Capston Project - Marketing campaign

January 23, 2026

[14]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
# Python
import warnings

# Suppress all warnings
warnings.filterwarnings("ignore")
```

[6]:

```
df= pd.read_csv("bank-full.csv", sep=";")
df.head()
```

[6]:

```
   age           job  marital  education default  balance housing loan  \
0   58    management  married   tertiary      no     2143     yes   no
1   44    technician   single  secondary      no       29     yes   no
2   33  entrepreneur  married  secondary      no        2     yes  yes
3   47   blue-collar  married    unknown      no     1506     yes   no
4   33       unknown   single    unknown      no        1      no   no

   contact  day month  duration  campaign  pdays  previous poutcome   y
0  unknown    5   may       261         1     -1         0  unknown  no
1  unknown    5   may       151         1     -1         0  unknown  no
2  unknown    5   may        76         1     -1         0  unknown  no
3  unknown    5   may        92         1     -1         0  unknown  no
4  unknown    5   may       198         1     -1         0  unknown  no
```

[7]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45211 entries, 0 to 45210
Data columns (total 17 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   age        45211 non-null  int64
 1   job        45211 non-null  object
 2   marital    45211 non-null  object
 3   education  45211 non-null  object
 4   default    45211 non-null  object
 5   balance    45211 non-null  int64
 6   housing    45211 non-null  object
 7   loan       45211 non-null  object
 8   contact    45211 non-null  object
 9   day        45211 non-null  int64
 10  month      45211 non-null  object
 11  duration   45211 non-null  int64
 12  campaign   45211 non-null  int64
 13  pdays      45211 non-null  int64
 14  previous   45211 non-null  int64
 15  poutcome   45211 non-null  object
 16  y          45211 non-null  object
dtypes: int64(7), object(10)
memory usage: 5.9+ MB
```

[ ]:

0.1 Univariate analysis for categorical and object variables

[10]:

```python
# Univariate analysis of categorical and object variables

def plot_object(dataframe, column_name):
 """
 Plots a bar chart showing category frequencies with both frequency (inside bar)
 and proportion (above bar) labels.
 Parameters:- dataframe: pandas DataFrame- column_name: str, name of the␣
 ↪categorical column to visualize
 """
 # Count frequencies and proportions
 value_counts = dataframe[column_name].value_counts()
 proportions = value_counts / len(dataframe)
 # Set plot style
 sns.set(style="whitegrid")
 plt.figure(figsize=(10, 6))
 # Bar plot
 palette1=sns.color_palette(palette='tab20')
 ax = sns.barplot(x=value_counts.index, y=value_counts.values, palette=palette1)
 # Annotate bars
 for i, (count, prop) in enumerate(zip(value_counts.values, proportions.
 ↪values)):
 # Frequency inside bar
   ax.text(i, count * 0.5, f'{count}', ha='center', va='center',fontsize=11,␣
 ↪color='black', fontweight='bold')
 # Proportion above bar
   ax.text(i, count + max(value_counts.values) * 0.02, f'{prop:.
 ↪2%}',ha='center', fontsize=10, color='black')
```

```
plt.title(f'Distribution of {column_name}')
plt.xlabel(column_name)
plt.xticks(rotation=90)
plt.ylabel('Frequency')
plt.tight_layout()
plt.show()
```

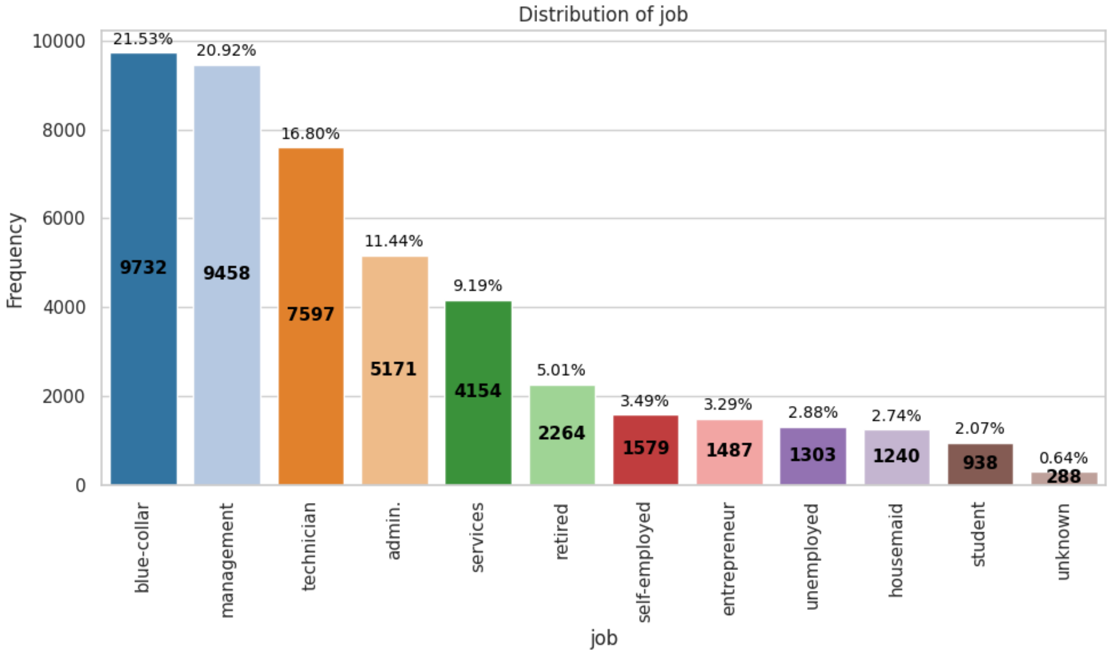
[ ]:

```

```

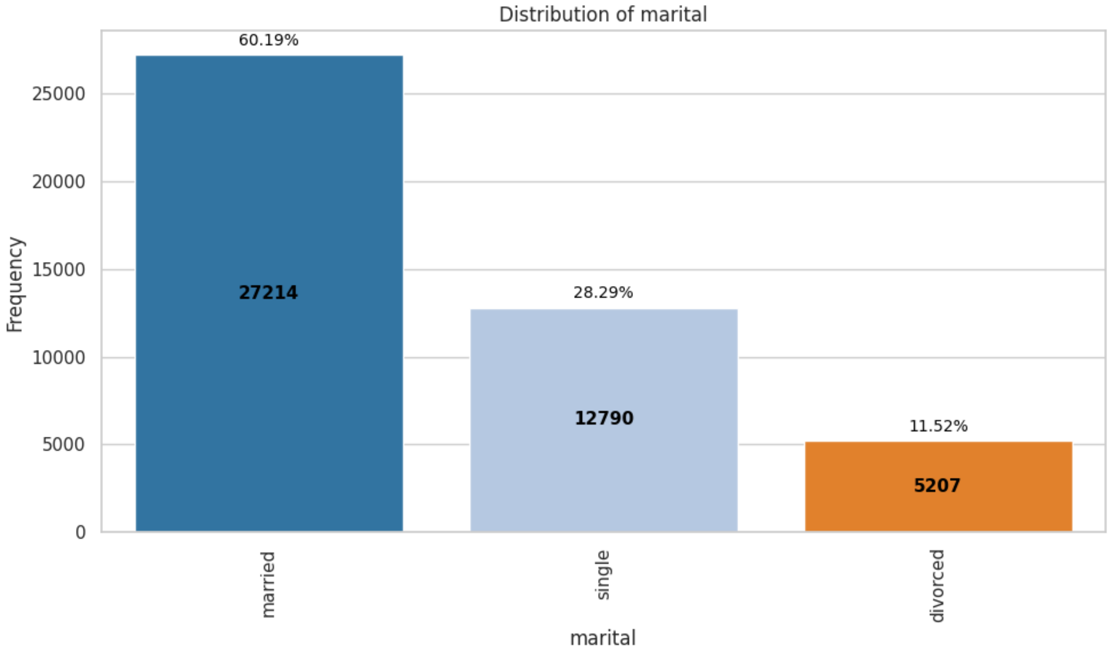
[13]:

```
plot_object(df, "job")
```

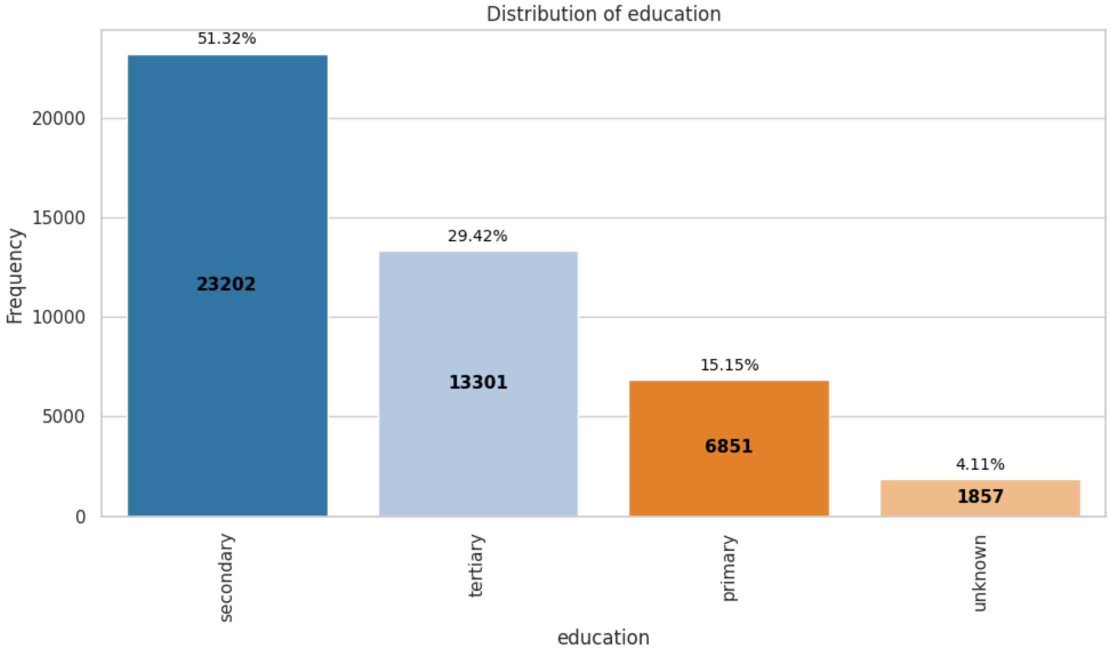


[16]:

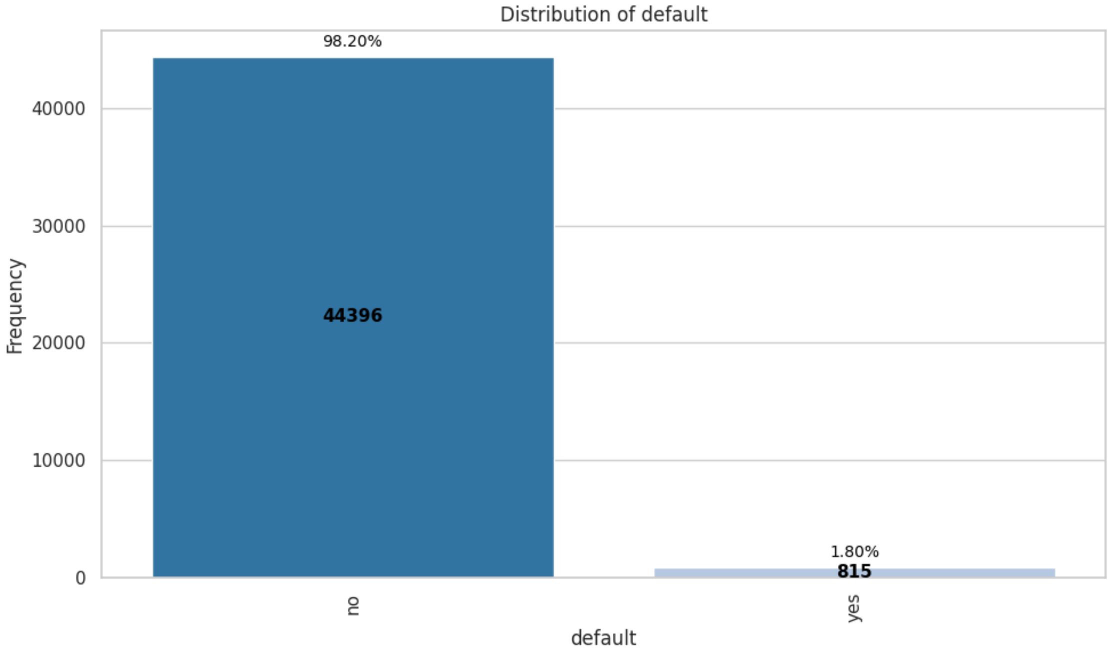
```
plot_object(df, 'marital')
```



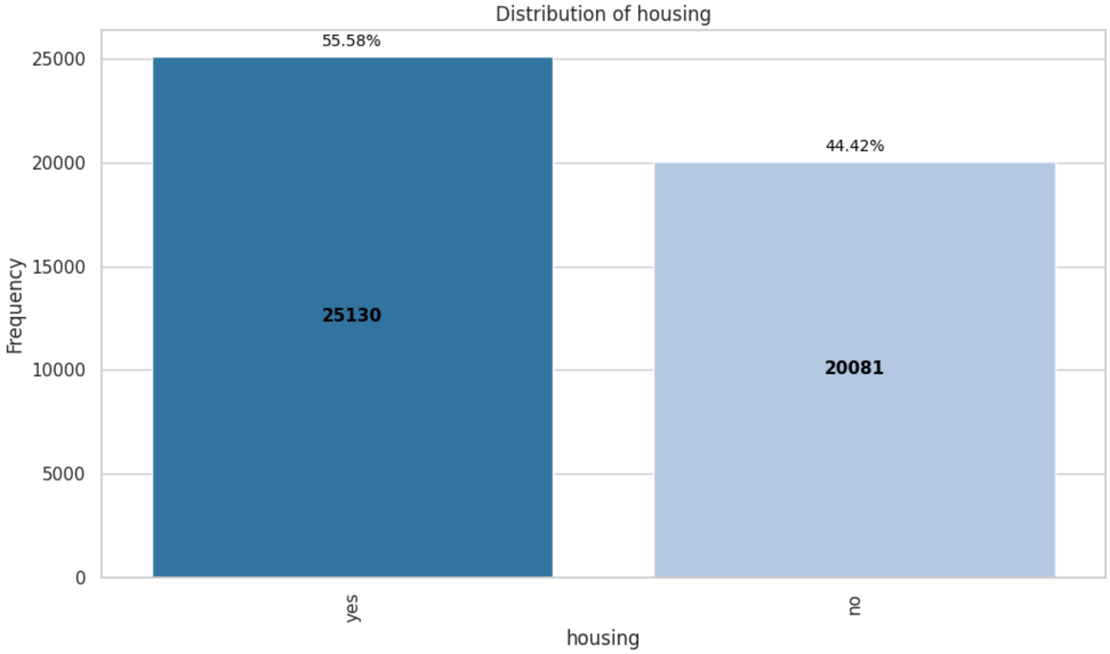
```
[17]: plot_object(df, 'education')
```



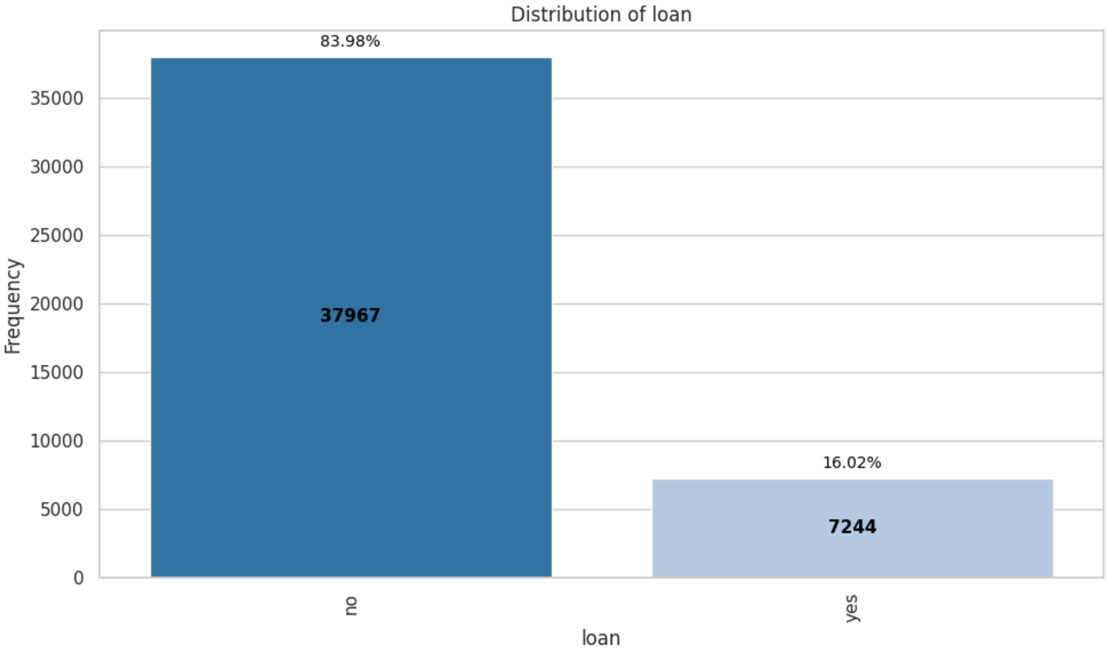
```
[18]: plot_object(df, 'default')
```



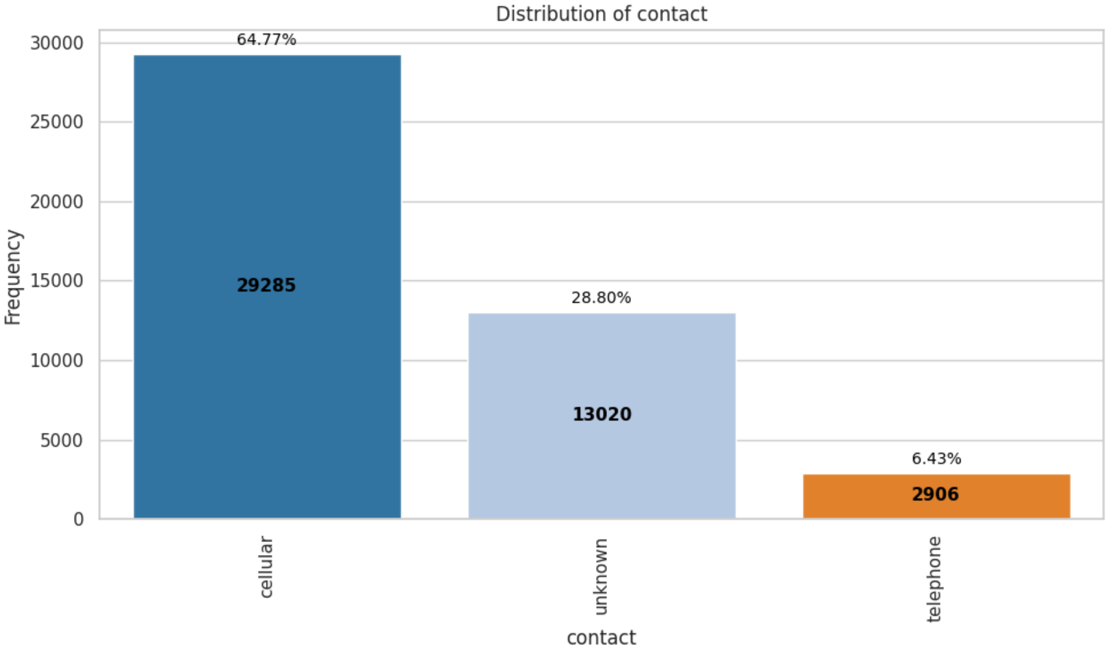
```
[19]: plot_object(df, 'housing')
```



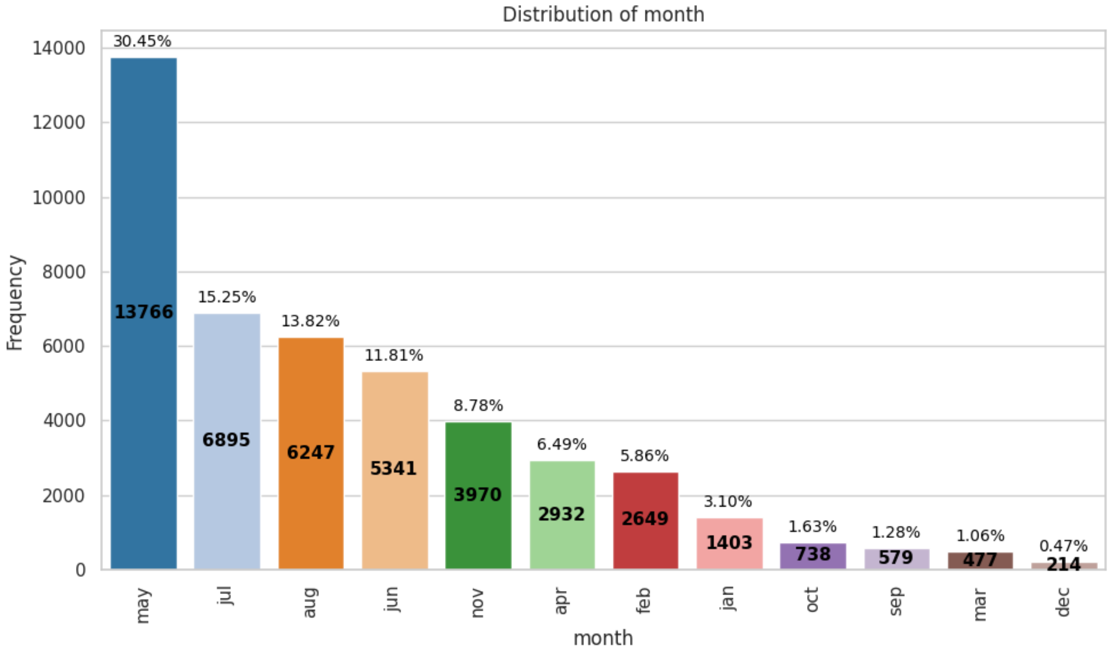
```
[20]: plot_object(df, 'loan')
```



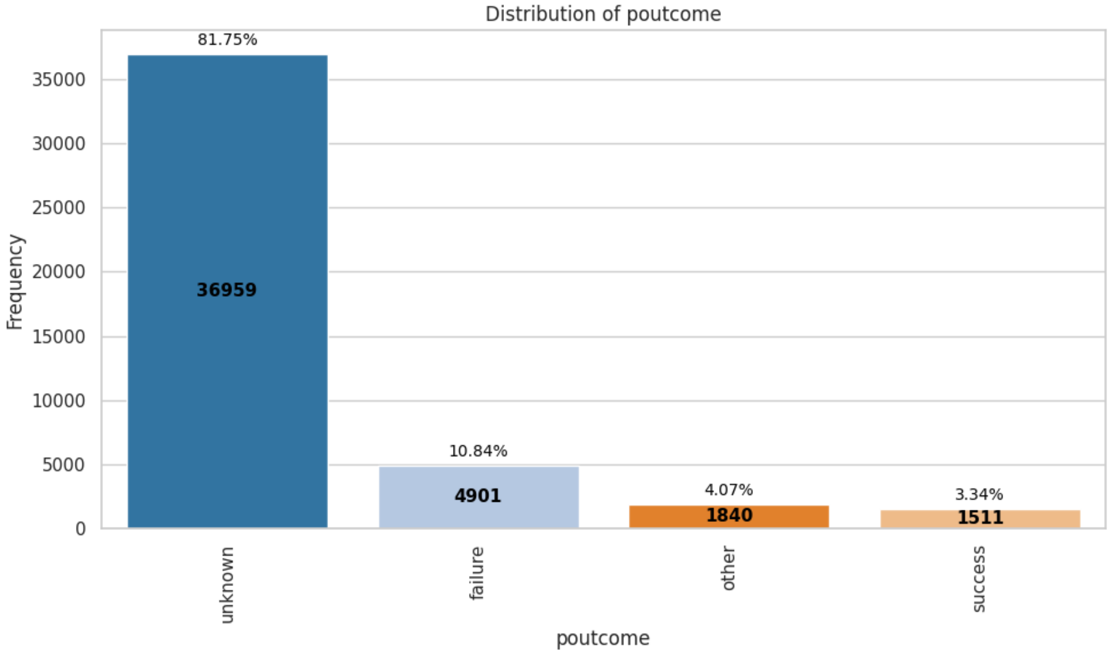
```
[22]: plot_object(df, 'contact')
```



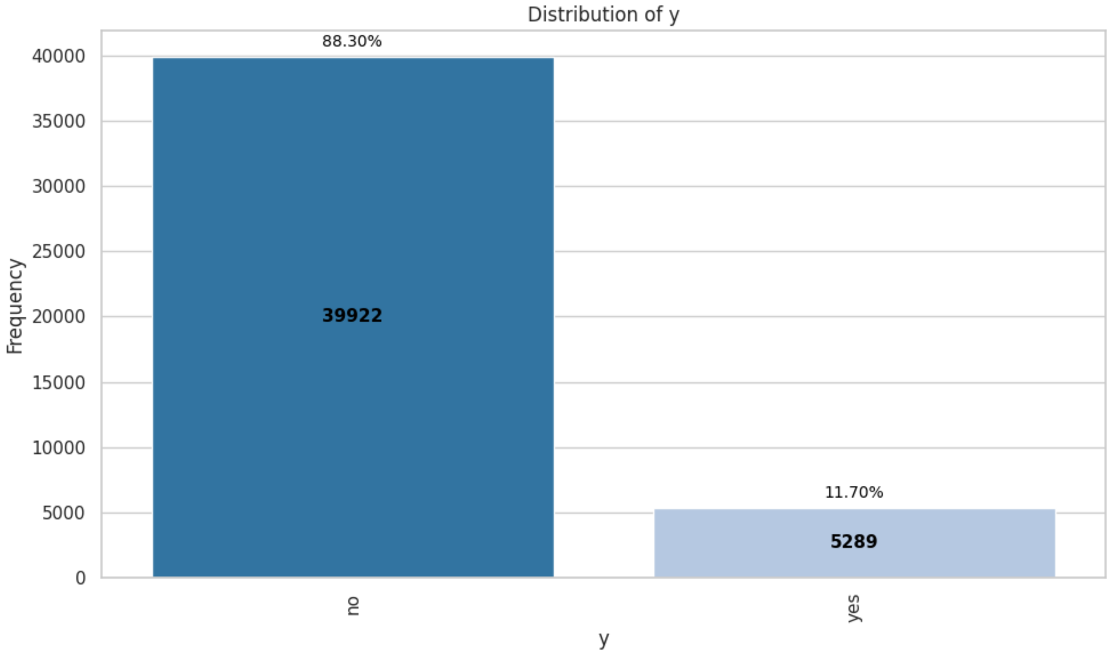
```
[23]: plot_object(df, 'month')
```



```
[24]: plot_object(df, 'poutcome')
```



```
[25]: plot_object(df, 'y')
```



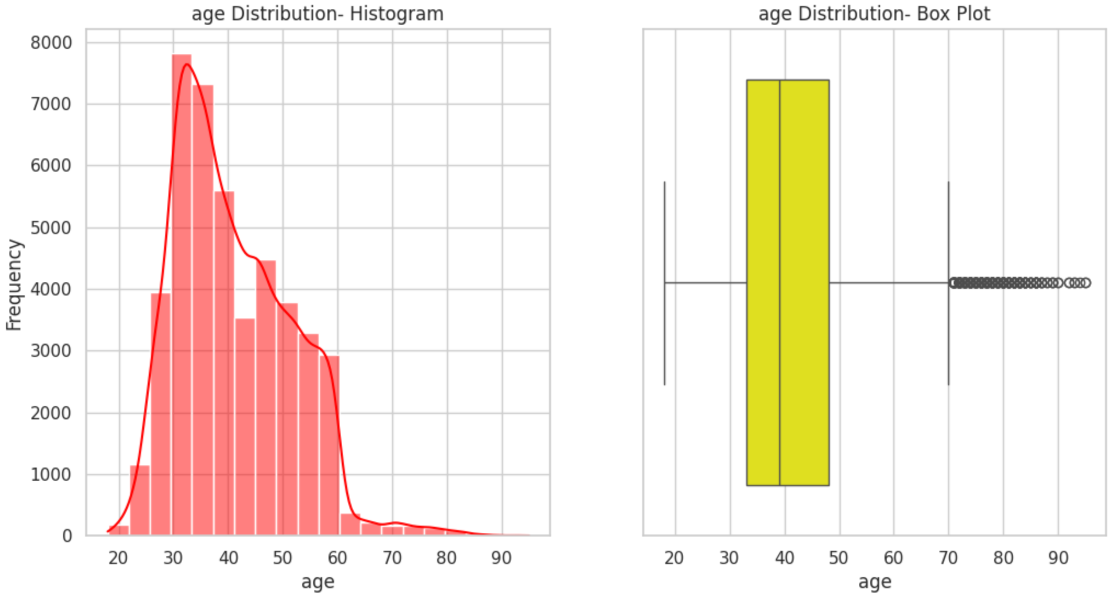
0.2 Univariate Analysis for numerical variables

[34]:
```
# univariate analysis of continuous variables
def cont_plot(df, var):
 #var="Age"
 # Set plot style
 sns.set(style="whitegrid")
 # Create a figure with two subplots: histogram and box plot
 plt.figure(figsize=(12, 6))
 # Histogram
 # Box plot
 plt.subplot(1, 2, 1)
 sns.histplot(df[var], bins=20, kde=True, color='red')
 plt.title(var+' Distribution- Histogram')
 plt.xlabel(var)
 plt.ylabel('Frequency')
 plt.subplot(1, 2, 2)
 sns.boxplot(x=df[var], color='yellow')
 plt.title(var+' Distribution- Box Plot')
```

[48]:
```
def NoOutlier(df,var):
    sns.boxplot(x=df[var], showfliers=False,color="yellow")
    plt.title("Boxplot of "+ var +" (without outliers)")
    plt.show()
```

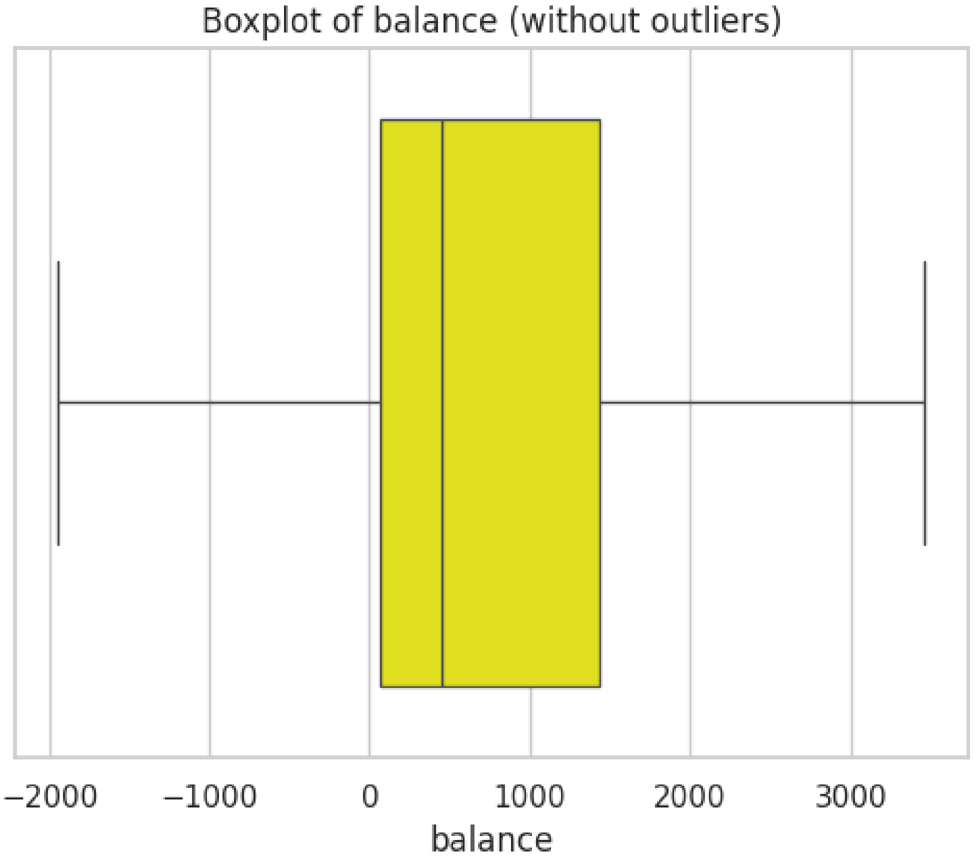
[35]:

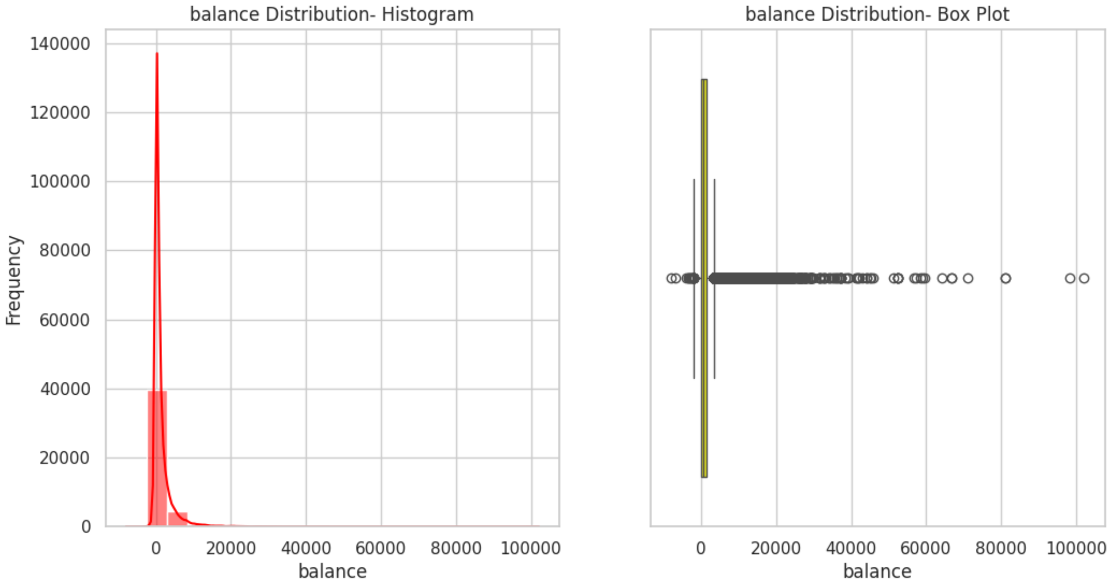
```
cont_plot(df,'age' )
```



[49]:

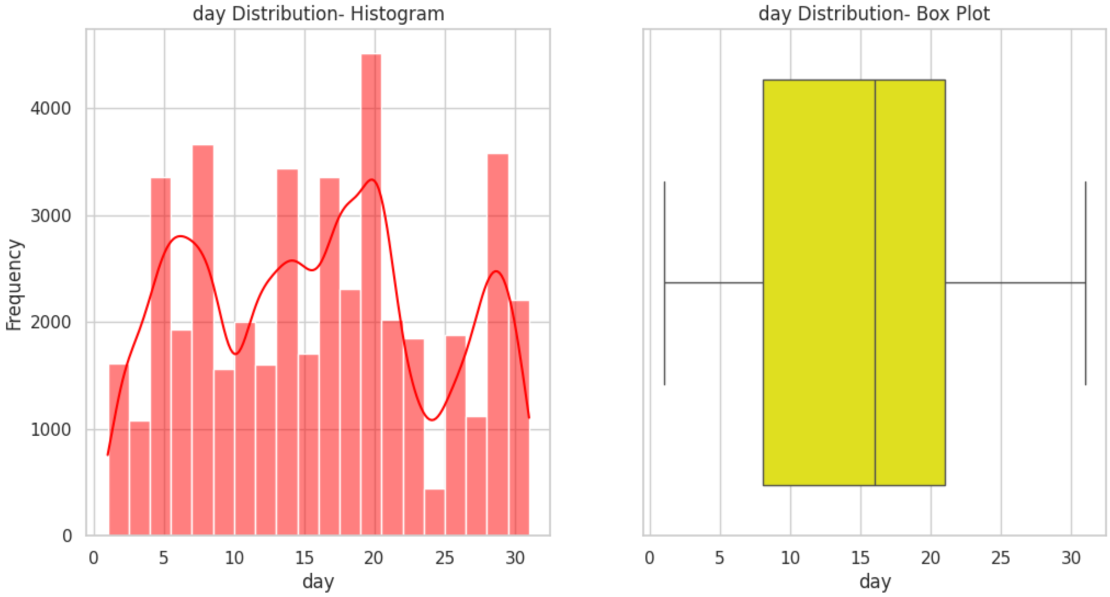
```
# Boxplot without outliers
NoOutlier(df,"balance")
cont_plot(df,'balance' )
```


Boxplot of balance (without outliers)
−2000
−1000
0
1000
2000
3000
balance


balance Distribution- Histogram
140000
120000
100000
80000
60000
40000
20000
0
Frequency
0
20000
40000
60000
80000
100000
balance
balance Distribution- Box Plot
0
20000
40000
60000
80000
100000
balance

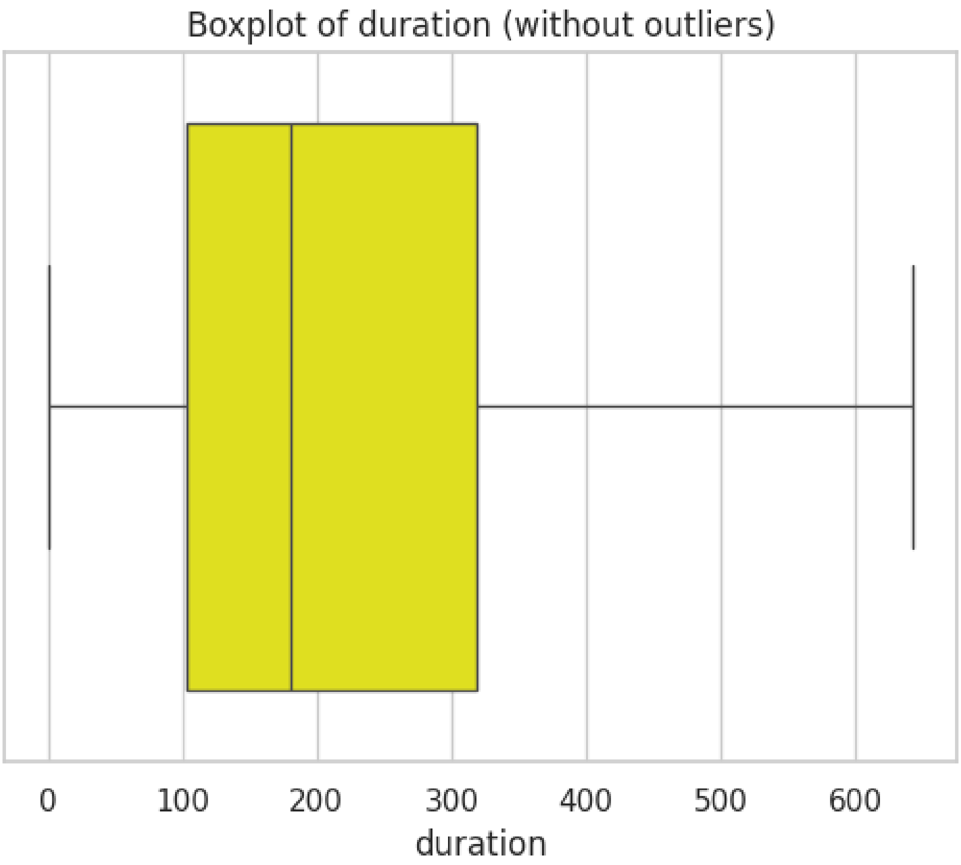
[44]:

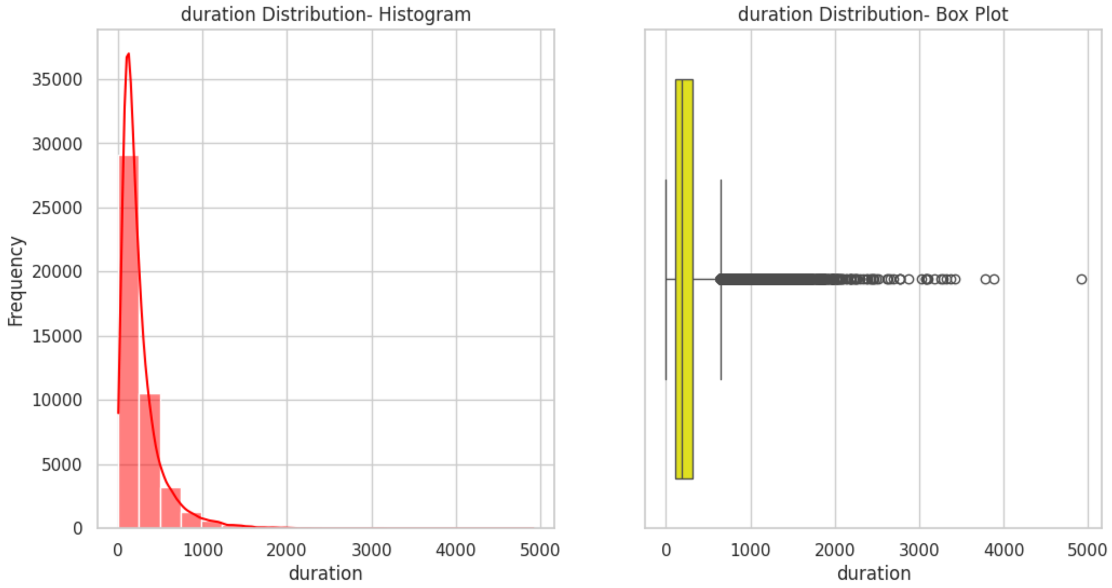
```
cont_plot(df,'day' )
```



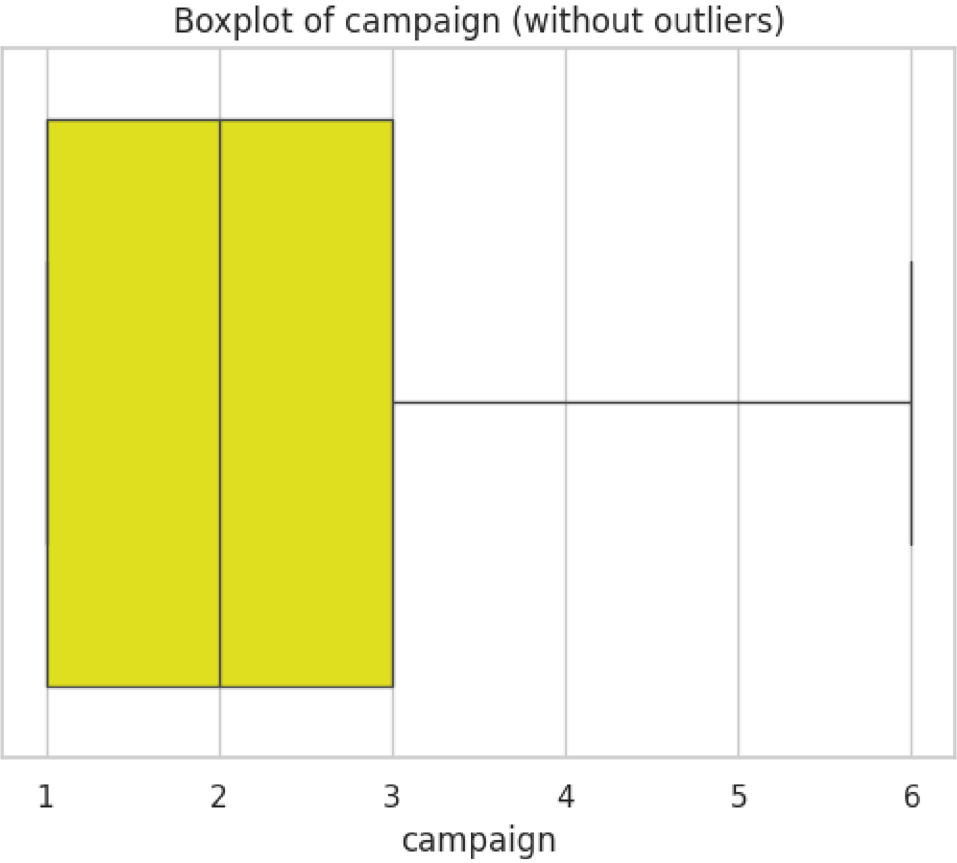
[50]:

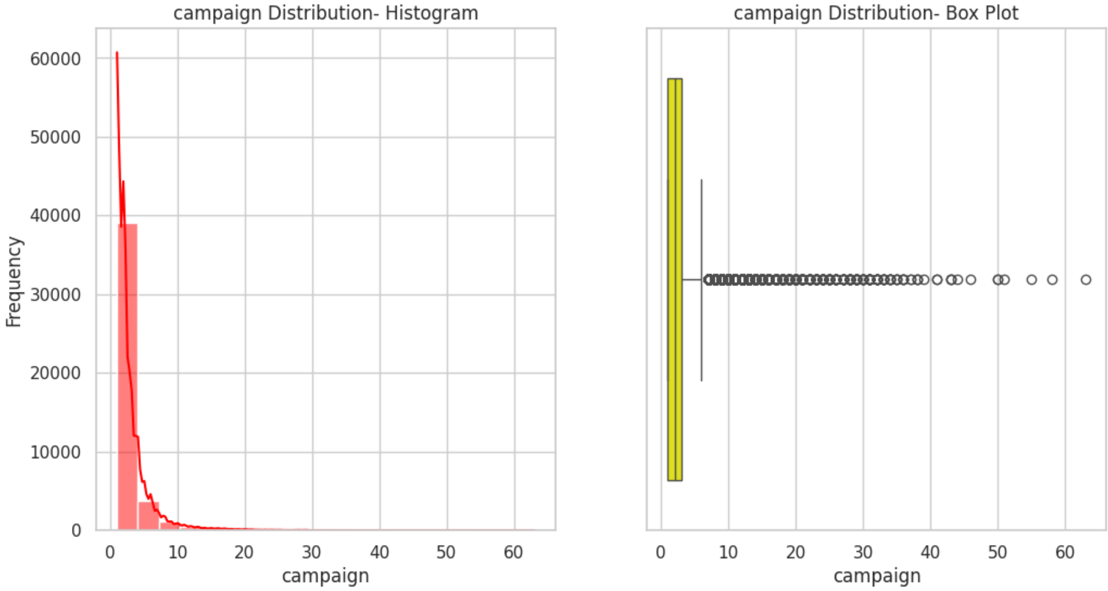
```
NoOutlier(df,"duration")
cont_plot(df,'duration' )
```


Boxplot of duration (without outliers)
0
100
200
300
400
500
600
duration


duration Distribution- Histogram
Frequency
35000
30000
25000
20000
15000
10000
5000
0
0
1000
2000
3000
4000
5000
duration
duration Distribution- Box Plot
0
1000
2000
3000
4000
5000
duration

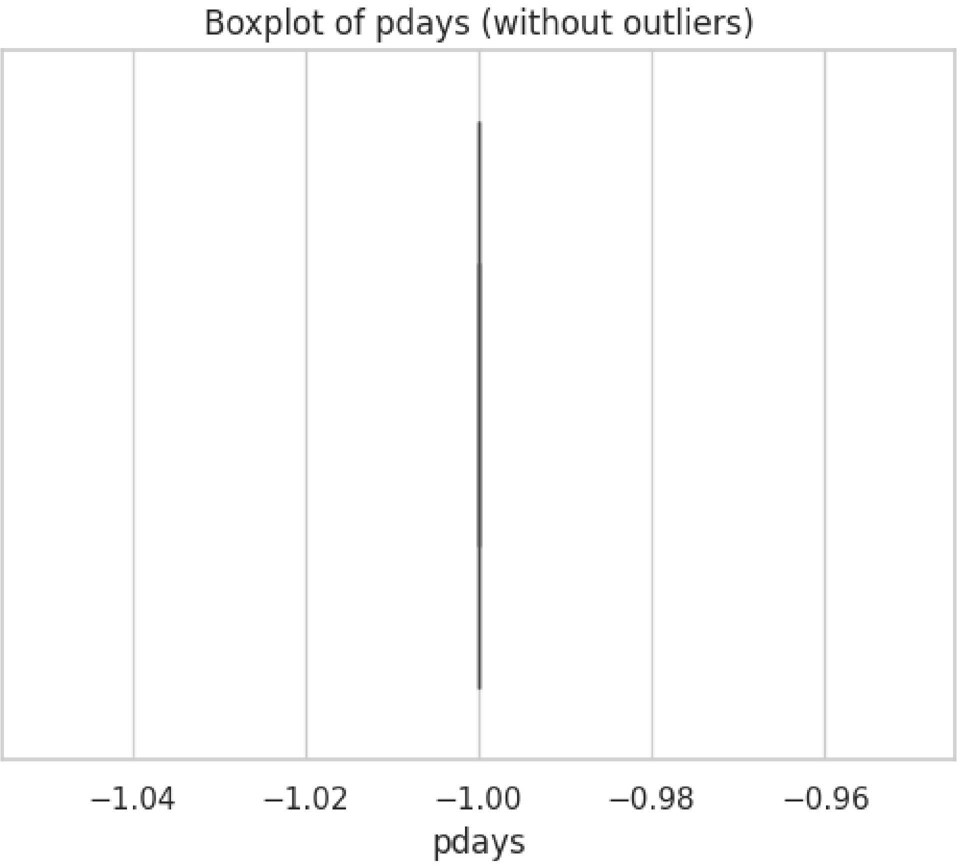
```
[52]: NoOutlier(df,"campaign")
      cont_plot(df,'campaign' )
```

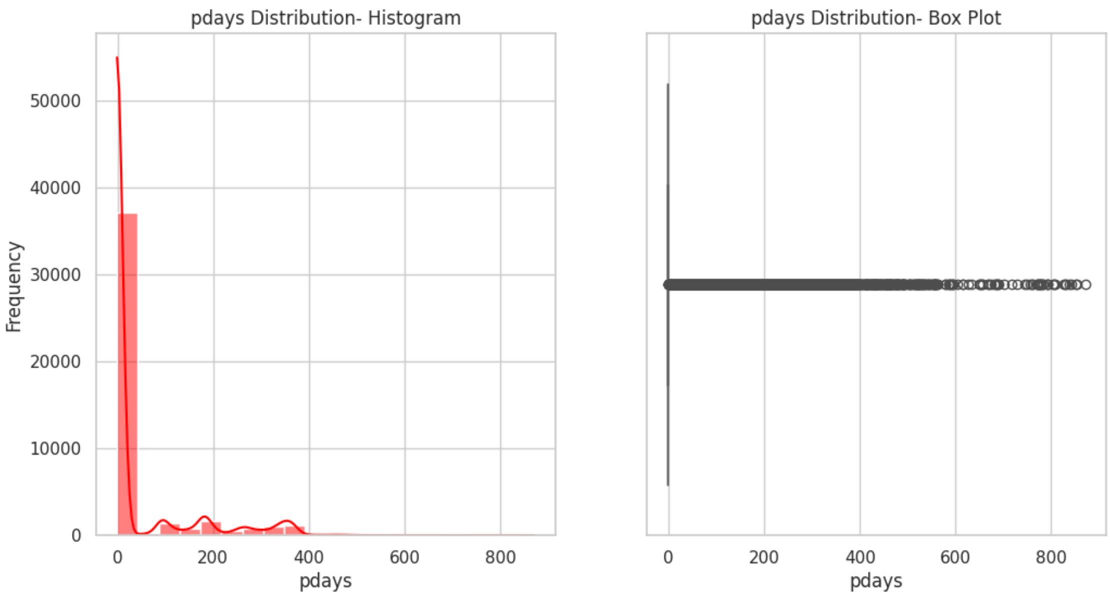




[53]:

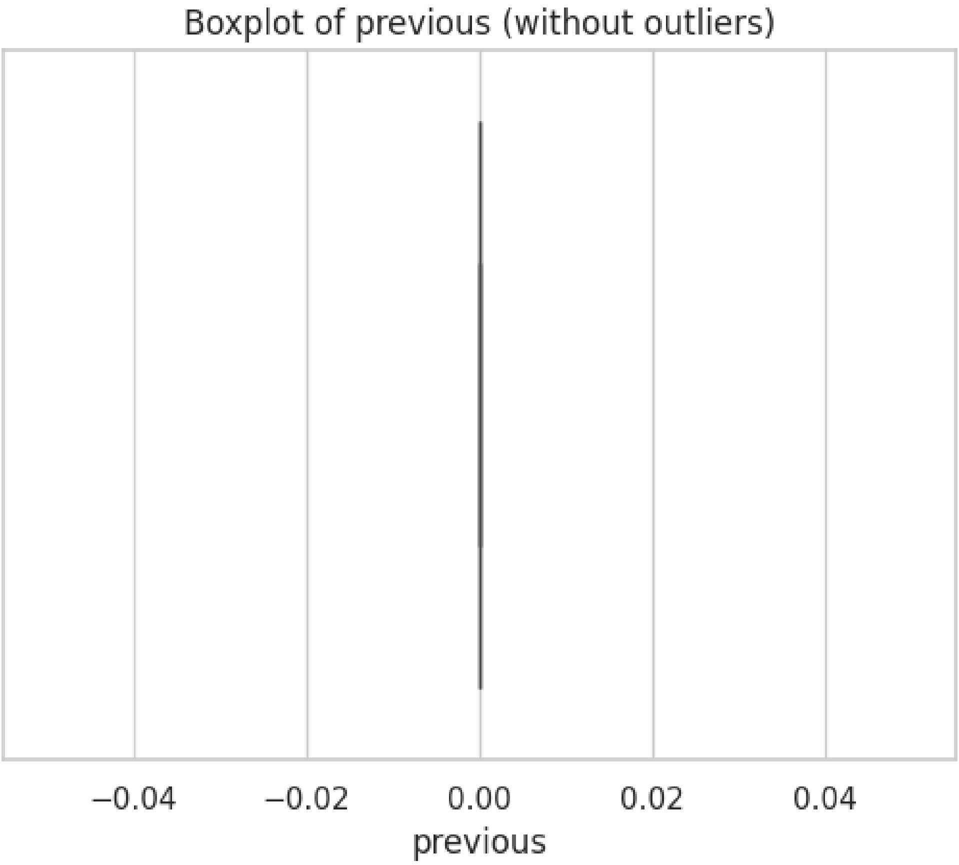
```
NoOutlier(df,"pdays")
cont_plot(df,'pdays' )
```

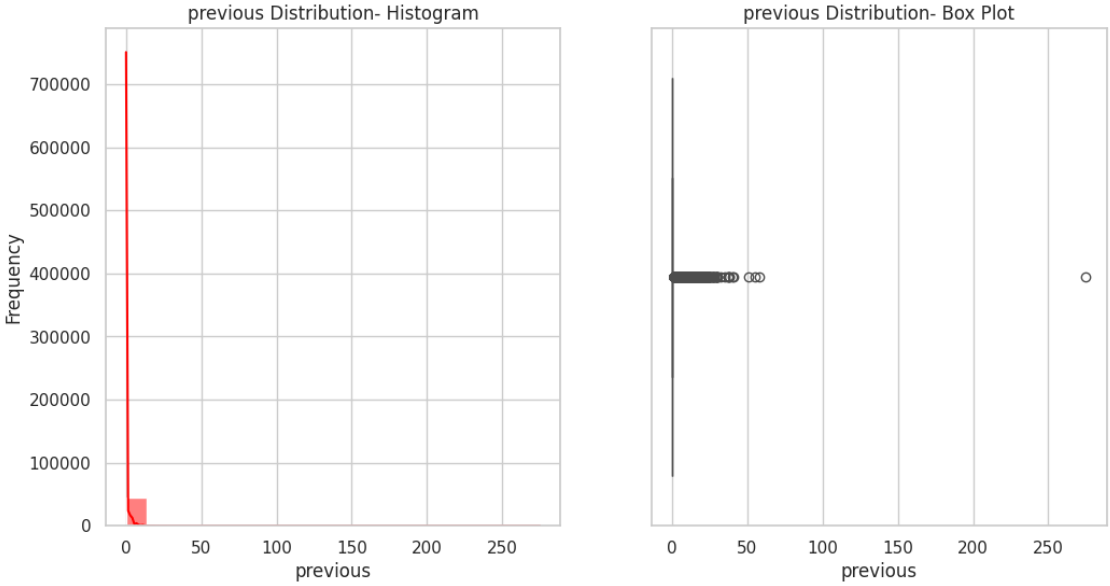




[54]:

```
NoOutlier(df,"previous")
cont_plot(df,'previous' )
```

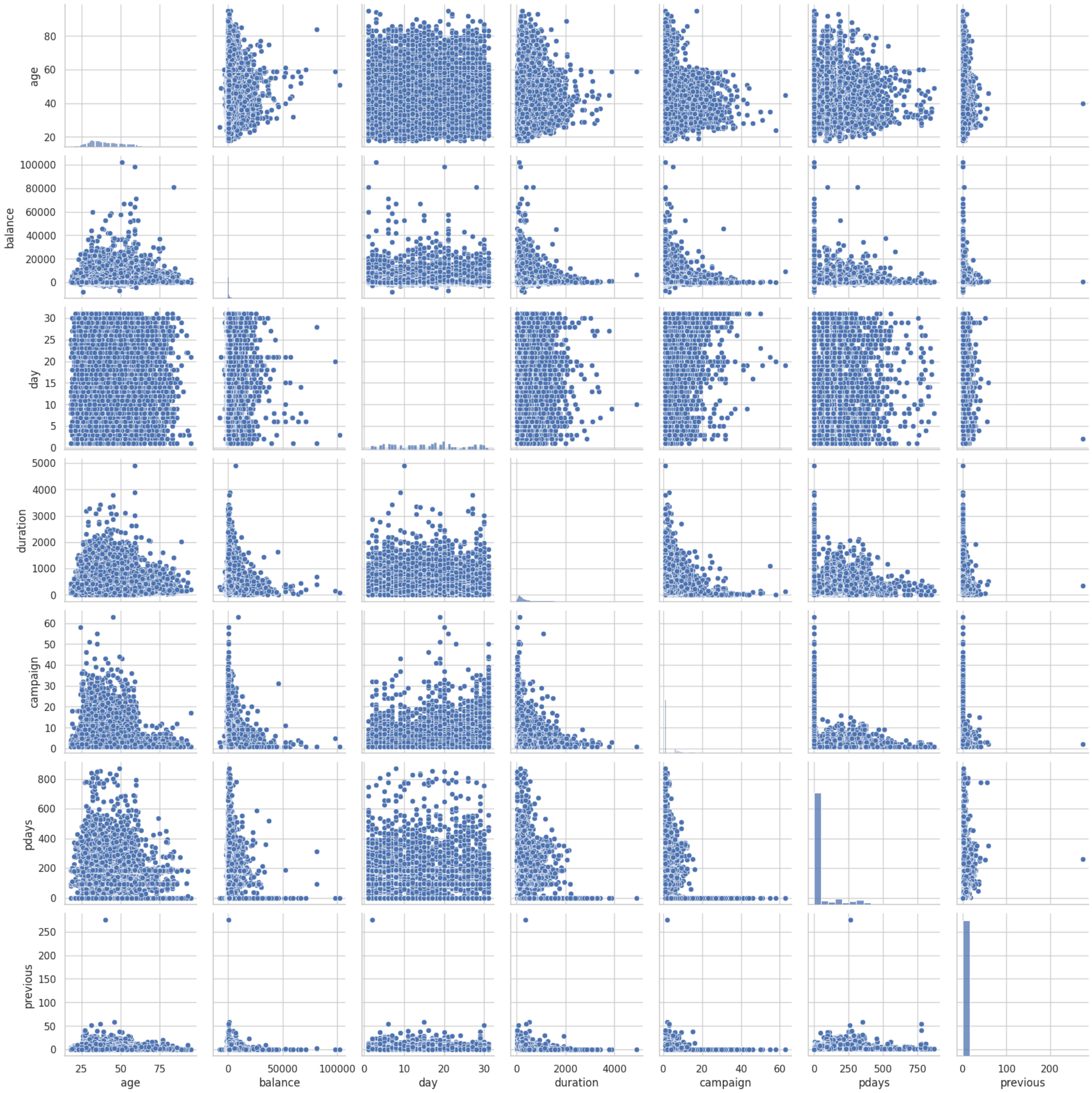




[55]:

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

# Pairwise scatter plots for numerical variables
sns.pairplot(df.select_dtypes(include=['number']))
plt.show()
```

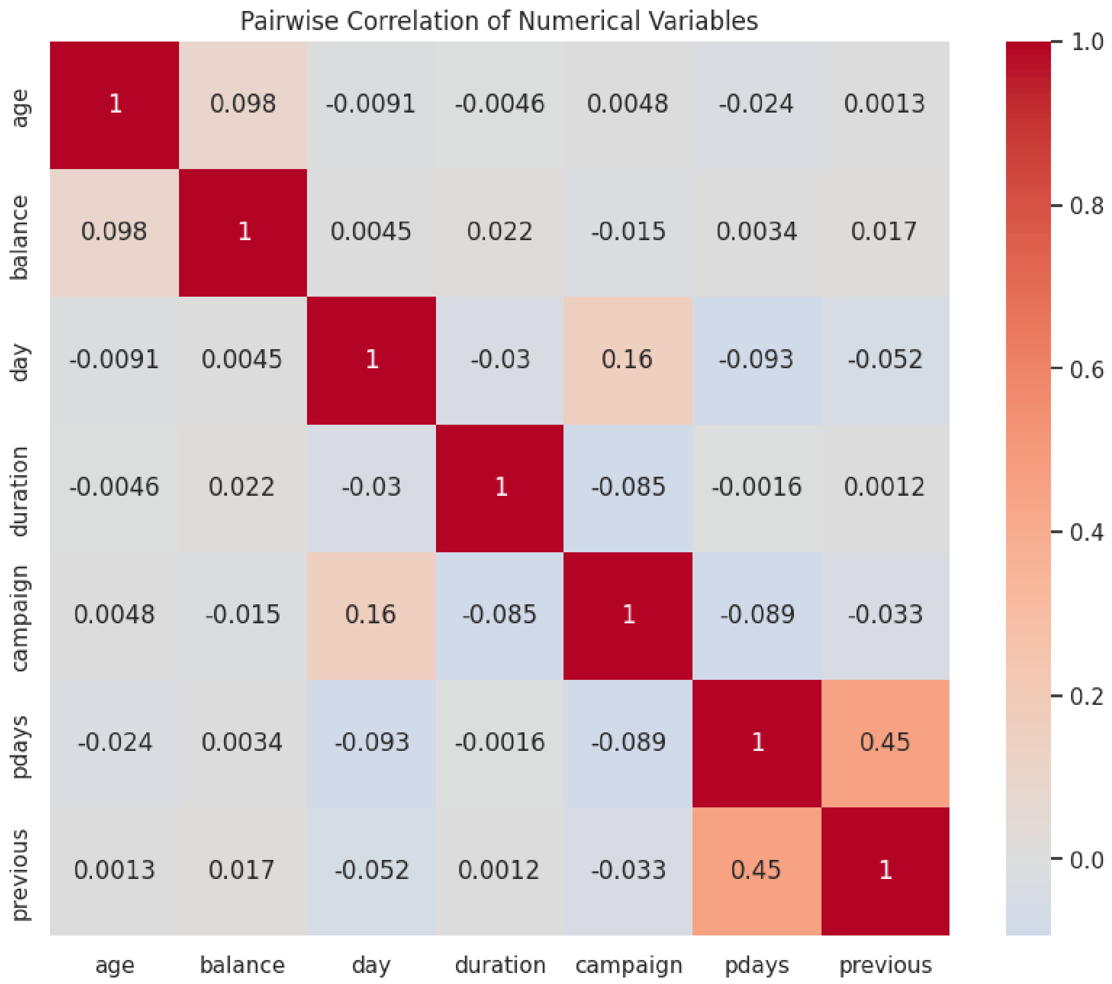


[56]:

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

# Compute correlation matrix
corr_matrix = df.select_dtypes(include=['number']).corr()

# Plot heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', center=0)
plt.title("Pairwise Correlation of Numerical Variables")
plt.show()
```



[30]:

```
int_columns = df.select_dtypes('int64').columns.tolist()

print(int_columns)
```

```
['age', 'balance', 'day', 'duration', 'campaign', 'pdays', 'previous']
```

[ ]:

[31]:

```
object_columns = df.select_dtypes(include=['object']).columns.tolist()

print(object_columns)
```

```
['job', 'marital', 'education', 'default', 'housing', 'loan', 'contact',
'month', 'poutcome', 'y']
```

[ ]:

[ ]: