

Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge.

PROJECT2: FREELANCER EARNING DATASET CLEANING & VISUALIZATION

STEP1 : Import libraries

```
1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 import seaborn as sns
```

STEP2 : Load a dataset

```
1 df = pd.read_csv("/content/freelancer_earnings_bd.csv")
```

STEP3 : Data cleaning

1. Finding statistical information

```
1 df.describe()
```

	Freelancer_ID	Job_Completed	Earnings_USD	Hourly_Rate	Job_Success_Rate	Client_Rating	Job_Duration_Days	Rehire_Rate	Marketing_Spend
count	1950.000000	1950.000000	1950.000000	1950.000000	1950.000000	1950.000000	1950.000000	1950.000000	1950.000000
mean	975.500000	150.864103	5017.566667	52.579774	74.951846	3.995892	44.704615	44.556913	24.704615
std	563.060832	85.480770	2926.279918	26.927584	14.615735	0.575976	26.022998	20.193842	14.704615
min	1.000000	5.000000	51.000000	5.020000	50.160000	3.000000	1.000000	10.000000	1.000000
25%	488.250000	76.000000	2419.000000	30.047500	61.917500	3.510000	22.000000	27.150000	11.000000
50%	975.500000	149.000000	5048.000000	52.285000	75.400000	3.990000	45.000000	43.920000	25.000000
75%	1462.750000	225.000000	7608.250000	75.125000	87.537500	4.500000	67.000000	61.690000	37.000000
max	1950.000000	299.000000	9991.000000	99.830000	99.990000	5.000000	89.000000	79.950000	49.000000

2. Finding all type information

```
1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1950 entries, 0 to 1949
Data columns (total 15 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Freelancer_ID         1950 non-null   int64
1   Job_Category          1950 non-null   object
2   Platform              1950 non-null   object
3   Experience_Level      1950 non-null   object
4   Client_Region         1950 non-null   object
5   Payment_Method        1950 non-null   object
6   Job_Completed         1950 non-null   int64
7   Earnings_USD          1950 non-null   int64
8   Hourly_Rate           1950 non-null   float64
9   Job_Success_Rate      1950 non-null   float64
10  Client_Rating          1950 non-null   float64
11  Job_Duration_Days      1950 non-null   int64
12  Project_Type           1950 non-null   object
13  Rehire_Rate            1950 non-null   float64
14  Marketing_Spend        1950 non-null   int64
dtypes: float64(4), int64(5), object(6)
memory usage: 228.6+ KB
```

3. Find missing values sum

```
1 df.isna().sum()
```

	0
Freelancer_ID	0
Job_Category	0
Platform	0
Experience_Level	0
Client_Region	0
Payment_Method	0
Job_Completed	0
Earnings_USD	0
Hourly_Rate	0
Job_Success_Rate	0
Client_Rating	0
Job_Duration_Days	0
Project_Type	0
Rehire_Rate	0
Marketing_Spend	0

df = df.astype(np.int64)

4. Find duplicates values

```
1 df.duplicated().sum()
```

np.int64(0)

5. Find duplicates value from freelancer Id column

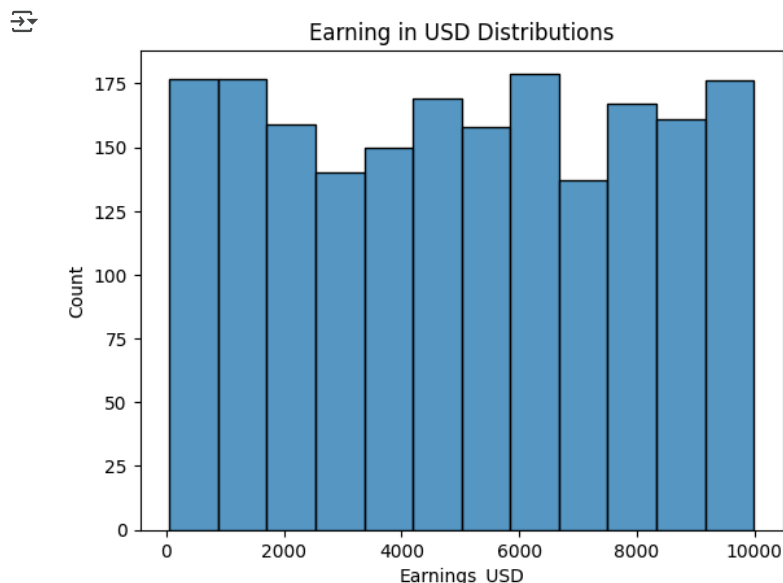
```
1 df["Freelancer_ID"].duplicated().sum()
```

np.int64(0)

✓ STEP4 : Data visualization

QUE1 : Which earning interval has the highest frequency in this distribution?

```
1 sns.histplot(data=df, x="Earnings_USD")
2 plt.title("Earning in USD Distributions")
3 plt.show()
```

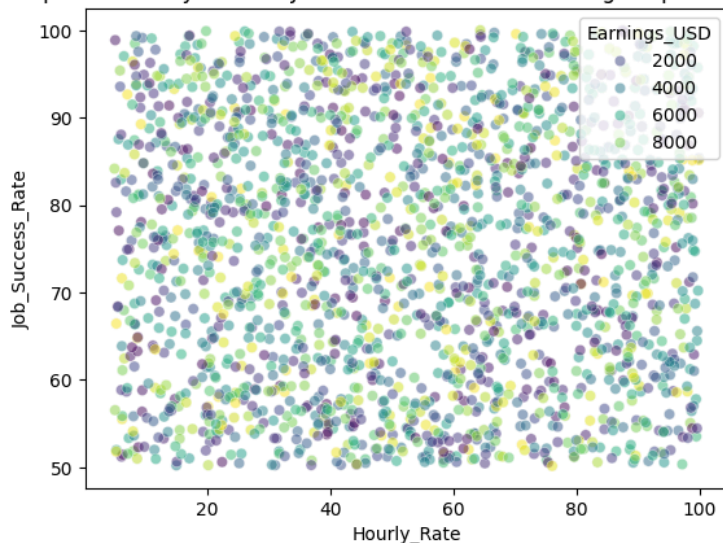


CONCLUSION : The bar chart reveals the frequency distribution of earnings in USD across different income intervals, with counts ranging from approximately 125 to 175.

QUE2 : Does the chart suggest any correlation between hourly rate and job success rate?

```
1 sns.scatterplot(data=df, x="Hourly_Rate", y="Job_Success_Rate", hue="Earnings_USD", palette="viridis", alpha=0.5)
2 plt.title("Impact of Hourly Rate on Job Success Rate with Earnings Representation")
3 plt.show()
```

Impact of Hourly Rate on Job Success Rate with Earnings Representation



CONCLUSION : The chart explores the relationship between hourly rates and job success rates, showing no clear connection as the data points are widely scattered.

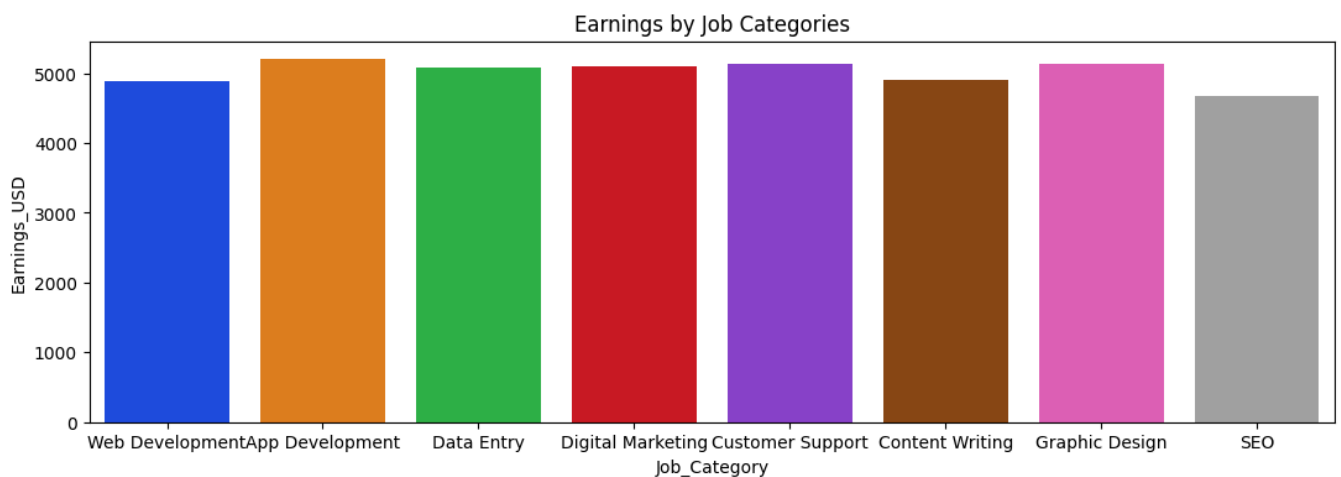
QUE3 : Which job category shown in the chart has the highest earnings?

```
1 plt.figure(figsize=(13, 4))
2 sns.barplot(data=df, x="Job_Category", y="Earnings_USD", palette="bright", errorbar=("ci",0))
3 plt.title("Earnings by Job Categories")
4 plt.show()
```

<ipython-input-15-593afc02f752>:2: FutureWarning:

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

```
sns.barplot(data=df, x="Job_Category", y="Earnings_USD", palette="bright", errorbar=("ci",0))
```



CONCLUSION : The bar chart displays earnings in USD across various job categories, with Digital Marketing having the highest earnings among them.

QUE4 : Which category has the most completed jobs according to the chart?

```
1 plt.figure(figsize=(13, 4))
2 gb = df.groupby("Job_Category")["Job_Completed"].count().reset_index()
```

```

3 print(gb)
4 sns.barplot(data=gb, x="Job_Category", y="Job_Completed", palette="colorblind", errorbar=("ci",0))
5 plt.title("Job Completed by Categories")
6 sns.despine()
7 plt.show()
8 #

```

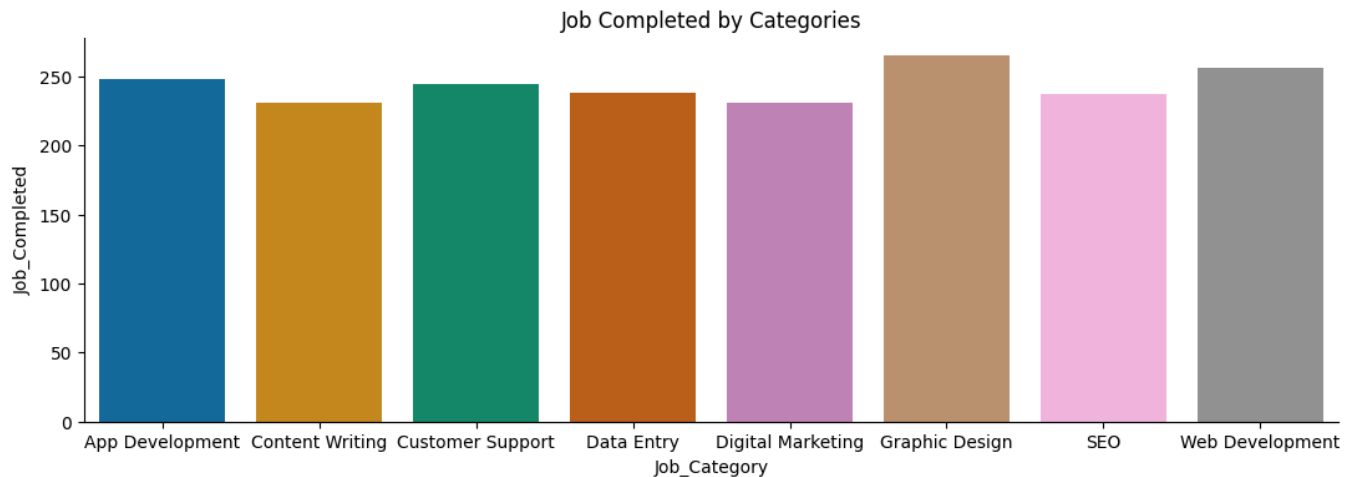
```

↗
   Job_Category  Job_Completed
0   App Development           248
1  Content Writing           231
2  Customer Support           244
3    Data Entry             238
4  Digital Marketing           231
5  Graphic Design           265
6                SEO           237
7  Web Development           256
<ipython-input-17-4478381b07a4>:4: FutureWarning:

```

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

```
sns.barplot(data=gb, x="Job_Category", y="Job_Completed", palette="colorblind", errorbar=("ci",0))
```



CONCLUSION : The bar chart titled "Job Completed by Categories" shows the number of jobs completed across various fields, ranging from approximately 200 to 250 jobs per category.

QUE5 : How does job duration vary among beginners, intermediates, and experts in the chart?

```

1 sns.boxplot(data=df, x="Experience_Level", y="Job_Duration_Days", palette="muted")
2 plt.title("Job Duration Trends for Experience levels")
3 plt.show()

```

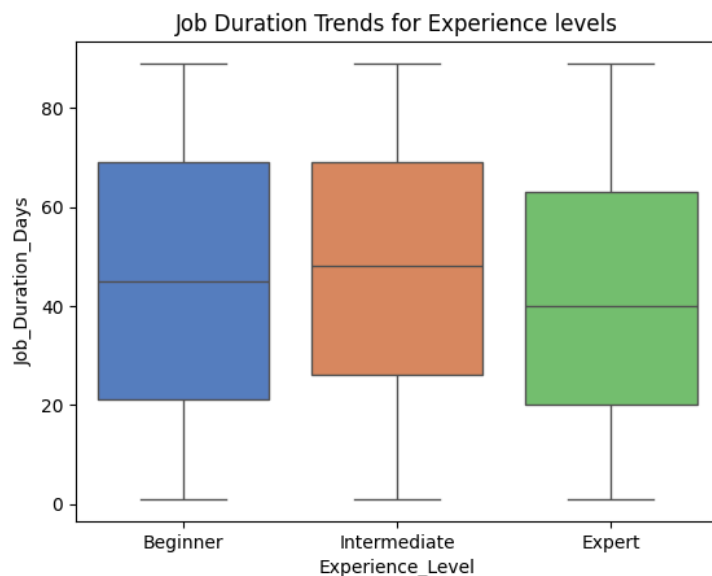
```

↗
<ipython-input-18-97ccf45b5d75>:1: FutureWarning:

```

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


```
sns.boxplot(data=df, x="Experience_Level", y="Job_Duration_Days", palette="muted")
```



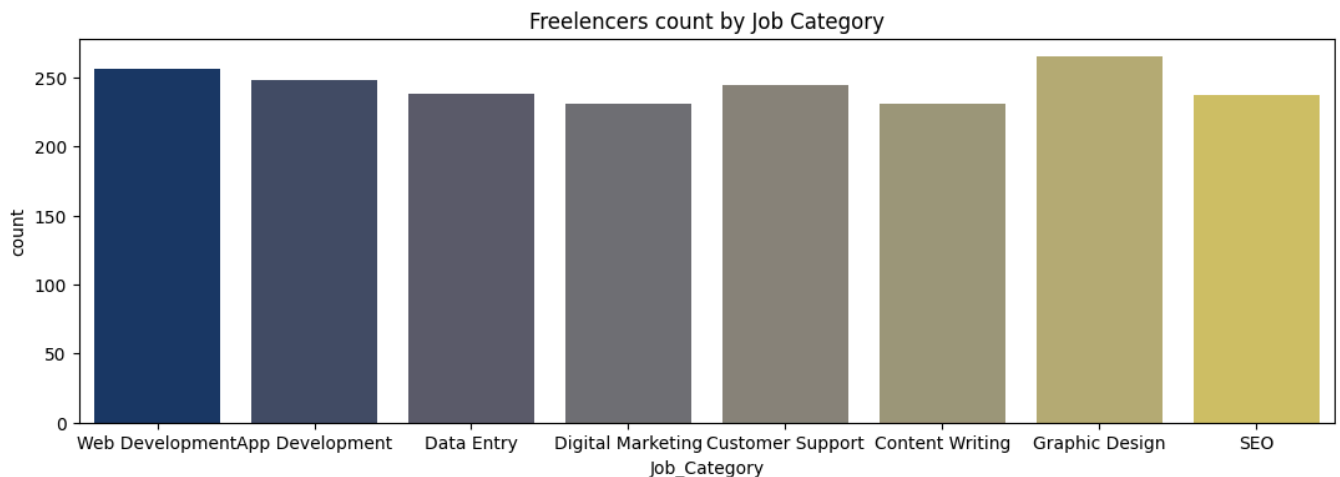
CONCLUSION : The chart highlights that job duration tends to decrease as experience level increases, with experts showing the shortest job durations on average.

QUE6 : Which job category has the highest number of freelancers, as per the chart?

```
1 plt.figure(figsize=(13, 4))
2 sns.countplot(data=df, x="Job_Category", palette="cividis")
3 plt.title("Freelancers count by Job Category")
4 plt.show()
```

 <ipython-input-20-2983d2959fb7>:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend`
 sns.countplot(data=df, x="Job_Category", palette="cividis")



CONCLUSION : The chart reveals that Graphic Design has the highest count of freelancers, while Digital Marketing has the lowest, with other categories distributed in between.

QUE7 : What does the chart suggest about the relationship between job success rates and client ratings?

```
1 plt.figure(figsize=(13, 4))
2 sns.scatterplot(data=df, y="Client_Rating", x="Job_Success_Rate", hue="Job_Success_Rate", palette="plasma")
3 plt.title("Client Rating vs. Job Success Rate")
4 plt.show()
```



CONCLUSION : The scatter plot reveals that higher job success rates generally correspond with higher client ratings, indicating a positive correlation.

QUE8 : Which freelancing platform has the highest median job success rate in the chart?

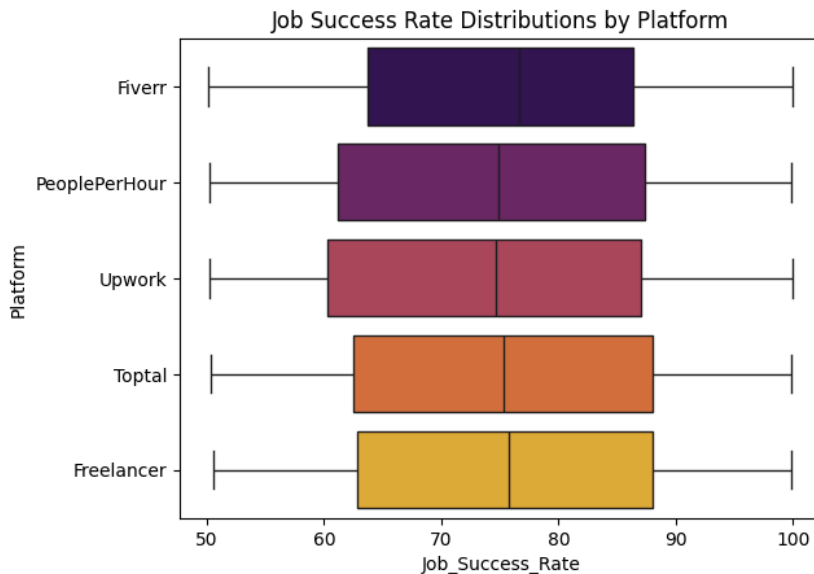
```
1 sns.boxplot(data=df, y="Platform", x="Job_Success_Rate", palette="inferno")
2 plt.title("Job Success Rate Distributions by Platform")
```

```
3 plt.show()
```

```
<ipython-input-23-cc5de287d5ed>:1: FutureWarning:
```

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

```
sns.boxplot(data=df, y="Platform", x="Job_Success_Rate", palette="inferno")
```



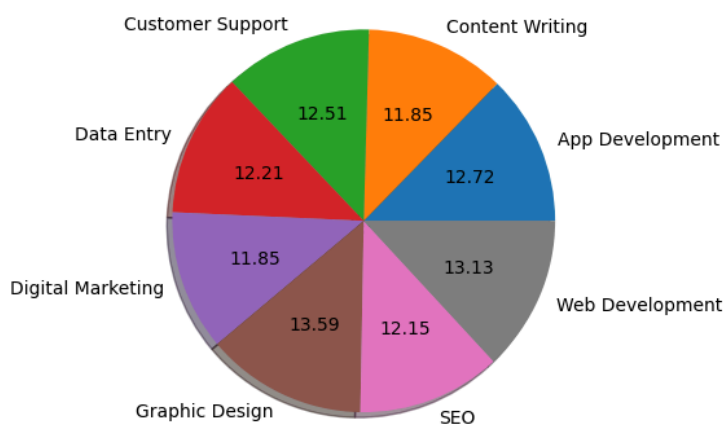
CONCLUSION : The box plot displays the job success rate distributions for five freelancing platforms, highlighting variations in median, range, and potential outliers.

QUE9 : Which job role holds the highest percentage in the distribution?

```
1 gb = df.groupby("Job_Category")["Job_Category"].count()
2 print(gb)
3 plt.pie(gb.values, labels=gb.index, startangle=0, shadow=True, autopct="%.2f")
4 plt.title("Freelancer Job Role Distribution")
5 plt.show()
```

```
Job_Category
App Development      248
Content Writing      231
Customer Support     244
Data Entry           238
Digital Marketing    231
Graphic Design       265
SEO                  237
Web Development      256
Name: Job_Category, dtype: int64
```

Freelancer Job Role Distribution



CONCLUSION : The pie chart depicts the percentage distribution of various job roles, with Graphic Design having the highest share (13.59%) and Content Writing and Digital Marketing tying for the lowest (11.85%).

QUE10 : Which platform has the highest number of freelancers according to the chart?

```

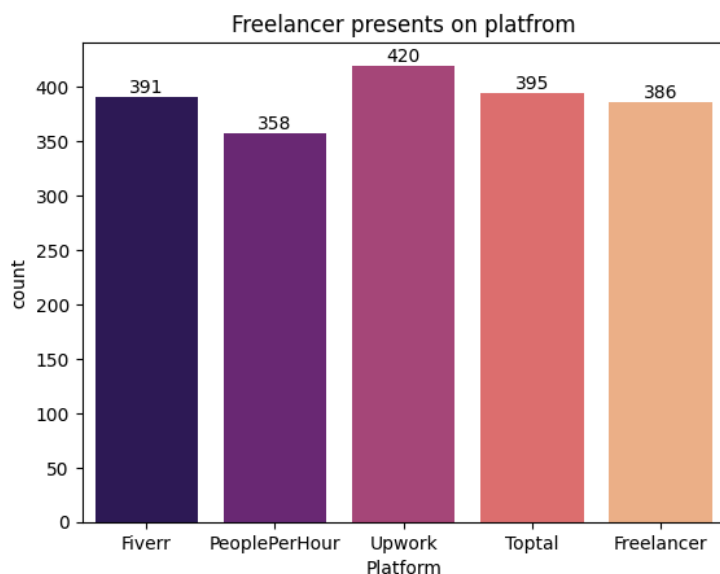
1 platform = sns.countplot(data=df, x="Platform", palette="magma")
2 for i in range(0, 5):
3     platform.bar_label(platform.containers[i])
4 plt.title("Freelancer presents on platform")
5 plt.show()

```

 <ipython-input-25-6aece0f1cb6f>:1: FutureWarning:

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

```
platform = sns.countplot(data=df, x="Platform", palette="magma")
```



CONCLUSION : The bar chart compares freelancer numbers across five platforms, with Upwork leading at 420 and PeoplePerHour trailing at 358.

QUE11 : Which region has the highest number of clients according to the chart?

```

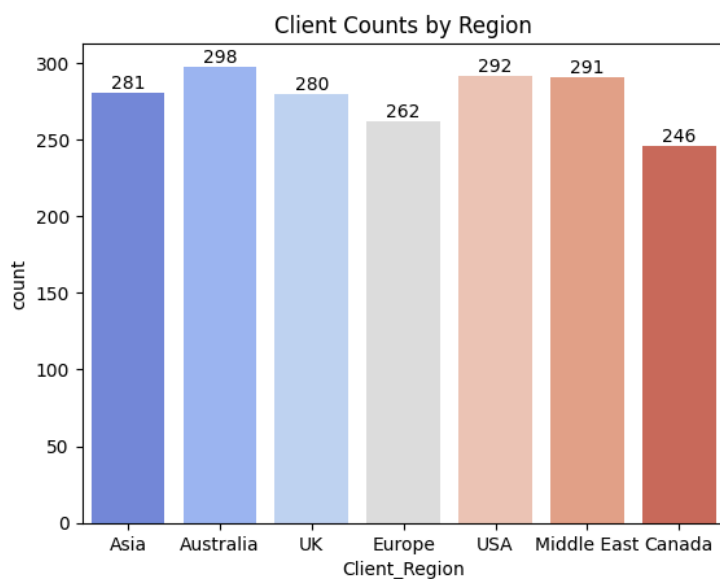
1 cl_Region = sns.countplot(data=df, x="Client_Region", palette="coolwarm")
2 for i in range(0, 7):
3     cl_Region.bar_label(cl_Region.containers[i])
4 plt.title("Client Counts by Region")
5 plt.show()

```

 <ipython-input-26-89a17f2ba20c>:1: FutureWarning:

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

```
cl_Region = sns.countplot(data=df, x="Client_Region", palette="coolwarm")
```



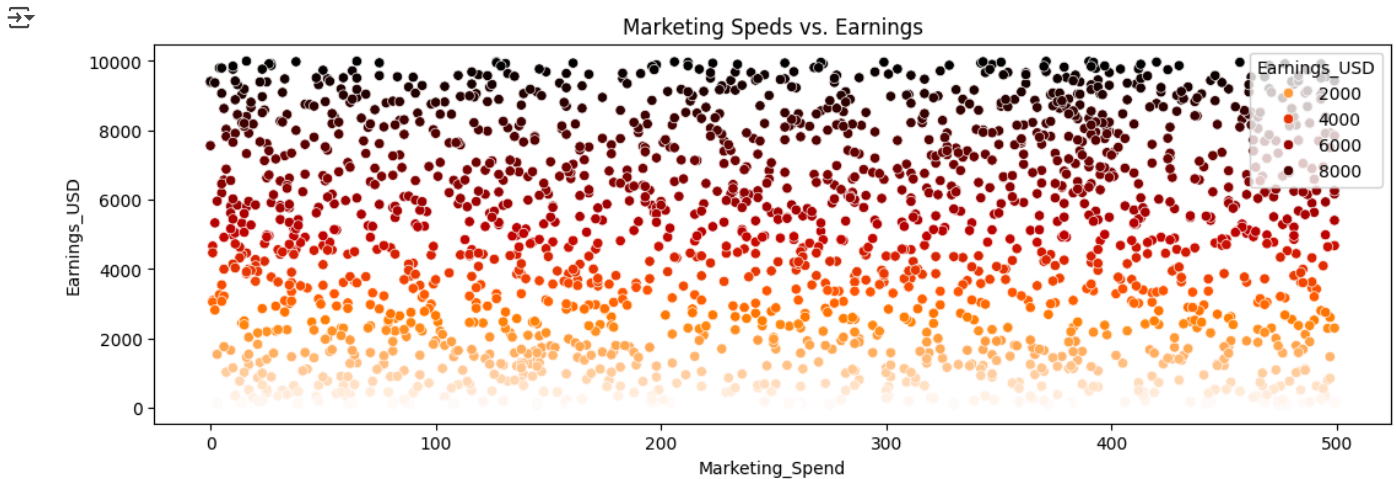
CONCLUSION : The bar chart shows the number of clients from various regions, with Australia leading at 298 and Canada trailing at 246.

QUE12 : How does marketing spend impact earnings according to the scatter plot?

```

1 plt.figure(figsize=(13, 4))
2 sns.scatterplot(data=df, x="Marketing_Spend", y="Earnings_USD", hue="Earnings_USD", palette="gist_heat_r")
3 plt.title("Marketing Speds vs. Earnings")
4 plt.show()

```



CONCLUSION : The scatter plot shows the relationship between marketing spend and earnings, with higher marketing spend generally associated with increased earnings.

QUE13 : What percentage of users prefer Crypto as their payment method according to the chart?

```

1 gb = df.groupby("Payment_Method")["Payment_Method"].count()
2 print(gb)
3 plt.pie(gb.values, labels=gb.index, startangle=0, shadow=True, autopct="%.2f")
4 plt.title("Most Common Payment Methods")
5 plt.show()

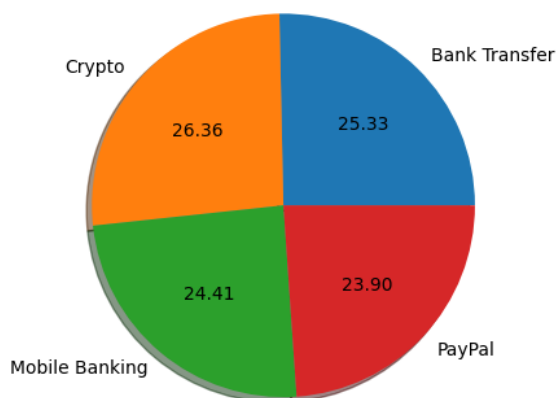
```

```

Payment_Method
Bank Transfer    494
Crypto           514
Mobile Banking   476
PayPal           466
Name: Payment_Method, dtype: int64

```

Most Common Payment Methods



CONCLUSION : The chart showcases the percentage split of common payment methods, with Crypto emerging as the top choice at 26.36%.

QUE14 : Which platform has the largest percentage of freelancers according to the pie chart?

```

1 gb = df.groupby("Platform")["Platform"].count()
2 print(gb)
3 plt.pie(gb.values, labels=gb.index, startangle=0, shadow=True, autopct="%.2f")
4 plt.title("Most Freelancers Platform")
5 plt.show()

```

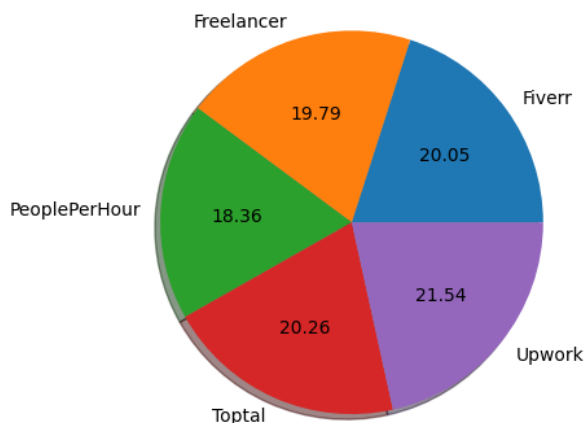


```

Platform
Fiverr      391
Freelancer  386
PeoplePerHour 358
Toptal      395
Upwork      420
Name: Platform, dtype: int64

```

Most Freelancers Platform



CONCLUSION : The pie chart displays the distribution of freelancers across five platforms, with Upwork leading at 21.54% and PeoplePerHour having the smallest share at 18.36%.

QUE15 : What percentage of freelancers fall into the Beginner category according to the pie chart?

```

1 gb = df.groupby("Experience_Level")["Experience_Level"].count()
2 print(gb)
3 plt.pie(gb.values, labels=gb.index, startangle=0, shadow=True, autopct="%.2f")
4 plt.title("Freelancers Experience level")
5 plt.show()

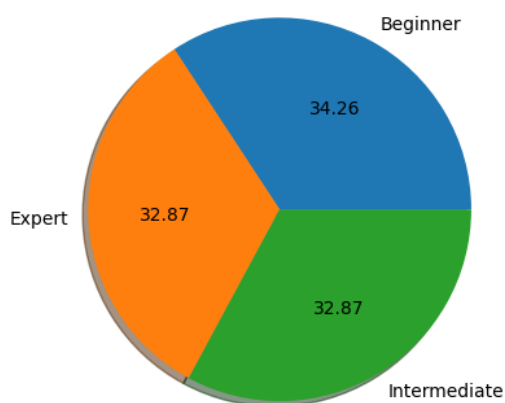
```

```

Experience_Level
Beginner      668
Expert        641
Intermediate   641
Name: Experience_Level, dtype: int64

```

Freelancers Experience level



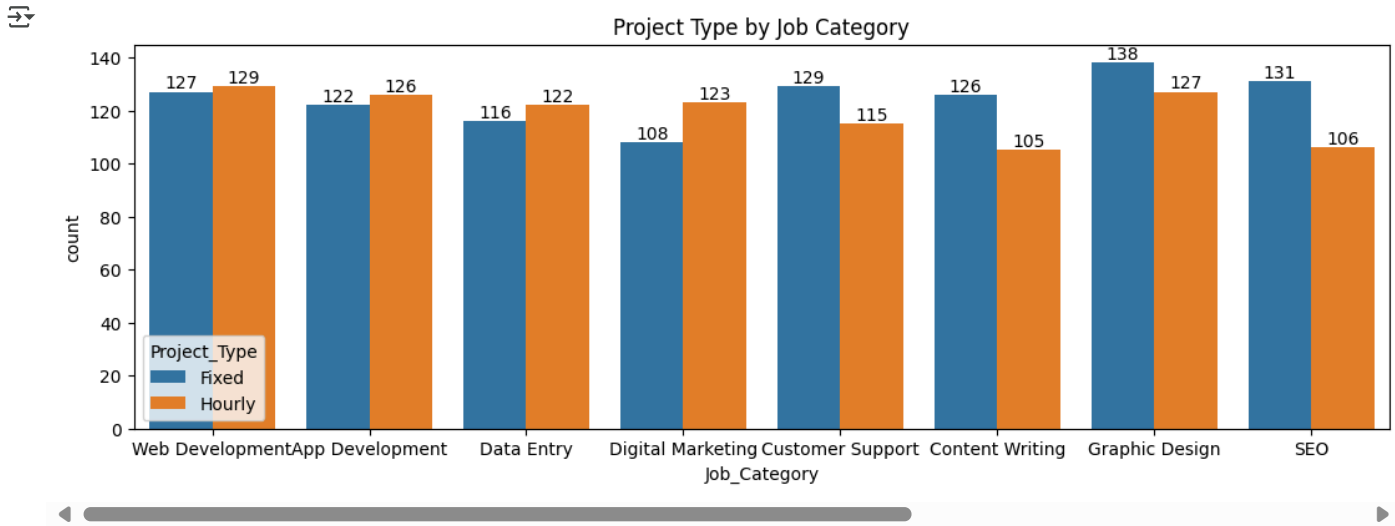
CONCLUSION : The pie chart depicts the distribution of freelancers based on experience levels, showing Beginners at 34.26%, with Intermediate and Experts each at 32.87%.

QUE16 : Which job categories have the highest counts for fixed and hourly projects according to the chart?

```

1 plt.figure(figsize=(13, 4))
2 project_type = sns.countplot(data=df, x="Job_Category", hue="Project_Type")
3 for i in range(0, 2):
4     project_type.bar_label(project_type.containers[i])
5 plt.title("Project Type by Job Category")
6 plt.show()

```



CONCLUSION : The bar chart compares fixed and hourly projects across job categories, with Graphic Design leading in fixed projects and App Development in hourly projects.

✓ STEP5 : Save clean data

```
1 df.to_csv("Freelancer Cleaned Data.csv", index=False)
2 print("Data Cleaning & Visualized Completed...")
3 print("Freelancer data Cleaning & Visualized Project Done!...")
```

Data Cleaning & Visualized Completed...
Freelancer data Cleaning & Visualized Project Done!...

Titanic Data Cleaning & Visualization Project Completed

Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge.