In [1]:
```
import pandas as pd
original_df = pd.read_csv("GEO557Tropical_Storm_Dataset.csv")
#
original_df.head(10)
```

Out[1]:

	Year	Name	BASIN	ISO_TIME_____	NATURE	LAT	LON	WMO WIND	WMO PRES	USA WIND	U PR
0	2023	IDALIA	NaN	2023-08-26 12:00:00	TS	20.8	-86.1	25.0	1006.0	25	1(
1	2023	IDALIA	NaN	15:00:00	TS	21.1	-86.1	NaN	NaN	25	1(
2	2023	IDALIA	NaN	18:00:00	TS	21.3	-86.2	25.0	1006.0	25	1(
3	2023	IDALIA	NaN	21:00:00	TS	21.3	-86.3	NaN	NaN	28	1(
4	2023	IDALIA	NaN	2023-08-27 0:00:00	TS	21.1	-86.4	30.0	1004.0	30	1(
5	2023	IDALIA	NaN	3:00:00	TS	20.8	-86.7	NaN	NaN	30	1(
6	2023	IDALIA	NaN	6:00:00	TS	20.5	-86.8	30.0	1002.0	30	1(
7	2023	IDALIA	NaN	9:00:00	TS	20.2	-86.6	NaN	NaN	33	1(
8	2023	IDALIA	NaN	12:00:00	TS	19.9	-86.3	35.0	999.0	35	9
9	2023	IDALIA	NaN	15:00:00	TS	19.9	-86.0	NaN	NaN	38	9

In [2]:
```
def populate_full_dates(df):
    #iterate through DF and fix dates
    # Initialize variable to hold the last full date encountered
    current_date = None

    # Iterate through the ISO_TIME column and update times based on the last full d
    for i, iso_time in enumerate(df['ISO_TIME']):
        if len(iso_time) > 8:  # Full datetime (YYYY-MM-DD HH:MM:SS)
            # Set current_date to the full date part of the timestamp
            current_date = iso_time[:10]  # Extract the date portion (YYYY-MM-DD)
        else:
            # If only time is present, add the current_date to create a full timest
            df.at[i, 'ISO_TIME'] = f"{current_date} {iso_time}"

    # Convert ISO_TIME column to datetime for consistency
    df['ISO_TIME'] = pd.to_datetime(df['ISO_TIME'])
    return df

# originally the data was gatherd from 23 different websites from NOAA historical h
# the name and year columns I added as I gathered the data.

# some issues with the dataset involve ISO time being seuqential, so the first one
# ISO_TIME____ column, the name and the data both need help.
# YYYY-MM-DD but every other measurement in that section doesn't have that until it
# we're going to do the following 3 things,
```

```
# 1. rename the iso_time column
# 2. add dates to match the TIME
df = pd.read_csv("GEO557Tropical_Storm_Dataset.csv")
# Step 1: Rename the ISO_TIME_________ column to ISO_TIME
df.rename(columns={'ISO_TIME_________': 'ISO_TIME'}, inplace=True)
df.head(5)

# Step 2: we have to iterate through the data set, and if ISO_TIME has a full date
df = populate_full_dates(df)
df.head(10)
```

Out[2]:

	Year	Name	BASIN	ISO_TIME	NATURE	LAT	LON	WMO WIND	WMO PRES	USA WIND	USA PRES
0	2023	IDALIA	NaN	2023-08-26 12:00:00	TS	20.8	-86.1	25.0	1006.0	25	1006
1	2023	IDALIA	NaN	2023-08-26 15:00:00	TS	21.1	-86.1	NaN	NaN	25	1006
2	2023	IDALIA	NaN	2023-08-26 18:00:00	TS	21.3	-86.2	25.0	1006.0	25	1006
3	2023	IDALIA	NaN	2023-08-26 21:00:00	TS	21.3	-86.3	NaN	NaN	28	1005
4	2023	IDALIA	NaN	2023-08-27 00:00:00	TS	21.1	-86.4	30.0	1004.0	30	1004
5	2023	IDALIA	NaN	2023-08-27 03:00:00	TS	20.8	-86.7	NaN	NaN	30	1003
6	2023	IDALIA	NaN	2023-08-27 06:00:00	TS	20.5	-86.8	30.0	1002.0	30	1002
7	2023	IDALIA	NaN	2023-08-27 09:00:00	TS	20.2	-86.6	NaN	NaN	33	1001
8	2023	IDALIA	NaN	2023-08-27 12:00:00	TS	19.9	-86.3	35.0	999.0	35	999
9	2023	IDALIA	NaN	2023-08-27 15:00:00	TS	19.9	-86.0	NaN	NaN	38	998

In [ ]:

In [ ]: