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 if data supports our concerning hypotheses. This information can be used for homebuyers and insurance companies.

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. Click below for data.

Follow to data

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 using Excel. Then using 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. Each tornadoes starting and ending counties were calculated in GetCounties.ipynb with the geopy library and exported to counties.csv

Analysis

Machine learning models will be created to predict the following:

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

Tools and Technology

Data Cleaning:
- Python 3.7.13 (pandas and geopy libraries)
- Jupyter Notebook 6.4.8

Database:
- PostgreSQL 11.16
- pgAdmin 4 v6.8

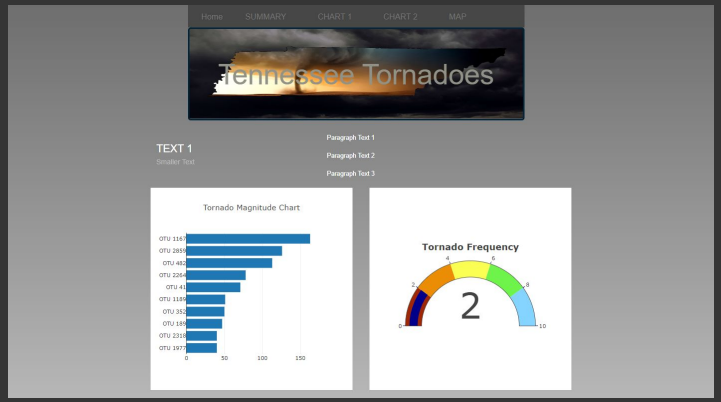
Connecting Database:
- Psycopg2

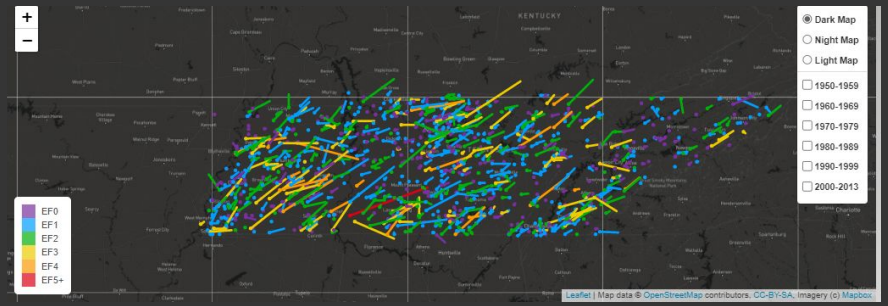
Machine Learning:
- Python (pandas, imbalance-learn, scikit-Learn, numpy libraries)
- Jupyter Notebook

Dashboard:
- Tableau
- Javascript
- Bootstrap
- Leaflet
- D3
- HTML
- CSS

Dashboard

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





Connecting the Database

Overview of ERD and screenshot of connection stream

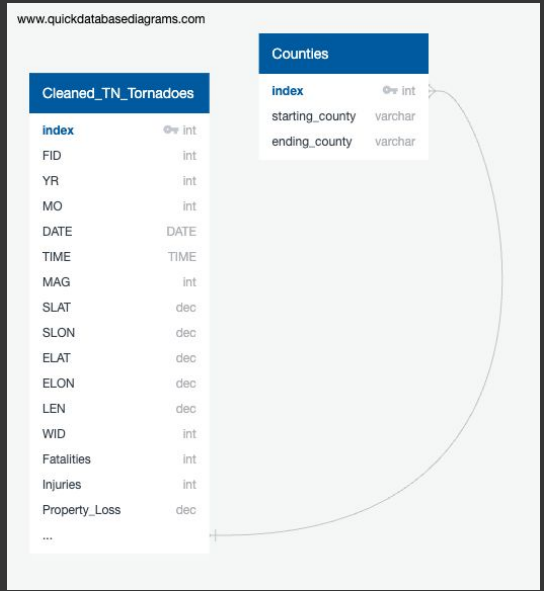
```
try:
    # declare a new PostgreSQL connection object
    conn = connect(
        dbname = "",
        user = "postgres",
        host = "tennesseetornadoes.cdilutmdgtwo.us-east-1.rds.amazonaws.com",
        port = "5432",
        password = password
    )

    # print the connection if successful
    print ("psycopg2 connection:", conn)

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

```
psycopg2 connection:
```

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

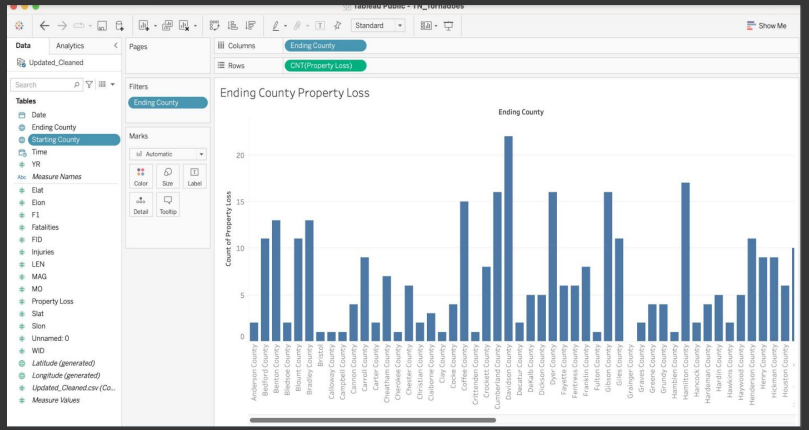
```
# Extract the column names
col_names = []
for elt in cr.description:
    col_names.append(elt[0])
```

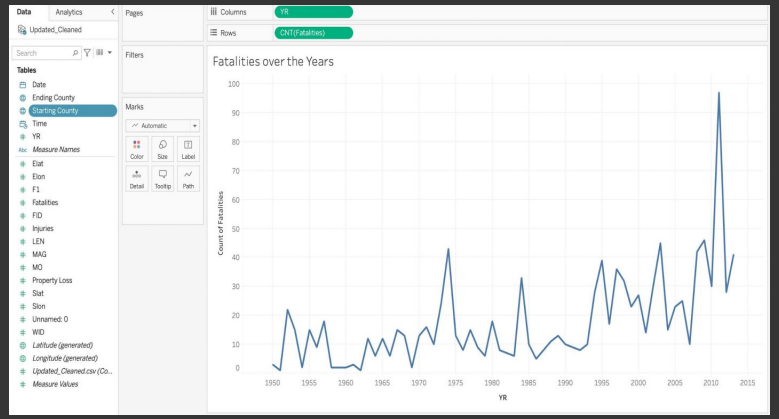
```
# Create the dataframe, passing in the list of col_names extracted from the description
df = pd.DataFrame(tmp, columns=col_names)

df.head()
```

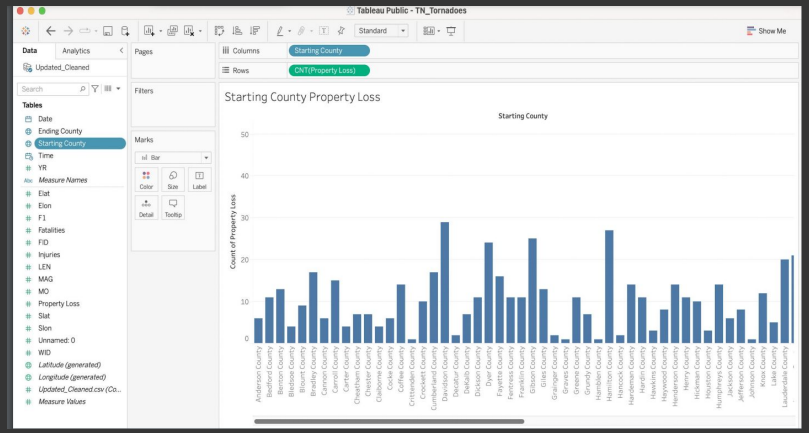


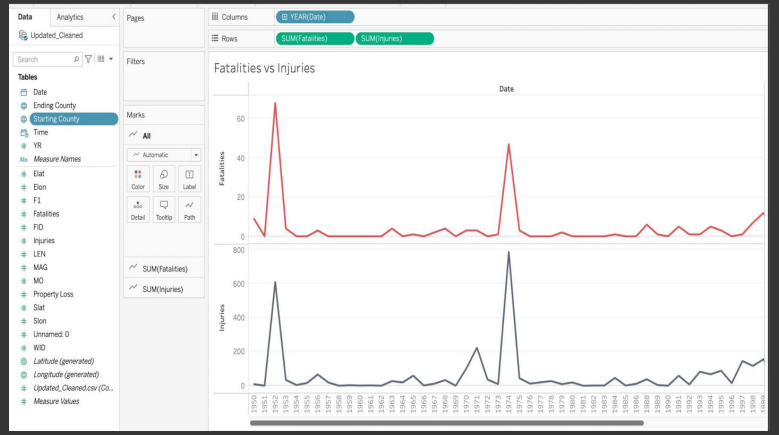
Database

Tables were created in Tableau









Machine Learning

```
# 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
```