

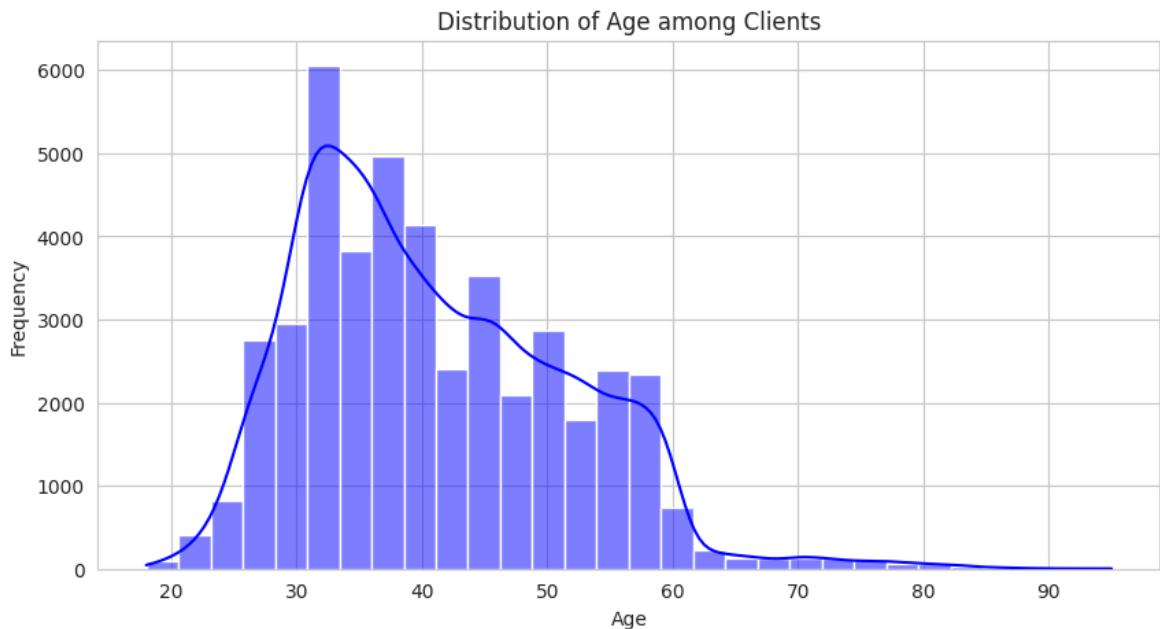
Q1. What is the distribution of age among the clients?

```
In [3]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

path="/content/drive/MyDrive/Dataset/banking_data.csv"
df = pd.read_csv(path)

sns.set_style("whitegrid")

plt.figure(figsize=(10, 5))
sns.histplot(df['age'], bins=30, kde=True, color='blue')
plt.xlabel("Age")
plt.ylabel("Frequency")
plt.title("Distribution of Age among Clients")
plt.show()
```



Q2. How does the job type vary among the clients?

```
In [4]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

path="/content/drive/MyDrive/Dataset/banking_data.csv"
df = pd.read_csv(path)

plt.figure(figsize=(12, 6))

sns.countplot(data=df, y="job", order=df["job"].value_counts().index, palette="viridis")

plt.xlabel("Count")
```

```

plt.ylabel("Job Type")
plt.title("Distribution of Job Types Among Clients")

plt.show()

```

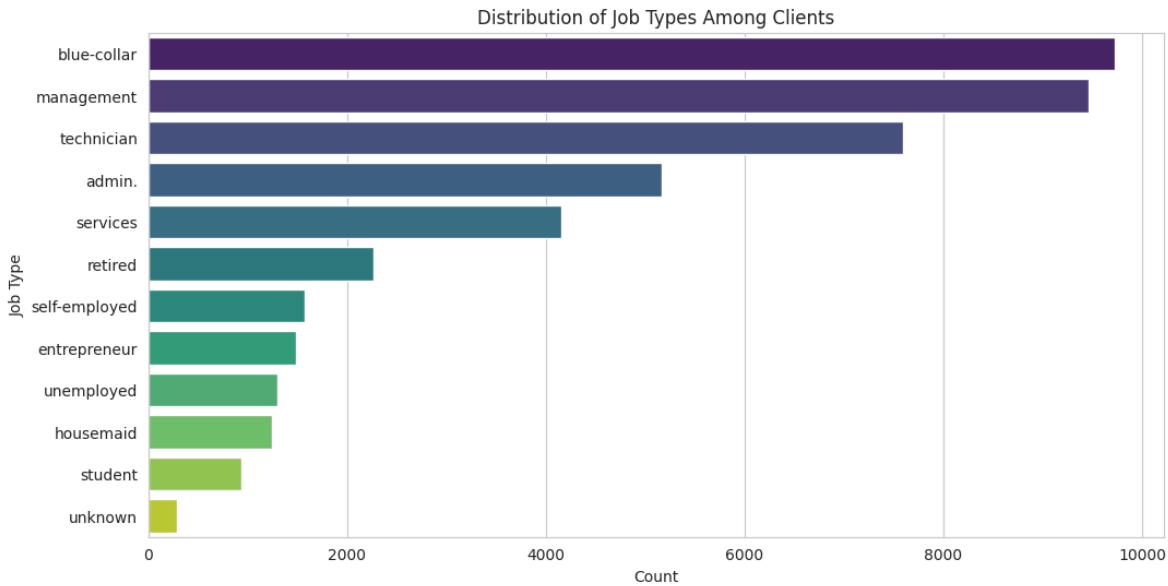
<ipython-input-4-8fba92a0eb77>:13: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v 0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

```

sns.countplot(data=df, y="job", order=df["job"].value_counts().index, palette
="viridis")

```



Q3.What is the marital status distribution of the clients?

```

In [5]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

path="/content/drive/MyDrive/Dataset/banking_data.csv"
df = pd.read_csv(path)

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

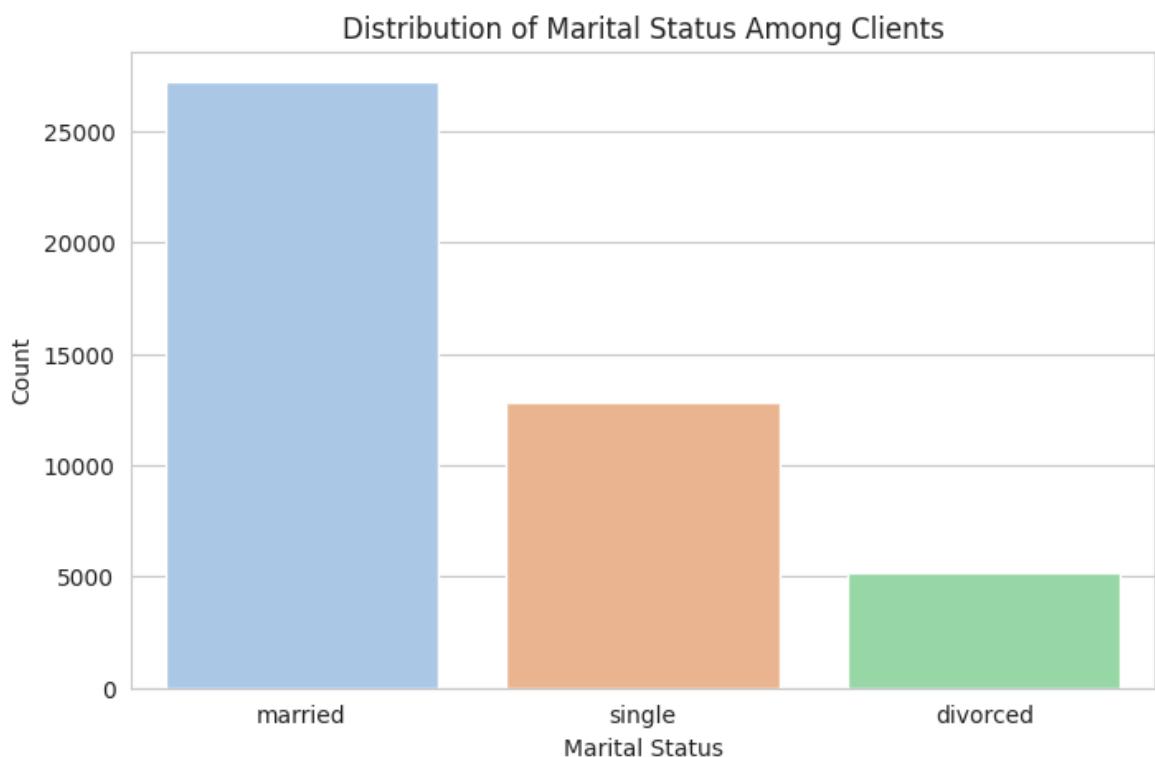
sns.countplot(data=df, x="marital", order=df["marital"].value_counts().index, pa
)

plt.xlabel("Marital Status")
plt.ylabel("Count")
plt.title("Distribution of Marital Status Among Clients")

plt.show()

```

```
<ipython-input-5-bb95d7162db3>:13: FutureWarning:  
  Passing `palette` without assigning `hue` is deprecated and will be removed in v  
  0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effe  
  ct.  
  
  sns.countplot(data=df, x="marital", order=df["marital"].value_counts().index, p  
  alette="pastel")
```



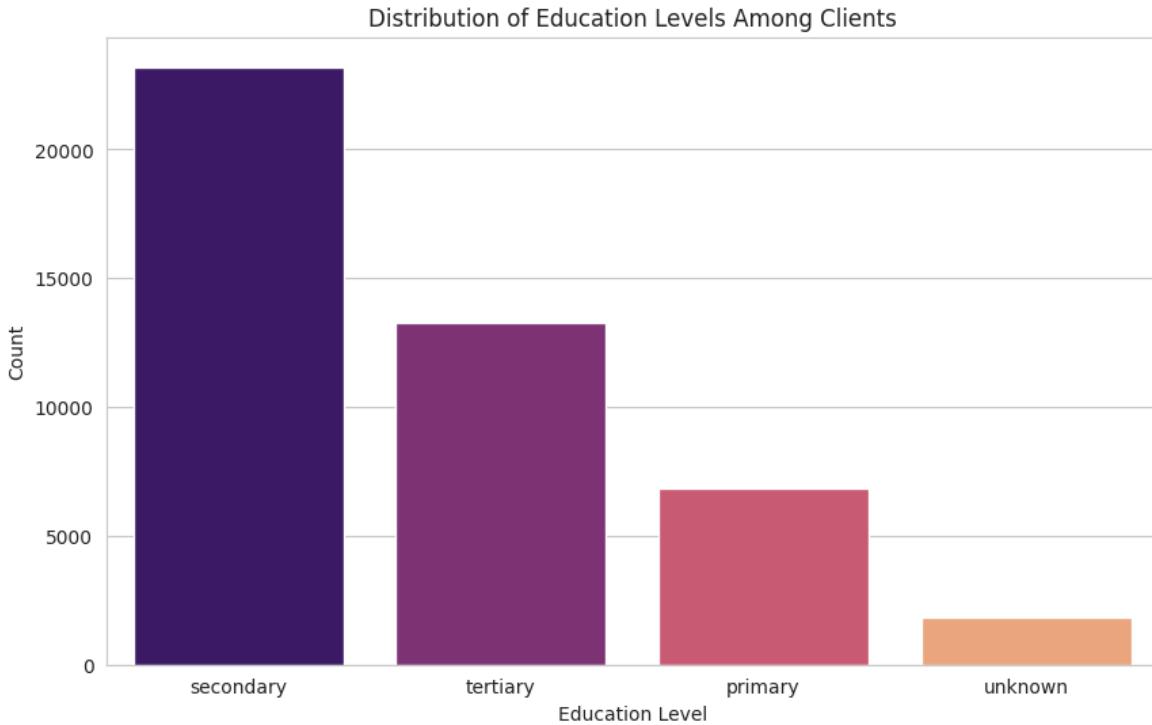
Q4. What is the level of education among the clients?

```
In [6]: import pandas as pd  
import matplotlib.pyplot as plt  
  
path="/content/drive/MyDrive/Dataset/banking_data.csv"  
df = pd.read_csv(path)  
  
plt.figure(figsize=(10, 6))  
  
sns.countplot(data=df, x="education", order=df["education"].value_counts().index)  
  
plt.xlabel("Education Level")  
plt.ylabel("Count")  
plt.title("Distribution of Education Levels Among Clients")  
plt.show()
```

```
<ipython-input-6-6b3dcd457529>:13: FutureWarning:
```

Passing `palette` without assigning `hue` is deprecated and will be removed in v 0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

```
sns.countplot(data=df, x="education", order=df["education"].value_counts().index, palette="magma")
```



Q5. What proportion of clients have credit in default?

```
In [7]: import pandas as pd

path="/content/drive/MyDrive/Dataset/banking_data.csv"
df = pd.read_csv(path)

default_proportion = df["default"].value_counts(normalize=True) * 100

print("Proportion of clients with credit in default:")
print(default_proportion)
```

Proportion of clients with credit in default:
 default
 no 98.197541
 yes 1.802459
 Name: proportion, dtype: float64

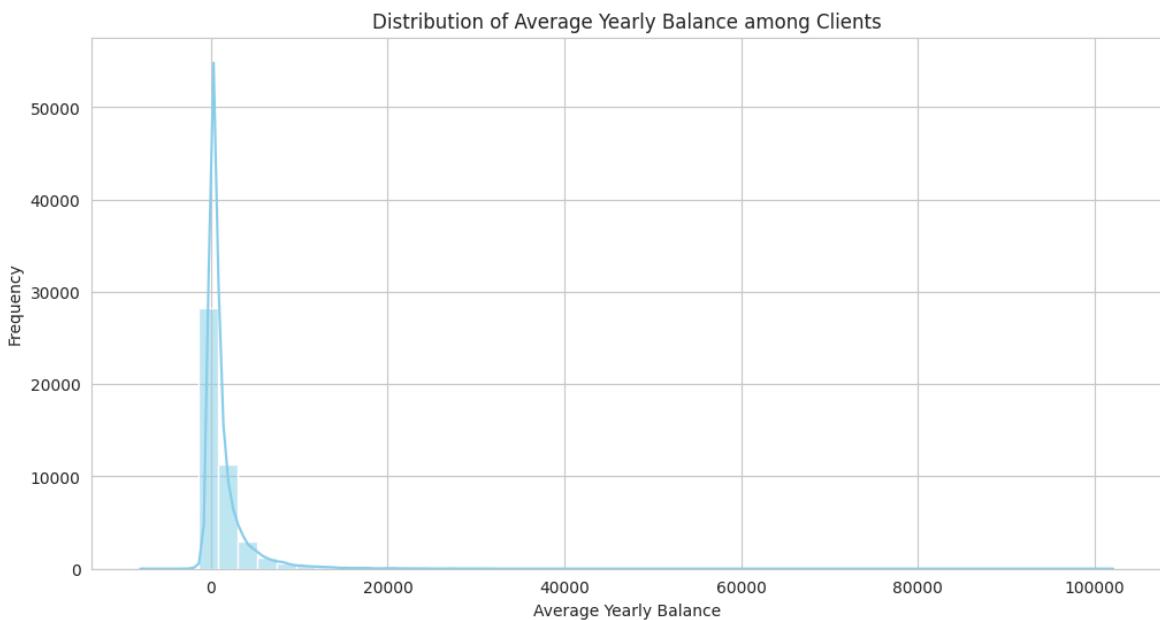
Q6. What is the distribution of average yearly balance among the clients?

```
In [8]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
path="/content/drive/MyDrive/Dataset/banking_data.csv"
df = pd.read_csv(path)

sns.set_style("whitegrid")

plt.figure(figsize=(12, 6))
sns.histplot(df['balance'], bins=50, kde=True, color='skyblue')
plt.xlabel("Average Yearly Balance")
plt.ylabel("Frequency")
plt.title("Distribution of Average Yearly Balance among Clients")
plt.show()
```



Q7. How many clients have housing loans?

```
In [9]: import pandas as pd

path="/content/drive/MyDrive/Dataset/banking_data.csv"
df = pd.read_csv(path)

housing_loan_count = df[df['housing'] == 'yes'].shape[0]

print(f"Number of clients with housing loans: {housing_loan_count}")
```

Number of clients with housing loans: 25130

Q8. How many clients have personal loans?

```
In [10]: import pandas as pd  
  
path="/content/drive/MyDrive/Dataset/banking_data.csv"  
df = pd.read_csv(path)  
  
personal loan count = df[df['loan'] == 'yes'].shape[0]
```

```
print(f"Number of clients with personal loans: {personal_loan_count}")
```

Number of clients with personal loans: 7244

Q9.What are the communication types used for contacting clients during the campaign?

```
In [13]: import pandas as pd
path="/content/drive/MyDrive/Dataset/banking_data.csv"
df = pd.read_csv(path)

print(f"These are the various communication types used for contacting clients du")
```

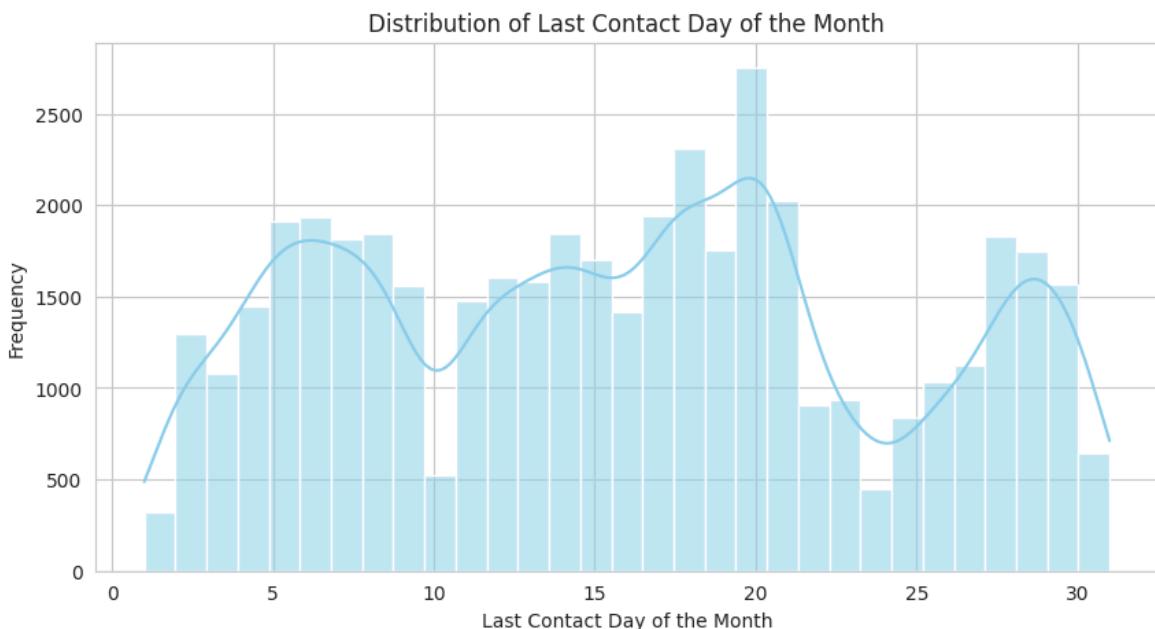
These are the various communication types used for contacting clients during the campaign ['unknown' 'cellular' 'telephone']

Q10.What is the distribution of the last contact day of the month?

```
In [14]: import pandas as pd
import matplotlib.pyplot as plt
path="/content/drive/MyDrive/Dataset/banking_data.csv"
df = pd.read_csv(path)

sns.set_style("whitegrid")

plt.figure(figsize=(10, 5))
sns.histplot(df['day'], bins=31, kde=True, color='skyblue') # Assuming 'day' rep
plt.xlabel("Last Contact Day of the Month")
plt.ylabel("Frequency")
plt.title("Distribution of Last Contact Day of the Month")
plt.show()
```



Q11.How does the last contact month vary among the clients?

```
In [15]: import pandas as pd
import matplotlib.pyplot as plt
path="/content/drive/MyDrive/Dataset/banking_data.csv"
df = pd.read_csv(path)

plt.figure(figsize=(10, 6))
```

```

sns.countplot(data=df, x="month", order=df["month"].value_counts().index, palette="magma")
plt.xlabel("Last Contact Month")
plt.ylabel("Count")
plt.title("Distribution of Last Contact Month Among Clients")
plt.show()

```

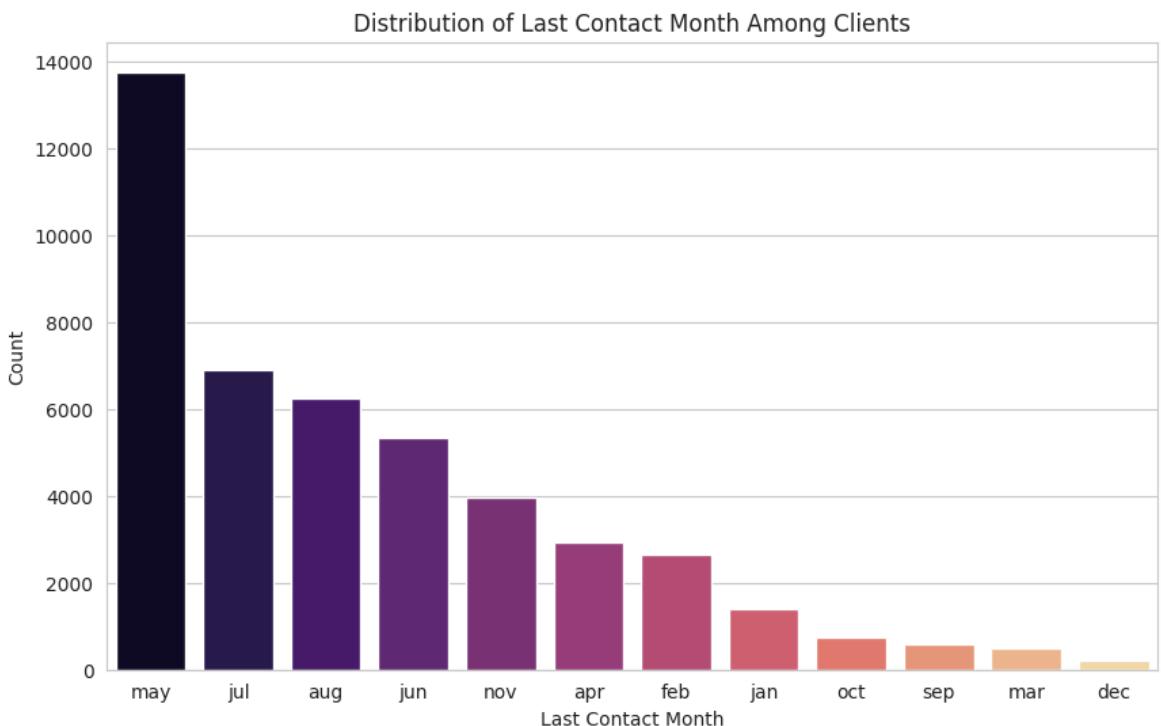
<ipython-input-15-30949b1da633>:13: FutureWarning:

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

```

sns.countplot(data=df, x="month", order=df["month"].value_counts().index, palette="magma")

```



Q12.What is the distribution of the duration of the last contact?

```

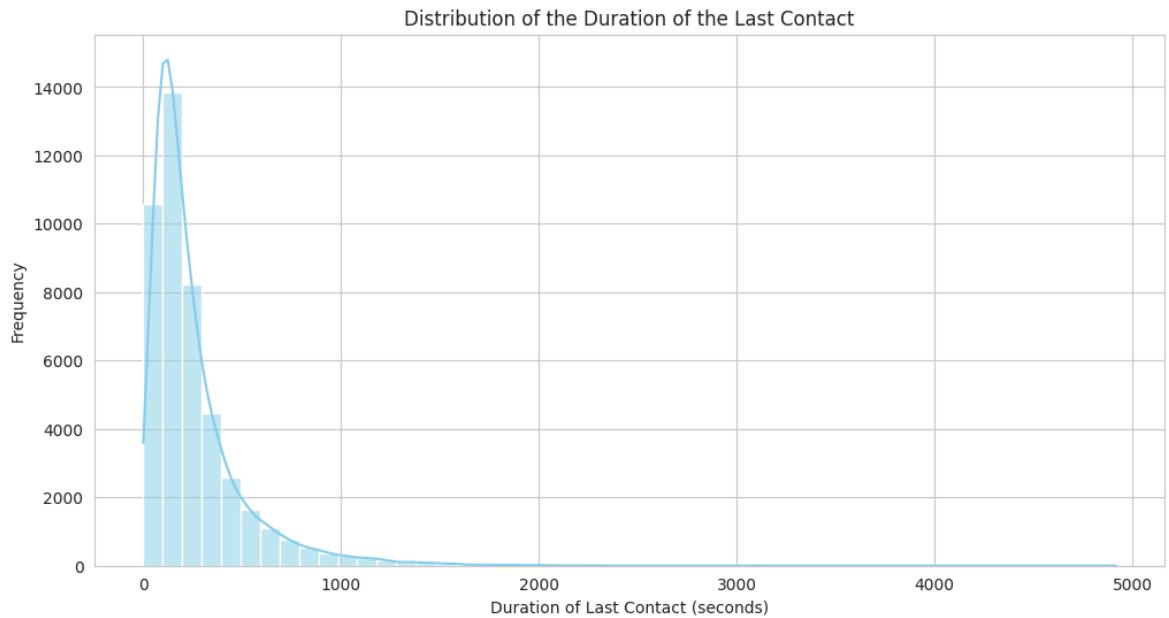
In [16]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

path="/content/drive/MyDrive/Dataset/banking_data.csv"
df = pd.read_csv(path)

sns.set_style("whitegrid")

plt.figure(figsize=(12, 6))
sns.histplot(df['duration'], bins=50, kde=True, color='skyblue')
plt.xlabel("Duration of Last Contact (seconds)")
plt.ylabel("Frequency")
plt.title("Distribution of the Duration of the Last Contact")
plt.show()

```



Q13.How many contacts were performed during the campaign for each client?

```
In [18]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

path="/content/drive/MyDrive/Dataset/banking_data.csv"
df = pd.read_csv(path)

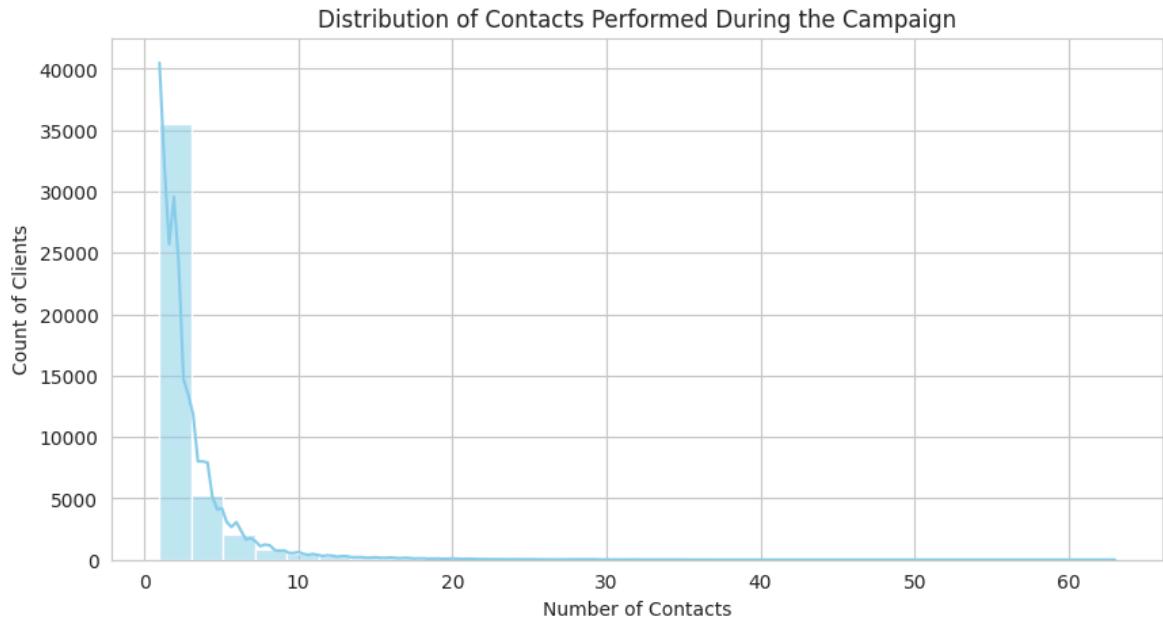
campaign_summary = df["campaign"].describe()
print("Summary Statistics for Number of Contacts per Client:")
print(campaign_summary)

plt.figure(figsize=(10, 5))
sns.histplot(df["campaign"], bins=30, kde=True, color="skyblue")

plt.xlabel("Number of Contacts")
plt.ylabel("Count of Clients")
plt.title("Distribution of Contacts Performed During the Campaign")

plt.show()
```

```
Summary Statistics for Number of Contacts per Client:
count    45216.000000
mean      2.763668
std       3.097896
min       1.000000
25%      1.000000
50%      2.000000
75%      3.000000
max      63.000000
Name: campaign, dtype: float64
```



Q14.What is the distribution of the number of days passed since the client was last contacted from a previous campaign?

In [19]:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

path="/content/drive/MyDrive/Dataset/banking_data.csv"
df = pd.read_csv(path)

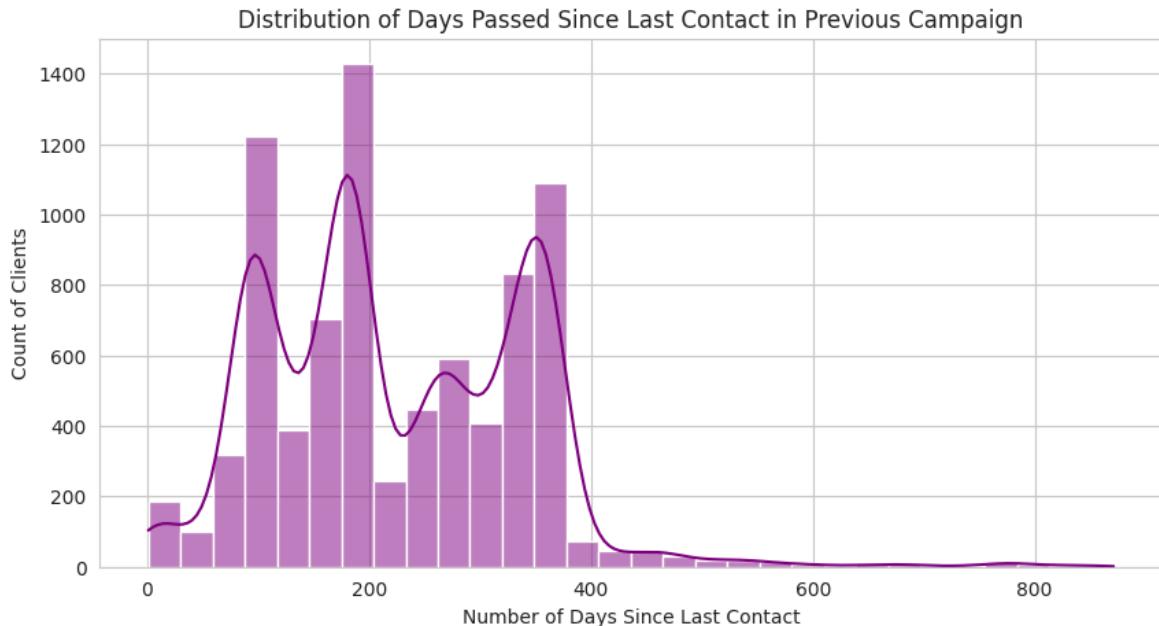
plt.figure(figsize=(10, 5))

sns.histplot(df[df["pdays"] != -1]["pdays"], bins=30, kde=True, color="purple")

plt.xlabel("Number of Days Since Last Contact")
plt.ylabel("Count of Clients")
plt.title("Distribution of Days Passed Since Last Contact in Previous Campaign")

plt.show()

no_previous_contact = (df["pdays"] == -1).mean() * 100
print(f"Percentage of clients who were not previously contacted: {no_previous_co
```



Percentage of clients who were not previously contacted: 81.73%

Q15. How many contacts were performed before the current campaign for each client?

In [21]:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

path="/content/drive/MyDrive/Dataset/banking_data.csv"
df = pd.read_csv(path)

previous_summary = df["previous"].describe()
print("Summary Statistics for Number of Previous Contacts:")
print(previous_summary)

plt.figure(figsize=(10, 5))
sns.histplot(df["previous"], bins=30, kde=True, color="orange")

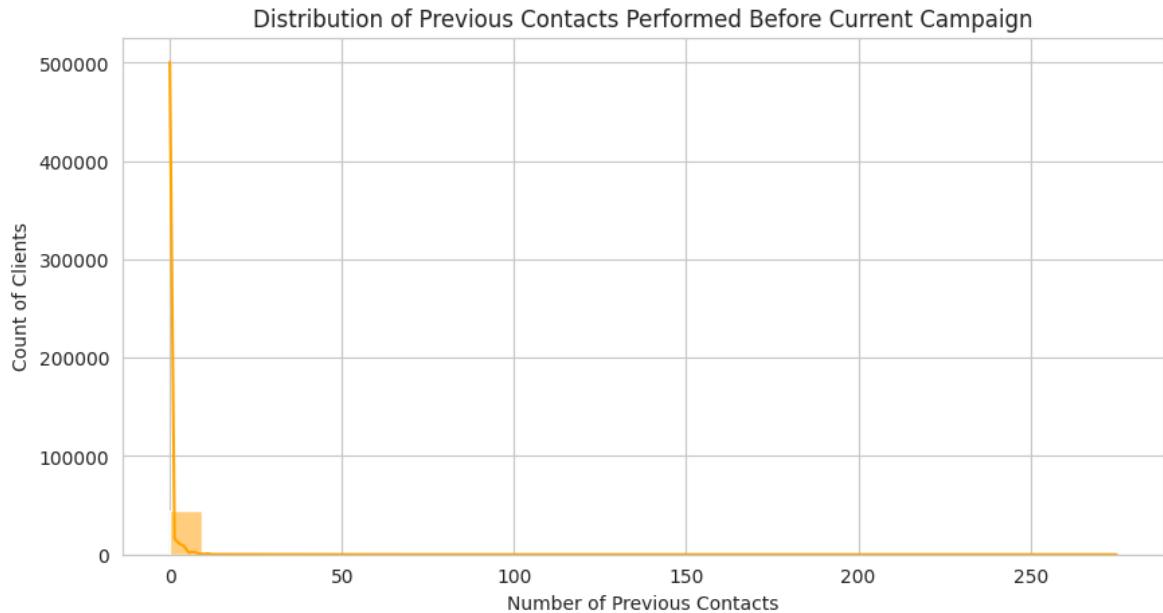
plt.xlabel("Number of Previous Contacts")
plt.ylabel("Count of Clients")
plt.title("Distribution of Previous Contacts Performed Before Current Campaign")

plt.show()
```

Summary Statistics for Number of Previous Contacts:

| | |
|-------|--------------|
| count | 45216.000000 |
| mean | 0.580657 |
| std | 2.303778 |
| min | 0.000000 |
| 25% | 0.000000 |
| 50% | 0.000000 |
| 75% | 0.000000 |
| max | 275.000000 |

Name: previous, dtype: float64



Q16.What were the outcomes of the previous marketing campaigns?

```
In [22]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

path="/content/drive/MyDrive/Dataset/banking_data.csv"
df = pd.read_csv(path)

poutcome_counts = df["poutcome"].value_counts()

print("Outcome Distribution of Previous Marketing Campaigns:")
print(poutcome_counts)

plt.figure(figsize=(8, 5))
sns.barplot(x=poutcome_counts.index, y=poutcome_counts.values, palette="muted")

plt.xlabel("Outcome of Previous Campaign")
plt.ylabel("Number of Clients")
plt.title("Distribution of Previous Campaign Outcomes")

plt.show()
```

Outcome Distribution of Previous Marketing Campaigns:

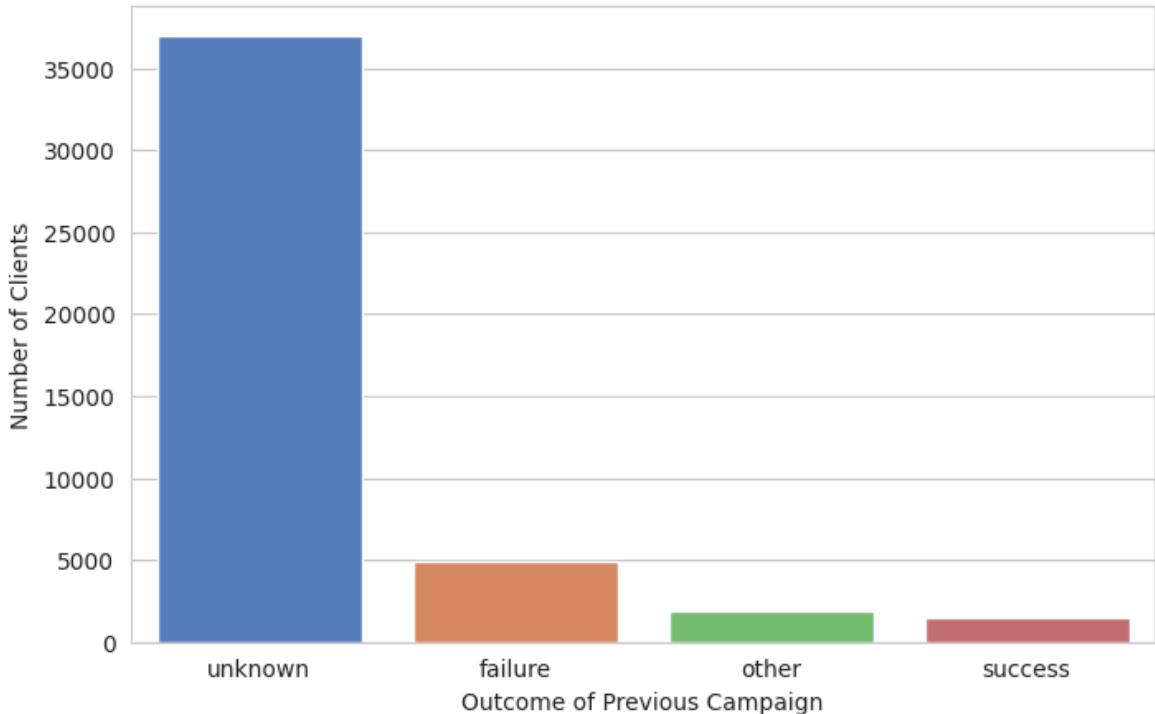
```
poutcome
unknown      36961
failure       4902
other        1840
success       1513
Name: count, dtype: int64
```

```
<ipython-input-22-4d679c582a91>:18: FutureWarning:
```

```
Passing `palette` without assigning `hue` is deprecated and will be removed in v
0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effe
ct.
```

```
sns.barplot(x=poutcome_counts.index, y=poutcome_counts.values, palette="muted")
```

Distribution of Previous Campaign Outcomes



Q17.What is the distribution of clients who subscribed to a term deposit vs. those who did not?

```
In [23]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

path="/content/drive/MyDrive/Dataset/banking_data.csv"
df = pd.read_csv(path)

subscription_counts = df["y"].value_counts()

print("Distribution of Clients Who Subscribed to a Term Deposit:")
print(subscription_counts)

plt.figure(figsize=(6, 5))
sns.barplot(x=subscription_counts.index, y=subscription_counts.values, palette=""

plt.xlabel("Subscription to Term Deposit")
plt.ylabel("Number of Clients")
plt.title("Distribution of Clients Who Subscribed vs. Did Not Subscribe")

plt.show()
```

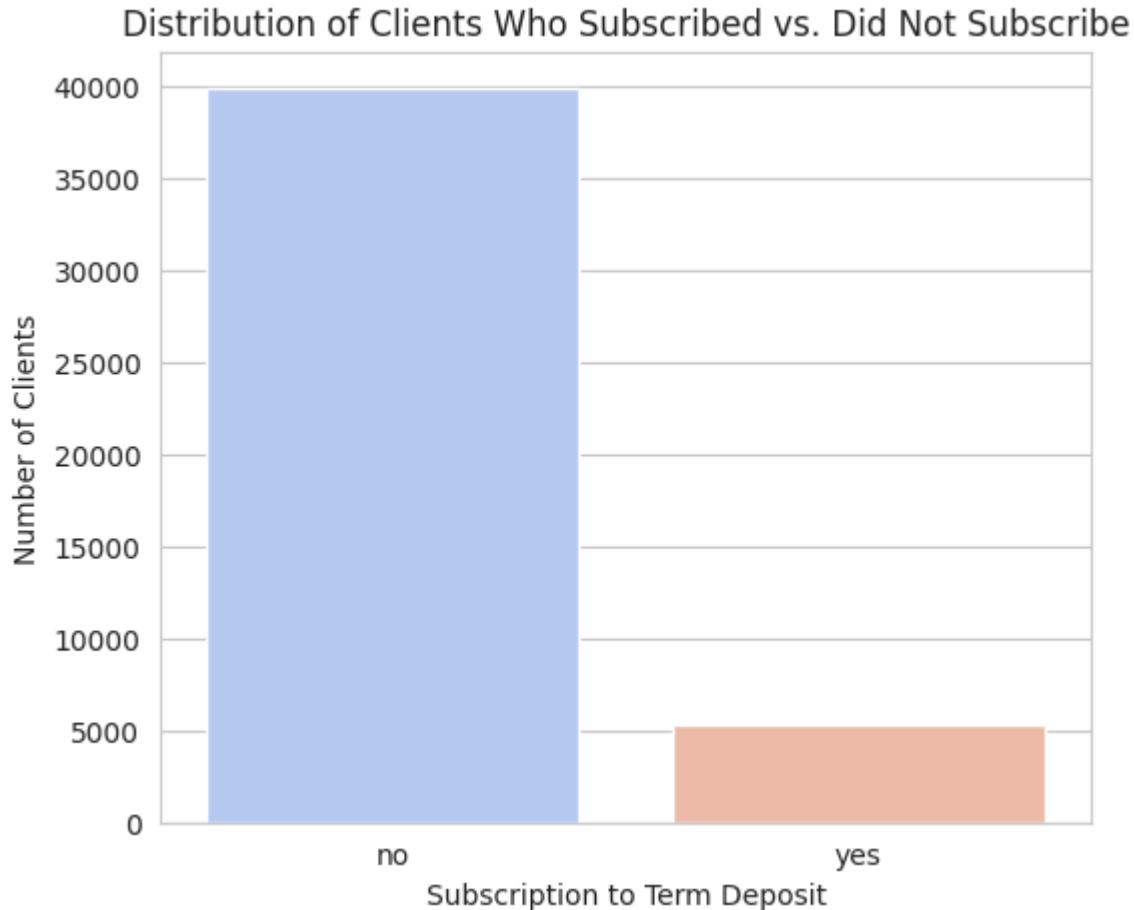
Distribution of Clients Who Subscribed to a Term Deposit:

```
y
no      39922
yes     5294
Name: count, dtype: int64
```

```
<ipython-input-23-8631f3664650>:18: FutureWarning:
```

Passing `palette` without assigning `hue` is deprecated and will be removed in v 0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

```
sns.barplot(x=subscription_counts.index, y=subscription_counts.values, palette = "coolwarm")
```



Q18. Are there any correlations between different attributes and the likelihood of subscribing to a term deposit?

In [31]:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

path="/content/drive/MyDrive/Dataset/banking_data.csv"
df = pd.read_csv(path)

df["y"] = df["y"].map({"yes": 1, "no": 0})

binary_cols = ["default", "housing", "loan"]
for col in binary_cols:
    df[col] = df[col].map({"yes": 1, "no": 0})

df[binary_cols] = df[binary_cols].fillna(0)

categorical_cols = ["job", "marital", "education", "contact", "month", "poutcome"]
df_encoded = pd.get_dummies(df, columns=categorical_cols, drop_first=True)

non_numeric_cols = df_encoded.select_dtypes(include=["object"]).columns
```

```

if len(non_numeric_cols) > 0:
    print("Warning: Non-numeric columns found:", non_numeric_cols)
    for col in non_numeric_cols:
        df_encoded[col] = pd.to_numeric(df_encoded[col], errors='coerce')

non_numeric_cols = df_encoded.select_dtypes(include=["object"]).columns
if len(non_numeric_cols) > 0:
    print("Error: Non-numeric columns still present after conversion:", non_numeric_cols)
    df_encoded = df_encoded.drop(columns=non_numeric_cols)

correlation_matrix = df_encoded.corr()

target_correlation = correlation_matrix["y"].sort_values(ascending=False)

print("Top attributes correlated with subscription to a term deposit:")
print(target_correlation.head(10))
print("\nLeast correlated attributes:")
print(target_correlation.tail(10))

plt.figure(figsize=(12, 8))
sns.heatmap(correlation_matrix, cmap="coolwarm", center=0, annot=False)

plt.title("Correlation Heatmap of Attributes with Subscription to Term Deposit")

plt.show()

```

Warning: Non-numeric columns found: Index(['marital_status', 'day_month'], dtype='object')

Top attributes correlated with subscription to a term deposit:

| y | 1.000000 |
|------------------|----------|
| duration | 0.394387 |
| poutcome_success | 0.307083 |
| month_mar | 0.129371 |
| month_oct | 0.128439 |
| month_sep | 0.123099 |
| pdays | 0.103699 |
| previous | 0.093576 |
| job_retired | 0.079928 |
| job_student | 0.076821 |

Name: y, dtype: float64

Least correlated attributes:

| marital_married | -0.060216 |
|------------------|-----------|
| loan | -0.068289 |
| job_blue-collar | -0.072211 |
| campaign | -0.073294 |
| month_may | -0.102656 |
| housing | -0.139445 |
| contact_unknown | -0.151062 |
| poutcome_unknown | -0.167284 |
| marital_status | Nan |
| day_month | Nan |

Name: y, dtype: float64

