In [70]:
```
import numpy as np
import pandas as pd
import matplotlib.pylab as plt
%matplotlib inline
import altair as alt
```

In [71]:
```
url = 'https://www.ncei.noaa.gov/orders/cdo/4182581.csv'
```

In [72]:
```
df = pd.read_csv(url)
```

In [73]:
```
df_queens = df
```

In [74]:
```
df.head()
```

Out[74]:

	STATION	STATION_NAME	ELEVATION	LATITUDE	LONGITUDE	DAT
0	COOP:305803	JFK INTERNATIONAL AIRPORT NY US	2.7	40.63915	-73.7639	2012102 00:0
1	COOP:305803	JFK INTERNATIONAL AIRPORT NY US	2.7	40.63915	-73.7639	2012102 08:0
2	COOP:305803	JFK INTERNATIONAL AIRPORT NY US	2.7	40.63915	-73.7639	2012102 09:0
3	COOP:305803	JFK INTERNATIONAL AIRPORT NY US	2.7	40.63915	-73.7639	2012102 15:0
4	COOP:305803	JFK INTERNATIONAL AIRPORT NY US	2.7	40.63915	-73.7639	2012102 16:0

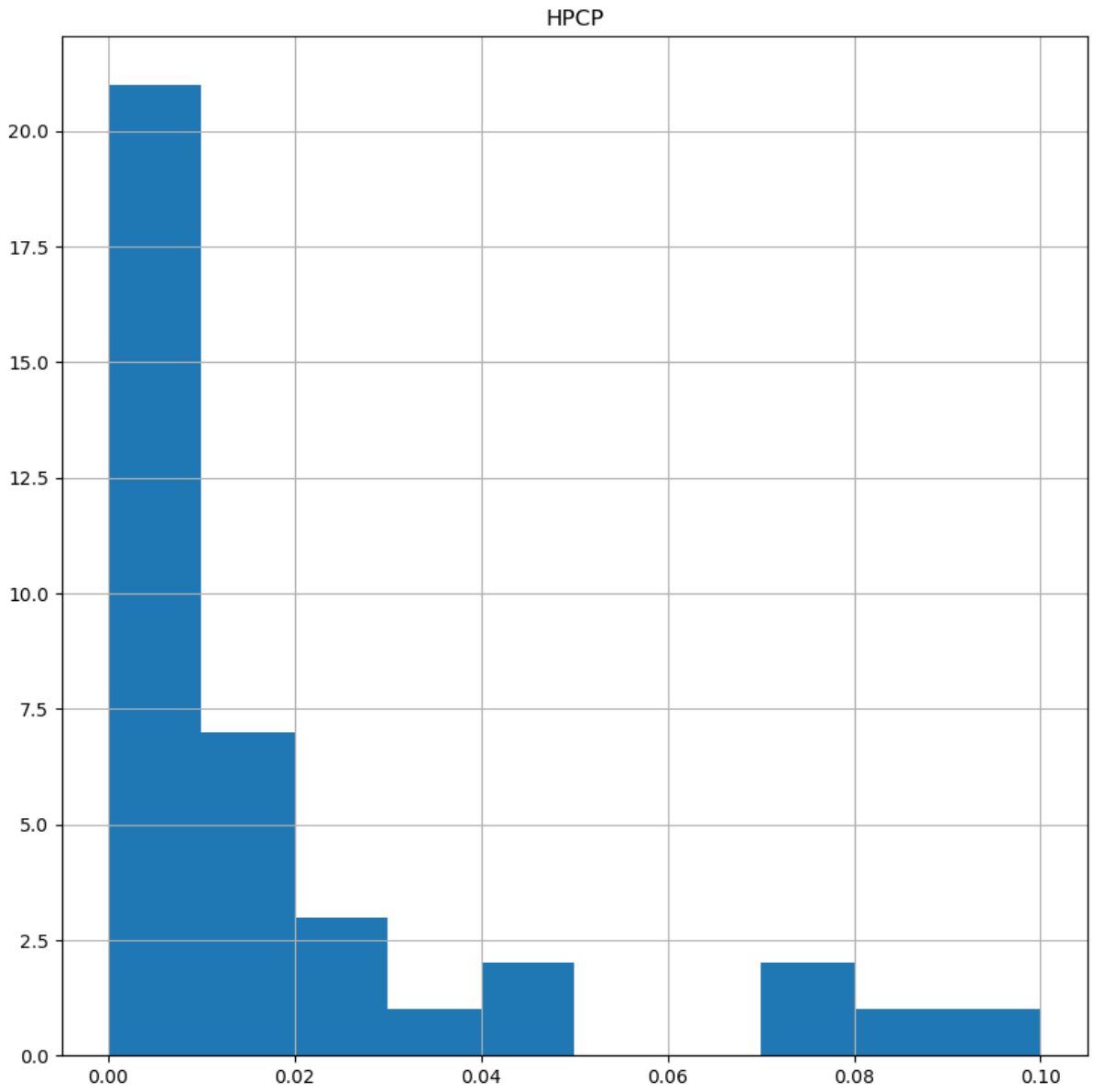
In [75]:
```
df.columns
```

Out[75]:
```
Index(['STATION', 'STATION_NAME', 'ELEVATION', 'LATITUDE', 'LONGITUD
E', 'DATE',
       'HPCP', 'Measurement Flag', 'Quality Flag'],
      dtype='object')
```

In [76]:
```
station_names = list(set(df_queens["STATION_NAME"]))
print("Station Names:", station_names)

station = list(set(df_queens["STATION"]))
print(station)
```

```
Station Names: ['JFK INTERNATIONAL AIRPORT NY US']
['COOP:305803']
```

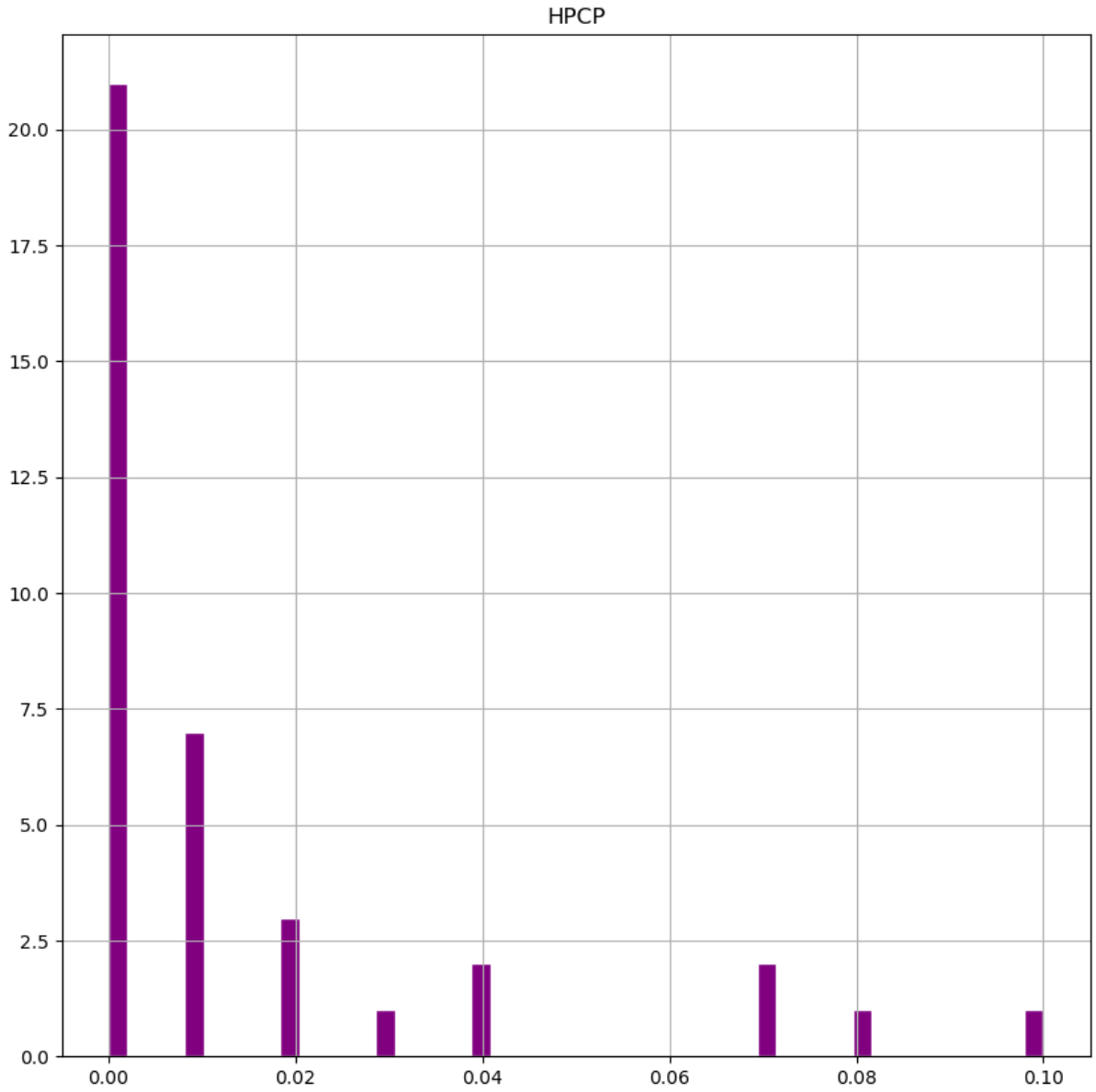
In [10]:
```
fig, ax = plt.subplots(figsize=(10,10))

df_queens.hist(column="HPCP", ax=ax)
plt.show()
```



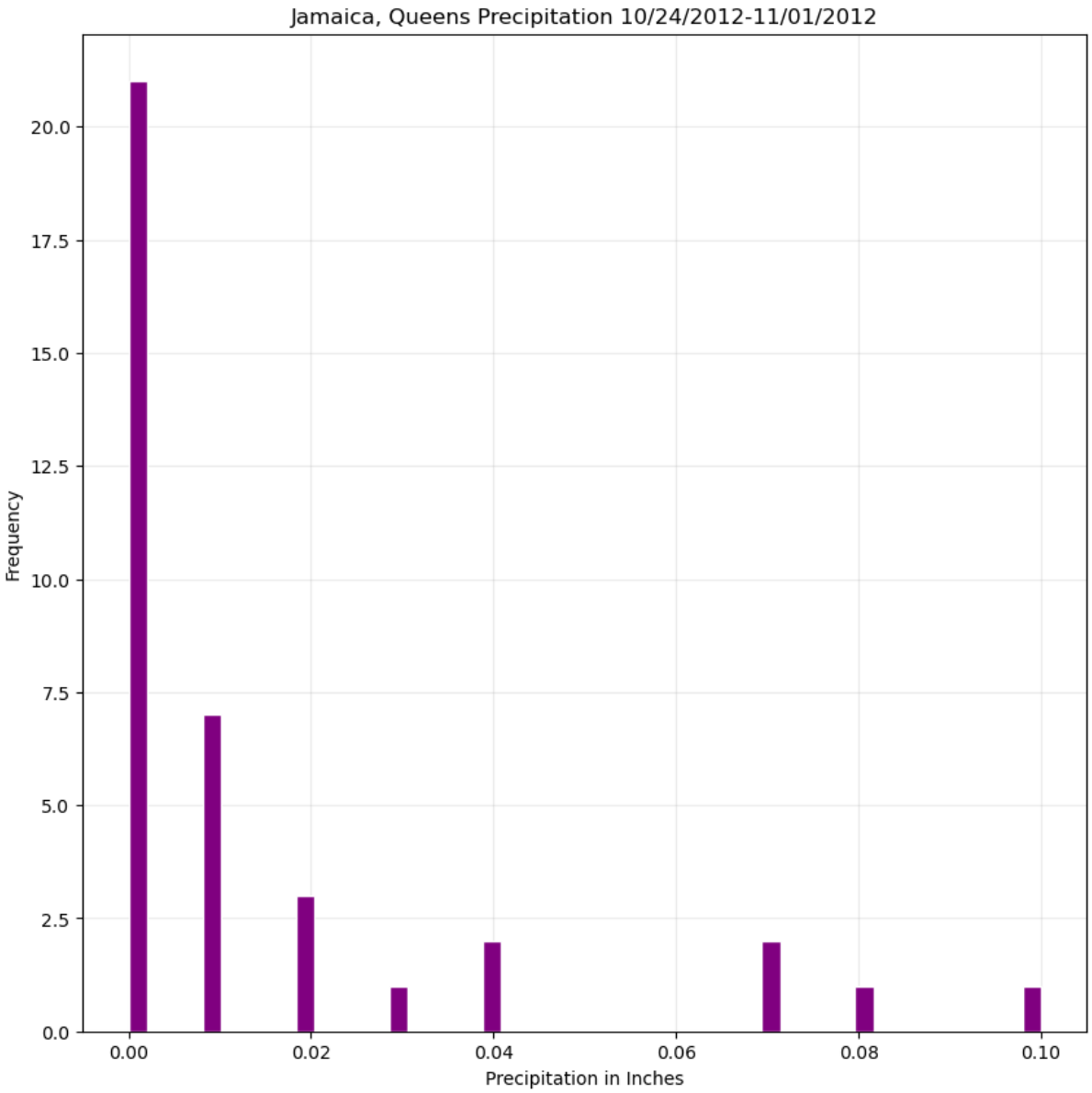
In [79]:
```
min_value = df_queens["HPCP"].min()
min_value
max_value = df_queens["HPCP"].max()
max_value


my_bins = np.linspace(min_value, max_value)
```

In [12]:
```
fig, ax = plt.subplots(figsize=(10,10))
```

```
df_queens.hist(column="HPCP", ax=ax, bins=my_bins, facecolor="purple",
plt.show()
```



In [13]:
```
fig, ax = plt.subplots(figsize=(10,10))

df_queens.hist(column="HPCP", ax=ax, bins=my_bins, facecolor="purple",

ax.set_title("Jamaica, Queens Precipitation 10/24/2012-11/01/2012")
ax.set_xlabel("Precipitation in Inches")
ax.set_ylabel("Frequency")
ax.grid(alpha = 0.25)
ax.set_axisbelow(True)


plt.show()
```



In [ ]:

In [14]:
```
df_queens['DATE'] = pd.to_datetime(df_queens['DATE'])


min_date = df_queens["DATE"].min()

max_date = df_queens["DATE"].max()

df_queens["DATE"].describe()
```

Out[14]:
```
count                            38
mean     2012-10-29 10:03:09.473684224
min                2012-10-24 00:00:00
25%                2012-10-29 03:30:00
50%                2012-10-29 13:30:00
75%                2012-10-30 12:45:00
max                2012-11-01 01:00:00
Name: DATE, dtype: object
```

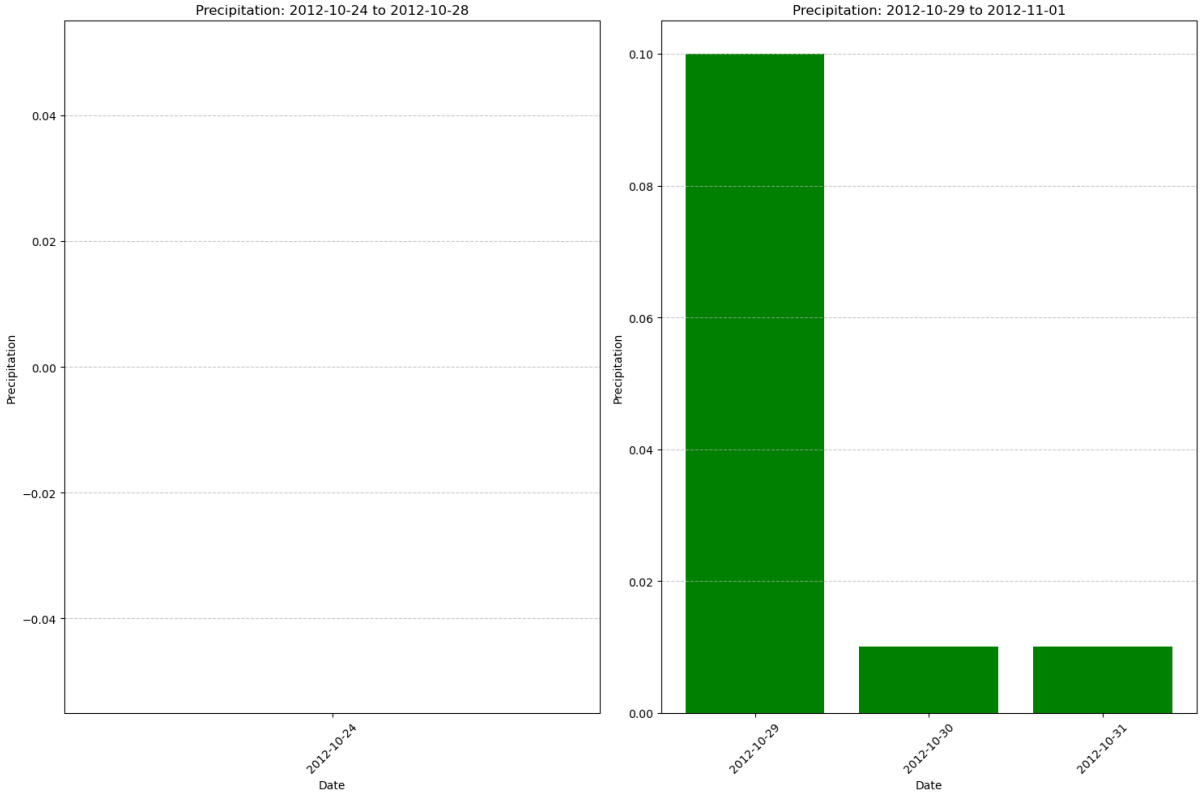
Precipitation Patterns The initial histograms display the distribution of precipitation (HPCP) values recorded at JFK International Airport, Queens, NY, between October 24 and November 1, 2012. Most precipitation values are clustered near zero, indicating that dry periods were common during this timeframe. The histograms use both default and custom binning, with purple coloring for enhanced visibility.

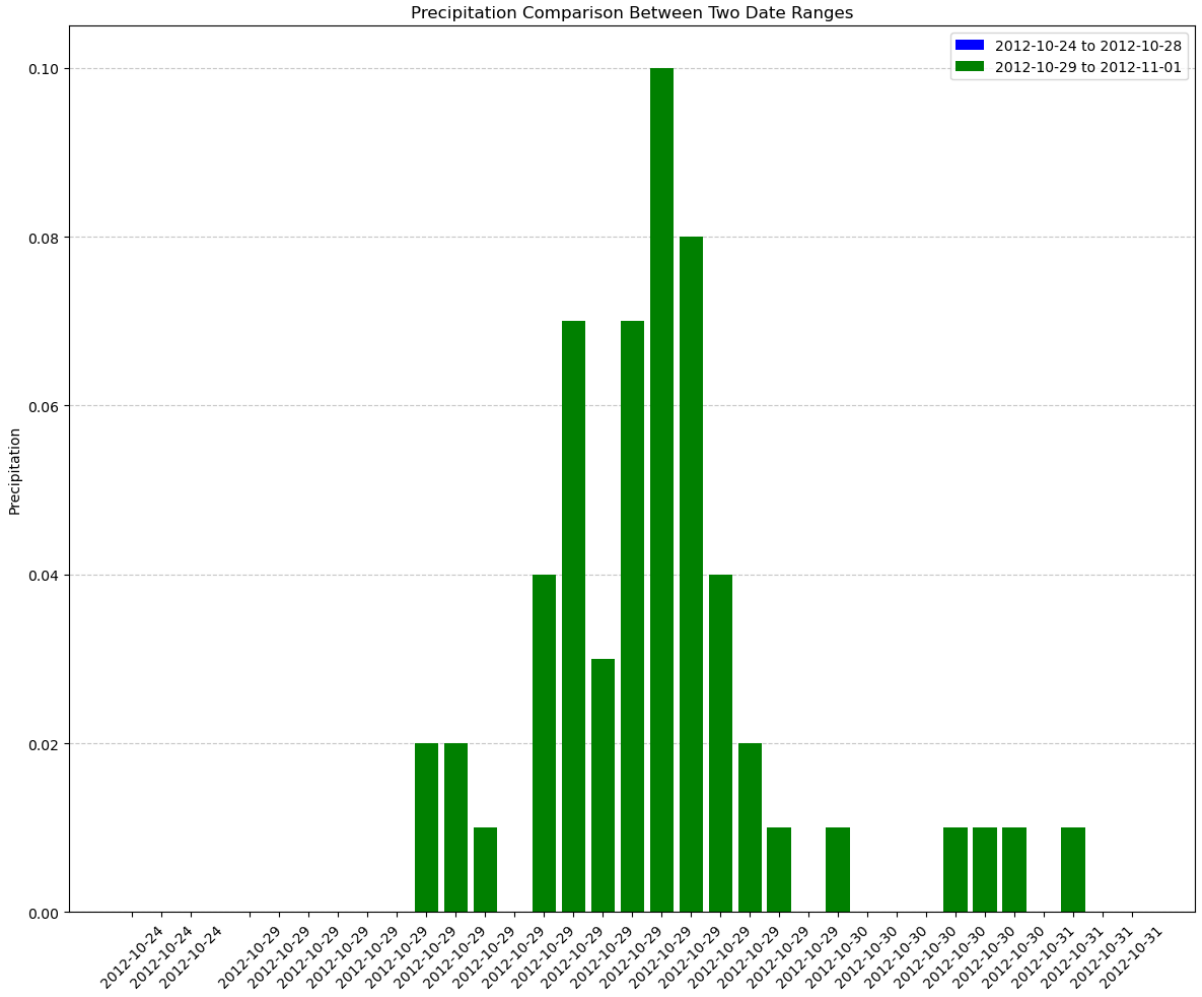
Bar plots compare precipitation across two periods: Oct 24–28, 2012: Precipitation was minimal, with most days showing little to no rainfall. Oct 29–Nov 1, 2012: There is a noticeable increase in precipitation, especially on October 29, which coincides with Hurricane Sandy's landfall. This is reflected in the higher frequency and magnitude of precipitation events during this period.

On October 29, 20 hourly measurements were recorded. The mean precipitation was 0.0255 inches, with a maximum of 0.1 inches in a single hour. Eight of the measurements were zero, indicating intermittent rainfall. The data confirms a spike in precipitation on this day, consistent with the impact of Hurricane Sandy.

In [15]:
```
start_date1 = '2012-10-24'
end_date1 = '2012-10-28'
start_date2 = '2012-10-29'
end_date2 = '2012-11-01'



range1_data = df_queens[(df_queens['DATE'] >= start_date1) & (df_queen
range2_data = df_queens[(df_queens['DATE'] >= start_date2) & (df_queen



fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(15, 10))



ax1.bar(range1_data['DATE'].dt.strftime('%Y-%m-%d'), range1_data['HPCP
ax1.set_title(f'Precipitation: {start_date1} to {end_date1}')
ax1.set_xlabel('Date')
ax1.set_ylabel('Precipitation')
ax1.tick_params(axis='x', rotation=45)
```

```
ax1.grid(axis='y', linestyle='--', alpha=0.7)



ax2.bar(range2_data['DATE'].dt.strftime('%Y-%m-%d'), range2_data['HPCP
ax2.set_title(f'Precipitation: {start_date2} to {end_date2}')
ax2.set_xlabel('Date')
ax2.set_ylabel('Precipitation')
ax2.tick_params(axis='x', rotation=45)
ax2.grid(axis='y', linestyle='--', alpha=0.7)

plt.tight_layout()
plt.show()



fig, ax = plt.subplots(figsize=(12, 10))



dates1 = range1_data['DATE'].dt.strftime('%Y-%m-%d')
dates2 = range2_data['DATE'].dt.strftime('%Y-%m-%d')


x1 = np.arange(len(dates1))
x2 = np.arange(len(dates2)) + len(dates1) + 1

ax.bar(x1, range1_data['HPCP'], color='blue', label=f'{start_date1} to
ax.bar(x2, range2_data['HPCP'], color='green', label=f'{start_date2} t


ax.set_title('Precipitation Comparison Between Two Date Ranges')
ax.set_ylabel('Precipitation')
ax.set_xticks(list(x1) + list(x2))
ax.set_xticklabels(list(dates1) + list(dates2), rotation=45)
ax.legend()
ax.grid(axis='y', linestyle='--', alpha=0.7)


ax.set_axisbelow(True)
plt.tight_layout()
plt.show()


plt.show()
```





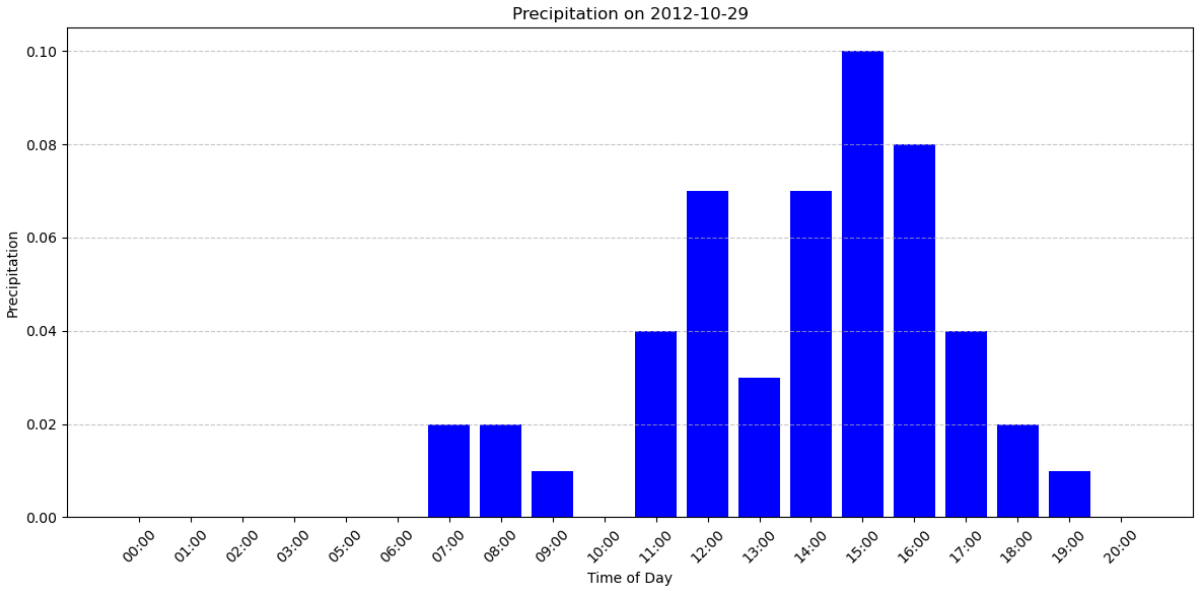
In [16]:
```
df_queens['DATE'] = pd.to_datetime(df_queens['DATE'])
```

```
target_date_str = '2012-10-29'
target_date_obj = pd.to_datetime(target_date_str).date()

filtered_day_data = df_queens[df_queens['DATE'].dt.date == target_date


print(f"Number of rows for {target_date_str}: {len(filtered_day_data)}


if len(filtered_day_data) > 0:
    print("\nSample data:")
    print(filtered_day_data[['DATE', 'HPCP']].describe())


    if 'HPCP' in filtered_day_data.columns:
        print("\nHPCP column exists")
        print(f"HPCP values for this date: {filtered_day_data['HPCP'].
        print(f"Data types in HPCP: {filtered_day_data['HPCP'].dtype}"

        # Check for NaN or zero values
        print(f"NaN values: {filtered_day_data['HPCP'].isna().sum()}")
        print(f"Zero values: {(filtered_day_data['HPCP'] == 0).sum()}"
    else:
        print("\nHPCP column doesn't exist. Available columns are:")
        print(filtered_day_data.columns.tolist())
else:
    print(f"No data found for date {target_date_str}")
```

```
Number of rows for 2012-10-29: 20

Sample data:
                      DATE       HPCP
count                   20  20.000000
mean   2012-10-29 10:18:00   0.025500
min    2012-10-29 00:00:00   0.000000
25%    2012-10-29 05:45:00   0.000000
50%    2012-10-29 10:30:00   0.015000
75%    2012-10-29 15:15:00   0.040000
max    2012-10-29 20:00:00   0.100000
std                    NaN   0.031368

HPCP column exists
HPCP values for this date: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.02, 0.02,
0.01, 0.0, 0.04, 0.07, 0.03, 0.07, 0.1, 0.08, 0.04, 0.02, 0.01, 0.0]
Data types in HPCP: float64
NaN values: 0
Zero values: 8
```

In [17]:
```
df_queens['DATE'] = pd.to_datetime(df_queens['DATE'])

target_date_str = '2012-10-29'
target_date_obj = pd.to_datetime(target_date_str).date()
```

```
day_data = df_queens[df_queens['DATE'].dt.date == target_date_obj]


if not day_data.empty:
    if len(day_data) == 1:
        plt.figure(figsize=(10, 6))
        plt.bar(target_date_str, day_data['HPCP'].values[0], color='bl
        plt.title(f'Precipitation on {target_date_str}')
        plt.ylabel('Precipitation')
        plt.grid(axis='y', linestyle='--', alpha=0.7)

        plt.text(target_date_str, day_data['HPCP'].values[0],
                 f"{day_data['HPCP'].values[0]:.2f}",
                 ha='center', va='bottom')

        plt.tight_layout()
        plt.show()


    else:
        plt.figure(figsize=(12, 6))
        if 'TIME' in day_data.columns:
            x_values = day_data['TIME']
            x_label = 'Time'
        else:
            x_values = day_data['DATE'].dt.strftime('%H:%M')
            x_label = 'Time of Day'

        plt.bar(x_values, day_data['HPCP'], color='blue')
        plt.title(f'Precipitation on {target_date_str}')
        plt.xlabel(x_label)
        plt.ylabel('Precipitation')

        plt.xticks(rotation=45)
        plt.grid(axis='y', linestyle='--', alpha=0.7)
        plt.tight_layout()
        plt.show()
else:
    print(f"No data found for the date: {target_date_str}")

plt.figtext(0.5, 0.01,
            "Note: October 29, 2012 was when Hurricane Sandy made land
            ha='center', fontsize=10, style='italic')
```



Out[17]:
```
Text(0.5, 0.01, 'Note: October 29, 2012 was when Hurricane Sandy made
landfall near Atlantic City, NJ')
```

In [ ]:
```
import seaborn as sns
```

In [19]:
```
url2 = 'https://www.ncei.noaa.gov/orders/cdo/4182627.csv'
```

In [20]:
```
df2 = pd.read_csv(url2)
```

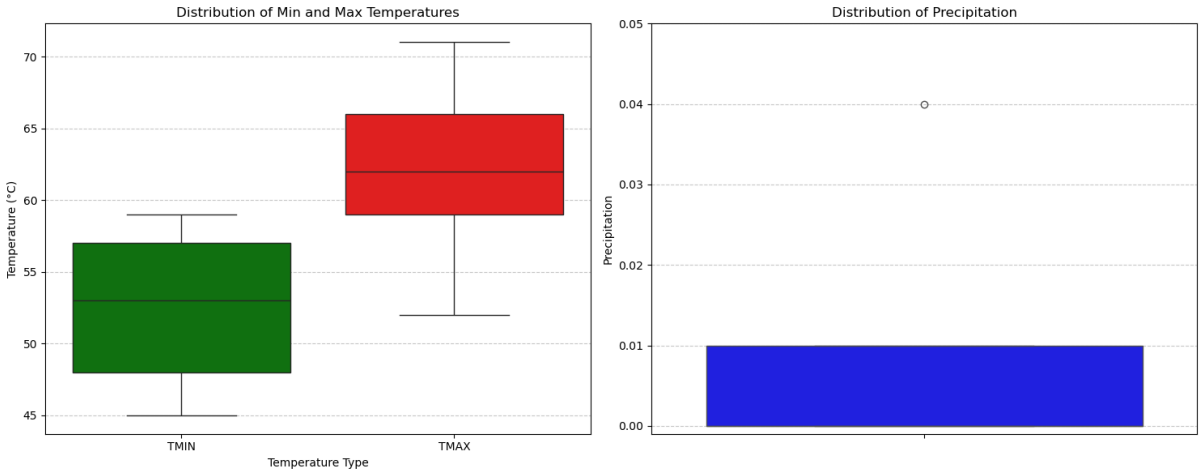
In [21]:
```
df_temp = df2
```

In [ ]:
```

```

In [22]:
```
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(15, 6))

temp_data = pd.melt(df_temp,
                    value_vars=['TMIN', 'TMAX'],
                    var_name='Temperature Type',
                    value_name='Temperature (°C)')


temp_colors = {'TMIN': 'green', 'TMAX': 'red'}
sns.boxplot(x='Temperature Type', y='Temperature (°C)', data=temp_data
ax1.set_title('Distribution of Min and Max Temperatures')
ax1.grid(axis='y', linestyle='--', alpha=0.7)


sns.boxplot(y=df_temp['PRCP'], ax=ax2, color = 'blue')
ax2.set_title('Distribution of Precipitation')
ax2.set_ylabel('Precipitation')
ax2.set_xlabel('')
ax2.grid(axis='y', linestyle='--', alpha=0.7)
ax2.set_ylim(-0.001, 0.05)
```

```
plt.tight_layout()
plt.show()


print("Temperature Statistics:")
print(df_temp[['TMIN', 'TMAX']].describe())
print("\nPrecipitation Statistics:")
print(df_temp['PRCP'].describe())
```

```
<Figure size 640x480 with 0 Axes>
```




```
Temperature Statistics:
             TMIN       TMAX
count    9.000000   9.000000
mean    52.333333  61.222222
std      5.049752   6.418290
min     45.000000  52.000000
25%     48.000000  59.000000
50%     53.000000  62.000000
75%     57.000000  66.000000
max     59.000000  71.000000

Precipitation Statistics:
count    9.000000
mean     0.062222
std      0.168432
min      0.000000
25%      0.000000
50%      0.000000
75%      0.010000
max      0.510000
Name: PRCP, dtype: float64
```

Temperature and Precipitation Distribution

The boxplots for minimum (TMIN) and maximum (TMAX) temperatures, as well as precipitation (PRCP) for a subset of days: Temperature: TMIN ranged from 45°F to 59°F, with a mean of 52.3°F. TMAX ranged from 52°F to 71°F, with a mean of

61.2°F. The temperature distributions are relatively tight, suggesting stable weather aside from the storm event. Precipitation: Most precipitation values are near zero, with a maximum of 0.51 inches. The boxplot confirms that significant rainfall was rare, except for the storm event.

The analysis highlights October 29, 2012, as a day of exceptional weather activity due to Hurricane Sandy. The precipitation spike and accompanying note in the document confirm the storm's local impact.

Most days in the period were dry, with only brief, isolated rainfall events. Temperatures were moderate, with little variation, except for the storm-related anomaly.

The data reveals that late October 2012 in Queens was marked by generally dry and mild weather, punctuated by a dramatic increase in precipitation on October 29 due to Hurricane Sandy. Temperature remained stable, and significant rainfall was otherwise rare. These visualizations provide clear evidence of the storm's impact and the overall weather stability in the region during this period.

In [78]:

In [ ]: