
Clare Robbins, Jordan Holley Riggs, Matt Riley, Sari Broudy, Savannah Posner

# Why Tennessee Tornadoes

The members of this team live or have lived in the state of Tennessee. We have noticed a change in the frequency of tornadoes in the state during our collective time living in the state.  The team was  curious to see what data supports our concerning hypotheses.

The data was obtained from data.world and represents tornado tracks from the United States, Puerto Rico, and the US Virgin Islands. For this project we filtered the data for tornadoes in the state of Tennessee. Follow the link below to access our data

https://github.com/clarerobb/Tennessee_Tornadoes/blob/main/Resources/cleaned_tn_tornadoes.csv

# What will the data show?

Our Research questions for the data to answer are:

1. Have tornadoes increased in intensity in the last 50 years in the state of Tennessee?
2. What counties are most likely to have more tornadoes?
3. Has the frequency of tornadoes in Tennessee increased since 1950?

# Data Exploration

After deciding which data to use, we narrowed our data to just the state of Tennessee. Then used Python and Pandas to search for missing values, removing columns that were providing information that wasn't needed for our research questions, and to adjust values that switched in 1996 with reporting protocols to match the previous years.

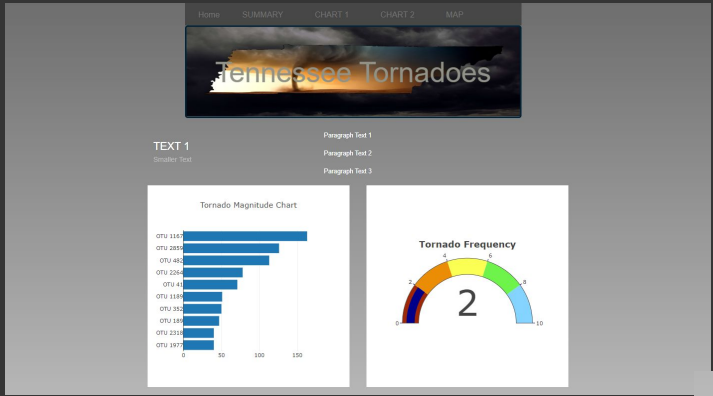
# Analysis

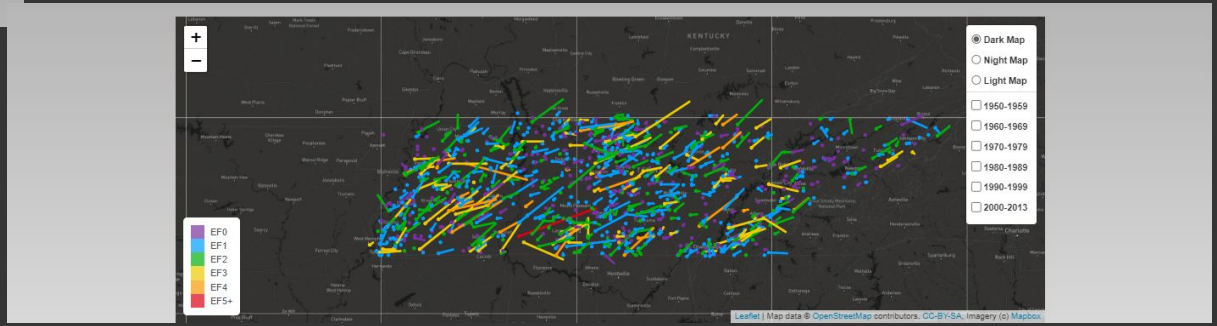
Machine learning models will be created to predict the following:

1. Magnitude of tornadoes
2. Location of tornadoes
3. Amount of property damage

# Dashboard

**The Dashboard has been created using Javascript to be displayed as an interactive webpage. CSS, D3, and Bootstrap components have been used to enhance the displays. The map was created using Leaflet**



# Database

Overview of tables and screenshot of connection stream

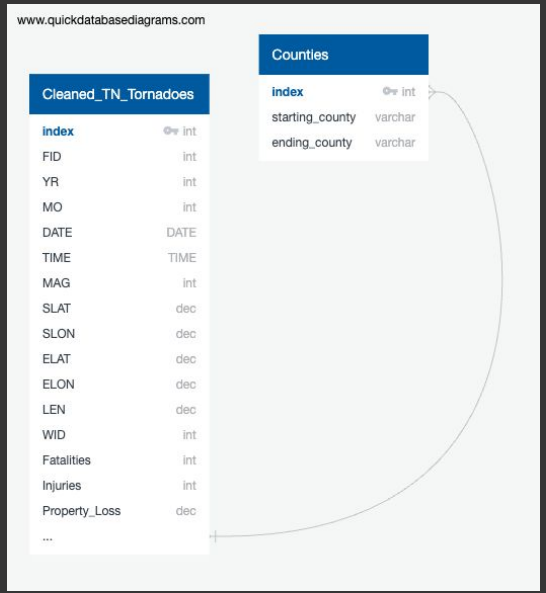
```
from getpass import getpass
password = getpass ('Enter database password')

try:
    conn = psycopg2.connect(
        host="localhost",
        database = "Tennessee_Tornadoes",
        user="postgres",
        password=password)
    print ('pyscopg2 connection:', conn)

except Exception as err:
    print ('psycopg2 connect() ERROR:', err)
    connect = None
```

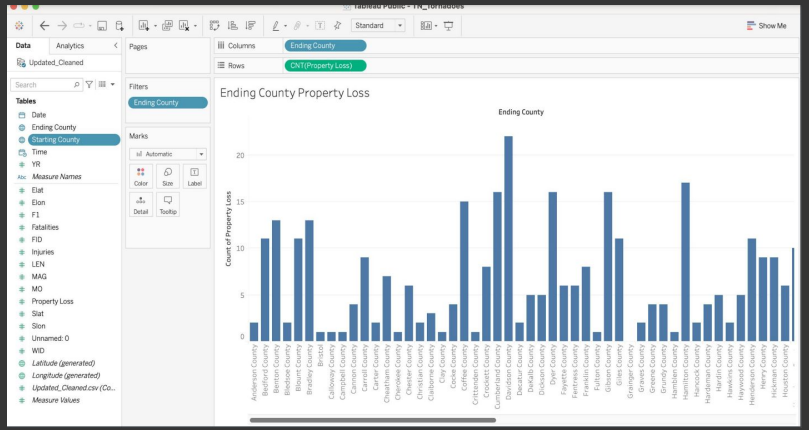
```
cr = conn.cursor()
cr.execute('SELECT * FROM cleaned_tn_tornadoes;')
tmp = cr.fetchall()
```

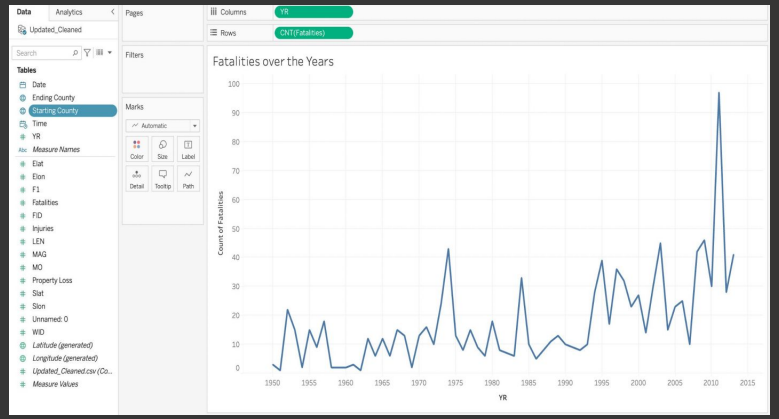
```
#extract the  column names
col_names = []
for db in cr.description:
    col_names.append(db[0])
```

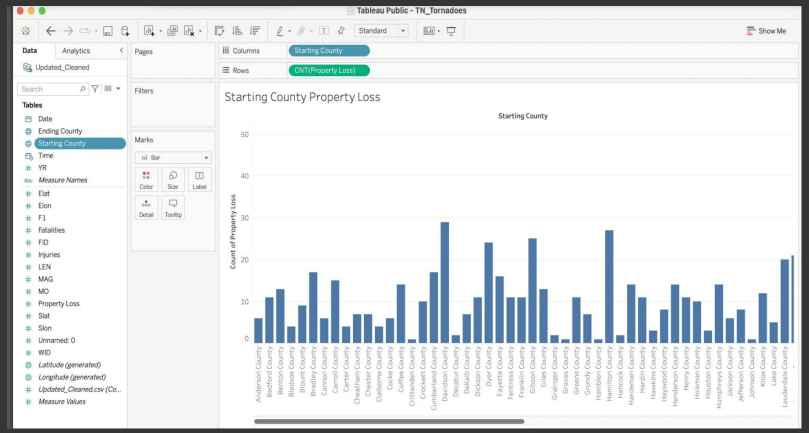


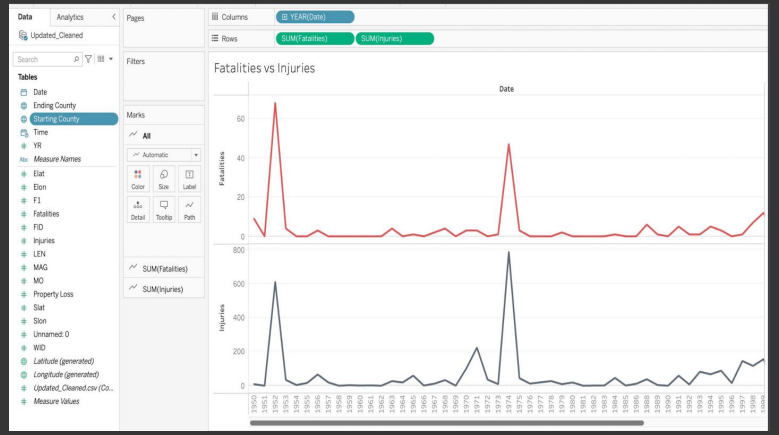
# Database

Tables were created in Tableau



# Machine Learning

```python
# prepare the dataframe

df_3 = df.drop(['Unnamed: 0','Unnamed: 0.1','TIME',"WID","starting county",
                'SLAT','SLON','ELAT','ELON','LEN','ending county','FID'], axis=1)
df.columns


target = 'Property Loss'
X = pd.get_dummies(df_3.drop([target],axis = 1))


# Create our target
y = df[target]


X_train, X_test, y_train, y_test = train_test_split(X,
                                                    y,
                                                    random_state=1)

from sklearn.pipeline import make_pipeline
from sklearn.linear_model import SGDClassifier
from sklearn.preprocessing import StandardScaler

model = make_pipeline(StandardScaler(),LogisticRegression(solver='newton-cg'))
model.fit(X_train, y_train)

# model = LogisticRegression(solver='newton-cg', random_state=1)

#'liblinear', 'sag', 'saga
```

```python
#'liblinear', 'sag', 'saga
model.fit(X_train, y_test)
# Display the confusion matrix

y_pred = model.predict(X_test)
print(f"The accuracy score: \n{balanced_accuracy_score(y_test, y_pred)}\n")
# Display the confusion matrix
print(f"Confusion Matrix:\n{confusion_matrix(y_test, y_pred)}\n")
# Print the imbalanced classification report
print(f"Classification Report:\n{classification_report_imbalanced(y_test, y_pred)}")
```