# ANALYSIS OF DATA PROFESSIONS

## ♦ About Dataset

This dataset aims to shed light on the salary statistics of employees in the Data field. It will focus on various aspects of employment, including work experience, job titles, and company locations. This dataset provides valuable insights into salary distributions within the industry.

## ♦ Objective of Analysis

This notebook aims at:

---

- Data processing
- Practice using libraries to visualize data
- Visualize data, provide explanations about the correlation between attributes
- Draw meaningful conclusions and insights

## 1. IMPORTING LIBRARIES AND DATA

In [49]:

```
#Importing of libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt # Visualization
import seaborn as sns # Visualization
from scipy.stats import shapiro #Test for normality
from scipy.stats import kruskal #Hypothesis Test
```

In [50]:

```
#Read data
data_salary = pd.read_csv('D:/Project_FoM/Analysis/Data Science Jobs Salaries.csv')
```

# 2. EXPLORATORY DATA ANALYSIS ( EDA )

## ♦ View Dataset

In [51]:
```
#Viewing part of the data
data_salary.head()
```

Out[51]:

| | work_year | experience_level | employment_type | job_title | salary | salary_currency | salary_in_usd |
|---|---|---|---|---|---|---|---|
| **0** | 2021e | EN | FT | Data Science Consultant | 54000 | EUR | 64369 |
| **1** | 2020 | SE | FT | Data Scientist | 60000 | EUR | 68428 |
| **2** | 2021e | EX | FT | Head of Data Science | 85000 | USD | 85000 |
| **3** | 2021e | EX | FT | Head of Data | 230000 | USD | 230000 |
| **4** | 2021e | EN | FT | Machine Learning Engineer | 125000 | USD | 125000 |

## Examing the Dataset

In [52]:
```
#Identifying all column headers
data_salary.columns
```

Out[52]:
```
Index(['work_year', 'experience_level', 'employment_type', 'job_title',
       'salary', 'salary_currency', 'salary_in_usd', 'employee_residence',
       'remote_ratio', 'company_location', 'company_size'],
      dtype='object')
```

In [53]:
```
#Identifying the various job titles
jobs_available=data_salary['job_title'].unique()
sum_unique=data_salary['job_title'].value_counts().count()
print(sum_unique, ' jobs can be found in the dataset')
```

```
43  jobs can be found in the dataset
```

In [54]:
```
#Identifying experience levels
data_salary['experience_level'].unique()
```

Out[54]:
```
array(['EN', 'SE', 'EX', 'MI'], dtype=object)
```

`Experience_level` :

- EN: Entry-level / Junior
- MI: Mid-level / Intermediate
- SE: Senior-level / Expert
- EX: Executive-level / Director

In [55]:
```
data_salary['employment_type'].unique()
```

Out[55]:
```
array(['FT', 'PT', 'CT', 'FL'], dtype=object)
```

`Employment_type` :

- FT: Full-Time
- PT: Part-Time
- CT: Contractor
- FL: Freelancer

In [56]:
```
#Identifying the company size
data_salary['company_size'].unique()
```

Out[56]:
```
array(['L', 'M', 'S'], dtype=object)
```

`Company_Size` :

- L: Large
- M: Medium
- S: Small

In [57]:
```
#Identifying the remote ratio
data_salary['remote_ratio'].unique()
```

Out[57]:
```
array([ 50, 100,   0], dtype=int64)
```

`Remote_ratio` :

- 0: None remote
- 50: Hybrid
- 100: Fully remote

# ♦ Cleaning the Dataset

Combining the 3 jobs we are working with.ie. Data Science, data analysis and data engineer.

In [58]:
```
data_scientist=data_salary[data_salary['job_title']=='Data Scientist']
data_analyst= data_salary[data_salary['job_title']=='Data Analyst']
data_engineer=data_salary[data_salary['job_title']=='Data Engineer']
work_data=pd.concat([data_scientist,data_analyst,data_engineer],axis=0)
```

In [59]:
```
#  Replace values in work_year column and change data type
work_data['work_year'] = work_data['work_year'].replace('2021e', '2021')
work_data['work_year'] = work_data['work_year'].astype('int64')

#  Replace values in experience-level column
work_data['experience_level'] = work_data['experience_level'].replace('EN', 'Entry-Lev
work_data['experience_level'] = work_data['experience_level'].replace('EX', 'Experienc
work_data['experience_level'] = work_data['experience_level'].replace('MI', 'Mid-Level
work_data['experience_level'] = work_data['experience_level'].replace('SE', 'Senior-Le
# Replace values in employment_type column
work_data['employment_type'] = work_data['employment_type'].replace('FT', 'Full-Time')
work_data['employment_type'] = work_data['employment_type'].replace('CT', 'Contractor'
work_data['employment_type'] = work_data['employment_type'].replace('FL', 'Freelancer'
work_data['employment_type'] = work_data['employment_type'].replace('PT', 'Part-Time')
# Replace values in Company size column
work_data['company_size'] = work_data['company_size'].replace('L', "Large")
work_data['company_size'] = work_data['company_size'].replace('M', "Medium")
work_data['company_size'] = work_data['company_size'].replace('S', "Small")
# Replace values in remote ratio column and change data type
work_data['remote_ratio'] = work_data['remote_ratio'].replace(0, "None remote")
work_data['remote_ratio'] = work_data['remote_ratio'].replace(50, "Hybrid")
work_data['remote_ratio'] = work_data['remote_ratio'].replace(100, "Fully remote")
work_data['remote_ratio'] = work_data['remote_ratio'].astype(object)


# New data
work_data=work_data.reset_index(drop = True)
work_data.head()
```

Out[59]:

| | work_year | experience_level | employment_type | job_title | salary | salary_currency | salary_in_usd | en |
|---|---|---|---|---|---|---|---|---|
| **0** | 2020 | Senior-Level | Full-Time | Data Scientist | 60000 | EUR | 68428 | |
| **1** | 2021 | Entry-Level | Full-Time | Data Scientist | 13400 | USD | 13400 | |
| **2** | 2021 | Mid-Level | Full-Time | Data Scientist | 95000 | CAD | 75966 | |
| **3** | 2021 | Mid-Level | Full-Time | Data Scientist | 150000 | USD | 150000 | |
| **4** | 2021 | Mid-Level | Full-Time | Data Scientist | 50000 | USD | 50000 | |

# ♦ Checking for null values

In [60]:
```
work_data.info()
data_salary[data_salary.isnull()].count()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 117 entries, 0 to 116
Data columns (total 11 columns):
 #   Column             Non-Null Count  Dtype 
---  ------             --------------  ----- 
 0   work_year          117 non-null    int64 
 1   experience_level   117 non-null    object
 2   employment_type    117 non-null    object
 3   job_title          117 non-null    object
 4   salary             117 non-null    int64 
 5   salary_currency    117 non-null    object
 6   salary_in_usd      117 non-null    int64 
 7   employee_residence 117 non-null    object
 8   remote_ratio       117 non-null    object
 9   company_location   117 non-null    object
 10  company_size       117 non-null    object
dtypes: int64(3), object(8)
memory usage: 10.2+ KB
```

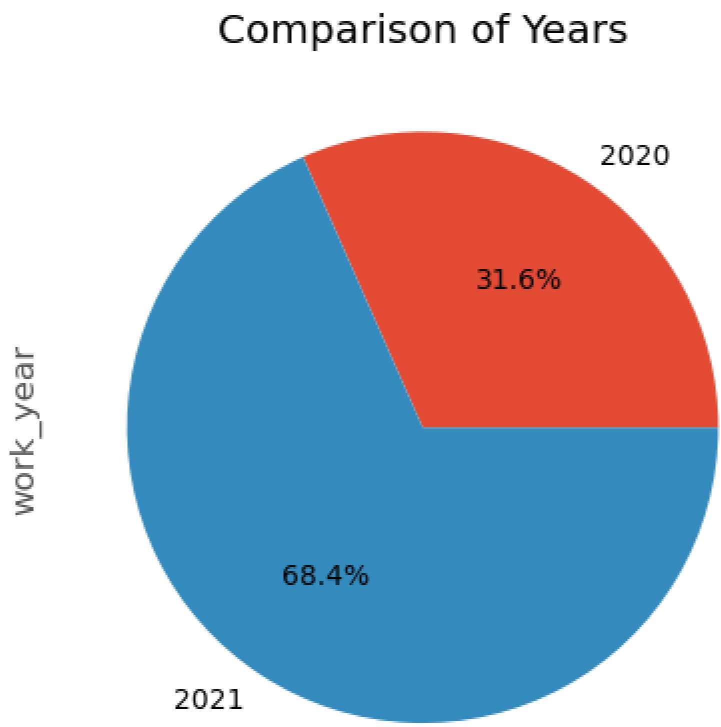
Out[60]:

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

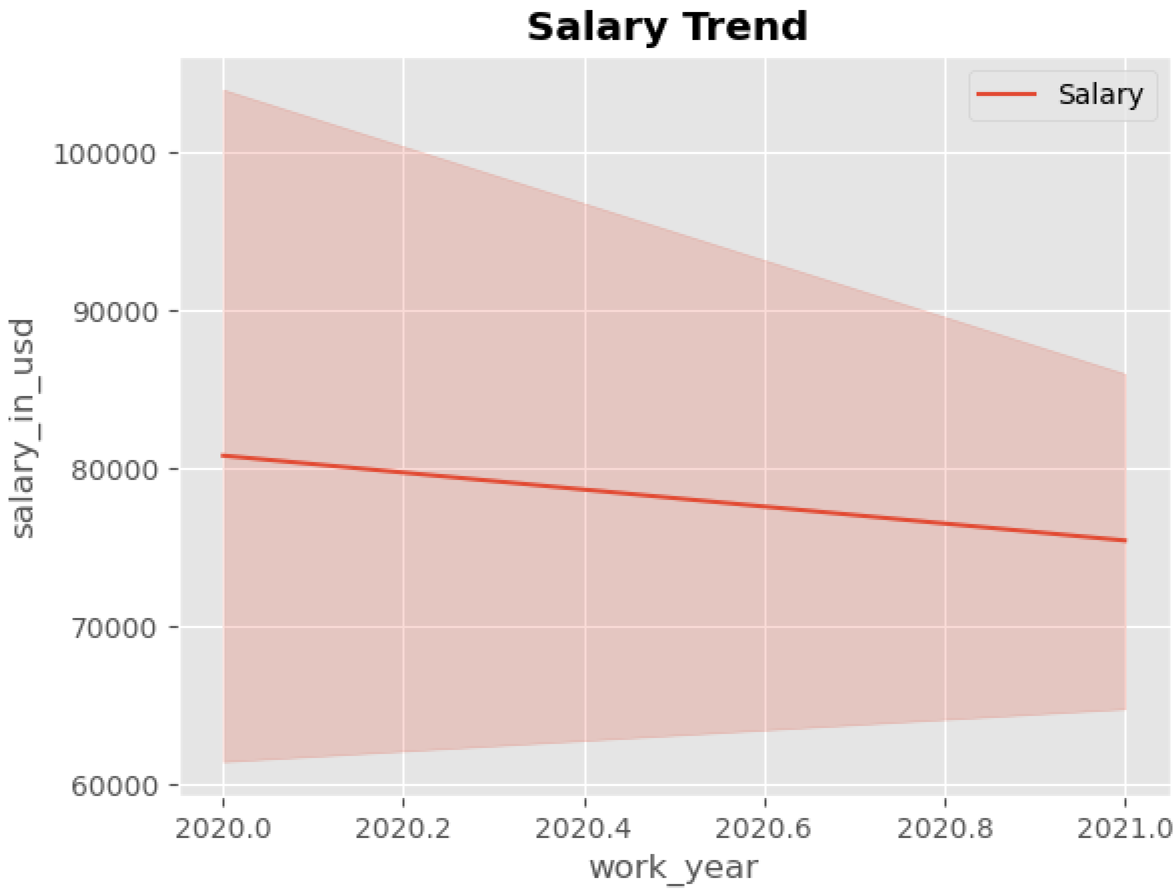
# ♦ Trend of the Data

In [61]:

```
#Year employees joined the domain
work_data.groupby('work_year')['work_year'].count().plot.pie(autopct='%1.1f%%')
plt.title('Comparison of Years')
plt.style.use('default')
```




In [63]:
```
#plottings work year by salary
sns.lineplot(data =work_data ,x = 'work_year', y = 'salary_in_usd')
plt.title('Salary Trend ', fontweight='bold')
plt.legend(['Salary'])
plt.show()
plt.style.use('ggplot')
```




- As the year increase from **2020** to **2021**, the job demand increases by almost 50% to that of the previous year.
- This leads to a decrease in the salary relatively to the increase in year.
- **Work_year** and **salary_in_usd** thus have a negative correlation

# ♦Summary Statistics

In [64]:
```
#Finding pearson correlation
work_data[['work_year','salary','salary_in_usd']].corr(method='pearson')
```

Out[64]:

| | work_year | salary | salary_in_usd |
|---|---|---|---|
| **work_year** | 1.000000 | 0.001543 | -0.045937 |
| **salary** | 0.001543 | 1.000000 | -0.113719 |
| **salary_in_usd** | -0.045937 | -0.113719 | 1.000000 |

- `Salary_in_usd` has a negative correlation of approximately -0.11 with `salary.` This suggests that as `Salary_in_usd` increases, `salary` tends to decrease slightly. This is accurate since the conversion rate of various currencies to USD are not equal.
- `Salary_in_usd` and `work_year` have a negative correlation of around -0.046. This indicates that as `Salary_in_usd` goes up, `work_year` also tends to decrease. This was proven in our figure when we were analyzing the trend.

In [65]:
```
print('--Summary statistics of the 3 job types--')
work_data.groupby('job_title')['salary_in_usd'].describe(include='all').T
```

```
--Summary statistics of the 3 job types--
```

Out[65]:

| job_title | Data Analyst | Data Engineer | Data Scientist |
|---|---|---|---|
| **count** | 20.000000 | 38.000000 | 59.000000 |
| **mean** | 69329.150000 | 82177.526316 | 76537.101695 |
| **std** | 40733.009666 | 50228.678867 | 61441.841082 |
| **min** | 6072.000000 | 4000.000000 | 2876.000000 |
| **25%** | 57654.250000 | 35555.500000 | 38460.000000 |
| **50%** | 71984.000000 | 73377.500000 | 62726.000000 |
| **75%** | 81250.000000 | 111943.750000 | 104477.000000 |
| **max** | 200000.000000 | 200000.000000 | 412000.000000 |

**Data Engineer** records the highest average salary with a count of 38 people.

In [66]:
```
print('--Further information after grouping by experience level and work year--')

work_data.groupby(['job_title', 'experience_level','work_year'])['salary_in_usd'].desc
```

```
--Further information after grouping by experience level and work year--
```

Out[66]:

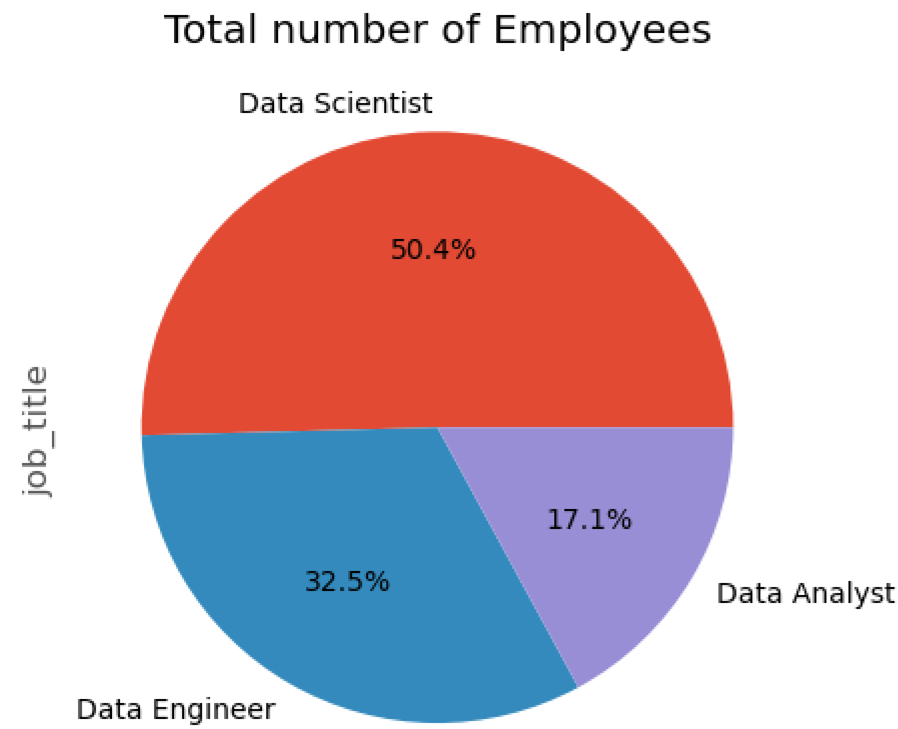
| job_title | experience_level | work_year | count | mean | std | min | 25% | 5 |
|---|---|---|---|---|---|---|---|---|
| **Data Analyst** | **Entry-Level** | **2020** | 4.0 | 44768.000000 | 43147.707116 | 6072.0 | 9018.00 | 410 |
| | | **2021** | 4.0 | 72400.250000 | 15111.743786 | 59601.0 | 59900.25 | 700 |
| | **Mid-Level** | **2020** | 3.0 | 46586.333333 | 38500.290393 | 8000.0 | 27379.50 | 467 |
| | | **2021** | 5.0 | 72362.800000 | 15975.109364 | 51814.0 | 62000.00 | 750 |
| | **Senior-Level** | **2021** | 4.0 | 104084.250000 | 64261.540579 | 64369.0 | 70068.25 | 759 |
| **Data Engineer** | **Entry-Level** | **2020** | 1.0 | 41689.000000 | NaN | 41689.0 | 41689.00 | 416 |
| | | **2021** | 4.0 | 47566.250000 | 25204.326816 | 21695.0 | 28305.50 | 480 |
| | **Mid-Level** | **2020** | 6.0 | 100656.833333 | 23700.953672 | 70139.0 | 82097.50 | 1080 |
| | | **2021** | 18.0 | 70173.722222 | 51703.125589 | 4000.0 | 28656.25 | 683 |
| | **Senior-Level** | **2020** | 3.0 | 89803.333333 | 85344.552517 | 33511.0 | 40705.00 | 478 |
| | | **2021** | 6.0 | 125719.000000 | 34636.205000 | 77481.0 | 101374.75 | 1325 |
| **Data Scientist** | **Entry-Level** | **2020** | 5.0 | 56126.400000 | 31246.459196 | 21669.0 | 39916.00 | 513 |
| | | **2021** | 8.0 | 45082.250000 | 34777.764591 | 4000.0 | 24706.25 | 333 |
| | **Mid-Level** | **2020** | 11.0 | 71256.000000 | 35724.454515 | 35735.0 | 41339.00 | 627 |
| | | **2021** | 24.0 | 73049.125000 | 47946.370410 | 2876.0 | 37082.75 | 674 |
| | **Senior-Level** | **2020** | 4.0 | 172916.250000 | 160779.832169 | 68428.0 | 85534.75 | 1056 |
| | | **2021** | 7.0 | 92248.428571 | 48160.829086 | 21843.0 | 65990.50 | 879 |

## ♦ Comparing the 3 job types

In [67]:

```
plotdata = work_data['job_title'].value_counts()
plotdata.plot.pie(autopct='%1.1f%%')
plt.title('Total number of Employees')
print(plotdata)
```

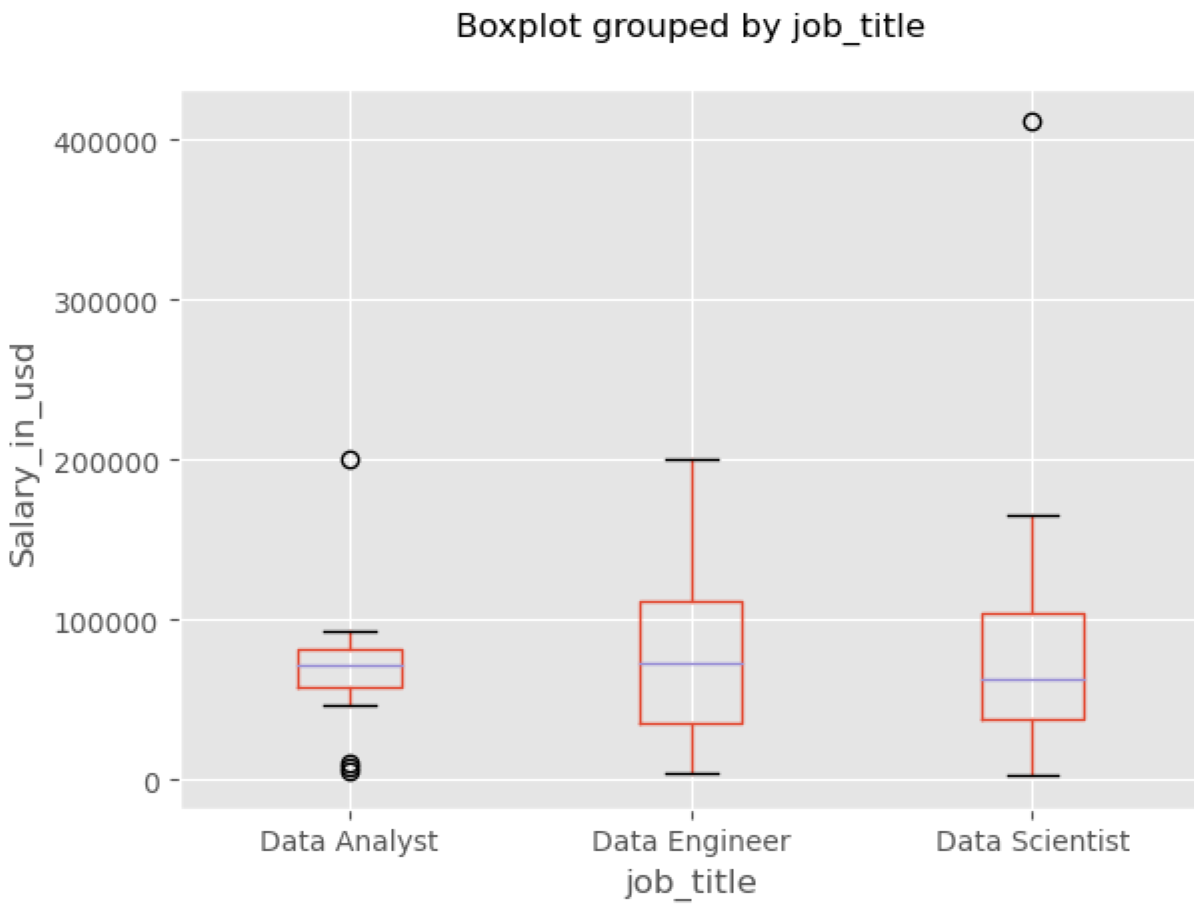
```
Data Scientist    59
Data Engineer     38
Data Analyst      20
Name: job_title, dtype: int64
```




In our dataset, we notice that data scientist have half the count of our cleaned dataset. This implies that, data science occupies a huge proportion than the other 2 data field.

In [68]:
```
work_data.boxplot(column='salary_in_usd',by='job_title')
plt.ylabel('Salary_in_usd')
plt.title('')
```

Out[68]:
```
Text(0.5, 1.0, '')
```

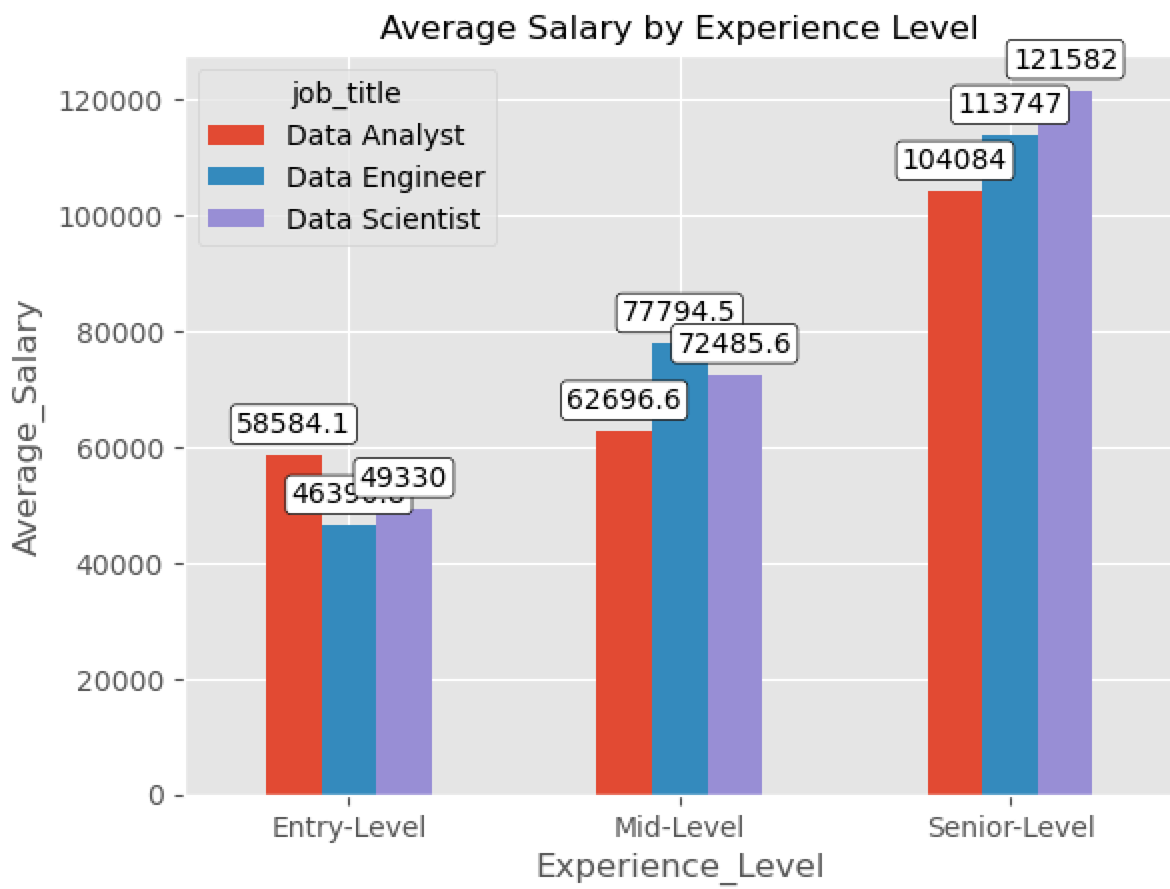



Overall, the salaries for data analysts, data engineers, and data scientists are all relatively high. Data Engineers have the highest median salary, followed by data analyst and then data scientist. There are a few possible explanations for these salary differences. One of these is the fact that data scientists are in higher demand than data analysts or data engineers. There are a few outliners in the data analyst and data scientist field indicating earnings beyond the maximum and the minimum salary.
This could have been caused by various factors such as individuals being extremely good at their work or on the other hand being bad at it. It could have also been as a result of improper record taking. A further analysis of the data will explain it further.

## ♦ Comparison by Experience Levels

In [69]:
```
exp=work_data.groupby([ 'experience_level','job_title'])['salary_in_usd'].mean().unsta
plt.ylabel('Average_Salary')
plt.xlabel('Experience_Level')
plt.style.use('default')
plt.title('Average Salary by Experience Level')
plt.xticks(rotation=0)
```

```
plt.style.use('ggplot')

#Place values above chart
for container in exp.containers:
    exp.bar_label(container, label_type="edge", color="black",
                 padding=6, bbox={'boxstyle': 'round,pad=0.2', 'facecolor': 'white', '
```




- In the Entry - Level, Data Analysis receive the highest average income
- In the Mid - Level, Data Engineer receive the highest average income.
- In the Senior - Level, Data Scientist receive the highest average income.

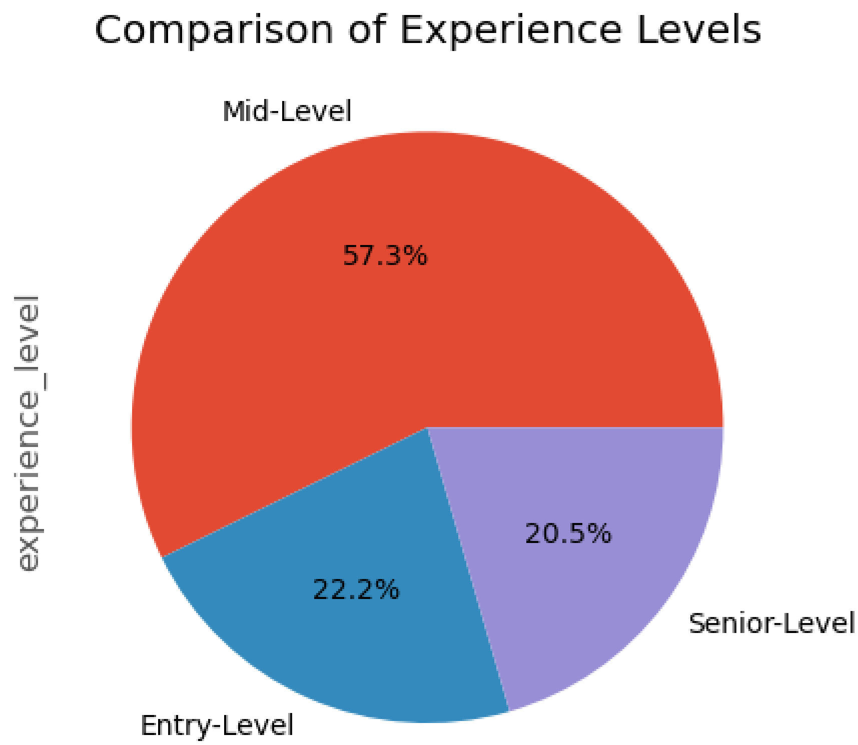
In [70]:
```
work_data[work_data['company_location']=='SG']
```

Out[70]:

| | work_year | experience_level | employment_type | job_title | salary | salary_currency | salary_in_usd | emp |
|---|---|---|---|---|---|---|---|---|

In [71]:
```
work_data['experience_level'].value_counts().plot.pie(autopct='%1.1f%%')
plt.title('Comparison of Experience Levels')
```

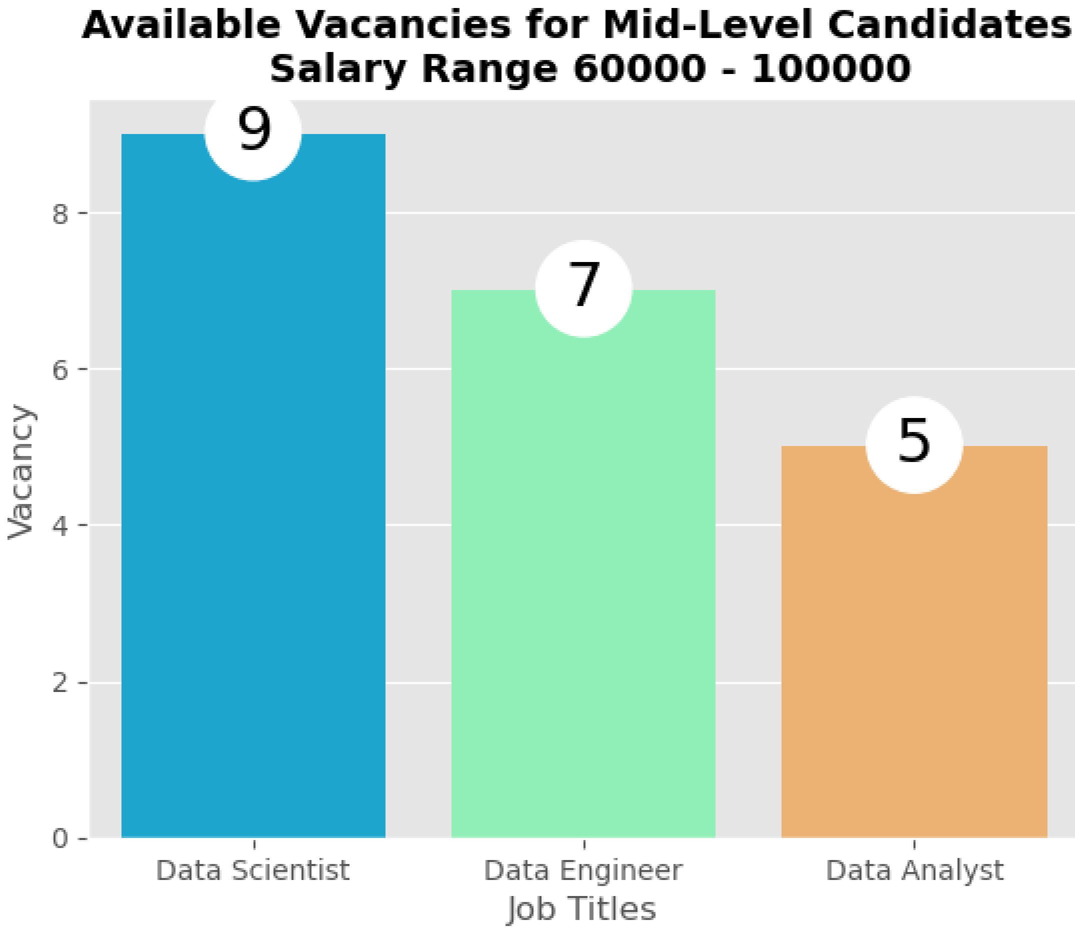
Out[71]:
```
Text(0.5, 1.0, 'Comparison of Experience Levels')
```




The skill set of most workers lie in the Mid - Level Tier

In [72]:

```
exp_level = 'Mid-Level' #Check for this experience level
salary_range = (60000, 100000)  # check this range
range_level = work_data[(work_data['experience_level'] == exp_level) &
                 (work_data['salary_in_usd'] >= salary_range[0]) &
                 (work_data['salary_in_usd'] <= salary_range[1])]
available = range_level['job_title'].value_counts().reset_index() # Count for each job
available.columns = ['Job Title', 'Count'] # Change headers

p=sns.barplot(y='Count', x='Job Title', data=available, palette = 'rainbow')
plt.ylabel('Vacancy')
plt.xlabel('Job Titles')
plt.title(f'Available Vacancies for {exp_level} Candidates \n Salary Range {salary_rar
plt.style.use('default')

for container in p.containers:
    p.bar_label(container, padding=-10,fontsize=22,
                 bbox={'boxstyle': 'circle,pad=0.3', 'facecolor': 'white', 'edgecolor'
```



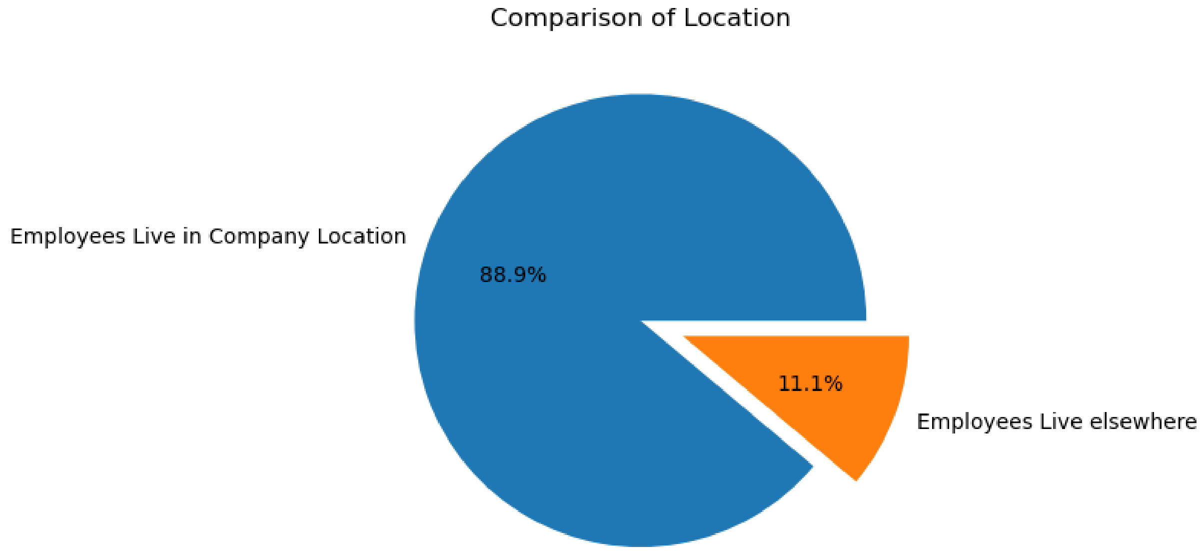

*Mid-Level vacancy available*

- Data Analyst - 5 job openings
- Data Scientist - 9 job openings
- Data Engineer -7 job openings

# ♦ Further Statistics of location

In [73]:
```
#Distribution of employees based on employees residence
in_loc = work_data[work_data['employee_residence']==work_data['company_location']]
out_loc = work_data[work_data['employee_residence']!= work_data['company_location']]
loc = in_loc.count()['work_year'], out_loc.count()['work_year']
plt.pie(loc,labels=['Employees Live in Company Location', 'Employees Live elsewhere'
plt.style.use('default')
plt.title('Comparison of Location')
```

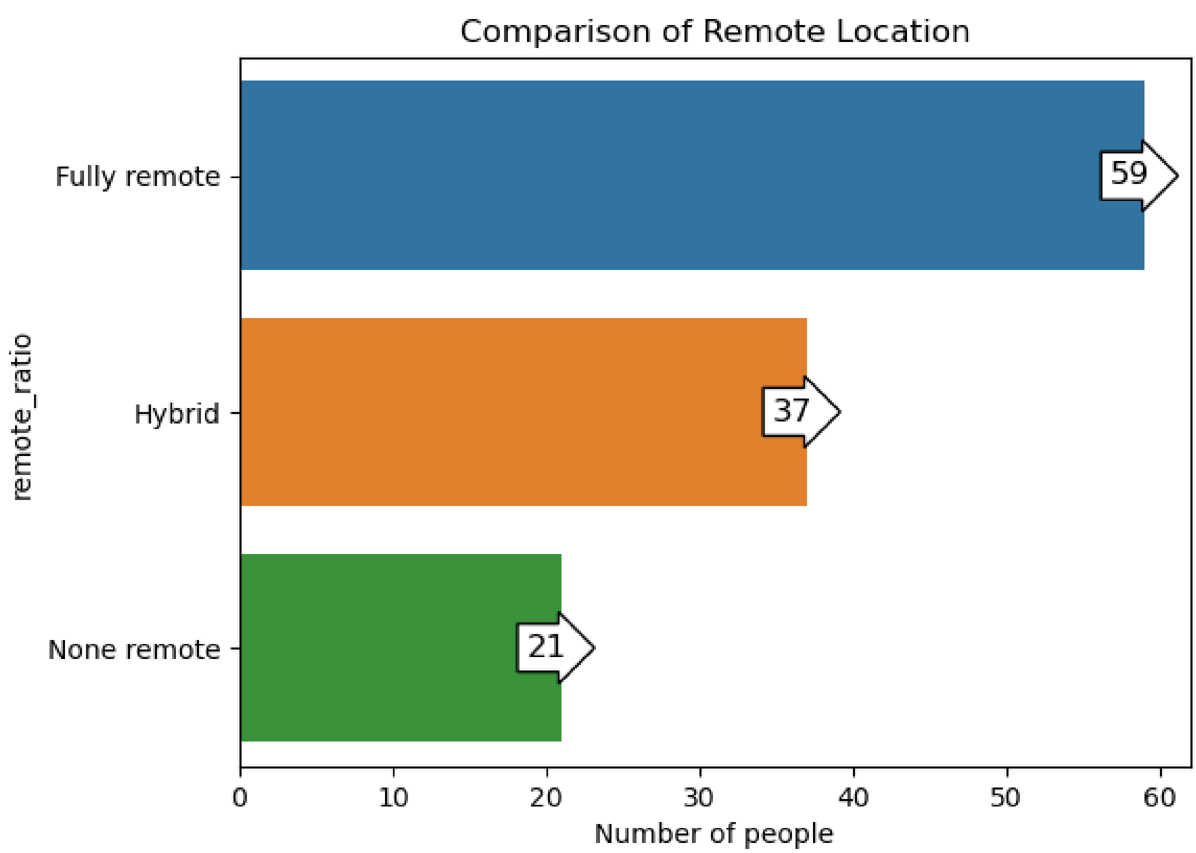
Out[73]:
```
Text(0.5, 1.0, 'Comparison of Location')
```




In [74]:
```
loc
```

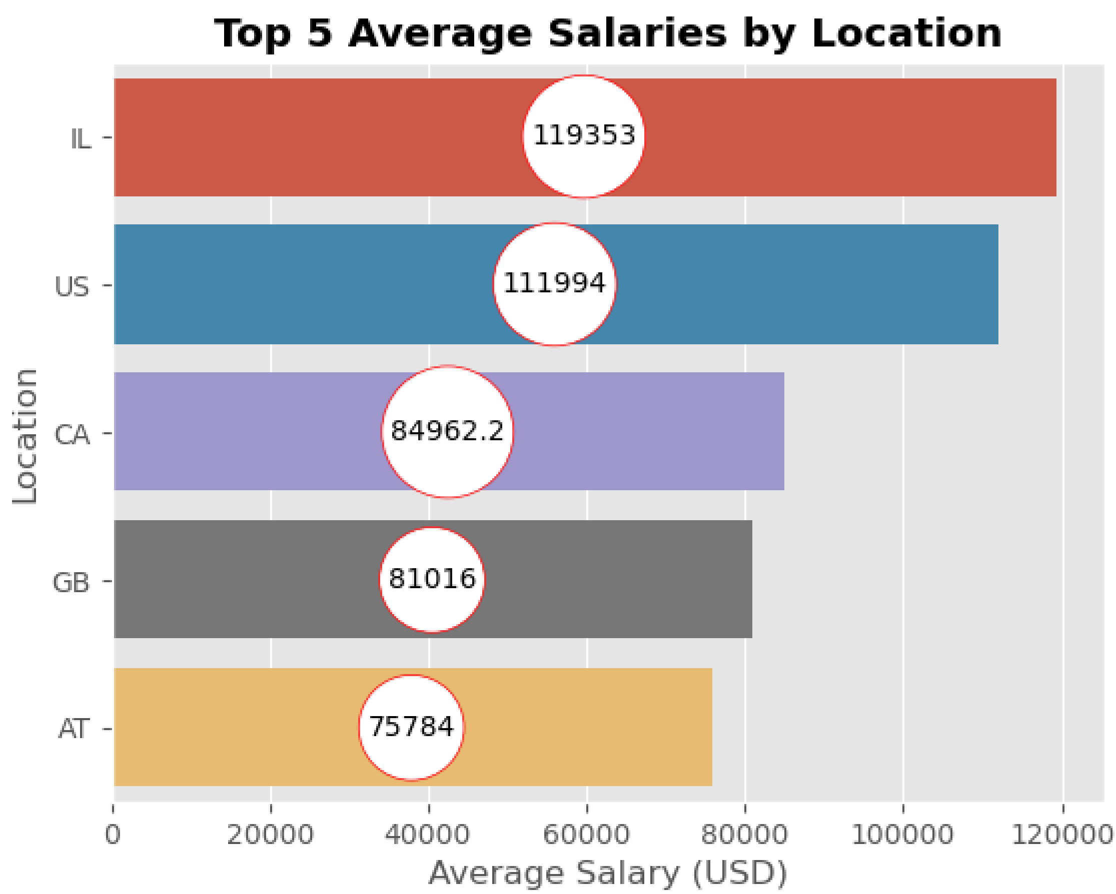
Out[74]:
```
(104, 13)
```

In [76]:
```
remote=work_data.groupby('remote_ratio')['remote_ratio'].count()
ratio=sns.barplot(y=remote.index,x=remote.values)
plt.xlabel('Number of people')
plt.title('Comparison of Remote Location')
for container in ratio.containers:
    ratio.bar_label(container, label_type="edge", color="black",
                 padding=-13,fontsize=12,bbox={'boxstyle': 'rarrow', 'facecolor': 'wh
```



Despite a huge population living in the country of their company's location, most prefer to work remotely

In [78]:
```
# Group the data by company_location and calculate the mean salary for each location
salary_location = work_data.groupby('company_location')['salary_in_usd'].mean().rese

# Sort the locations by average salary in descending order
salary_location = salary_location.sort_values(by='salary_in_usd', ascending=False)
# Create a bar chart to visualize average salaries by country
location = sns.barplot(x='salary_in_usd', y='company_location', data=salary_location
plt.title('Top 5 Average Salaries by Location', fontweight='bold' )
plt.xlabel('Average Salary (USD)')
plt.ylabel('Location')
plt.style.use('ggplot')
for container in location.containers:
    location.bar_label(container, bbox = {'boxstyle': 'circle', 'edgecolor': 'red',
                label_type="center",
               )
```




*Top 5 Countries*

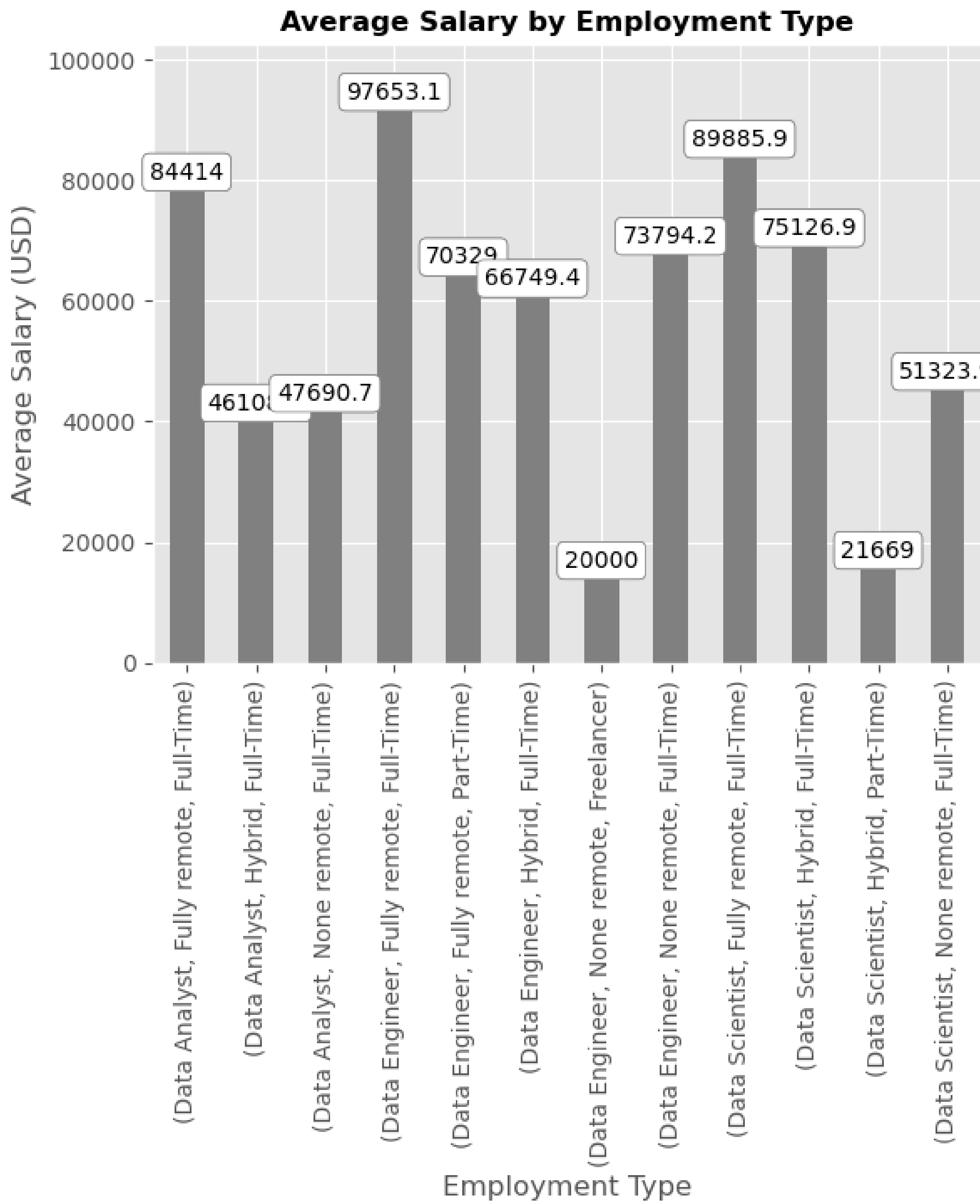
- **Illinois (IL)** records the highest average data salary at approximately *119353 USD.*

- **United States (US)** and **Canada(CA)** also offers a competitive average salaries, with approximately *111994 USD and 84962.2 USD*, respectively.
- **Great Britain (GB)** and **Austria (AT)** round up the top 5 locations with varying average salaries of *81016 USD and 75784 USD* .

# ♦ Further Statistics

In [79]:

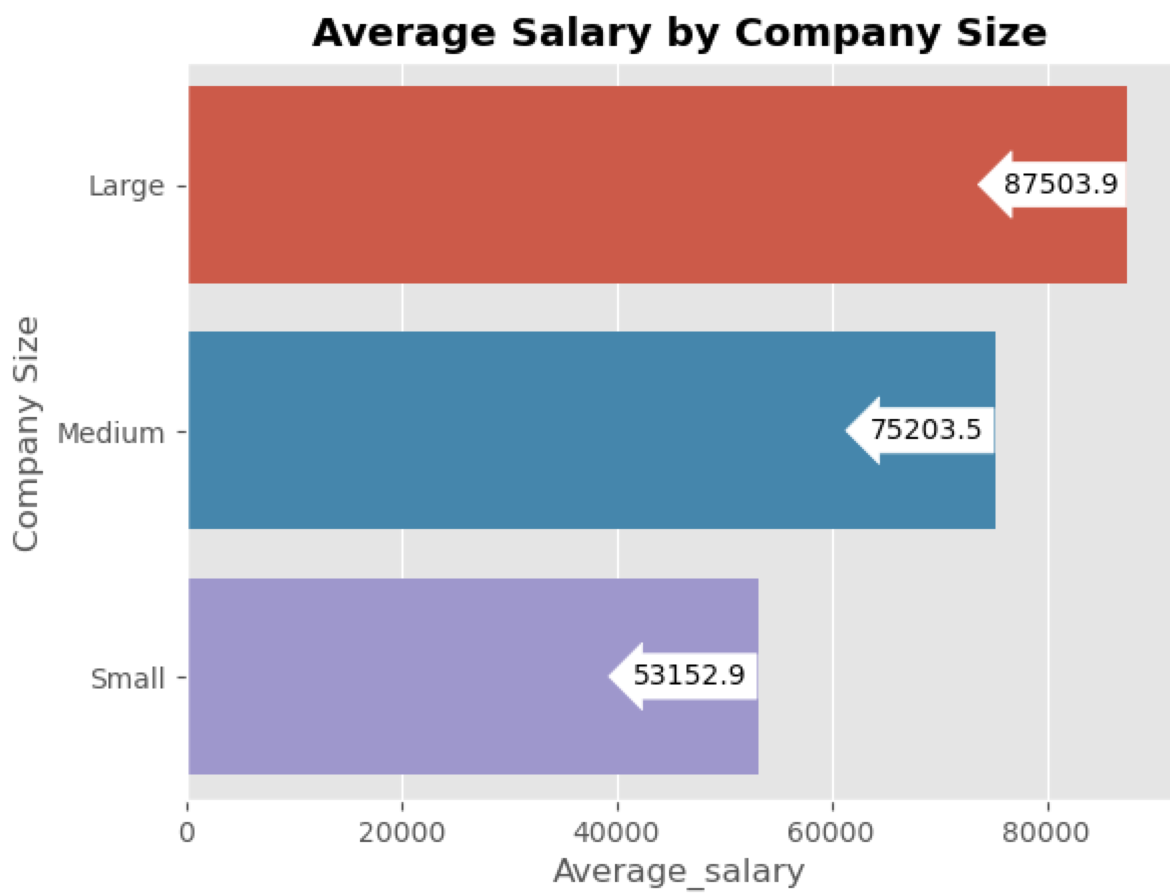
```
#Group data by 'employment_type' and calculate the average salary for each type
emp_salary = work_data.groupby(['job_title','remote_ratio','employment_type'])['sal

emp = emp_salary.plot(kind='bar',color='gray')
plt.title('Average Salary by Employment Type', fontsize=12, fontweight='bold')
plt.xlabel('Employment Type')
plt.ylabel('Average Salary (USD)')
plt.style.use('ggplot')
emp_salary = work_data.groupby(['job_title','remote_ratio','employment_type'])['sal

emp = emp_salary.plot(kind='bar',color='gray')
plt.title('Average Salary by Employment Type', fontsize=12, fontweight='bold')
plt.xlabel('Employment Type')
plt.ylabel('Average Salary (USD)')
plt.style.use('ggplot')

# labels on chart
for container in emp.containers:
    emp.bar_label(container, label_type="edge", color="black",padding=-13,
                  bbox={'boxstyle': 'round', 'facecolor': 'white', 'edgecolor': 'gra
```




Grouping the **remote_ratio** by **employment_type**, those working **Fully remote and Full-Time** obtain the highest average salary with **91298.2 USD**. This is earned by **Data Engineers**.

In [80]:
```
company_size_salary = work_data.groupby('company_size')['salary_in_usd'].mean()
p = sns.barplot(y=company_size_salary.index, x=company_size_salary.values)
plt.title('Average Salary by Company Size',  fontweight='bold')
plt.ylabel('Company Size')
plt.xlabel('Average_salary')
```

```
# labels on chart
for container in p.containers:
    p.bar_label(container, label_type="edge", color="black",
                 padding=-45,bbox={'boxstyle': 'larrow', 'facecolor': 'white', 'edg
```




Large companies tend to pay a higher average of about 87500 to their employees than that of Medium and Small companies.

# 3. HYPOTHESIS TEST

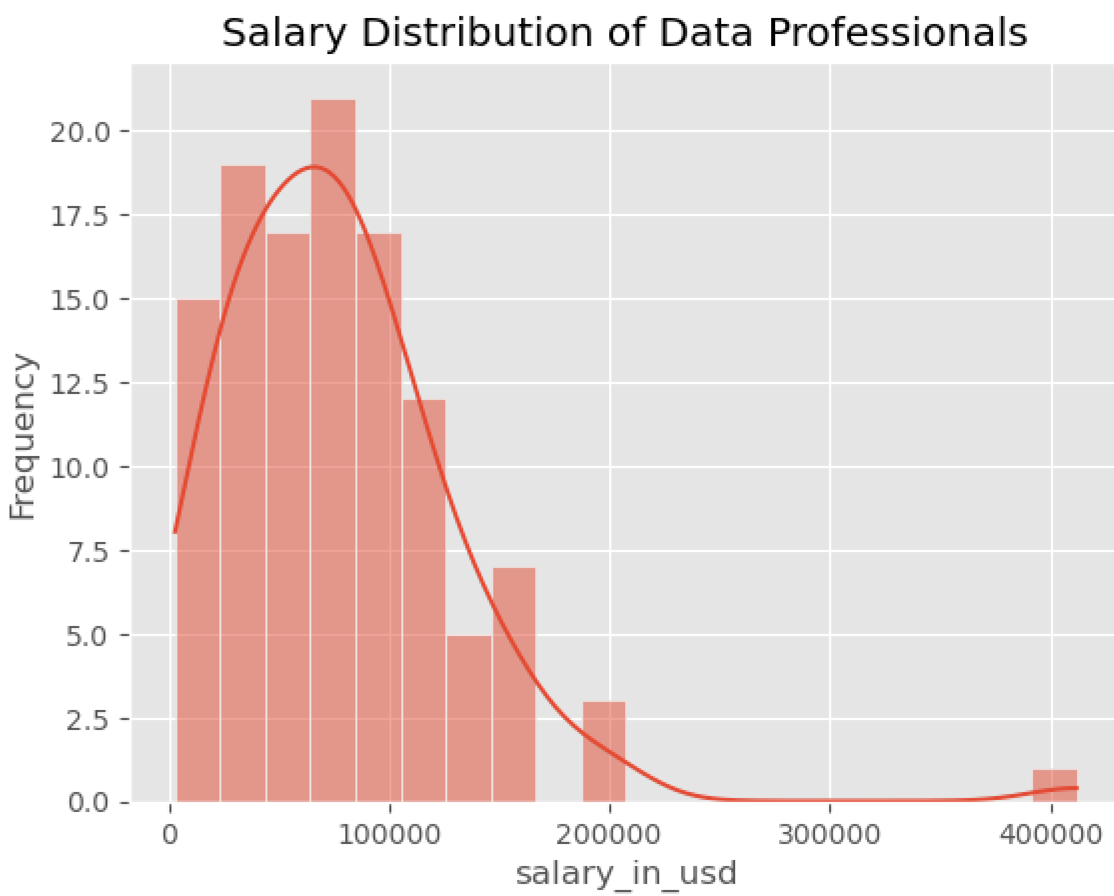
*TESTING OF THE HYPOTHESIS*

**Null hypothesis:** Data scientist earn a high amount of money in USD compared to the other parallel professions.
**Alternate hypothesis:** Data scientist do not earn a high amount of money in USD compared to the other parallel professions.

# ♦ Checking for normality whith Histogram

In [81]:
```
sns.histplot(work_data['salary_in_usd'], kde=True,bins=20)
plt.title("Salary Distribution of Data Professionals")
plt.style.use('ggplot')
plt.ylabel('Frequency')
```

Out[81]:
```
Text(0, 0.5, 'Frequency')
```




The histogram is skewed to the right and thus, the data is not symmetric. This implies that, it is not normally distributed

# ♦ Checking for normality with Shapiro Test

In [82]:
```
# Check for normality
x=data_scientist['salary_in_usd']
```

```
y=data_analyst['salary_in_usd']
z=data_engineer['salary_in_usd']
stat, p=shapiro(x)
print(' Stat = ',round(stat,2) ,'\n P-value = ', round(p,8))
if p > 0.05:
    print(' The data is normally distributed')
else:
    print(' The data is not normally distributed')
```

```
Stat =  0.77
P-value =  3e-08
The data is not normally distributed
```

Since the data is not not normally distributed, we therefore cannot use parametric test for the hypothesis test. We will have to use non-parametric test to check the hypothesis.

# ♦ Using non-parametric test (Kruskal Wallis)

In [83]:

```
statistics , p_value=kruskal(x,y,z)
print(' Stat = ',round(statistics,2) ,'\n P-value = ', round(p_value,2))
if p_value < 0.05:
    print(' We reject the null hypothesis. \n There is significant evidence to sugg
else:
    print(' We fail to reject the null hypothesis. \n There is no significant evide
```

```
Stat =  0.97
P-value =  0.62
We fail to reject the null hypothesis.
There is no significant evidence to suggest differences between the salaries.
```

This means that our null hypothesis holds. That is: **data scientist earn a high amount of money in USD compared to the other parallel professions.**

*The end*

By: *Musah Faridu Oubda*