```
from google.colab import files
uploaded = files.upload()
```

Choose Files Customer Churn.csv

- **Customer Churn.csv**(text/csv) - 977501 bytes, last modified: 3/2/2025 - 100% done

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

```
df = pd.read_csv('Customer Churn.csv')
df
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes
2	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Yes
4	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No
...	...	...	...	...	...	...	...	...	...	...
7038	6840-RESVB	Male	0	Yes	Yes	24	Yes	Yes	DSL	Yes
7039	2234-XADUH	Female	0	Yes	Yes	72	Yes	Yes	Fiber optic	No
7040	4801-JZAZL	Female	0	Yes	Yes	11	No	No phone service	DSL	Yes
7041	8361-LTMKD	Male	1	Yes	No	4	Yes	Yes	Fiber optic	No
7042	3186-AJIEK	Male	0	No	No	66	Yes	No	Fiber optic	Yes

7043 rows × 21 columns

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   customerID        7043 non-null   object
 1   gender            7043 non-null   object
 2   SeniorCitizen     7043 non-null   int64
 3   Partner           7043 non-null   object
 4   Dependents        7043 non-null   object
 5   tenure            7043 non-null   int64
 6   PhoneService      7043 non-null   object
 7   MultipleLines     7043 non-null   object
 8   InternetService   7043 non-null   object
 9   OnlineSecurity    7043 non-null   object
 10  OnlineBackup      7043 non-null   object
 11  DeviceProtection  7043 non-null   object
 12  TechSupport       7043 non-null   object
 13  StreamingTV       7043 non-null   object
 14  StreamingMovies   7043 non-null   object
 15  Contract          7043 non-null   object
 16  PaperlessBilling  7043 non-null   object
 17  PaymentMethod     7043 non-null   object
 18  MonthlyCharges    7043 non-null   float64
 19  TotalCharges      7043 non-null   object
 20  Churn             7043 non-null   object
dtypes: float64(1), int64(2), object(18)
memory usage: 1.1+ MB
```

```
df.size
```

```
147903
```

```
df.columns
```

```
Index(['customerID', 'gender', 'SeniorCitizen', 'Partner', 'Dependents',
       'tenure', 'PhoneService', 'MultipleLines', 'InternetService',
       'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport',
       'StreamingTV', 'StreamingMovies', 'Contract', 'PaperlessBilling',
       'PaymentMethod', 'MonthlyCharges', 'TotalCharges', 'Churn'],
      dtype='object')
```

```
df.isnull().sum()
```

	0
customerID	0
gender	0
SeniorCitizen	0
Partner	0
Dependents	0
tenure	0
PhoneService	0
MultipleLines	0
InternetService	0
OnlineSecurity	0
OnlineBackup	0
DeviceProtection	0
TechSupport	0
StreamingTV	0
StreamingMovies	0
Contract	0
PaperlessBilling	0
PaymentMethod	0
MonthlyCharges	0
TotalCharges	0
Churn	0

Replacing blanks with 0 as tenure is 0 and no total charges are recorded

```
df['TotalCharges'] = df['TotalCharges'].replace(" ", "0")
df['TotalCharges'] = df['TotalCharges'].astype("float")
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   customerID        7043 non-null   object
 1   gender            7043 non-null   object
 2   SeniorCitizen     7043 non-null   int64
 3   Partner           7043 non-null   object
 4   Dependents        7043 non-null   object
 5   tenure            7043 non-null   int64
 6   PhoneService      7043 non-null   object
 7   MultipleLines     7043 non-null   object
 8   InternetService   7043 non-null   object
 9   OnlineSecurity    7043 non-null   object
 10  OnlineBackup      7043 non-null   object
 11  DeviceProtection  7043 non-null   object
 12  TechSupport       7043 non-null   object
 13  StreamingTV       7043 non-null   object
 14  StreamingMovies   7043 non-null   object
```

```
 15  Contract          7043 non-null   object
 16  PaperlessBilling  7043 non-null   object
 17  PaymentMethod     7043 non-null   object
 18  MonthlyCharges    7043 non-null   float64
 19  TotalCharges      7043 non-null   float64
 20  Churn             7043 non-null   object
dtypes: float64(2), int64(2), object(17)
memory usage: 1.1+ MB
```

```
df.describe()
```

	SeniorCitizen	tenure	MonthlyCharges	TotalCharges
count	7043.000000	7043.000000	7043.000000	7043.000000
mean	0.162147	32.371149	64.761692	2279.734304
std	0.368612	24.559481	30.090047	2266.794470
min	0.000000	0.000000	18.250000	0.000000
25%	0.000000	9.000000	35.500000	398.550000
50%	0.000000	29.000000	70.350000	1394.550000
75%	0.000000	55.000000	89.850000	3786.600000
max	1.000000	72.000000	118.750000	8684.800000

```
df["customerID"].duplicated().sum()
```

```
0
```

Converted 0 and 1 values of senior citizen to yes/no to make it easier to understand

```
def conv(value):
  if value== 1:
    return "Yes"
  else:
      return "No"

df["SeniorCitizen"] = df["SeniorCitizen"].apply(conv)
```

```
df.head(20)
```

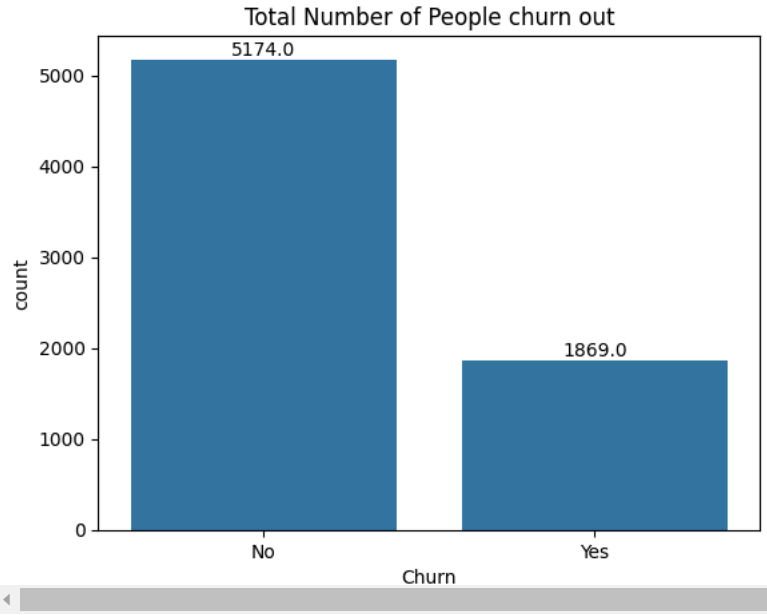
D	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...
3	Female	No	Yes	No	1	No	No phone service	DSL	No	...
E	Male	No	No	No	34	Yes	No	DSL	Yes	...
K	Male	No	No	No	2	Yes	No	DSL	Yes	...
V	Male	No	No	No	45	No	No phone service	DSL	Yes	...
J	Female	No	No	No	2	Yes	No	Fiber optic	No	...
C	Female	No	No	No	8	Yes	Yes	Fiber optic	No	...
K	Male	No	No	Yes	22	Yes	Yes	Fiber optic	No	...
C	Female	No	No	No	10	No	No phone service	DSL	Yes	...
D	Female	No	Yes	No	28	Yes	Yes	Fiber optic	No	...
J	Male	No	No	Yes	62	Yes	No	DSL	Yes	...
D	Male	No	Yes	Yes	13	Yes	No	DSL	Yes	...
I	Male	No	No	No	16	Yes	No	No	No internet service	...
X	Male	No	Yes	No	58	Yes	Yes	Fiber optic	No	...
X	Male	No	No	No	49	Yes	Yes	Fiber optic	No	...
S	Male	No	No	No	25	Yes	No	Fiber optic	Yes	...
Z	Female	No	Yes	Yes	69	Yes	Yes	Fiber optic	Yes	...
3	Female	No	No	No	52	Yes	No	No	No internet service	...
T	Male	No	No	Yes	71	Yes	Yes	Fiber optic	Yes	...
V	Female	No	Yes	Yes	10	Yes	No	DSL	No	...
B	Female	No	No	No	21	Yes	No	Fiber optic	No	...

umns

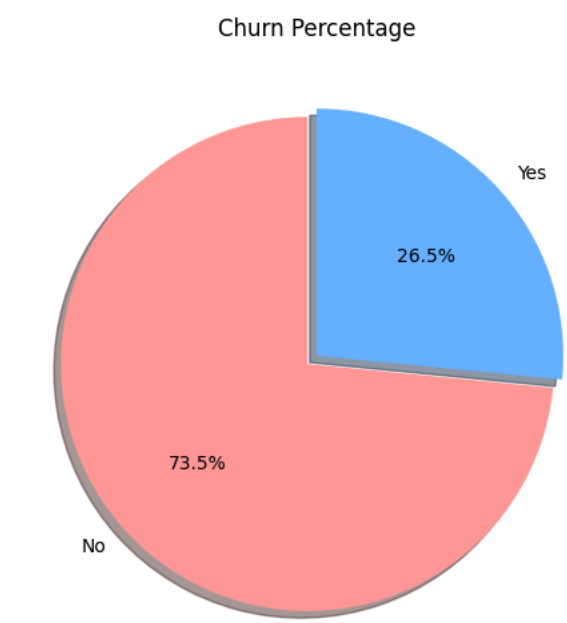
```
churn_counts = df['Churn'].value_counts()
print(churn_counts)
```

```
Churn
No     5174
Yes    1869
Name: count, dtype: int64
```

```
ax = sns.countplot(x=df['Churn'])
for p in ax.patches:
    ax.annotate(f'{p.get_height()}', (p.get_x() + p.get_width() / 2, p.get_height()),
                ha='center', va='bottom')
plt.title("Total Number of People churn out")

plt.show()
```



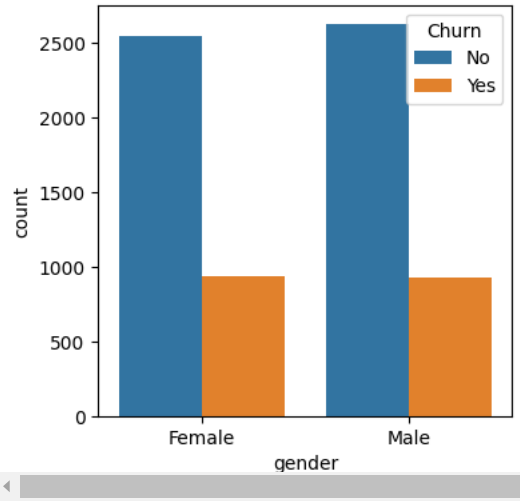

```
churn_counts = df['Churn'].value_counts()

# Labels for the pie chart
labels = churn_counts.index  # "Yes" and "No"

# Colors for better visualization
colors = ['#ff9999', '#66b3ff']

# Plot the pie chart
plt.figure(figsize=(6, 6))
plt.pie(churn_counts, labels=labels, autopct='%1.1f%%', colors=colors, startangle=90, shadow=True, explode=[0.05, 0])

# Title of the chart
plt.title("Churn Percentage")

# Show the chart
plt.show()
```



From this pie chart we can conclude that 26.5% of our customers have churned out

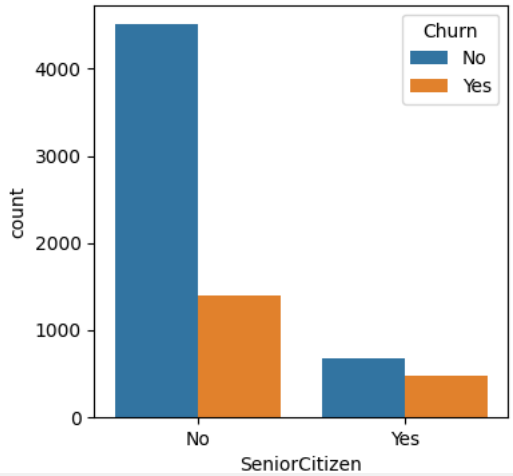
```
plt.figure(figsize=(4, 4))
sns.countplot(x='gender', hue='Churn', data=df)
```

```
<Axes: xlabel='gender', ylabel='count'>
```




```
plt.figure(figsize=(4, 4))
sns.countplot(x='SeniorCitizen', hue='Churn', data=df)
```

```
<Axes: xlabel='SeniorCitizen', ylabel='count'>
```




```
import matplotlib.pyplot as plt

# Count churned vs. not churned for Senior Citizens
senior_churn = df[df["SeniorCitizen"] == 1]["Churn"].value_counts()
print(senior_churn)
```

```
Series([], Name: count, dtype: int64)
```

```
print(df["Churn"].unique())
```

```
['No' 'Yes']
```

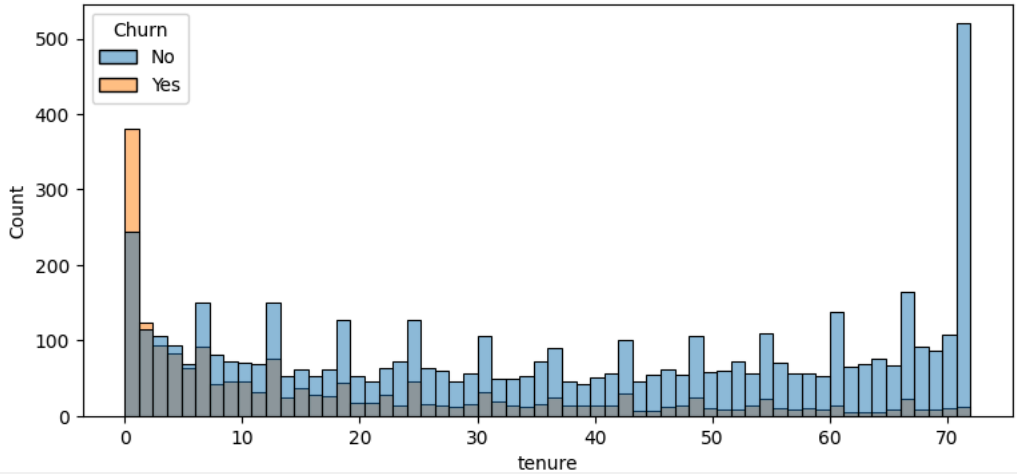
```
senior_churn = df[df["SeniorCitizen"] == 1]["Churn"].value_counts()

# Check if senior_churn is empty
if senior_churn.empty:
    print("No senior citizens found in the dataset.")
else:
    labels = senior_churn.index
    colors = ["#ff9999", "#66b3ff"]
    explode = [0.05] * len(senior_churn)

    plt.figure(figsize=(6, 6))
    plt.pie(senior_churn, labels=labels, autopct='%1.1f%%', colors=colors,
            startangle=90, shadow=True, explode=explode)

    plt.title("Senior Citizens Churn Percentage")
    plt.show()
```

```
No senior citizens found in the dataset.
```

```
senior_count = df[df["SeniorCitizen"] == 1].shape[0]
print(f"Total Senior Citizens: {senior_count}")
```

```
Total Senior Citizens: 0
```

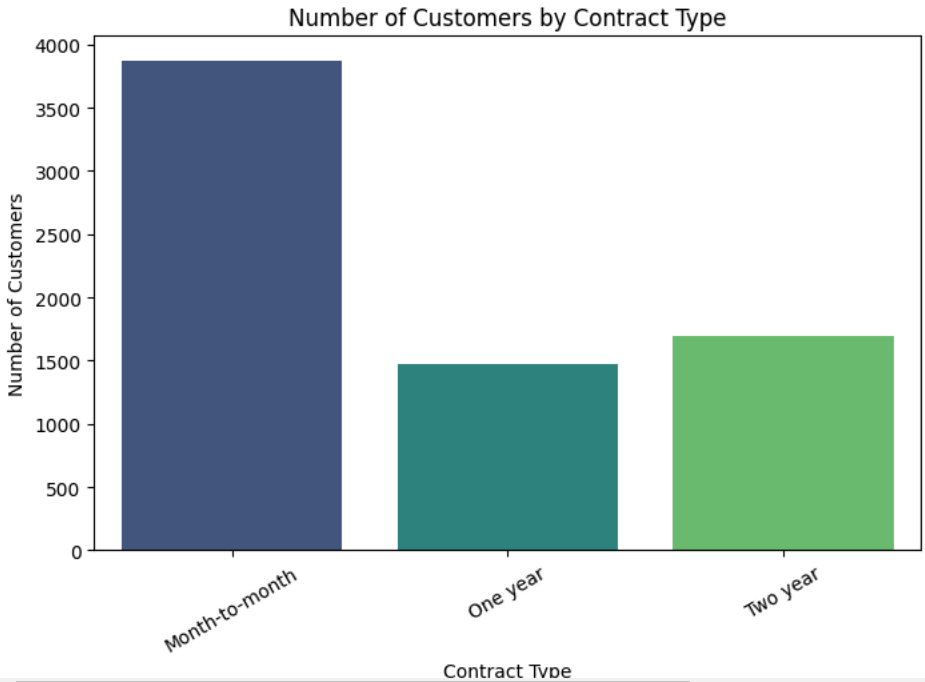
```
plt.figure(figsize=(9, 4))
sns.histplot(x='tenure', data=df,bins=60, hue="Churn" )
plt.show()
```



People who have used our services for long time has stayed and people who have used our services for 1 or 2 months have churned

```
plt.figure(figsize=(8, 5))
sns.countplot(x=df["Contract"], palette="viridis")
plt.title("Number of Customers by Contract Type")
plt.xlabel("Contract Type")
plt.ylabel("Number of Customers")
plt.xticks(rotation=30)
plt.show()
```

```
<ipython-input-34-017b081869e3>: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 `le

  sns.countplot(x=df["Contract"], palette="viridis")
```



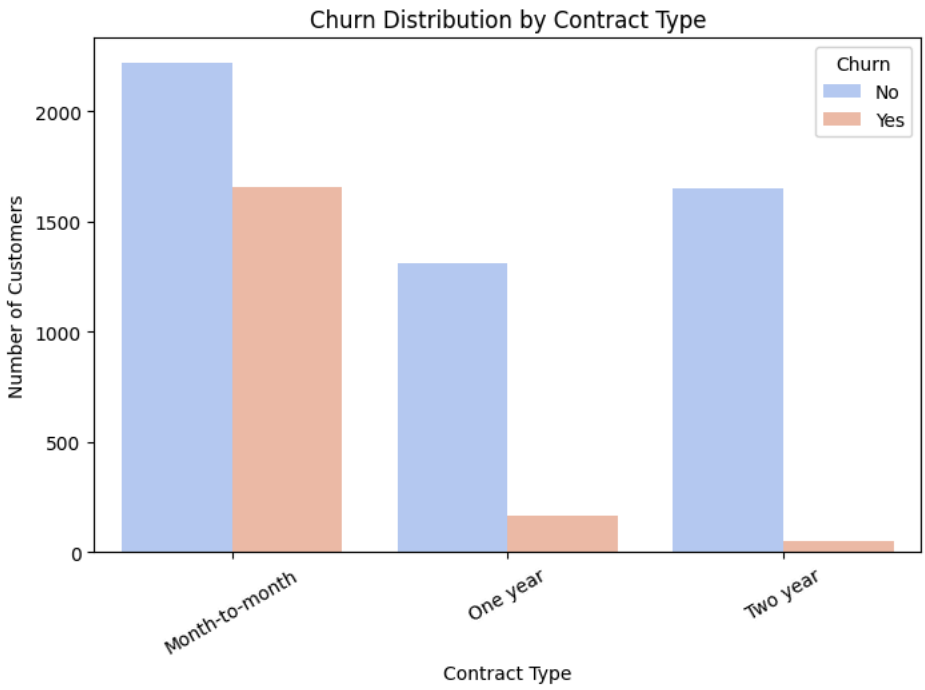
```
churn_by_contract = df[df["Churn"] == "Yes"]["Contract"].value_counts()
print(churn_by_contract)
```

```
Contract
Month-to-month    1655
One year           166
Two year            48
Name: count, dtype: int64
```

```
contract_churn_rate = df.groupby("Contract")["Churn"].value_counts(normalize=True).unstack() * 100
print(contract_churn_rate)
```

```
Churn                  No        Yes
Contract
Month-to-month  57.290323  42.709677
One year        88.730482  11.269518
Two year        97.168142   2.831858
```

```
plt.figure(figsize=(8, 5))
sns.countplot(x="Contract", hue="Churn", data=df, palette="coolwarm")
plt.title("Churn Distribution by Contract Type")
plt.xlabel("Contract Type")
plt.ylabel("Number of Customers")
plt.legend(title="Churn")
plt.xticks(rotation=30)
plt.show()
```



people who have month to month contract are likely to churn then who have stayed longer

```
df.columns.values
```

```
array(['customerID', 'gender', 'SeniorCitizen', 'Partner', 'Dependents',
       'tenure', 'PhoneService', 'MultipleLines', 'InternetService',
       'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',
       'TechSupport', 'StreamingTV', 'StreamingMovies', 'Contract',
       'PaperlessBilling', 'PaymentMethod', 'MonthlyCharges',
       'TotalCharges', 'Churn'], dtype=object)
```

```
# List of service-related columns
service_columns = ['PhoneService', 'MultipleLines', 'InternetService',
                   'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',
                   'TechSupport', 'StreamingTV', 'StreamingMovies']

# Set figure size
fig, axes = plt.subplots(3, 3, figsize=(11, 11))  # 3x3 grid for subplots
fig.suptitle("Customer Churn Based on Services", fontsize=16)
```

```
# Plot countplots for each service with hue='Churn'
for ax, col in zip(axes.flat, service_columns):
    sns.countplot(x=col, data=df, hue="Churn", ax=ax, palette="coolwarm")  # Churn as hue
    ax.set_title(col)
    ax.set_xlabel("")
    ax.set_ylabel("Count")
    ax.set_xticklabels(ax.get_xticklabels(), rotation=20)

plt.tight_layout(rect=[0, 0, 1, 0.96])  # Adjust layout
plt.show()
```

```
<ipython-input-43-8b033348661d>:16: UserWarning: set_ticklabels() should only be used with a fixed number of ticks, i.e. after set_t
  ax.set_xticklabels(ax.get_xticklabels(), rotation=20)
<ipython-input-43-8b033348661d>:16: UserWarning: set_ticklabels() should only be used with a fixed number of ticks, i.e. after set_t
```

```
# Count churned customers based on PaymentMethod
churn_counts_by_payment = df[df["Churn"] == "Yes"]["PaymentMethod"].value_counts()

# Display the result
print(churn_counts_by_payment)
```

```
PaymentMethod
Electronic check             1071
Mailed check                  308
Bank transfer (automatic)     258
Credit card (automatic)       232
Name: count, dtype: int64
```

```
plt.figure(figsize=(8, 5))
sns.barplot(x=churn_counts_by_payment.index, y=churn_counts_by_payment.values, palette="coolwarm")
plt.xlabel("Payment Method")
plt.ylabel("Number of Churned Customers")
plt.title("Churn Count by Payment Method")
plt.xticks(rotation=20)
plt.show()
```

```
<ipython-input-45-c9c12ab16f53>: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 `le

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