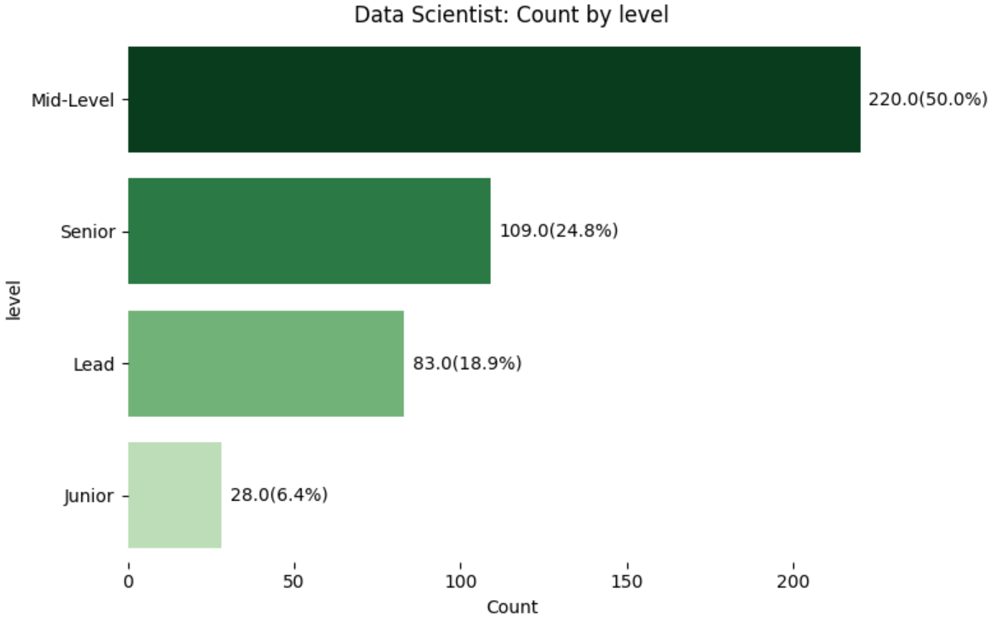
```
columns_level = ('level',)

plot_multiple_bars(columns_level,figsize=(8,5),count_percentage=True)
```

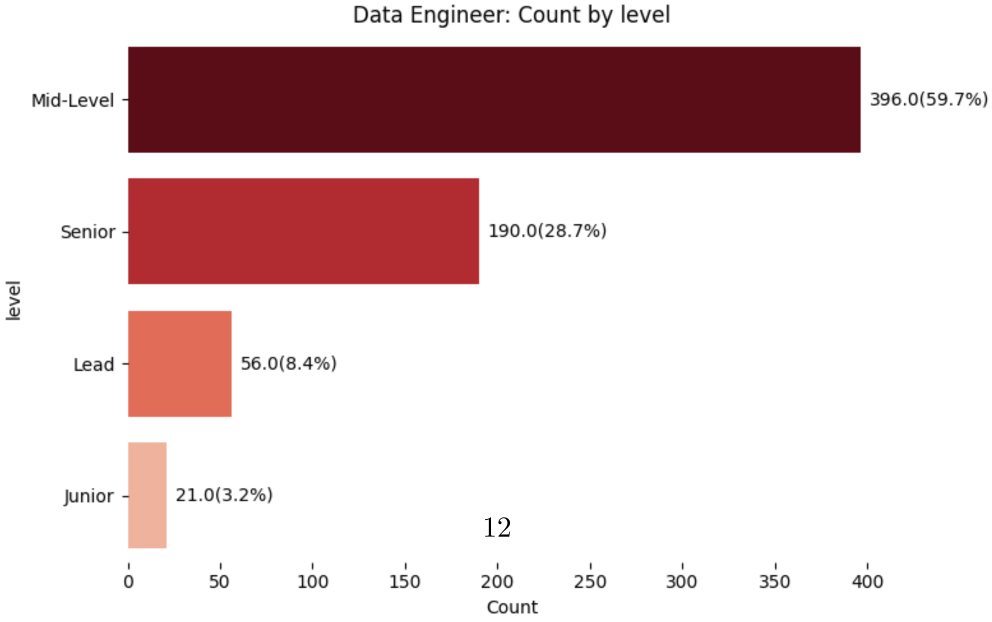


[14]:

```
plot_multiple_bars(columns_level,category='job_group',figsize=(8,15),count_percentage=True)
```

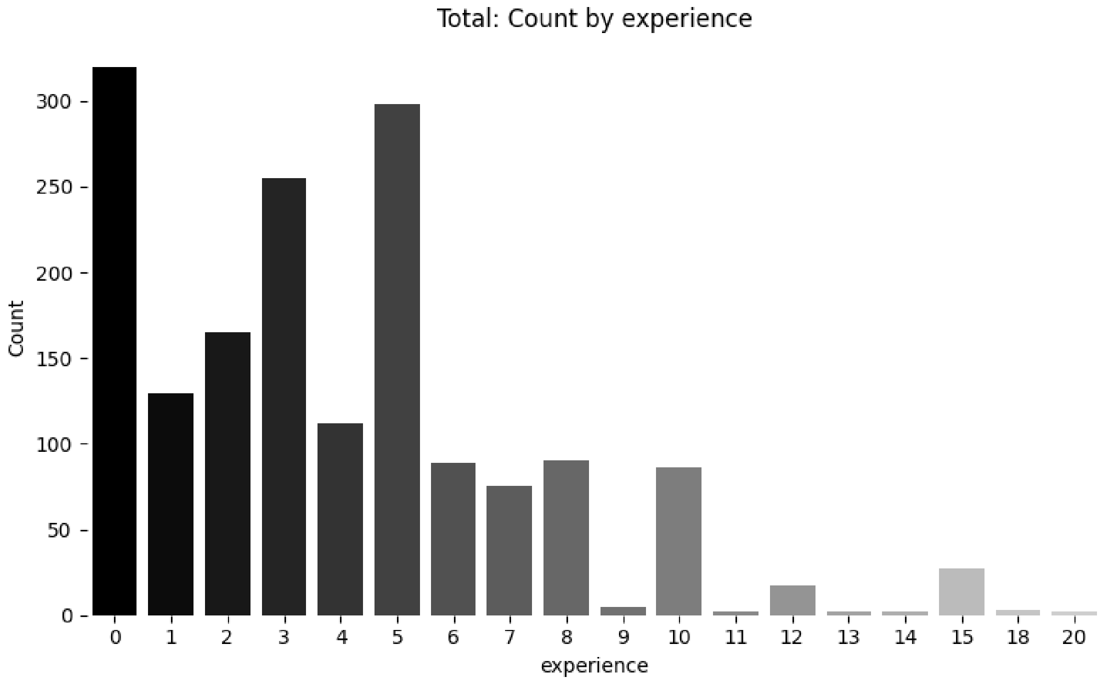

Data Analyst: Count by level
Mid-Level
Senior
Lead
Junior
level
373.0(64.8%)
144.0(25.0%)
34.0(5.9%)
25.0(4.3%)
0
50
100
150
200
250
300
350
Count


Data Scientist: Count by level
Mid-Level
Senior
Lead
Junior
level
220.0(50.0%)
109.0(24.8%)
83.0(18.9%)
28.0(6.4%)
0
50
100
150
200
Count

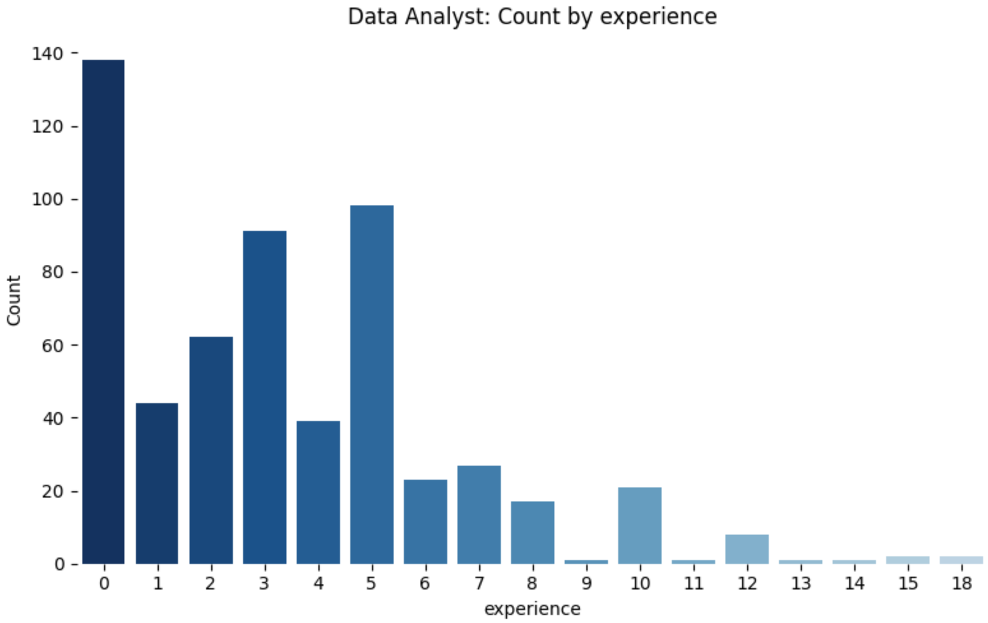

Data Engineer: Count by level
Mid-Level
Senior
Lead
Junior
level
396.0(59.7%)
190.0(28.7%)
56.0(8.4%)
21.0(3.2%)
0
50
100
150
200
250
300
350
400
Count

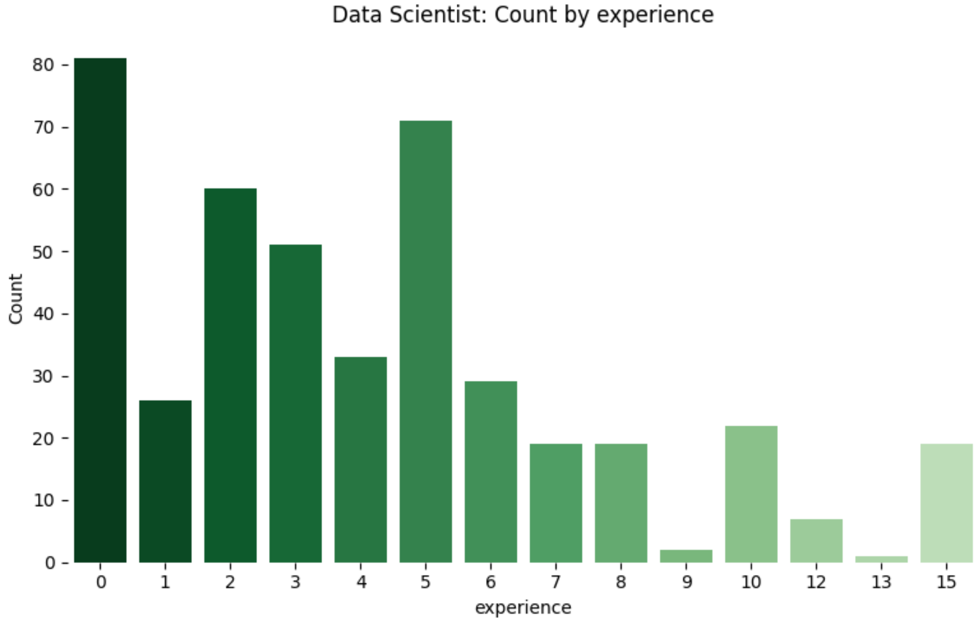
4.3 Experience

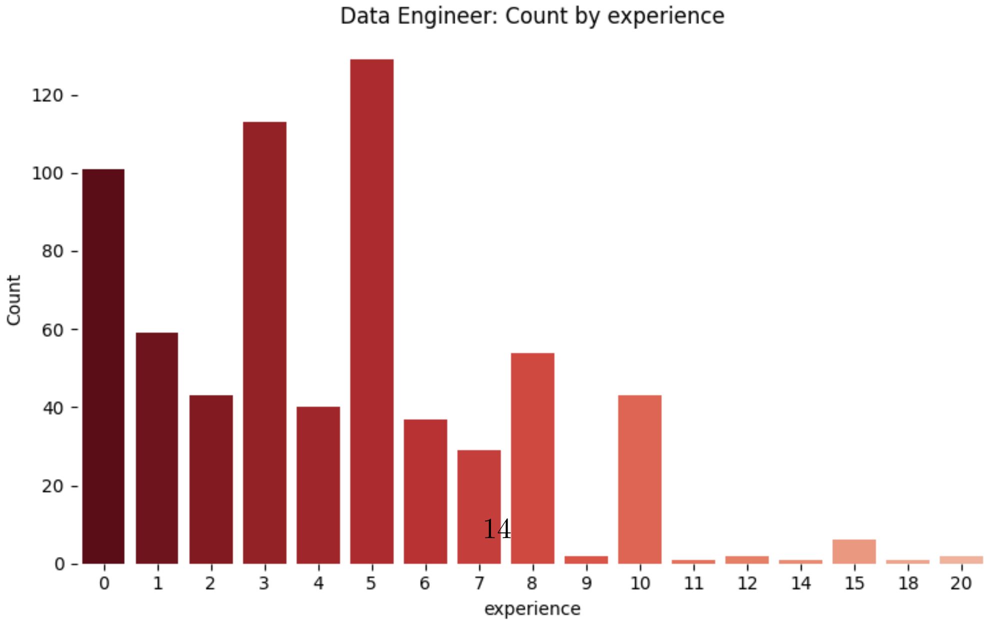
[15]:
```
columns_experience = ('experience',)

plot_multiple_bars(columns_experience,figsize=(8,5),x_y=True)
```



[16]:
```
plot_multiple_bars(columns_experience,category='job_group',figsize=(8,15),x_y=True)
```

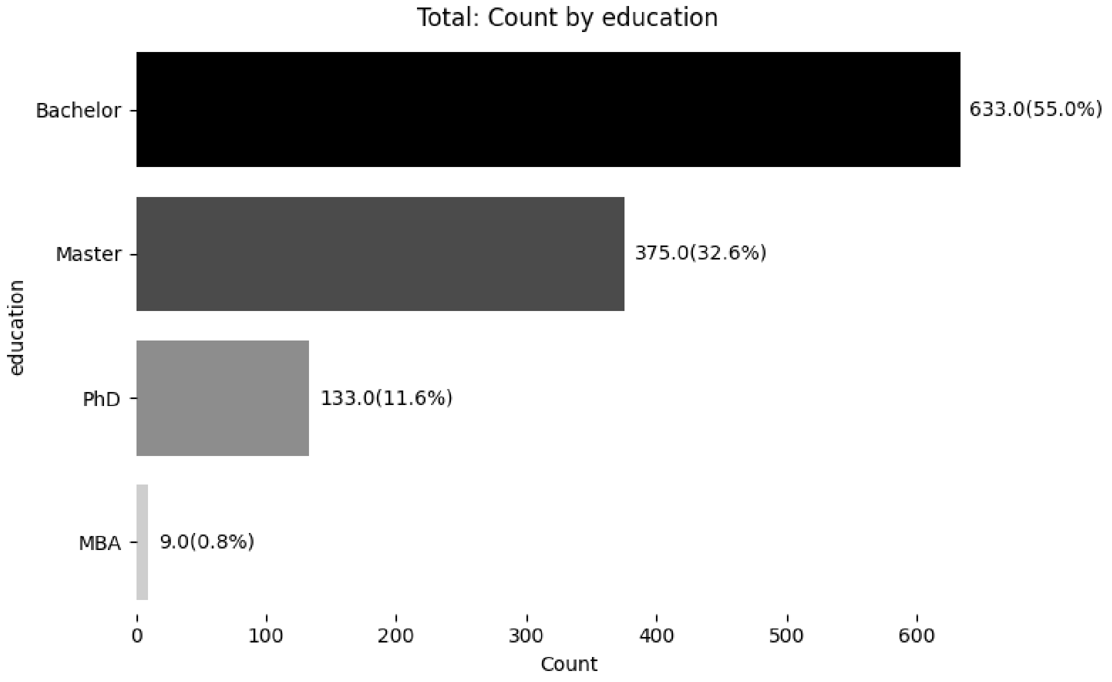

Data Analyst: Count by experience
Count
0
20
40
60
80
100
120
140
experience
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
18


Data Scientist: Count by experience
Count
0
10
20
30
40
50
60
70
80
experience
0
1
2
3
4
5
6
7
8
9
10
12
13
15

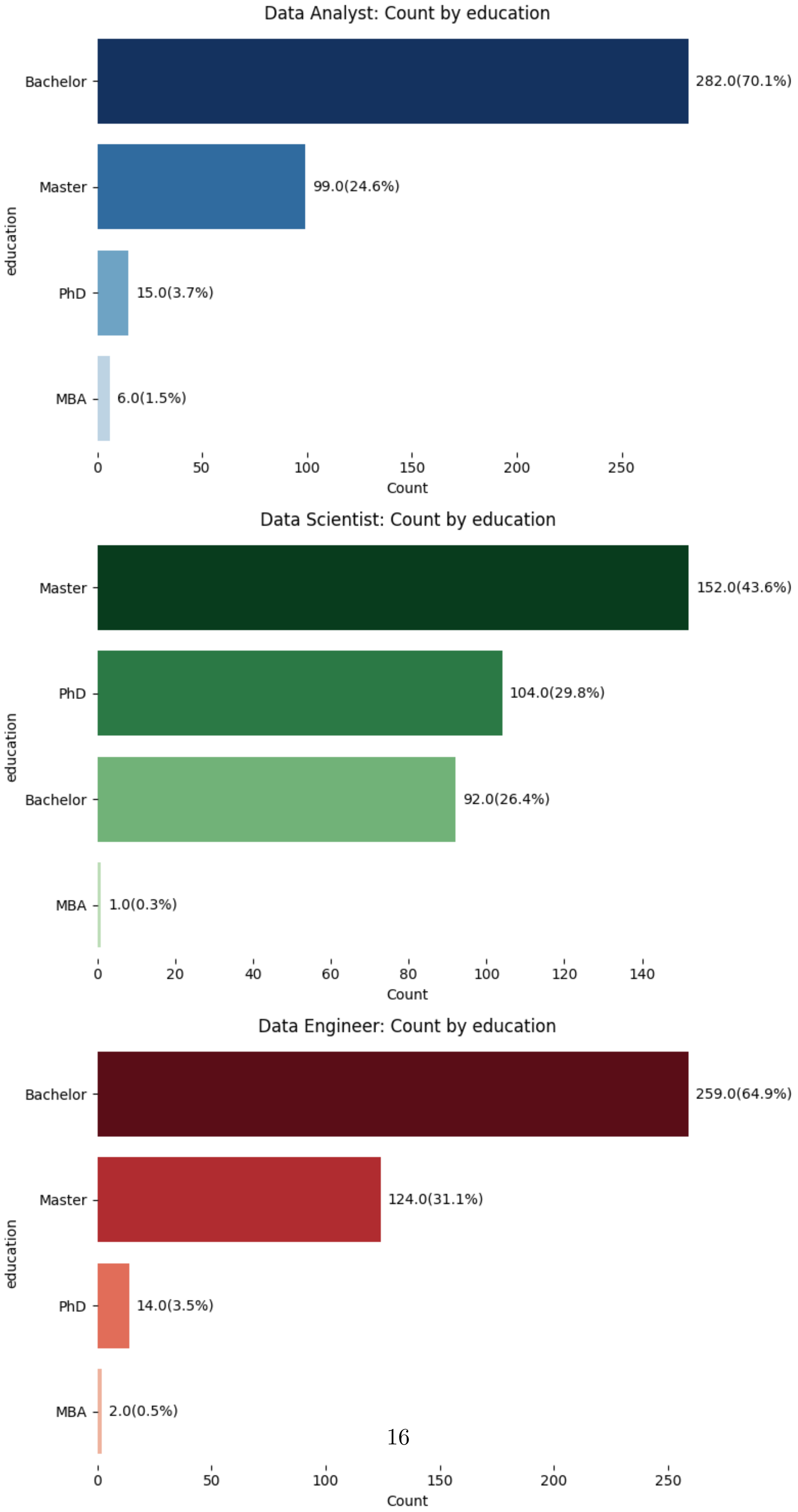

Data Engineer: Count by experience
Count
0
20
40
60
80
100
120
experience
0
1
2
3
4
5
6
7
8
9
10
11
12
14
15
18
20

4.4 Education

```
[17]: columns_education = ('education',)

      plot_multiple_bars(columns_education,figsize=(8,5),count_percentage=True)
```

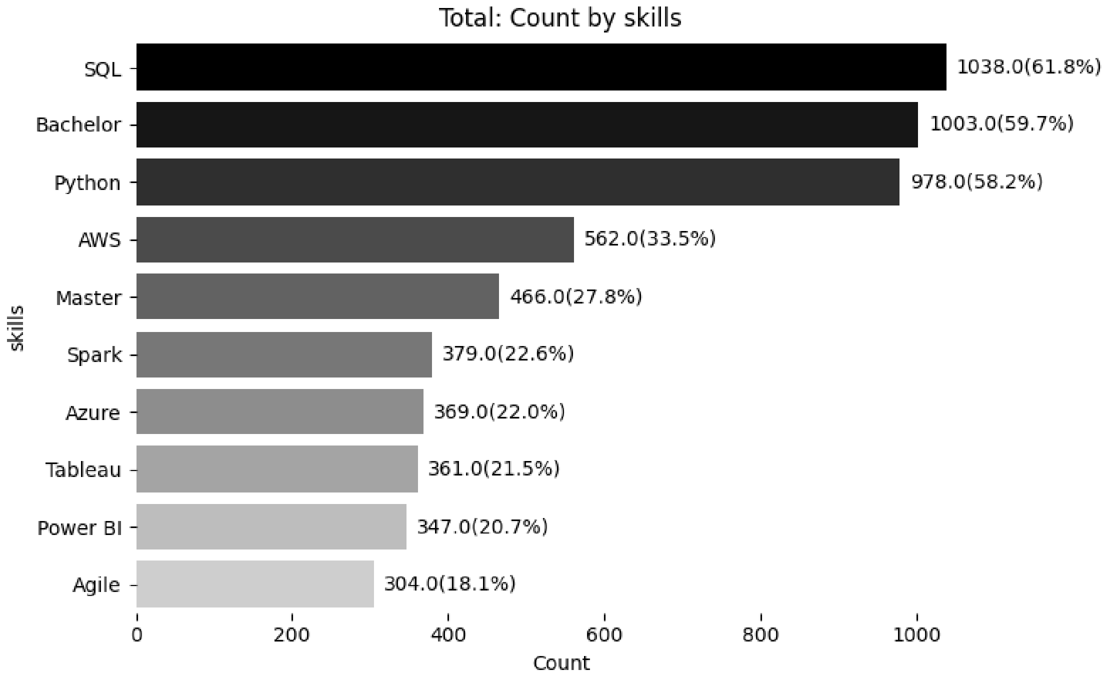


```
[18]: plot_multiple_bars(columns_education,category='job_group',figsize=(8,15),count_percentage=True
```

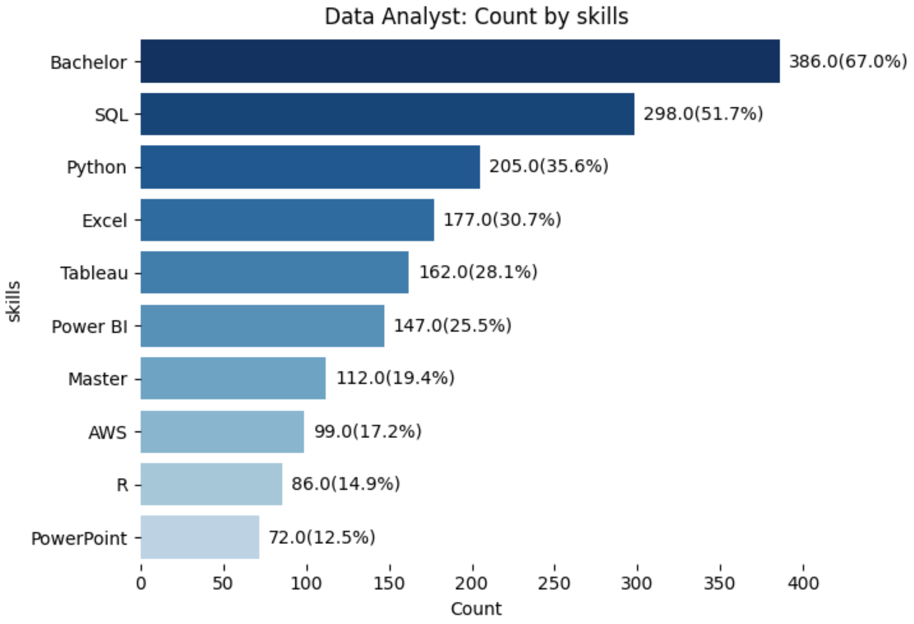

Data Analyst: Count by education
education
Bachelor
Master
PhD
MBA
282.0(70.1%)
99.0(24.6%)
15.0(3.7%)
6.0(1.5%)
0
50
100
150
200
250
Count
Data Scientist: Count by education
education
Master
PhD
Bachelor
MBA
152.0(43.6%)
104.0(29.8%)
92.0(26.4%)
1.0(0.3%)
0
20
40
60
80
100
120
140
Count
Data Engineer: Count by education
education
Bachelor
Master
PhD
MBA
259.0(64.9%)
124.0(31.1%)
14.0(3.5%)
2.0(0.5%)
0
50
100
150
200
250
Count

4.5 Skills

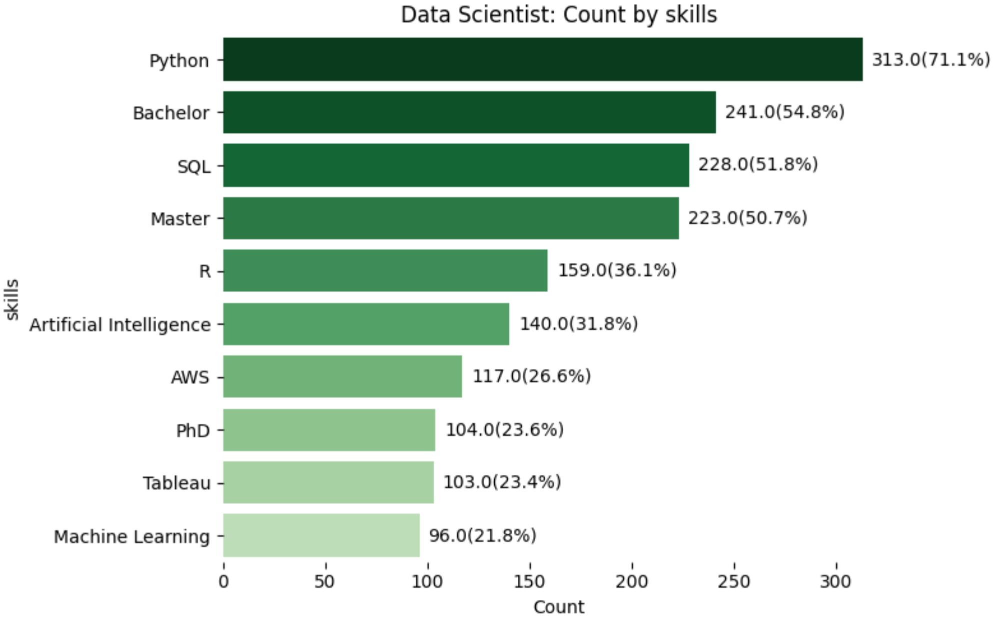
```
[19]: columns_skills = ['skills',]

plot_multiple_bars(columns_skills,figsize=(8,5),list_values=True,count_percentage=True)
```



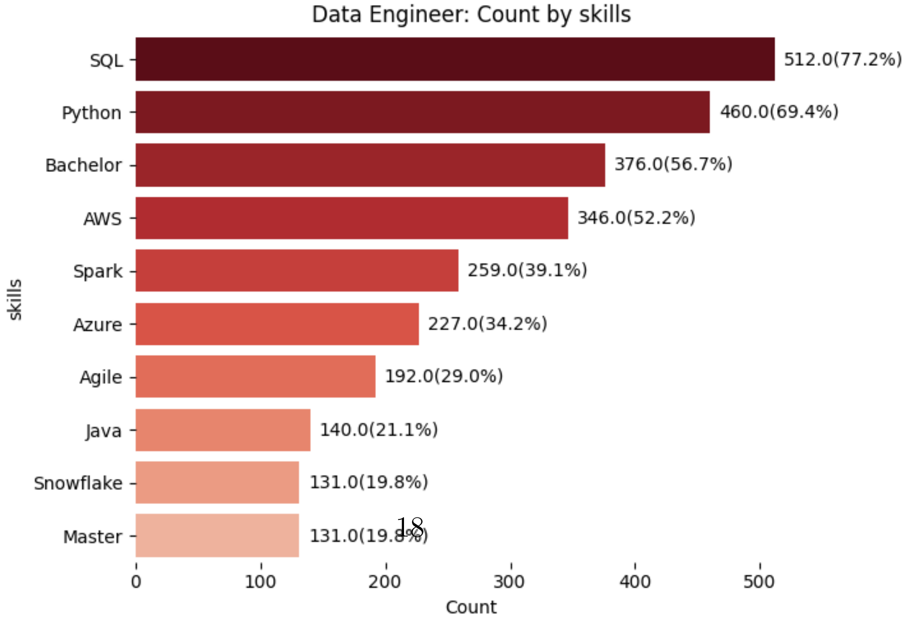
```
[20]: plot_multiple_bars(columns_skills,category='job_group',figsize=(8,15),list_values=True,count_p
```



Data Analyst: Count by skills



Data Scientist: Count by skills

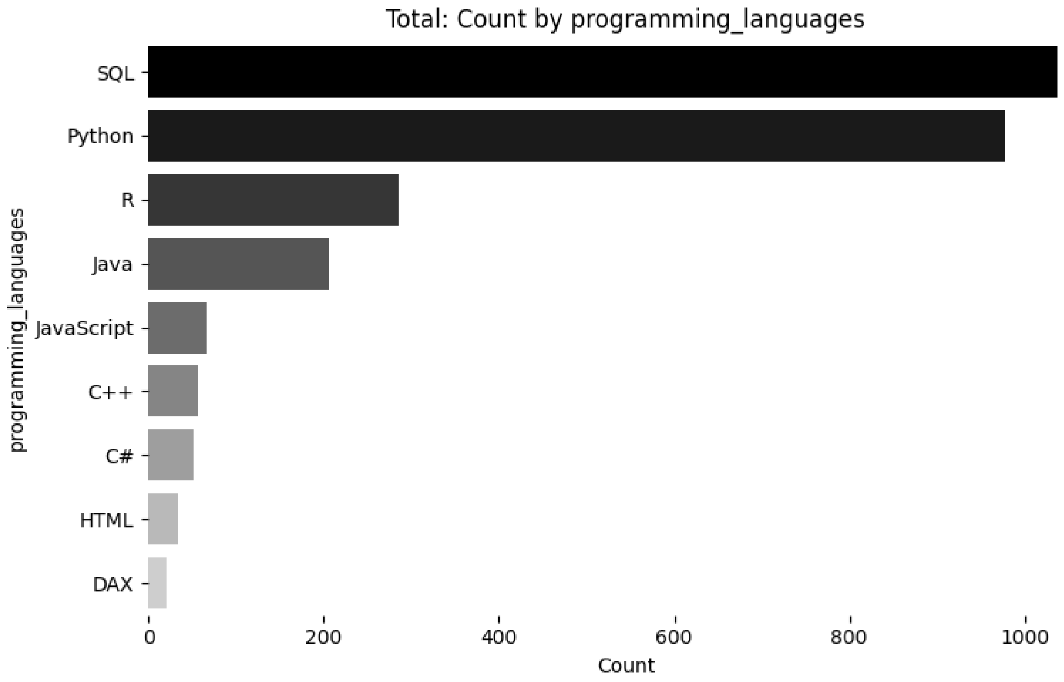


Data Engineer: Count by skills

4.6 Programming language

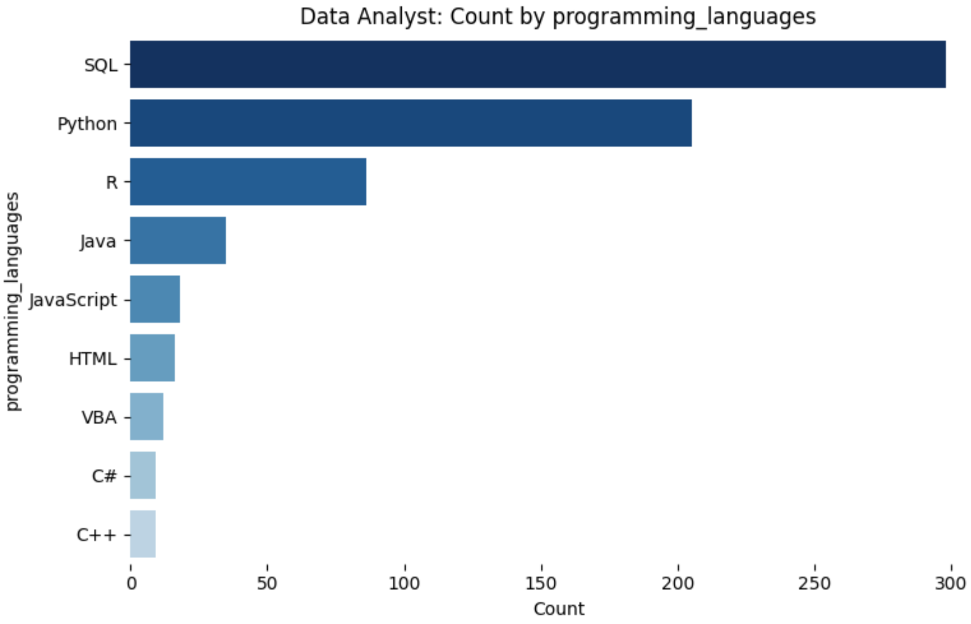
[21]:

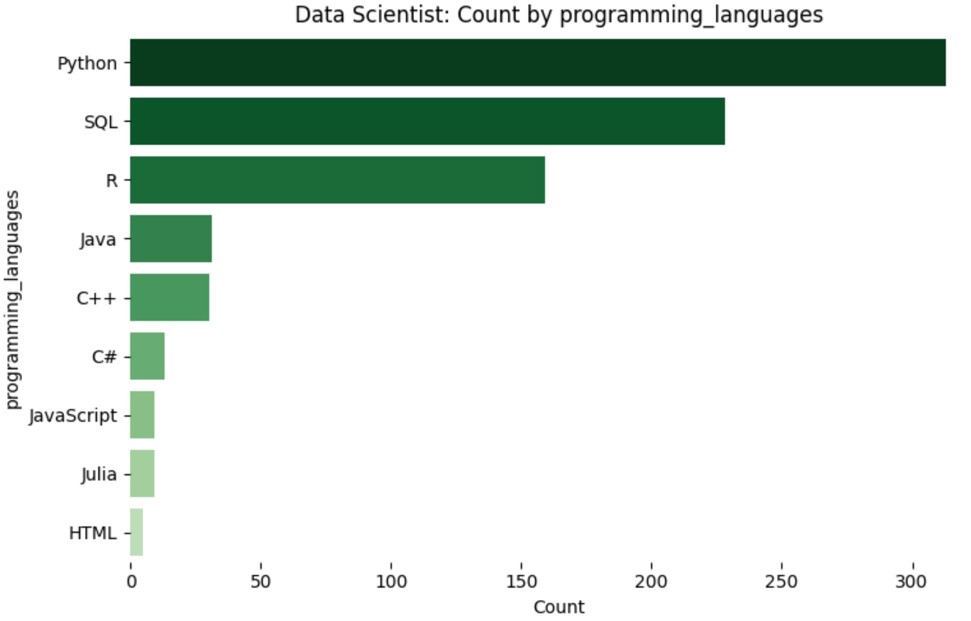
```
column_programming_language = ('programming_languages',)

plot_multiple_bars(column_programming_language,list_values=True,top_10=True,figsize=(8,5))
```

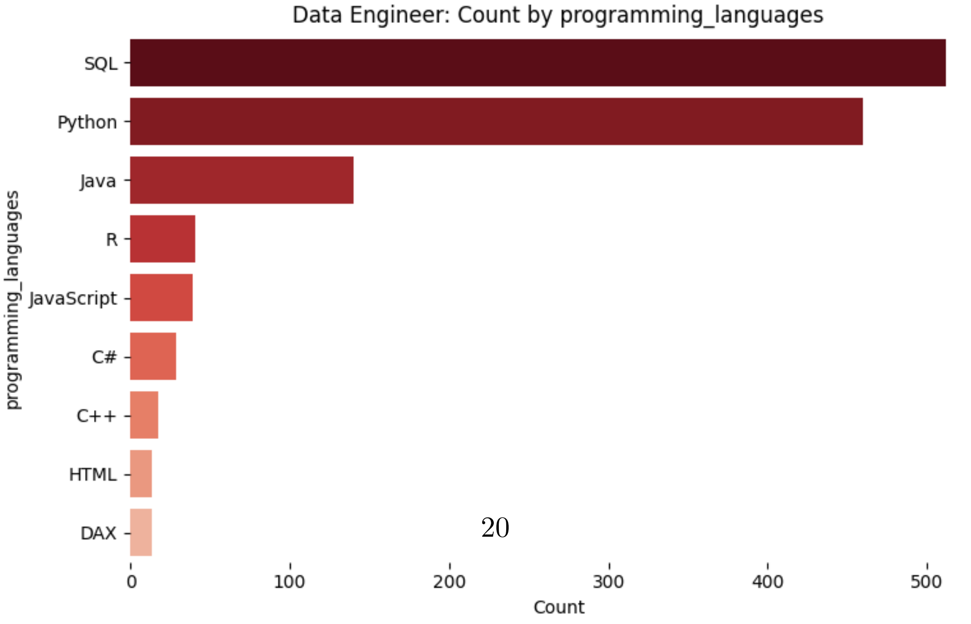


[22]:

```
plot_multiple_bars(column_programming_language,category='job_group',list_values=True,top_10=Tr
```


Data Analyst: Count by programming_languages
programming_languages
SQL
Python
R
Java
JavaScript
HTML
VBA
C#
C++
0
50
100
150
200
250
300
Count

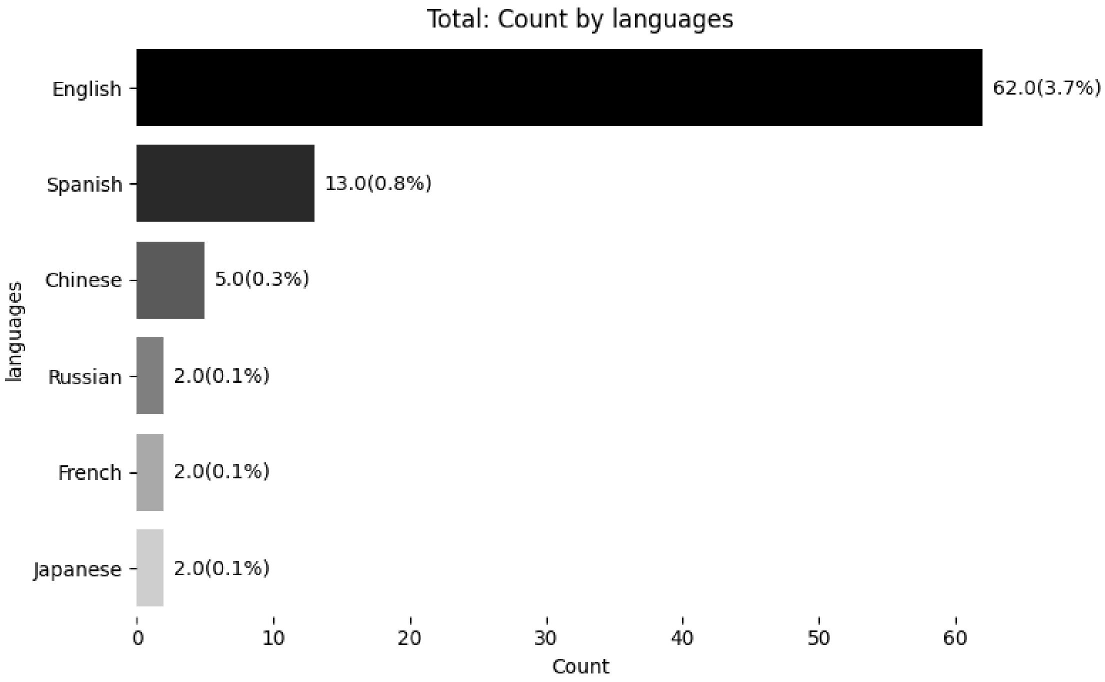

Data Scientist: Count by programming_languages
programming_languages
Python
SQL
R
Java
C++
C#
JavaScript
Julia
HTML
0
50
100
150
200
250
300
Count


Data Engineer: Count by programming_languages
programming_languages
SQL
Python
Java
R
JavaScript
C#
C++
HTML
DAX
0
100
200
300
400
500
Count

4.7 Languages

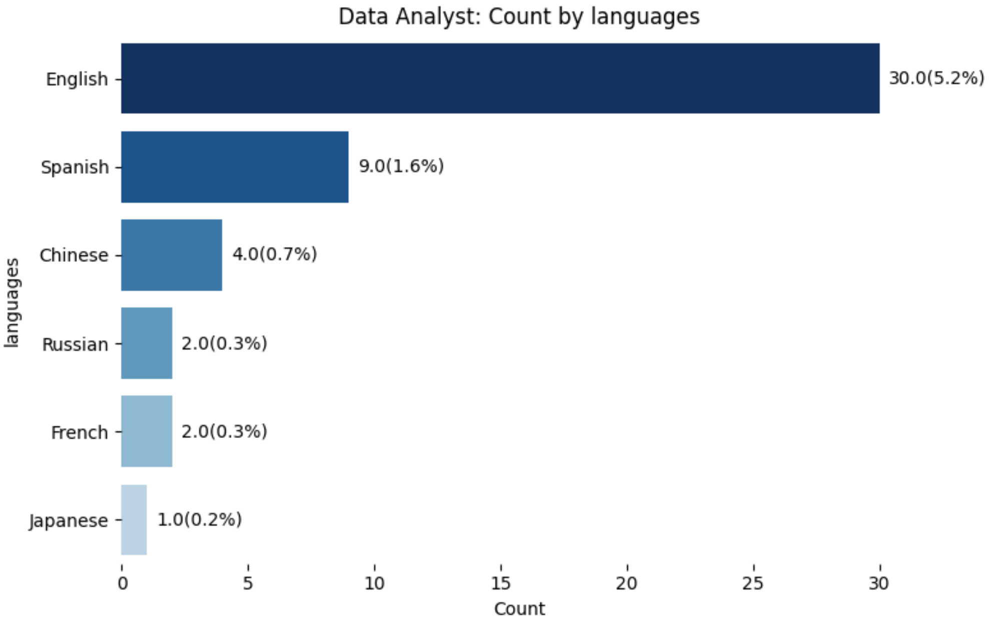
[24]:

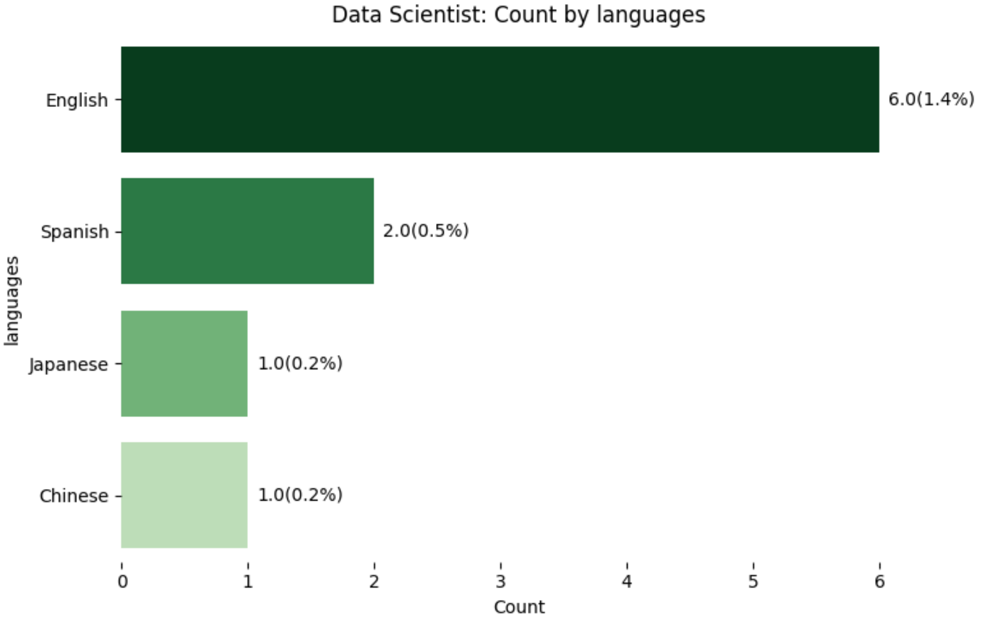
```
column_languages = ('languages',)

plot_multiple_bars(column_languages,list_values=True,figsize=(8,5),count_percentage=True)
```

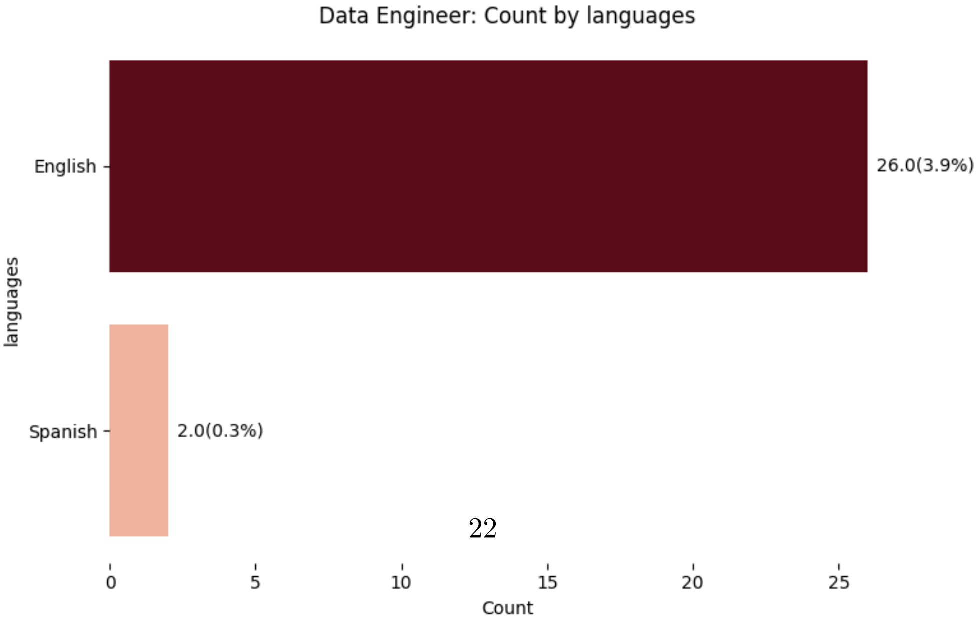


[25]:

```
plot_multiple_bars(column_languages,category =␣
↪'job_group',list_values=True,figsize=(8,15),count_percentage=True)
```


Data Analyst: Count by languages
languages
English
Spanish
Chinese
Russian
French
Japanese
30.0(5.2%)
9.0(1.6%)
4.0(0.7%)
2.0(0.3%)
2.0(0.3%)
1.0(0.2%)
0
5
10
15
20
25
30
Count


Data Scientist: Count by languages
languages
English
Spanish
Japanese
Chinese
6.0(1.4%)
2.0(0.5%)
1.0(0.2%)
1.0(0.2%)
0
1
2
3
4
5
6
Count

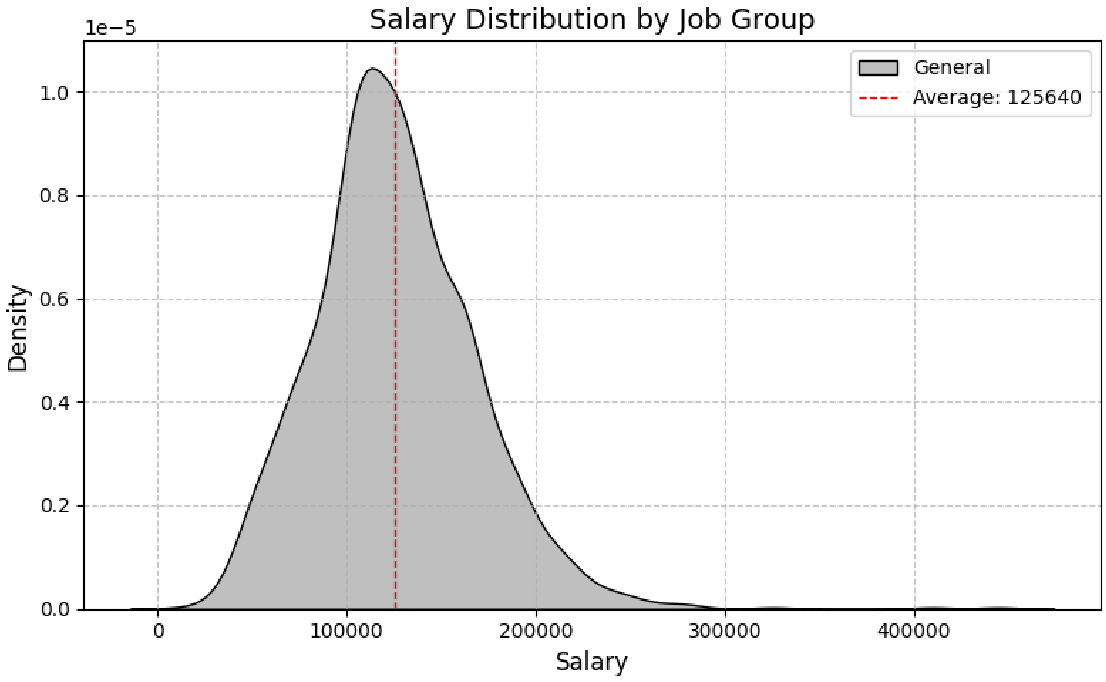

Data Engineer: Count by languages
languages
English
Spanish
26.0(3.9%)
2.0(0.3%)
0
5
10
15
20
25
Count

5 Salary

[31]:

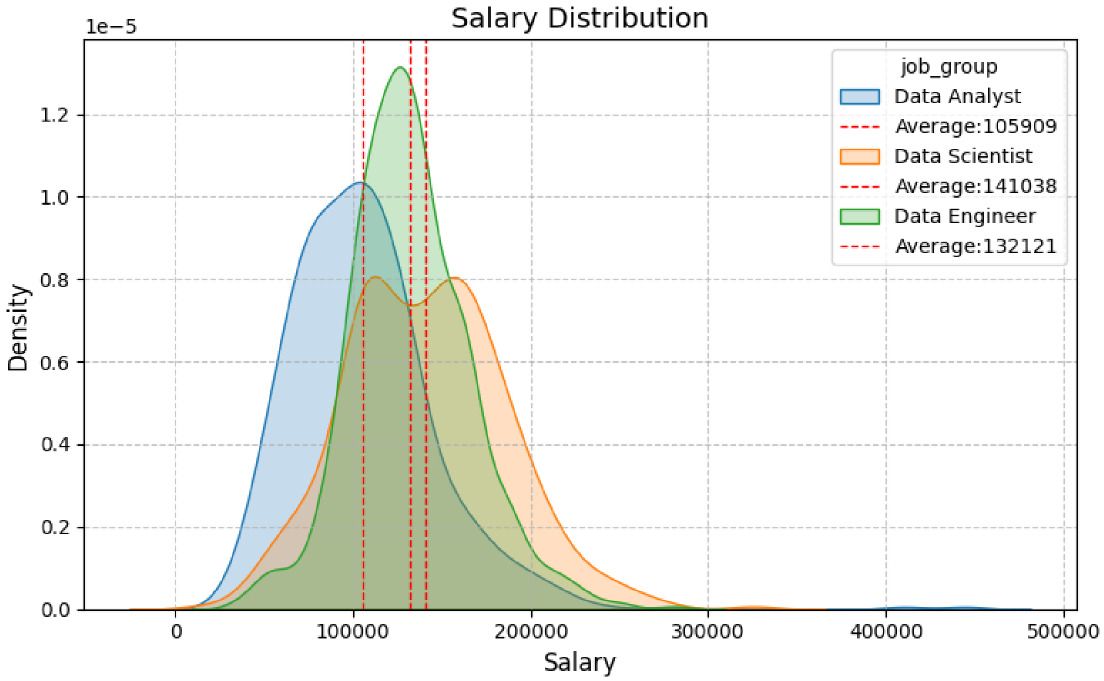
```
def plot_salary_kde(df,job_group_column=None,salary_column='salary',**kwargs):

    plt.figure(figsize=(8,5))

    if job_group_column:
        for job_group in df[job_group_column].unique():
            group_data = df[df[job_group_column] == job_group]

            mean_salary = group_data[salary_column].mean()

            sns.kdeplot(group_data[salary_column], shade=True, label=job_group,␣
 ↪**kwargs)

            plt.
 ↪axvline(mean_salary,color='red',linestyle='dashed',linewidth=1,label=f'Average:
 ↪{mean_salary:.0f}')

            plt.title('Salary Distribution', fontsize = 14)

    else:

        mean_salary = df[salary_column].mean()

        sns.kdeplot(df[salary_column],shade=True,color='black',label='General',␣
 ↪**kwargs)

        plt.axvline(mean_salary, color='red', linestyle='dashed', linewidth=1,␣
 ↪label=f'Average: {mean_salary:.0f}')


        plt.title('Salary Distribution by Job Group', fontsize=14)


    plt.xlabel('Salary',fontsize=12)
    plt.ylabel('Density',fontsize=12)
    plt.legend(title=job_group_column if job_group_column else None)
    plt.grid(True,linestyle='--',alpha=0.7)

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

```
plot_salary_kde(df,salary_column='mean_salary')
```



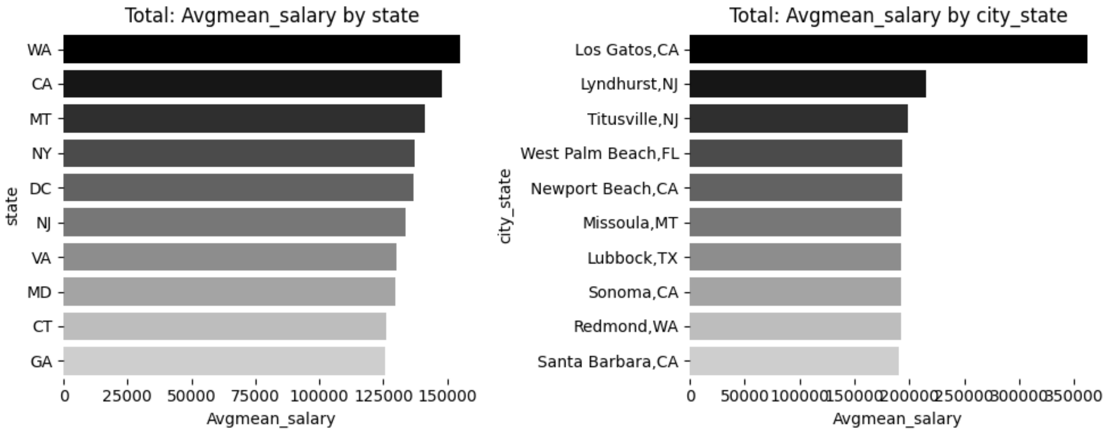
[32]:

```
plot_salary_kde(df,job_group_column='job_group',salary_column='mean_salary')
```

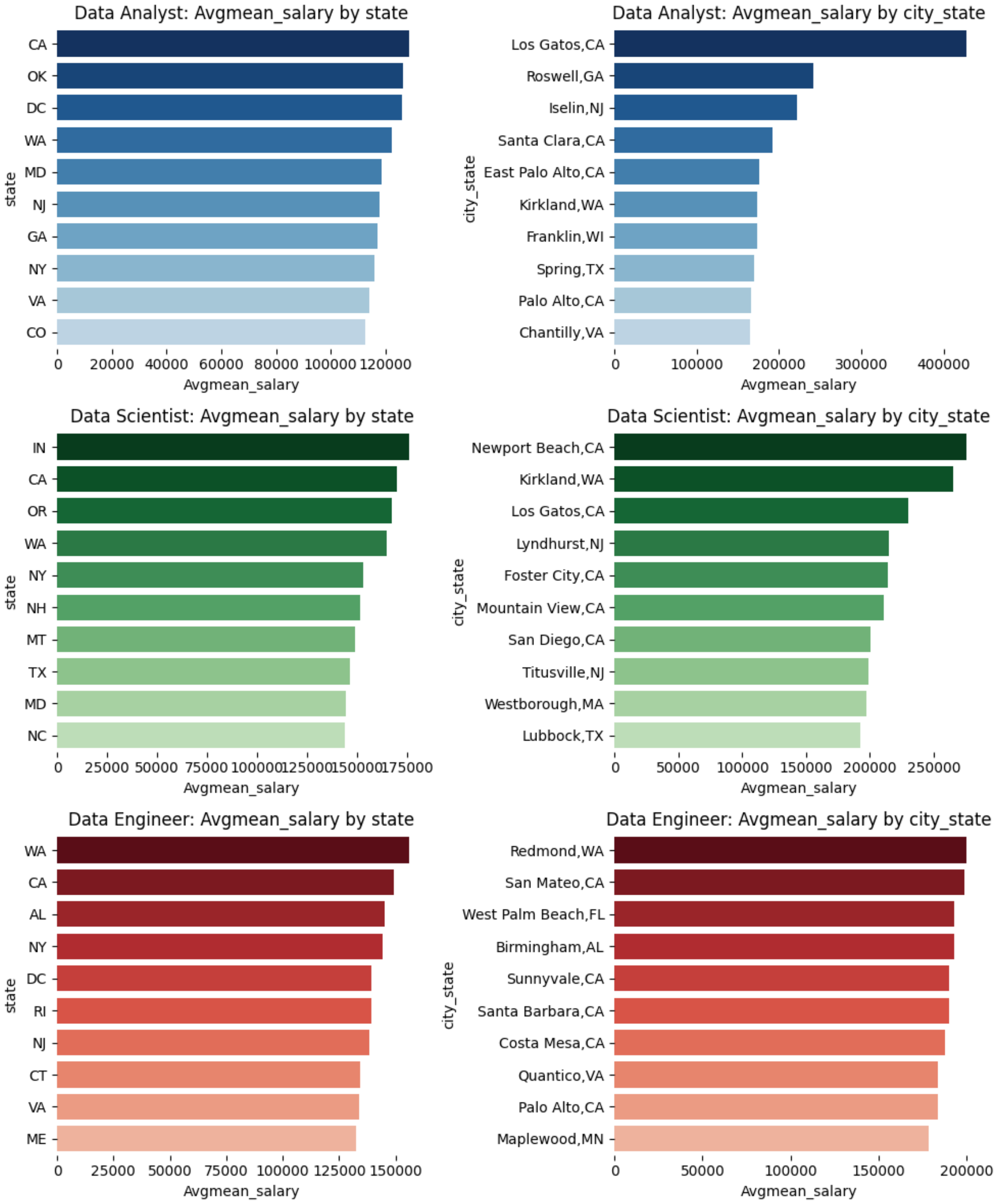


5.1 Location

```
[37]: plot_multiple_bars(columns_location,figsize=(10,4),top_10=True,aggregate_column='mean_salary')
```

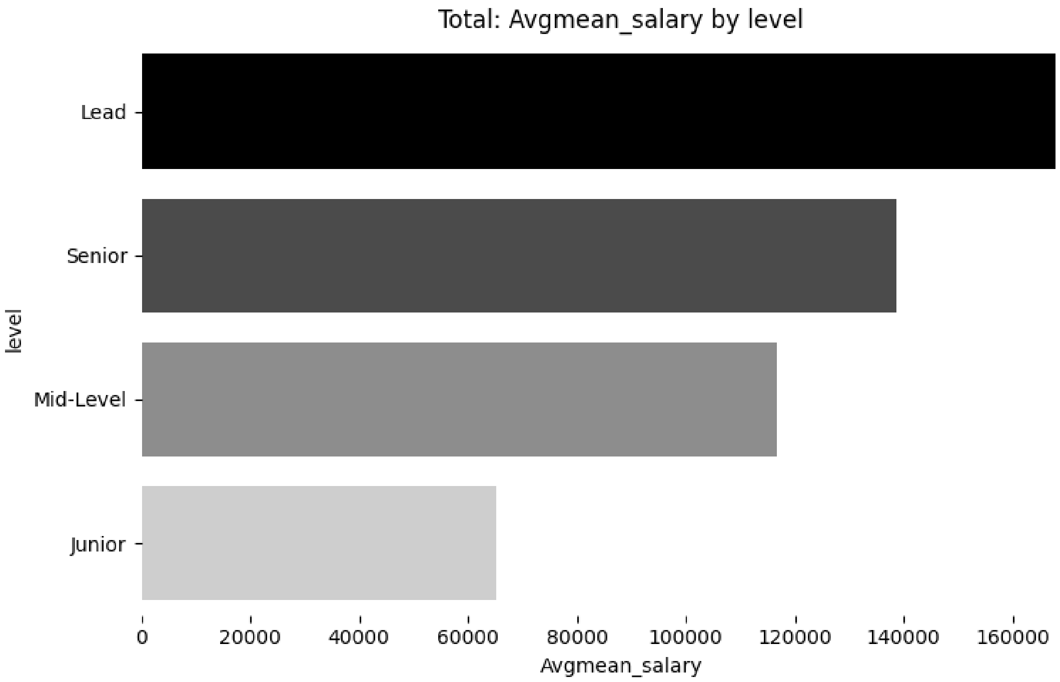


```
[38]: plot_multiple_bars(columns_location,category='job_group',figsize=(10,12),top_10=True,aggregate
```



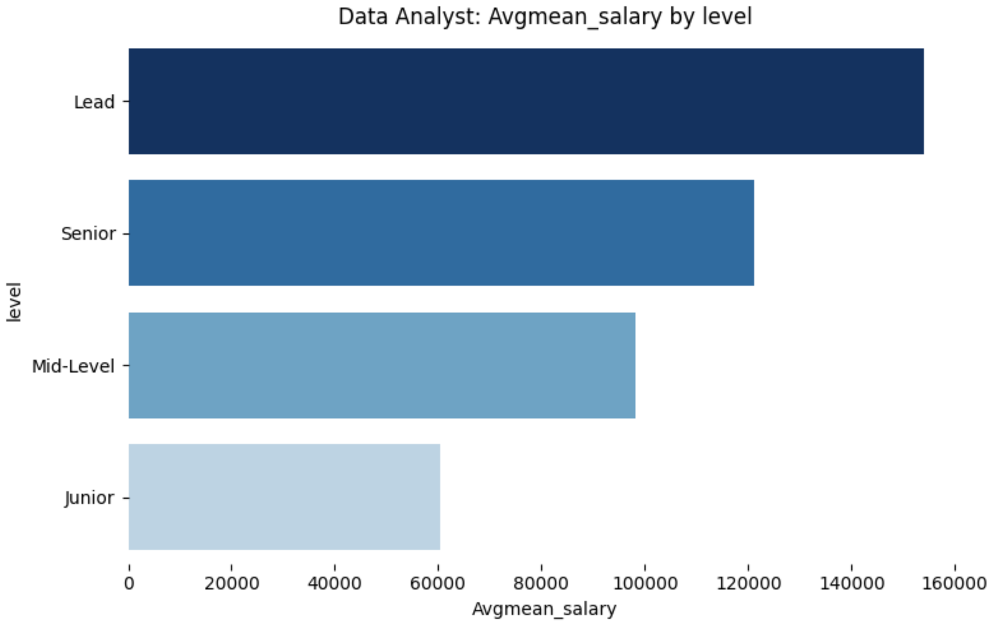
5.2 Level

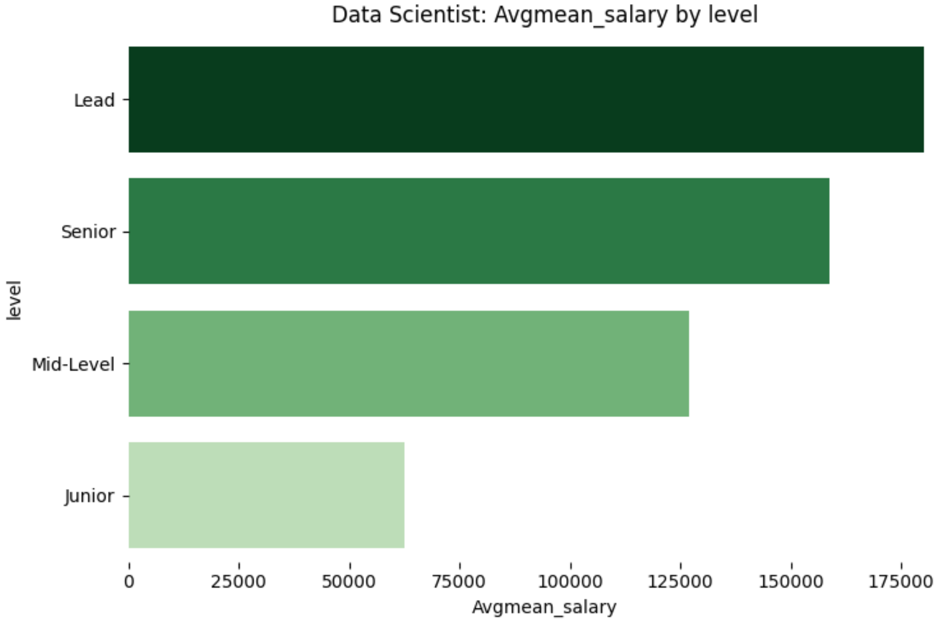
[39]:
```
plot_multiple_bars(columns_level,figsize=(8,5),aggregate_column='mean_salary')
```

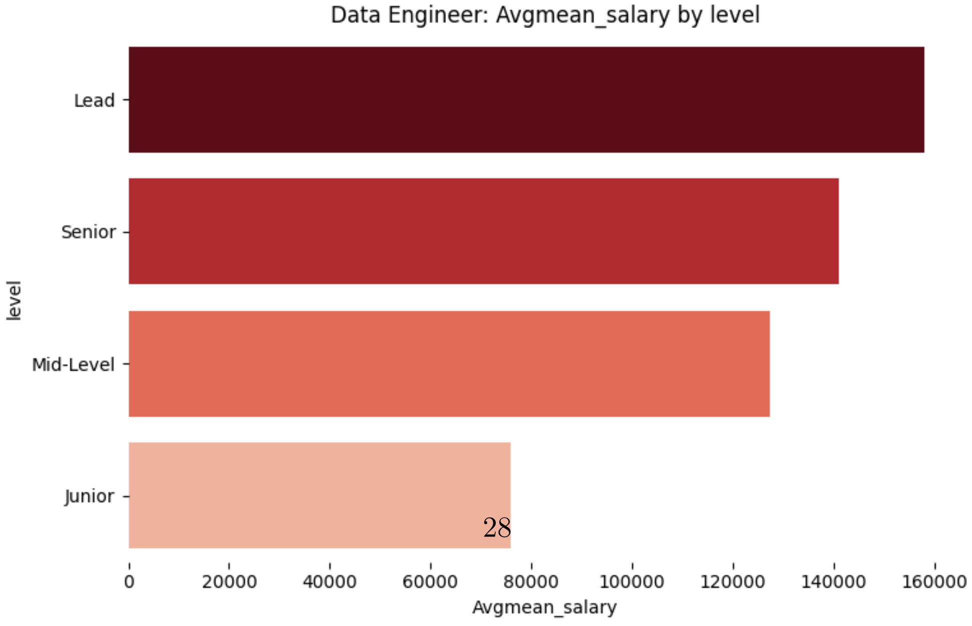


[40]:

```
plot_multiple_bars(columns_level,category='job_group',figsize=(8,15),aggregate_column='mean_sa
```

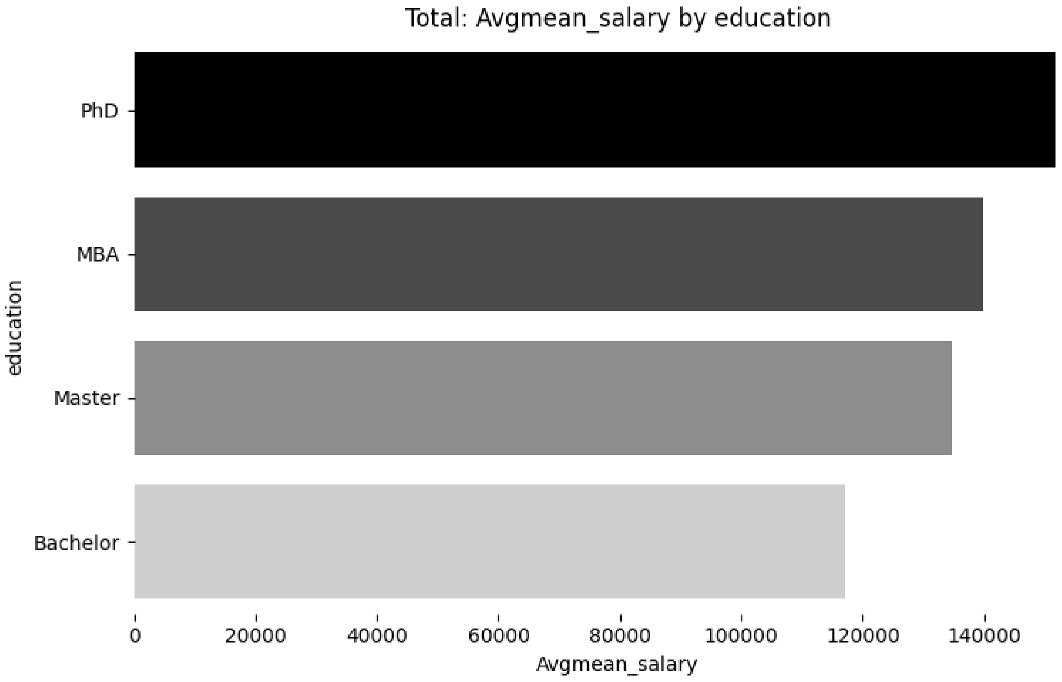

Data Analyst: Avgmean_salary by level
level
Lead
Senior
Mid-Level
Junior
0
20000
40000
60000
80000
100000
120000
140000
160000
Avgmean_salary


Data Scientist: Avgmean_salary by level
level
Lead
Senior
Mid-Level
Junior
0
25000
50000
75000
100000
125000
150000
175000
Avgmean_salary

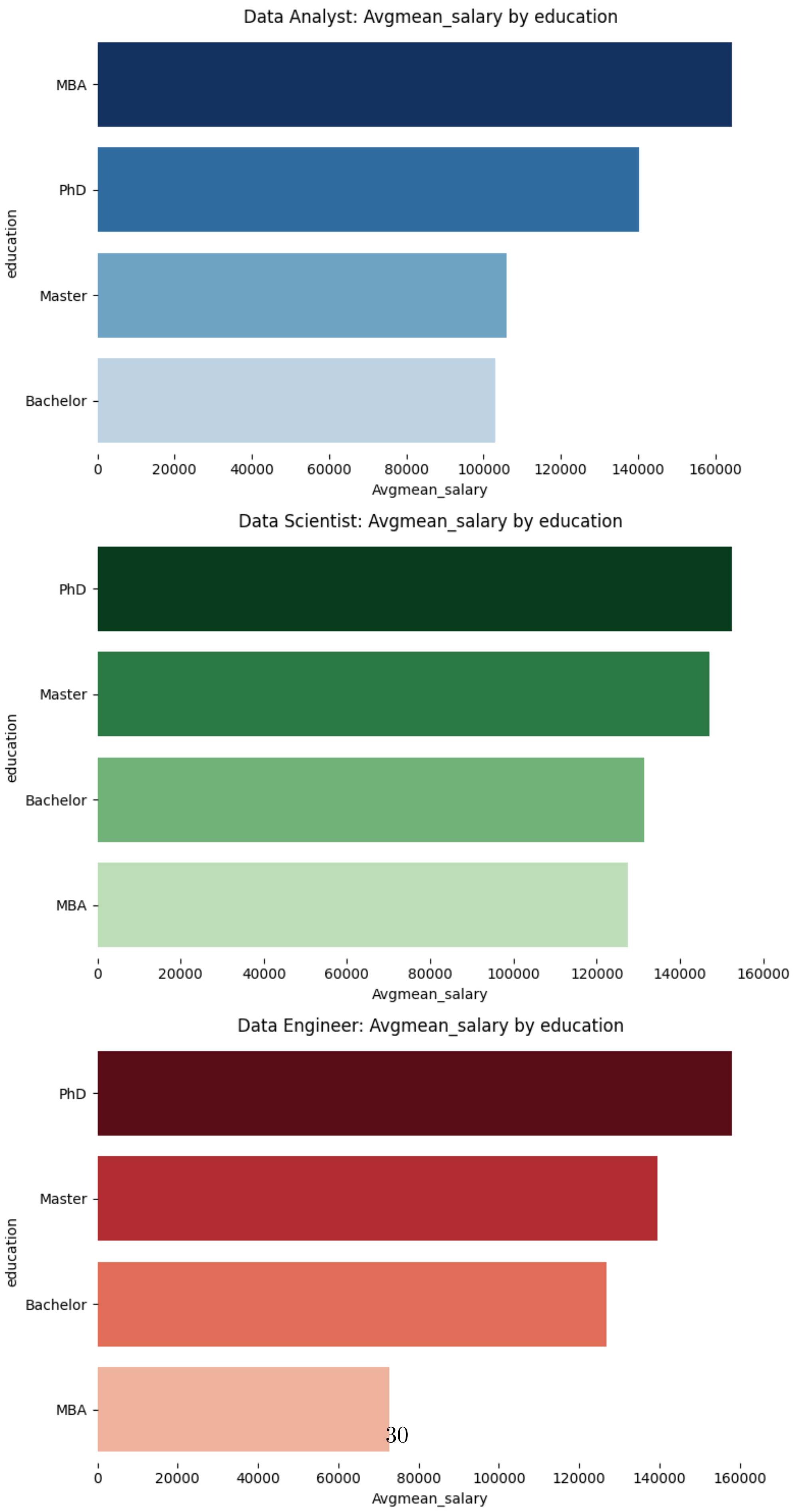

Data Engineer: Avgmean_salary by level
level
Lead
Senior
Mid-Level
Junior
0
20000
40000
60000
80000
100000
120000
140000
160000
Avgmean_salary

5.3 Education

```
[41]: plot_multiple_bars(columns_education,figsize=(8,5),aggregate_column='mean_salary')
```



```
[43]: plot_multiple_bars(columns_education,category='job_group',figsize=(8,15),aggregate_column='mea
```

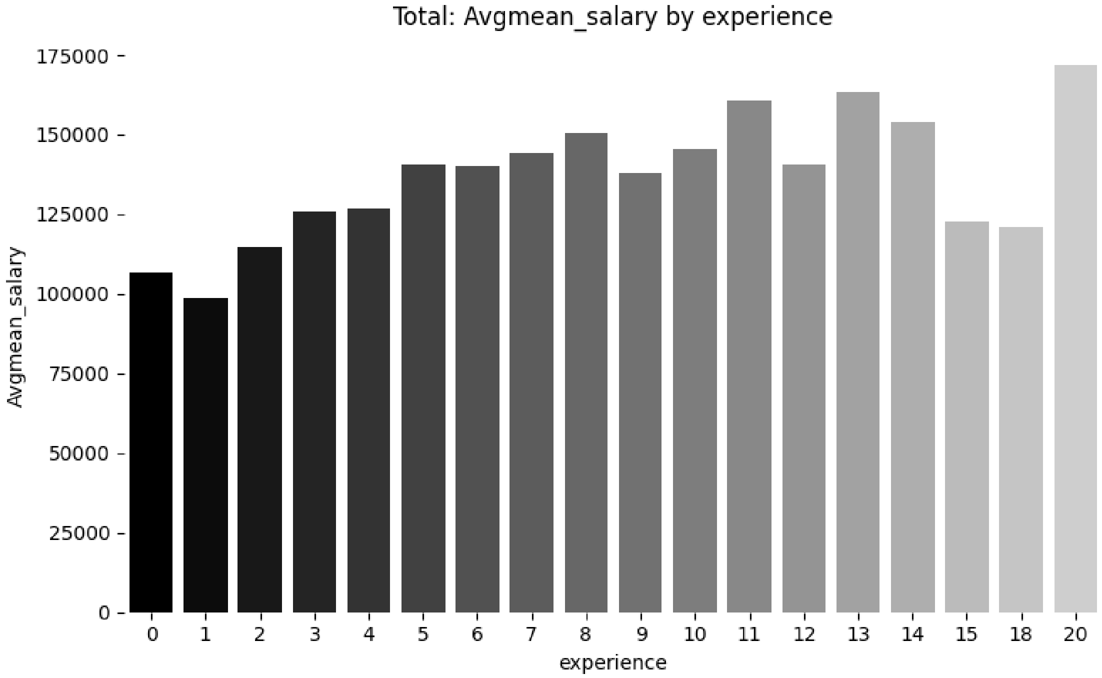

Data Analyst: Avgmean_salary by education

Data Scientist: Avgmean_salary by education

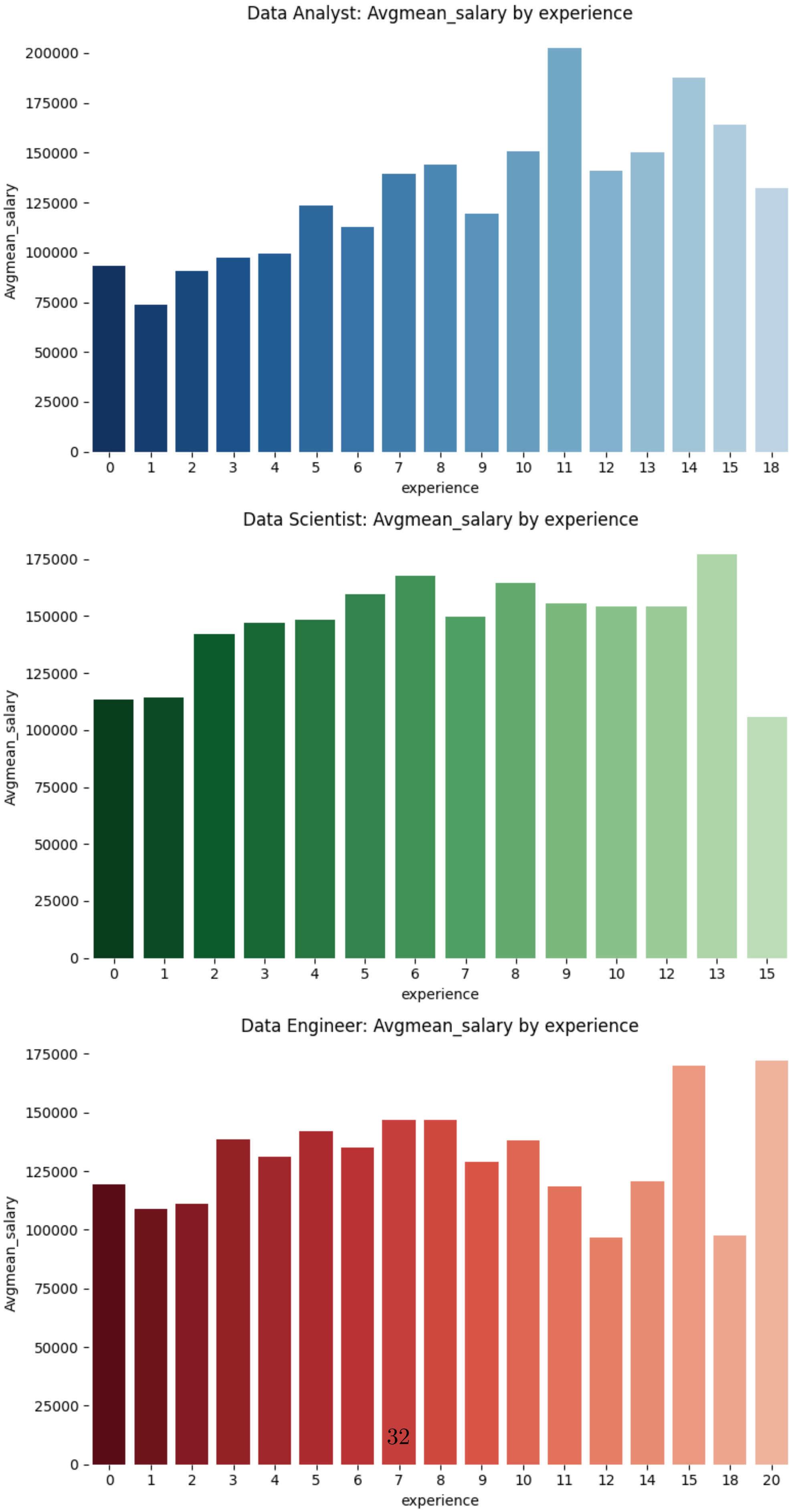
Data Engineer: Avgmean_salary by education

5.4 Experience

[44]:
```
plot_multiple_bars(columns_experience,figsize=(8,5), x_y= True,␣
↪aggregate_column='mean_salary')
```



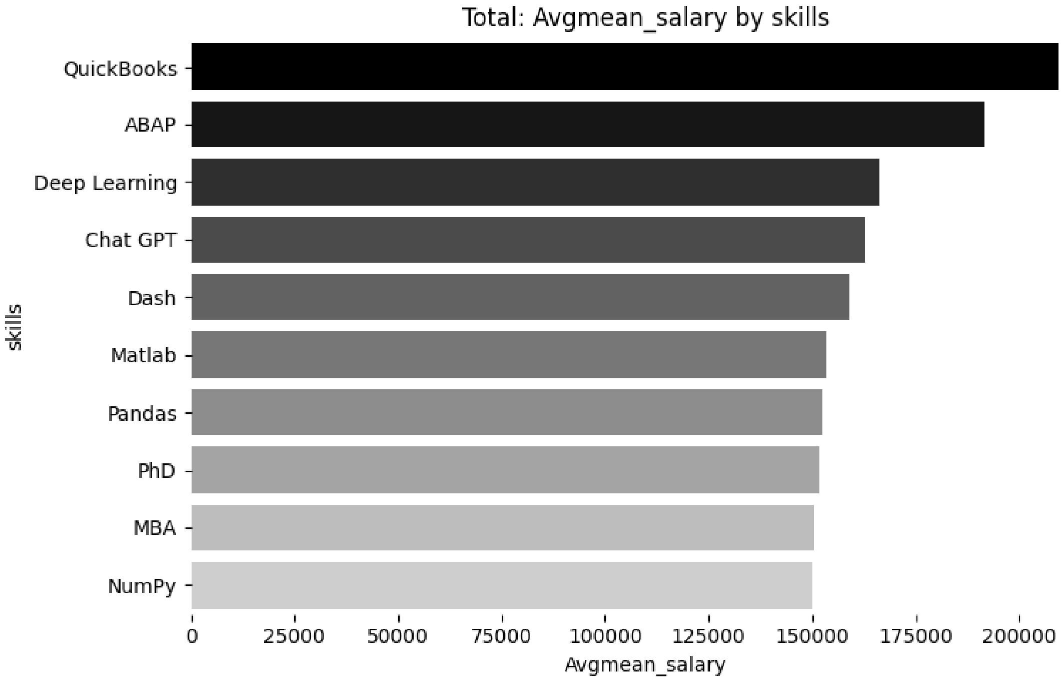
[ ]:
```
plot_multiple_bars(columns_experience,category='job_group',figsize=(8,15), x_y=␣
↪True, aggregate_column='mean_salary')
```


Data Analyst: Avgmean_salary by experience
Avgmean_salary
200000
175000
150000
125000
100000
75000
50000
25000
0
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
18
experience
Data Scientist: Avgmean_salary by experience
Avgmean_salary
175000
150000
125000
100000
75000
50000
25000
0
0
1
2
3
4
5
6
7
8
9
10
12
13
15
experience
Data Engineer: Avgmean_salary by experience
Avgmean_salary
175000
150000
125000
100000
75000
50000
25000
0
0
1
2
3
4
5
6
7
8
9
10
11
12
14
15
18
20
experience

5.5 Skills

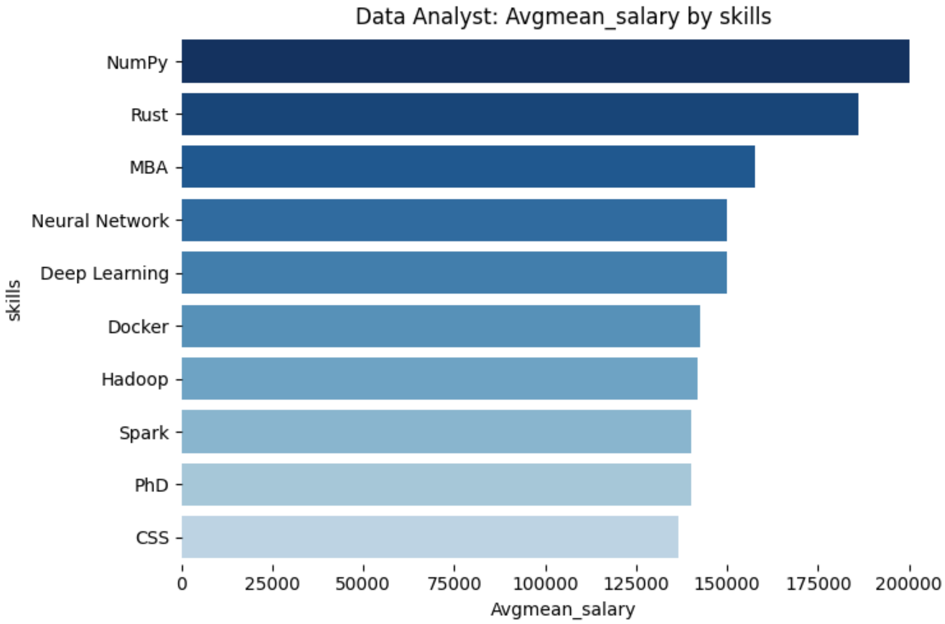
[46]:

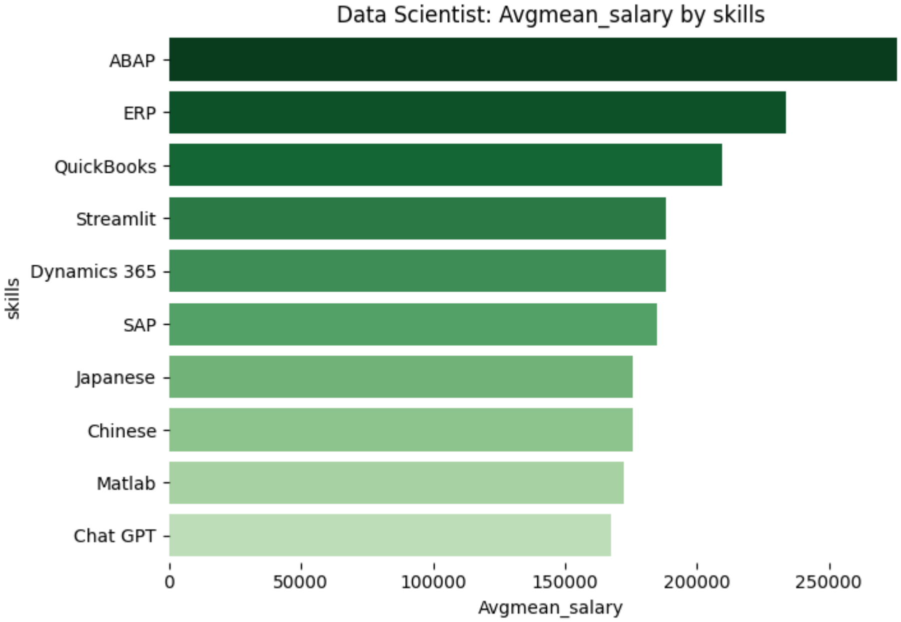
```
plot_multiple_bars(columns_skills,figsize=(8,5), list_values=True, top_10=True,␣
  ↪aggregate_column='mean_salary')
```

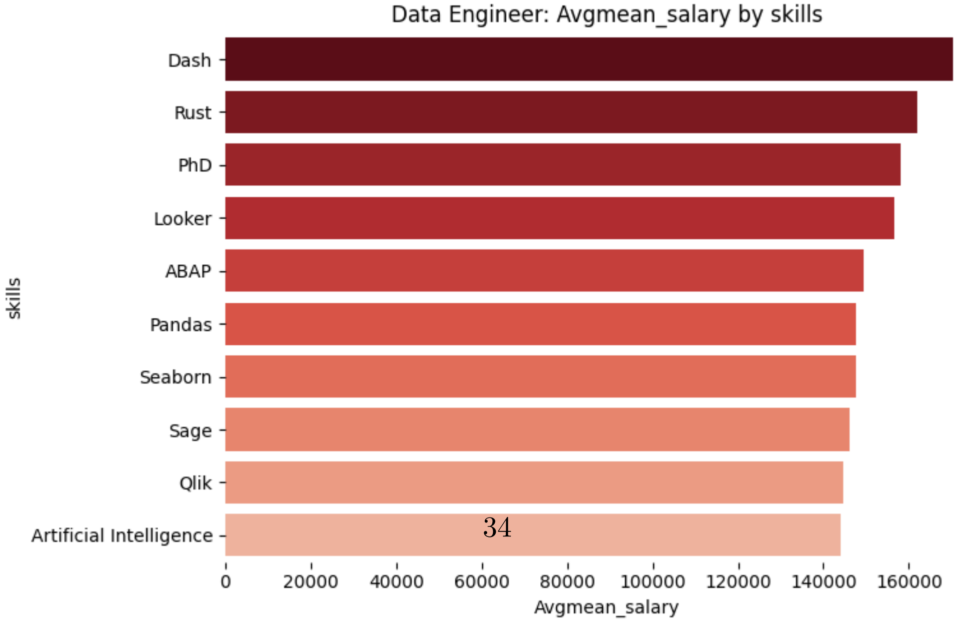


[48]:

```
plot_multiple_bars(columns_skills,category='job_group',figsize=(8,15),␣
  ↪list_values=True, top_10=True, aggregate_column='mean_salary')
```


Data Analyst: Avgmean_salary by skills

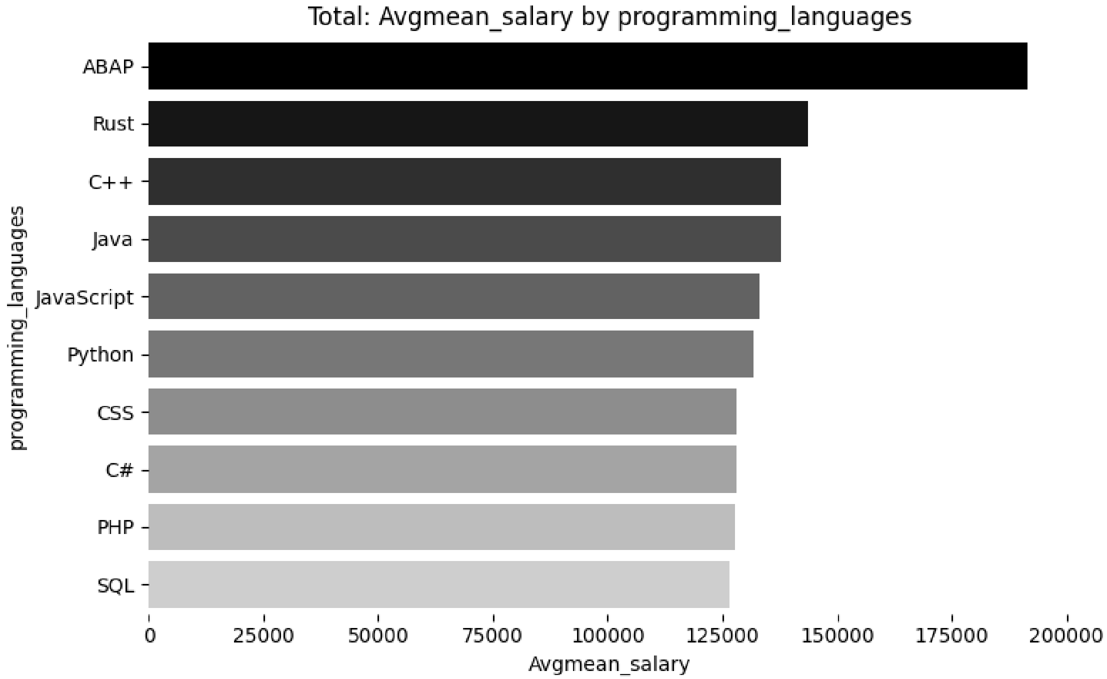

Data Scientist: Avgmean_salary by skills


Data Engineer: Avgmean_salary by skills

5.6 Programming languages

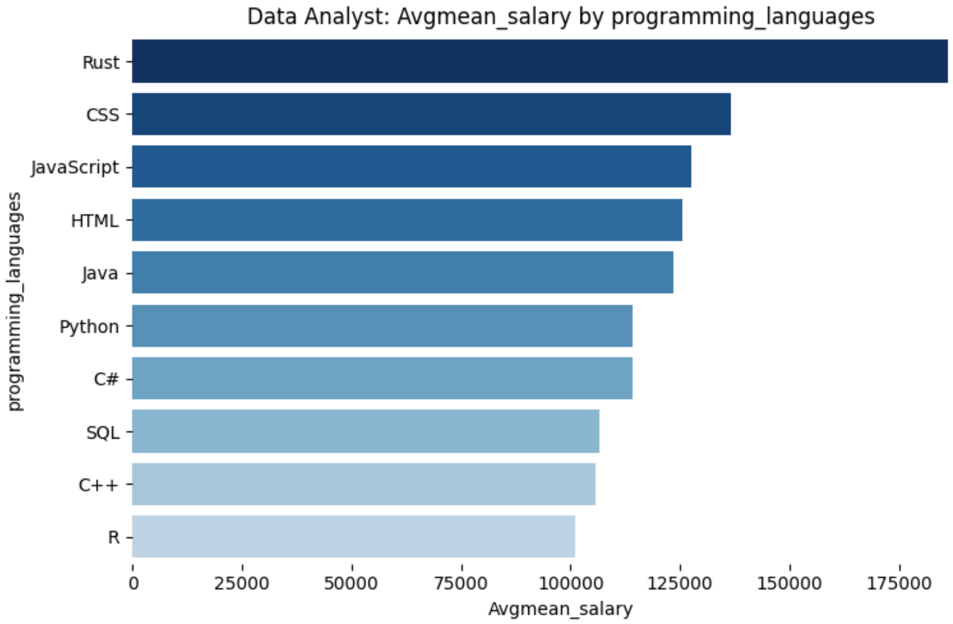
[49]:

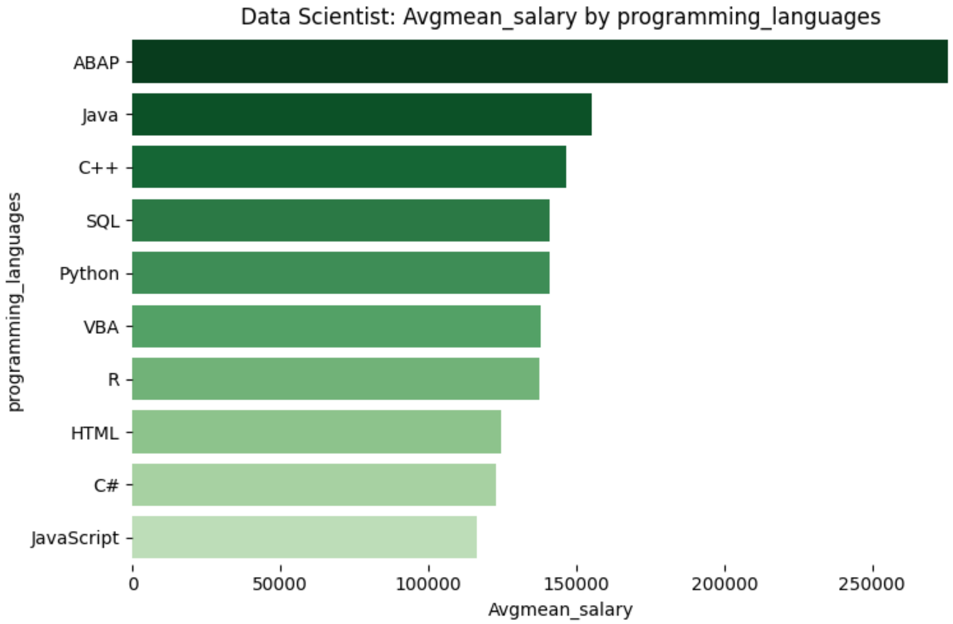
```
plot_multiple_bars(column_programming_language,figsize=(8,5), list_values=True, top_10=True, aggregate_column='mean_salary')
```

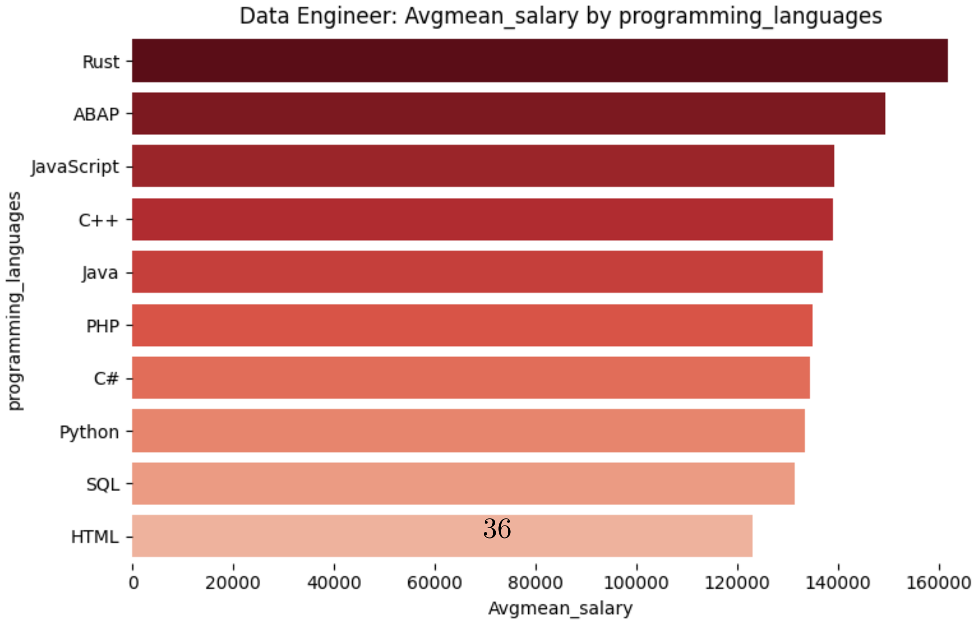


[50]:

```
plot_multiple_bars(column_programming_language,category='job_group',figsize=(8,15), list_values=True, top_10=True, aggregate_column='mean_salary')
```


Data Analyst: Avgmean_salary by programming_languages
programming_languages
Rust
CSS
JavaScript
HTML
Java
Python
C#
SQL
C++
R
0
25000
50000
75000
100000
125000
150000
175000
Avgmean_salary

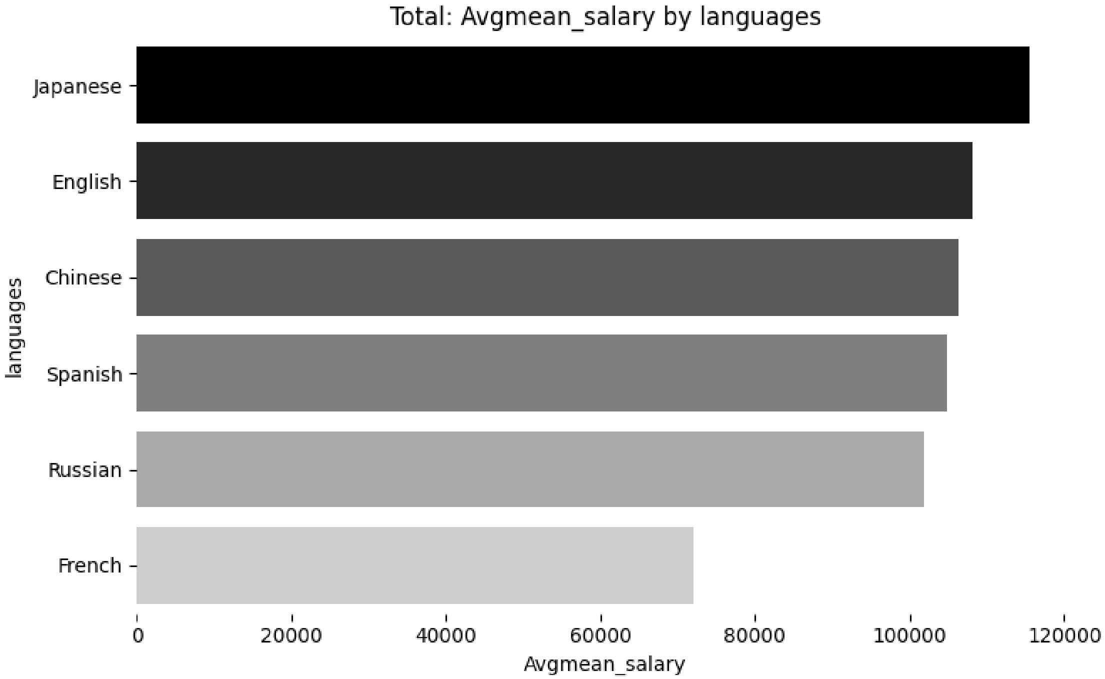

Data Scientist: Avgmean_salary by programming_languages
programming_languages
ABAP
Java
C++
SQL
Python
VBA
R
HTML
C#
JavaScript
0
50000
100000
150000
200000
250000
Avgmean_salary


Data Engineer: Avgmean_salary by programming_languages
programming_languages
Rust
ABAP
JavaScript
C++
Java
PHP
C#
Python
SQL
HTML
0
20000
40000
60000
80000
100000
120000
140000
160000
Avgmean_salary

5.7 Languages

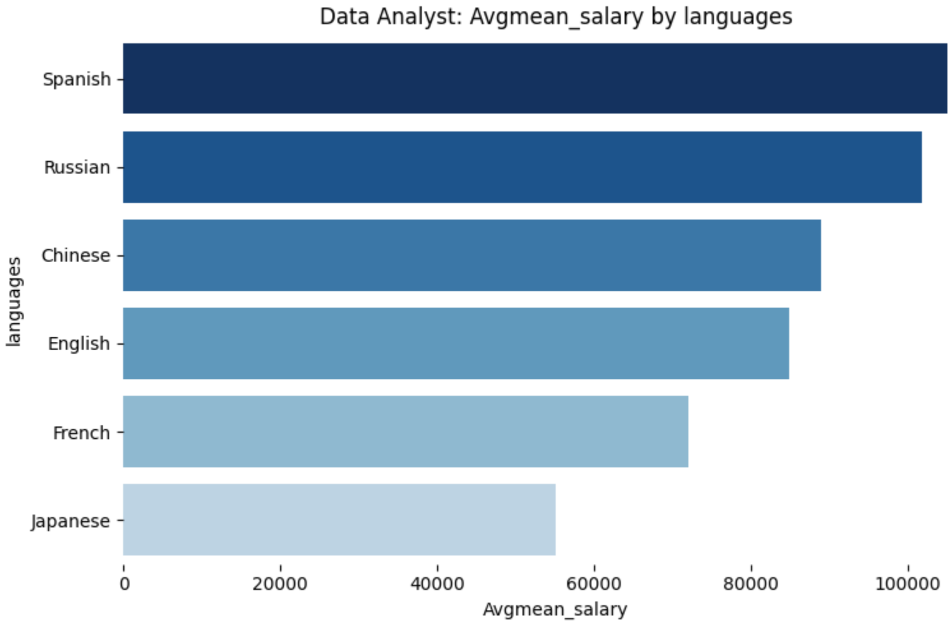
[51]:

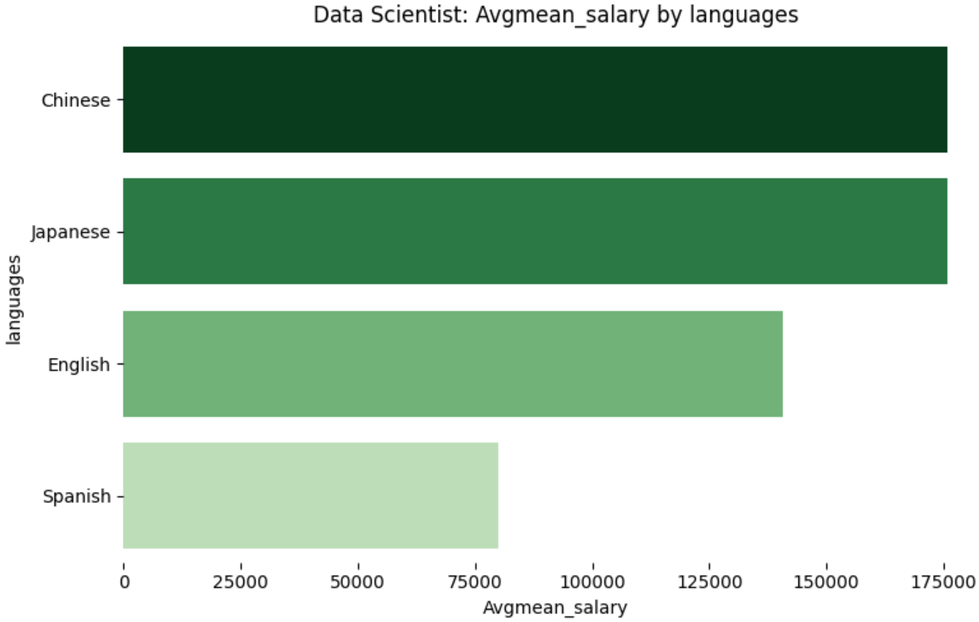
```
plot_multiple_bars(column_languages,figsize=(8,5), list_values=True,␣
 ↪top_10=True, aggregate_column='mean_salary')
```

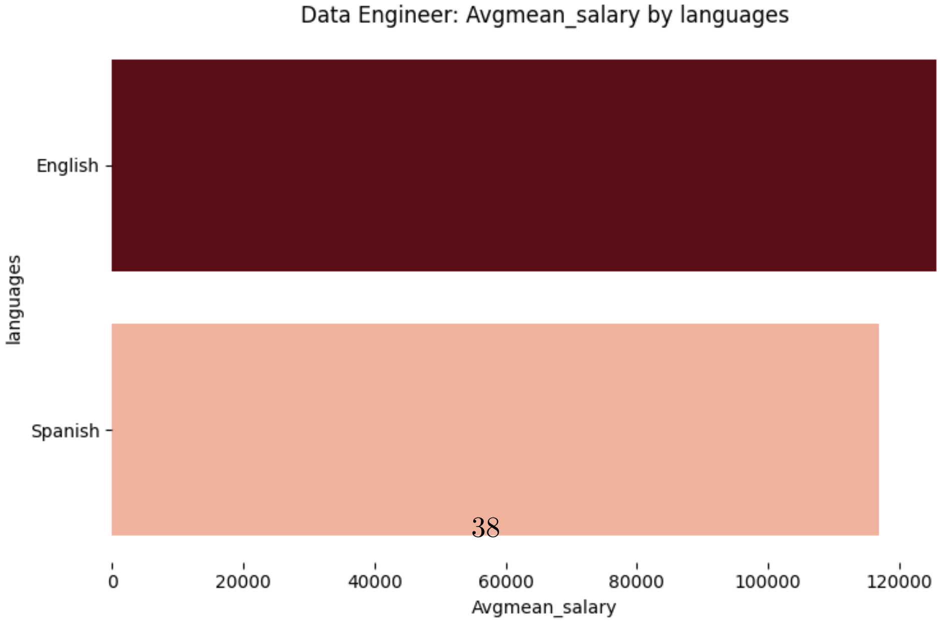


[52]:

```
plot_multiple_bars(column_languages,category='job_group',figsize=(8,15),␣
 ↪list_values=True, top_10=True, aggregate_column='mean_salary')
```


Data Analyst: Avgmean_salary by languages
languages
Spanish
Russian
Chinese
English
French
Japanese
0
20000
40000
60000
80000
100000
Avgmean_salary

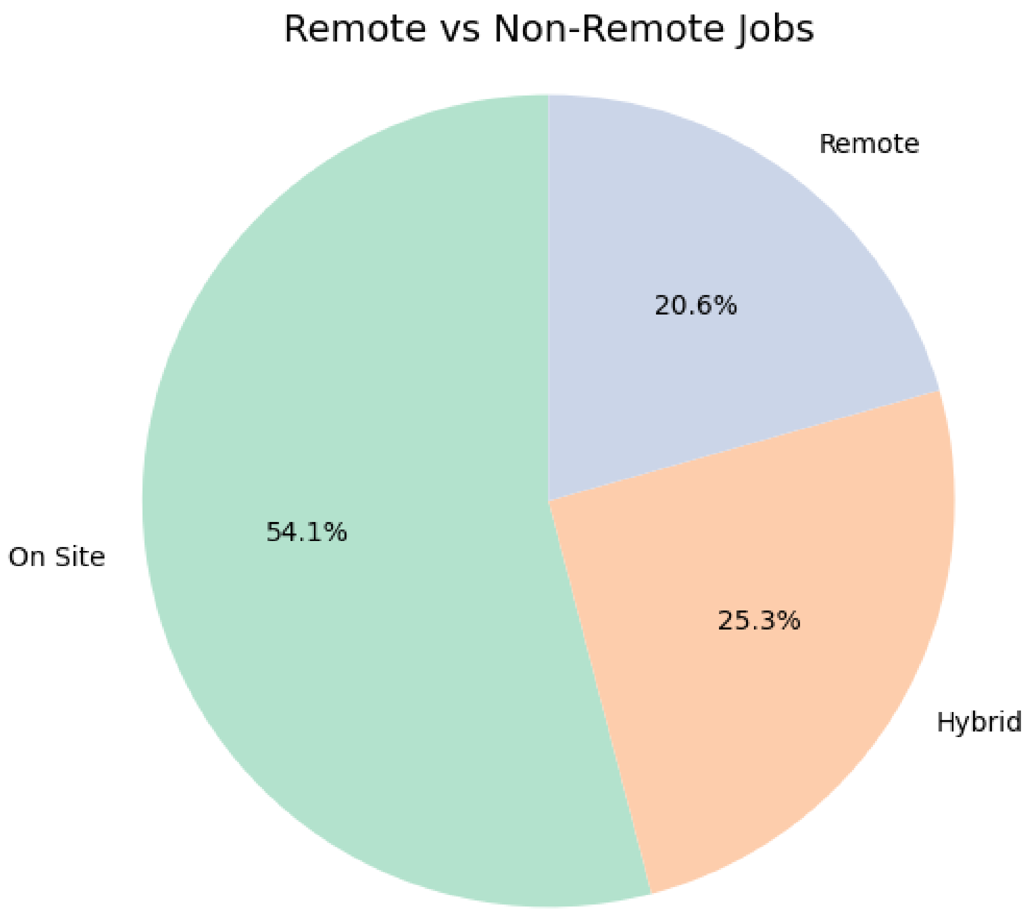

Data Scientist: Avgmean_salary by languages
languages
Chinese
Japanese
English
Spanish
0
25000
50000
75000
100000
125000
150000
175000
Avgmean_salary


Data Engineer: Avgmean_salary by languages
languages
English
Spanish
0
20000
40000
60000
80000
100000
120000
Avgmean_salary

6 Remote trend

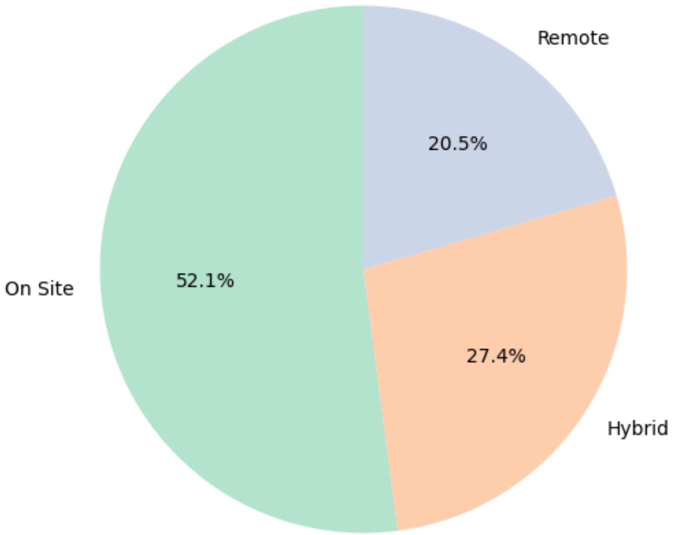
[53]:

```
def plot_remote_distribution(df,column_name,title,category=None,figsize=(6,6)):

    def create_pie(data,chart_title,ax):
        counts = data.value_counts()
        ax.pie(
            counts,
            labels=counts.index,
            autopct='%1.1f%%',
            colors=plt.cm.Pastel2.colors,
            startangle=90
        )
        ax.set_title(chart_title,fontsize=14)
        ax.axis('equal')

    if category:
        unique_categories = df[category].unique()
        n_categories = len(unique_categories)

        fig,axes = plt.subplots(n_categories,1,figsize=figsize)
        if n_categories == 1:
            axes = [axes]

        for i, job_group in enumerate(unique_categories):
            df_grouped = df[df[category] == job_group]
            title_grouped = f"{job_group}:{title}"
            create_pie(df_grouped[column_name], title_grouped, axes[i])

        plt.tight_layout()
        plt.show()

    else:

        fig, ax = plt.subplots(figsize=figsize)
        create_pie(df[column_name],title,ax)
        plt.show()

plot_remote_distribution(df=df,column_name='remote',title='Remote vs Non-Remote␣
 ↪Jobs')
```

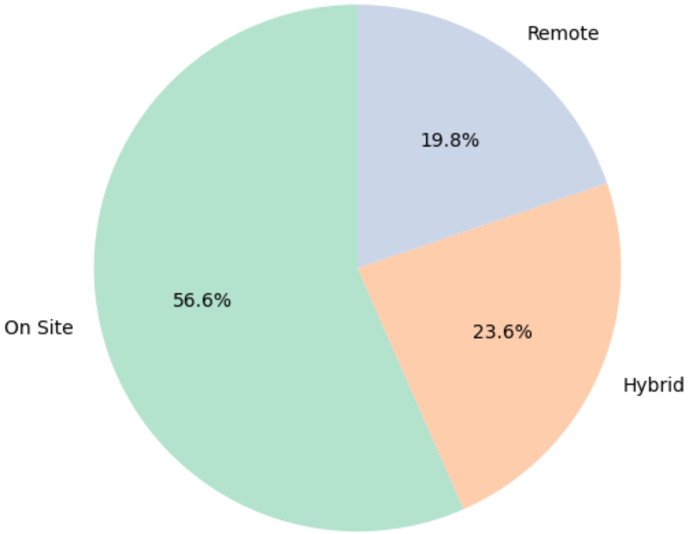


```
[54]: plot_remote_distribution(df=df,column_name='remote',title='Remote vs Non-Remote␣
        ↪Jobs', figsize=(6,14),category='job_group')
```

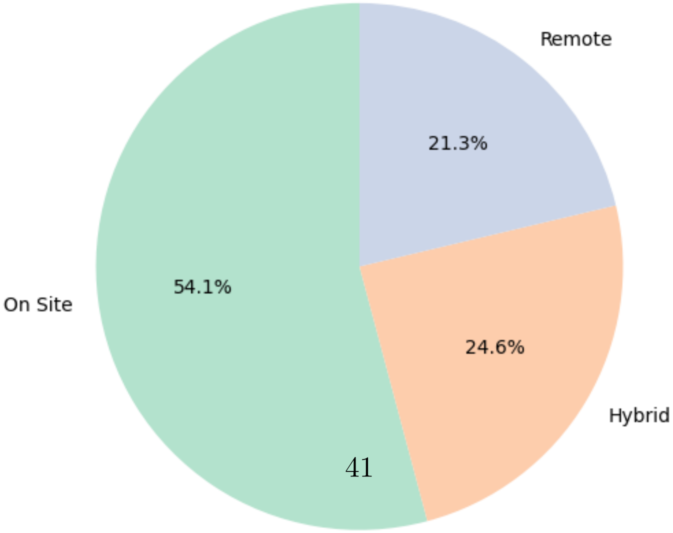
Data Analyst:Remote vs Non-Remote Jobs



Data Scientist:Remote vs Non-Remote Jobs



Data Engineer:Remote vs Non-Remote Jobs


Remote
On Site
Hybrid