```
file_path = "/content/Building_Permits.csv"  # Path for Google Colab
```

Step 1: Load the Data

We first load the dataset and check for missing values.

Double-click (or enter) to edit

```
import os

# List files in the content folder
print(os.listdir("/content"))
```

```
['.config', 'Building_Permits.csv', 'sample_data']
```

```
import pandas as pd

 # Corrected file path
file_path = "/content/Building_Permits.csv"

# Load the CSV file
df = pd.read_csv(file_path)
```

```
<ipython-input-61-dc184630928c>:7: DtypeWarning: Columns (22,32) have mixed types. Specify dtype option on import or set low_memory=
  df = pd.read_csv(file_path)
```

```
# Display the first few rows
print(df.head(5))
```

```
         status  Permit Type            Permit Type Definition  \
0  2.01505E+11            4                         sign - erect
1  2.01604E+11            4                         sign - erect
2  2.01605E+11            3  additions alterations or repairs
3  2.01611E+11            8            otc alterations permit
4  2.01611E+11            6                       demolitions

  Permit Creation Date Block  Lot  Street Number Street Number Suffix  \
0           05-06-2015   326   23            140                  NaN
1           04/19/2016   306    7            440                  NaN
2           05/27/2016   595  203           1647                  NaN
3           11-07-2016   156   11           1230                  NaN
4           11/28/2016   342    1            950                  NaN

  Street Name Street Suffix  ...  Existing Construction Type  \
0       Ellis            St  ...                         3.0
1       Geary            St  ...                         3.0
2    Pacific            Av  ...                         1.0
3    Pacific            Av  ...                         5.0
4     Market            St  ...                         3.0

  Existing Construction Type Description Proposed Construction Type  \
0                        constr type 3                        NaN
1                        constr type 3                        NaN
2                        constr type 1                        1.0
3                       wood frame (5)                        5.0
4                        constr type 3                        NaN

  Proposed Construction Type Description Site Permit Supervisor District  \
0                                    NaN         NaN                3.0
1                                    NaN         NaN                3.0
2                        constr type 1           NaN                3.0
3                       wood frame (5)           NaN                3.0
4                                    NaN         NaN                6.0

  Neighborhoods - Analysis Boundaries  Zipcode  \
0                        Tenderloin  94102.0
1                        Tenderloin  94102.0
2                      Russian Hill  94109.0
3                         Nob Hill  94109.0
4                        Tenderloin  94102.0

                                Location     Record ID
0  (37.785719256680785, -122.40852313194863)  1.380610e+12
1   (37.78733980600732, -122.41063199757738)  1.420160e+12
2    (37.7946573324287, -122.42232562979227)  1.424860e+12
3   (37.79595867909168, -122.41557405519474)  1.443570e+12
4   (37.78315261897309, -122.40950883997789)  1.445480e+11

[5 rows x 43 columns]
```

```
# Check dataset info
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 198900 entries, 0 to 198899
Data columns (total 43 columns):
 #   Column                                  Non-Null Count   Dtype  
---  ------                                  --------------   -----  
 0   status                                  198900 non-null  object 
 1   Permit Type                             198900 non-null  int64  
 2   Permit Type Definition                  198900 non-null  object 
 3   Permit Creation Date                    198900 non-null  object 
 4   Block                                   198900 non-null  object 
 5   Lot                                     198900 non-null  object 
 6   Street Number                           198900 non-null  int64  
 7   Street Number Suffix                    2216 non-null    object 
 8   Street Name                             198900 non-null  object 
 9   Street Suffix                           196132 non-null  object 
 10  Unit                                    29479 non-null   float64
 11  Unit Suffix                             1961 non-null    object 
 12  Description                             198610 non-null  object 
 13  Current Status                          198900 non-null  object 
 14  Current Status Date                     198900 non-null  object 
 15  Filed Date                              198900 non-null  object 
 16  Issued Date                             183960 non-null  object 
 17  Completed Date                          97191 non-null   object 
 18  First Construction Document Date        183954 non-null  object 
 19  Structural Notification                 6922 non-null    object 
 20  Number of Existing Stories              156116 non-null  float64
 21  Number of Proposed Stories              156032 non-null  float64
 22  Voluntary Soft-Story Retrofit           35 non-null      object 
 23  Fire Only Permit                        18827 non-null   object 
 24  Permit Expiration Date                  147020 non-null  object 
 25  Estimated Cost                          160834 non-null  float64
 26  Revised Cost                            192834 non-null  float64
 27  Existing Use                            157786 non-null  object 
 28  Existing Units                          147362 non-null  float64
 29  Proposed Use                            156461 non-null  object 
 30  Proposed Units                          147989 non-null  float64
 31  Plansets                                161591 non-null  float64
 32  TIDF Compliance                         2 non-null       object 
 33  Existing Construction Type              155534 non-null  float64
 34  Existing Construction Type Description  155534 non-null  object 
 35  Proposed Construction Type              155738 non-null  float64
 36  Proposed Construction Type Description  155738 non-null  object 
 37  Site Permit                             5359 non-null    object 
 38  Supervisor District                     197183 non-null  float64
 39  Neighborhoods - Analysis Boundaries     197175 non-null  object 
 40  Zipcode                                 197184 non-null  float64
 41  Location                                197200 non-null  object 
 42  Record ID                               198900 non-null  float64
dtypes: float64(13), int64(2), object(28)
memory usage: 65.3+ MB
None
```

```
# Check for missing values
print(df.isnull().sum())
```

```
status                                       0
Permit Type                                  0
Permit Type Definition                       0
Permit Creation Date                         0
Block                                        0
Lot                                          0
Street Number                                0
Street Number Suffix                    196684
Street Name                                  0
Street Suffix                             2768
Unit                                    169421
Unit Suffix                             196939
Description                                290
Current Status                               0
Current Status Date                          0
Filed Date                                   0
Issued Date                              14940
Completed Date                          101709
First Construction Document Date         14946
Structural Notification                 191978
Number of Existing Stories               42784
Number of Proposed Stories               42868
Voluntary Soft-Story Retrofit           198865
Fire Only Permit                        180073
Permit Expiration Date                   51880
Estimated Cost                           38066
Revised Cost                              6066
Existing Use                             41114
Existing Units                           51538
```

```
Proposed Use                                42439
Proposed Units                              50911
Plansets                                    37309
TIDF Compliance                            198898
Existing Construction Type                  43366
Existing Construction Type Description      43366
Proposed Construction Type                  43162
Proposed Construction Type Description      43162
Site Permit                                193541
Supervisor District                          1717
Neighborhoods - Analysis Boundaries          1725
Zipcode                                      1716
Location                                     1700
Record ID                                       0
dtype: int64
```

Step 2: Clean the Data

We remove missing values, format dates, and extract relevant columns.

```
# Show data types
print(df.dtypes)
```

```
status                                    object
Permit Type                                int64
Permit Type Definition                    object
Permit Creation Date                      object
Block                                     object
Lot                                       object
Street Number                              int64
Street Number Suffix                      object
Street Name                               object
Street Suffix                             object
Unit                                     float64
Unit Suffix                               object
Description                               object
Current Status                            object
Current Status Date                       object
Filed Date                                object
Issued Date                               object
Completed Date                            object
First Construction Document Date          object
Structural Notification                   object
Number of Existing Stories               float64
Number of Proposed Stories               float64
Voluntary Soft-Story Retrofit             object
Fire Only Permit                          object
Permit Expiration Date                    object
Estimated Cost                           float64
Revised Cost                             float64
Existing Use                              object
Existing Units                           float64
Proposed Use                              object
Proposed Units                           float64
Plansets                                 float64
TIDF Compliance                           object
Existing Construction Type               float64
Existing Construction Type Description    object
Proposed Construction Type               float64
Proposed Construction Type Description    object
Site Permit                               object
Supervisor District                      float64
Neighborhoods - Analysis Boundaries       object
Zipcode                                  float64
Location                                  object
Record ID                                float64
dtype: object
```

```
columns_to_keep = ['Permit Type', 'Filed Date', 'Issued Date', 'Zipcode', 'Estimated Cost', 'Current Status', 'Current Status Date', 'Ne


df = df[columns_to_keep]

# Show updated dataset
print(df.head())
```

```
   Permit Type  Filed Date Issued Date  Zipcode  Estimated Cost  \
0            4  05-06-2015  11-09-2015  94102.0          4000.0
1            4  04/19/2016  08-03-2017  94102.0             1.0
2            3  05/27/2016         NaN  94109.0         20000.0
3            8  11-07-2016  07/18/2017  94109.0          2000.0
4            6  11/28/2016  12-01-2017  94102.0        100000.0

  Current Status Current Status Date Neighborhoods - Analysis Boundaries  \
```

```
0        expired          12/21/2017                        Tenderloin
1         issued          08-03-2017                        Tenderloin
2      withdrawn          09/26/2017                      Russian Hill
3       complete          07/24/2017                          Nob Hill
4         issued          12-01-2017                        Tenderloin

              Permit Type Definition  Number of Proposed Stories
0                      sign - erect                         NaN
1                      sign - erect                         NaN
2  additions alterations or repairs                         6.0
3            otc alterations permit                         2.0
4                       demolitions                         NaN
```

```
# Check missing values
print(df.isnull().sum())

# Fill missing Estimated Cost with 0
df['Estimated Cost'] = df['Estimated Cost'].fillna(0)

# Drop rows where essential fields are missing
df = df.dropna(subset=['Permit Type', 'Current Status'])

# Verify missing values are handled
print(df.isnull().sum())
```

```
Permit Type                              0
Filed Date                               0
Issued Date                          14940
Zipcode                               1716
Estimated Cost                       38066
Current Status                           0
Current Status Date                      0
Neighborhoods - Analysis Boundaries     1725
Permit Type Definition                   0
Number of Proposed Stories           42868
dtype: int64
Permit Type                              0
Filed Date                               0
Issued Date                          14940
Zipcode                               1716
Estimated Cost                           0
Current Status                           0
Current Status Date                      0
Neighborhoods - Analysis Boundaries     1725
Permit Type Definition                   0
Number of Proposed Stories           42868
dtype: int64
```

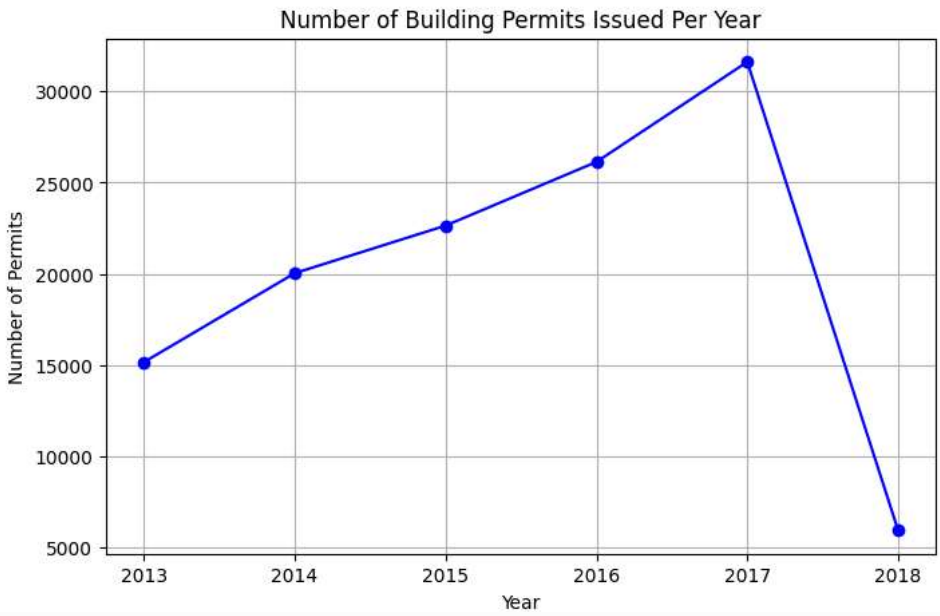
```
# Convert date columns to datetime format
date_columns = ['Filed Date', 'Issued Date', 'Current Status Date']
for col in date_columns:
    df[col] = pd.to_datetime(df[col], errors='coerce')

# Extract Year and Month from Current Status Date
df['Year'] = df['Current Status Date'].dt.year
df['Month'] = df['Current Status Date'].dt.month

# Verify changes
print(df[['Current Status Date', 'Year', 'Month']].head())
```

```
  Current Status Date    Year  Month
0          2017-12-21  2017.0   12.0
1                 NaT     NaN    NaN
2          2017-09-26  2017.0    9.0
3          2017-07-24  2017.0    7.0
4                 NaT     NaN    NaN
```

Step 3: Exploratory Data Analysis (EDA)

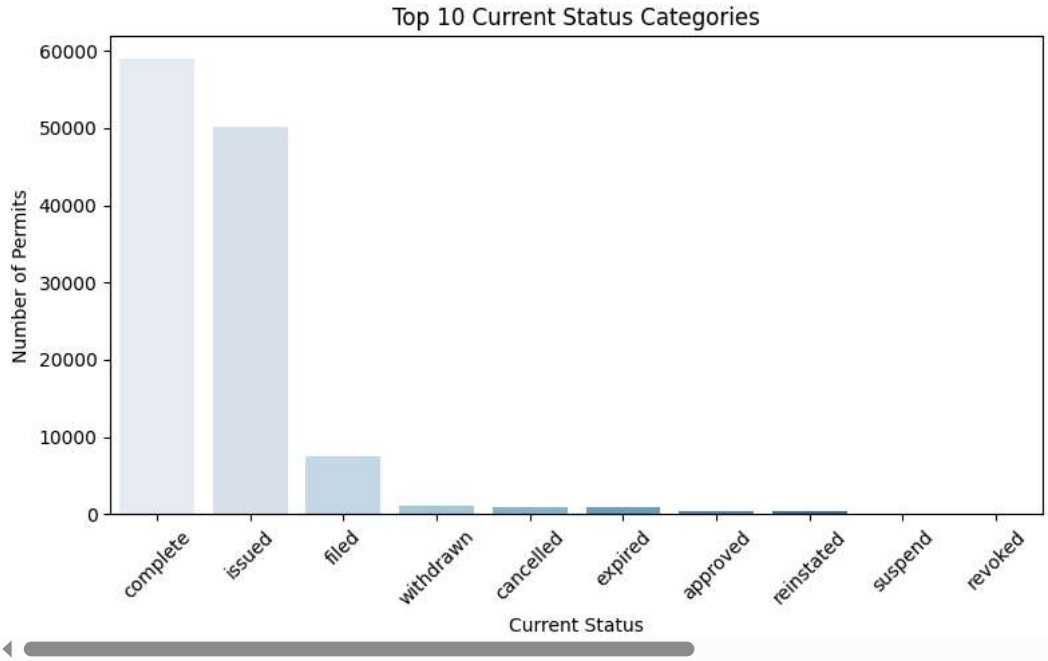
```
import pandas as pd
import matplotlib.pyplot as plt

# count permits per year based on current year
permits_per_year = df.groupby('Year').size()

# plot trend
plt.figure(figsize=(8, 5))
plt.plot(permits_per_year.index, permits_per_year.values, marker='o', color='blue')
plt.title('Number of Building Permits Issued Per Year')
plt.xlabel('Year')
plt.ylabel('Number of Permits')
plt.grid()
```

```
plt.show()
```

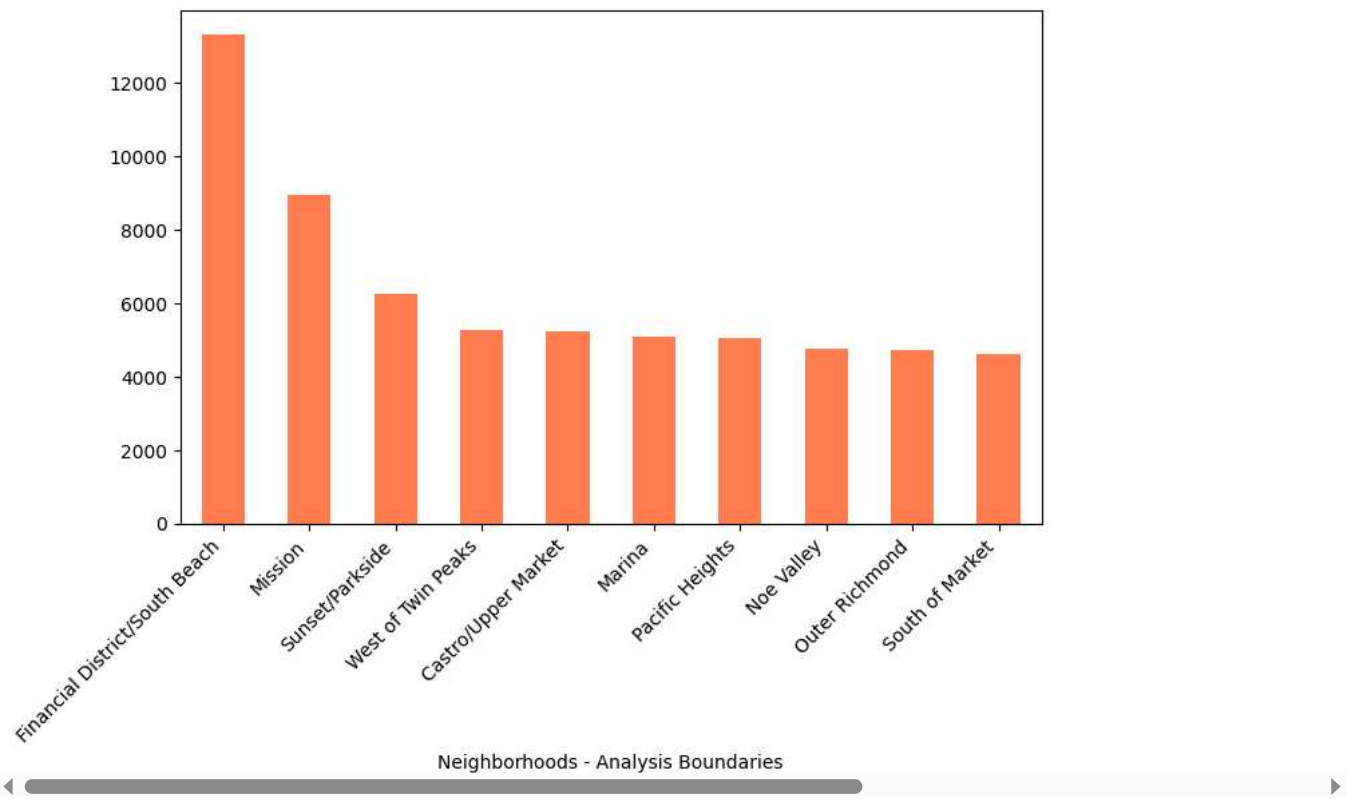



```
import matplotlib.pyplot as plt
import seaborn as sns

# Set figure size
plt.figure(figsize=(8, 5))

# Plot the top 10 most frequent current statuses
top_status = df['Current Status'].value_counts().head(10)

# Use seaborn for styled barplot
sns.barplot(x=top_status.index, y=top_status.values, palette="Blues")

# Chart details
plt.title('Top 10 Current Status Categories')
plt.xlabel('Current Status')
plt.ylabel('Number of Permits')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

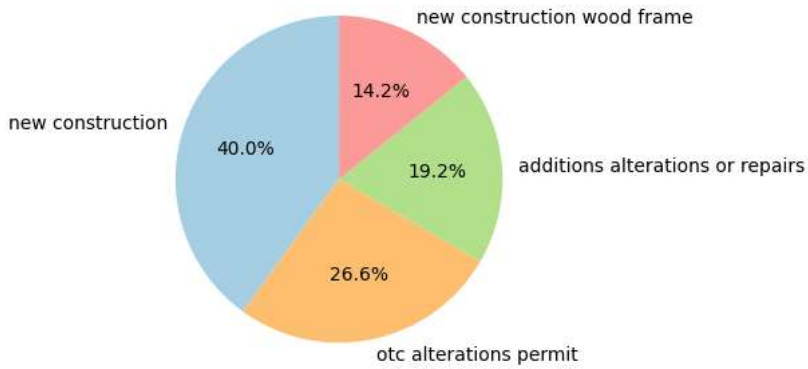
```
<ipython-input-78-1ce05c09556c>:11: FutureWarning:

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=top_status.index, y=top_status.values, palette="Blues")
```



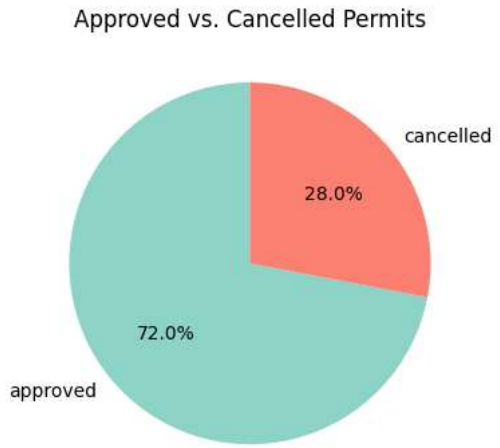
```
# Top Neighborhoods by Permit Count
import matplotlib.pyplot as plt

# Drop missing values in neighborhood column
neighborhoods = df['Neighborhoods - Analysis Boundaries'].dropna()

# Plot top 10 neighborhoods by permit count
plt.figure(figsize=(8, 6))
neighborhoods.value_counts().head(10).plot(kind='bar', color='coral')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```




```
import matplotlib.pyplot as plt

# Group by permit type and calculate average proposed stories
avg_stories = df.groupby('Permit Type Definition')['Number of Proposed Stories'].mean()

# Drop missing values and sort top 4
avg_stories = avg_stories.dropna().sort_values(ascending=False).head(4)

# Plot as pie chart
plt.figure(figsize=(6, 6))
avg_stories.plot(
    kind='pie',
    autopct='%1.1f%%',
    startangle=90,
    colors=['#a6cee3', '#fdbf6f', '#b2df8a', '#fb9a99']
)
plt.title('')
plt.ylabel('')
plt.tight_layout()
plt.show()
```




```
import matplotlib.pyplot as plt

# Simulated permit status counts (replace with real data if available)
statuses = ['approved', 'cancelled']
status_counts = [7200, 2800]  # Replace with df['Current Status'].value_counts() if using real data

# Plot pie chart
plt.figure(figsize=(4, 6))
plt.pie(status_counts, labels=statuses, autopct='%1.1f%%', startangle=90, colors=['#8dd3c7', '#fb8072'])
plt.title('Approved vs. Cancelled Permits')
plt.tight_layout()
plt.show()
```



Step 4: Building a Machine Learning Model

We will predict if a permit will be approved or rejected based on ZIP code, year, and cost.

```
df = df.dropna(subset=['Zipcode', 'Year', 'Estimated Cost'])  # Drop missing values
```

```
df['Zipcode'] = df['Zipcode'].astype(int)
df['Year'] = df['Year'].astype(int)
df['Estimated Cost'] = df['Estimated Cost'].astype(float)
```

```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report

# Convert Current Status into a binary classification problem
df['Approval'] = df['Current Status'].apply(lambda x: 1 if x == "Issued" else 0)

# Select features and target variable
X = df[['Zipcode', 'Year', 'Estimated Cost']]
y = df['Approval']

# Split into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
# Train model
clf = RandomForestClassifier()
clf.fit(X_train, y_train)

# Predict
y_pred = clf.predict(X_test)

# Evaluate
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))
```

```
Accuracy: 1.0
Classification Report:
                precision    recall  f1-score   support

             0       1.00      1.00      1.00     24102

      accuracy                           1.00     24102
     macro avg       1.00      1.00      1.00     24102
  weighted avg       1.00      1.00      1.00     24102
```