In [1]:
```
# Import necessary libraries
import pandas as pd
import matplotlib.pyplot as plt

# Load the dataset
dataset_path = "C:\\Users\\ARYAN PARIKH\\Downloads\\archive (1)\\AB_NYC_2019.c
df = pd.read_csv(dataset_path)

# Display basic information about the dataset
print("Original Dataset Info:")
print(df.info())

# Data Integrity: Ensure the accuracy, consistency, and reliability of data
# Example: Convert 'last_review' to datetime format
df['last_review'] = pd.to_datetime(df['last_review'])

# Missing Data Handling: Deal with missing values
# Example: Fill missing values in 'reviews_per_month' with the mean
df['reviews_per_month'].fillna(df['reviews_per_month'].mean(), inplace=True)

# Duplicate Removal: Identify and eliminate duplicate records
# Example: Drop duplicate rows based on all columns
df.drop_duplicates(inplace=True)

# Standardization: Consistent formatting and units across the dataset
# Example: Convert 'price' to numeric and remove symbols
df['price'] = pd.to_numeric(df['price'].replace('[\$,]', '', regex=True))

# Outlier Detection: Identify and address outliers
# Example: Remove rows where 'price' is an outlier (considering a threshold)
price_threshold = 1000
df = df[df['price'] <= price_threshold]

# Display information about the cleaned dataset
print("\nCleaned Dataset Info:")
print(df.info())

# Save the cleaned dataset (if needed)
# df.to_csv("cleaned_AB_NYC_2019.csv", index=False)

# Visualization: Plotting a histogram for 'price'
plt.figure(figsize=(10, 6))
plt.hist(df['price'], bins=50, color='skyblue', edgecolor='black')
plt.title('Distribution of Prices')
plt.xlabel('Price')
plt.ylabel('Frequency')
plt.show()
```

```
C:\Users\ARYAN PARIKH\AppData\Roaming\Python\Python311\site-packages\pandas
\core\arrays\masked.py:60: UserWarning: Pandas requires version '1.3.6' or n
ewer of 'bottleneck' (version '1.3.5' currently installed).
  from pandas.core import (
```

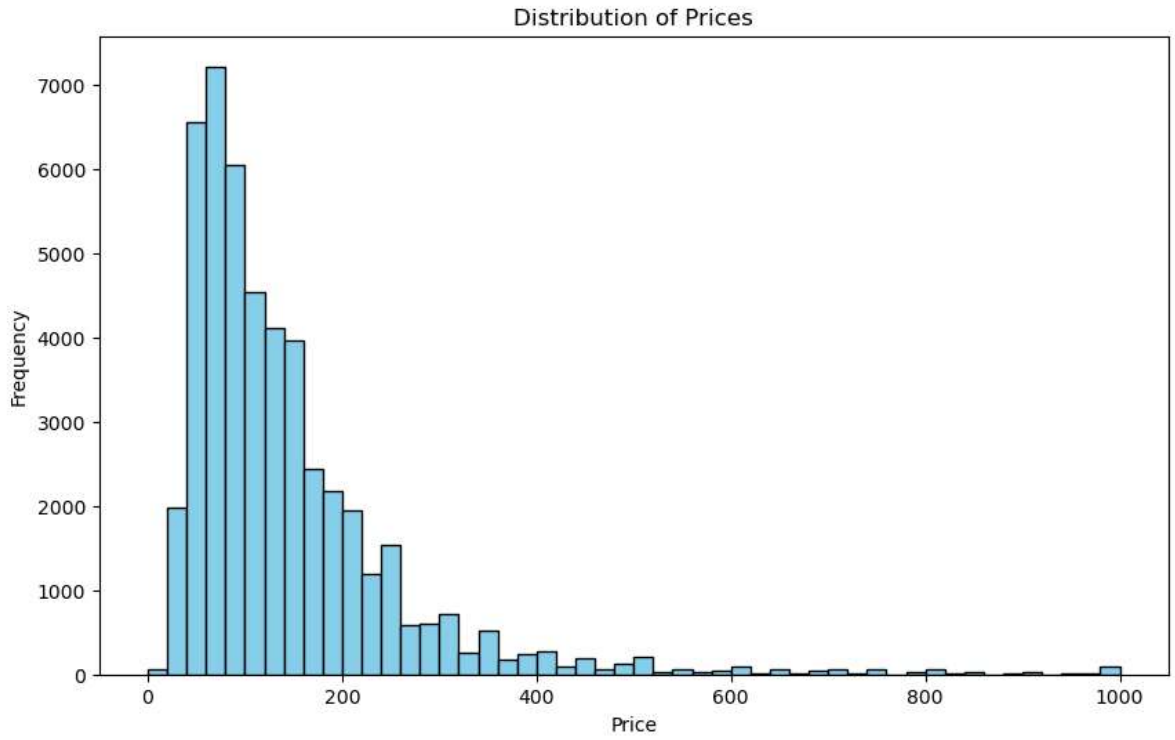
```
Original Dataset Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48895 entries, 0 to 48894
Data columns (total 16 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   id                              48895 non-null  int64  
 1   name                            48879 non-null  object 
 2   host_id                         48895 non-null  int64  
 3   host_name                       48874 non-null  object 
 4   neighbourhood_group             48895 non-null  object 
 5   neighbourhood                   48895 non-null  object 
 6   latitude                        48895 non-null  float64
 7   longitude                       48895 non-null  float64
 8   room_type                       48895 non-null  object 
 9   price                           48895 non-null  int64  
 10  minimum_nights                  48895 non-null  int64  
 11  number_of_reviews               48895 non-null  int64  
 12  last_review                     38843 non-null  object 
 13  reviews_per_month               38843 non-null  float64
 14  calculated_host_listings_count  48895 non-null  int64  
 15  availability_365                48895 non-null  int64  
dtypes: float64(3), int64(7), object(6)
memory usage: 6.0+ MB
None

Cleaned Dataset Info:
<class 'pandas.core.frame.DataFrame'>
Index: 48656 entries, 0 to 48894
Data columns (total 16 columns):
 #   Column                          Non-Null Count  Dtype         
---  ------                          --------------  -----         
 0   id                              48656 non-null  int64         
 1   name                            48640 non-null  object        
 2   host_id                         48656 non-null  int64         
 3   host_name                       48635 non-null  object        
 4   neighbourhood_group             48656 non-null  object        
 5   neighbourhood                   48656 non-null  object        
 6   latitude                        48656 non-null  float64       
 7   longitude                       48656 non-null  float64       
 8   room_type                       48656 non-null  object        
 9   price                           48656 non-null  int64         
 10  minimum_nights                  48656 non-null  int64         
 11  number_of_reviews               48656 non-null  int64         
 12  last_review                     38736 non-null  datetime64[ns]
 13  reviews_per_month               48656 non-null  float64       
 14  calculated_host_listings_count  48656 non-null  int64         
 15  availability_365                48656 non-null  int64         
dtypes: datetime64[ns](1), float64(3), int64(7), object(5)
memory usage: 6.3+ MB
None
```

```
C:\Users\ARYAN PARIKH\AppData\Local\Temp\ipykernel_1804\1829462070.py:19: Fu
tureWarning: A value is trying to be set on a copy of a DataFrame or Series
through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work
because the intermediate object on which we are setting values always behave
s as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'd
f.method({col: value}, inplace=True)' or df[col] = df[col].method(value) ins
tead, to perform the operation inplace on the original object.


  df['reviews_per_month'].fillna(df['reviews_per_month'].mean(), inplace=Tru
e)
```



In [ ]: