Analyzing wildfire activities in Australia

October 19, 2024

1 Practice Assignement Part I

[3]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import folium
%matplotlib inline
```

1.1 Dataset

Historical Wildfires This wildfire dataset contains data on fire activities in Australia starting from 2005. Variables: * Region: the 7 regions * Date: in UTC and provide the data for 24 hours ahead * Estimated_fire_area: daily sum of estimated fire area for presumed vegetation fires with a confidence > 75% for a each region in km2 * Mean_estimated_fire_brightness: daily mean (by flagged fire pixels(=count)) of estimated fire brightness for presumed vegetation fires with a confidence level > 75% in Kelvin * Mean_estimated_fire_radiative_power: daily mean of estimated radiative power for presumed vegetation fires with a confidence level > 75% for a given region in megawatts * Mean_confidence: daily mean of confidence for presumed vegetation fires with a confidence level > 75% * Std_confidence: standard deviation of estimated fire radiative power in megawatts * Var_confidence: Variance of estimated fire radiative power in megawatts * Count: daily numbers of pixels for presumed vegetation fires with a confidence level of larger than 75% for a given region * Replaced: Indicates with an Y whether the data has been replaced with standard quality data when they are available (usually with a 2-3 month lag). Replaced data has a slightly higher quality in terms of locations

Importing Data

[5]:
```
URL = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/
↪IBMDeveloperSkillsNetwork-DV0101EN-SkillsNetwork/Data%20Files/
↪Historical_Wildfires.csv"
df = pd.read_csv(URL)
print('Data read into a pandas dataframe!')
```

```
Data read into a pandas dataframe!
```

[6]:
```
df.head()
```

```
[6]:   Region      Date  Estimated_fire_area  Mean_estimated_fire_brightness  \
     0    NSW  1/4/2005              8.68000                      312.266667   
     1    NSW  1/5/2005             16.61125                      322.475000   
     2    NSW  1/6/2005              5.52000                      325.266667   
     3    NSW  1/7/2005              6.26400                      313.870000   
     4    NSW  1/8/2005              5.40000                      337.383333   

        Mean_estimated_fire_radiative_power  Mean_confidence  Std_confidence  \
     0                            42.400000        78.666667        2.886751   
     1                            62.362500        85.500000        8.088793   
     2                            38.400000        78.333333        3.214550   
     3                            33.800000        92.200000        7.529940   
     4                           122.533333        91.000000        7.937254   

        Var_confidence  Count Replaced  
     0        8.333333      3        R  
     1       65.428571      8        R  
     2       10.333333      3        R  
     3       56.700000      5        R  
     4       63.000000      3        R  
```

```
[7]: df.columns
```

```
[7]: Index(['Region', 'Date', 'Estimated_fire_area',
            'Mean_estimated_fire_brightness', 'Mean_estimated_fire_radiative_power',
            'Mean_confidence', 'Std_confidence', 'Var_confidence', 'Count',
            'Replaced'],
           dtype='object')
```

```
[8]: df.dtypes
```

```
[8]: Region                                  object
     Date                                    object
     Estimated_fire_area                    float64
     Mean_estimated_fire_brightness         float64
     Mean_estimated_fire_radiative_power    float64
     Mean_confidence                        float64
     Std_confidence                         float64
     Var_confidence                         float64
     Count                                    int64
     Replaced                                object
     dtype: object
```

```
[9]: import datetime as dt

     df['Year'] = pd.to_datetime(df['Date']).dt.year
     df['Month'] = pd.to_datetime(df['Date']).dt.month
```

[10]:
```
df.columns
```

[10]:
```
Index(['Region', 'Date', 'Estimated_fire_area',
       'Mean_estimated_fire_brightness', 'Mean_estimated_fire_radiative_power',
       'Mean_confidence', 'Std_confidence', 'Var_confidence', 'Count',
       'Replaced', 'Year', 'Month'],
      dtype='object')
```

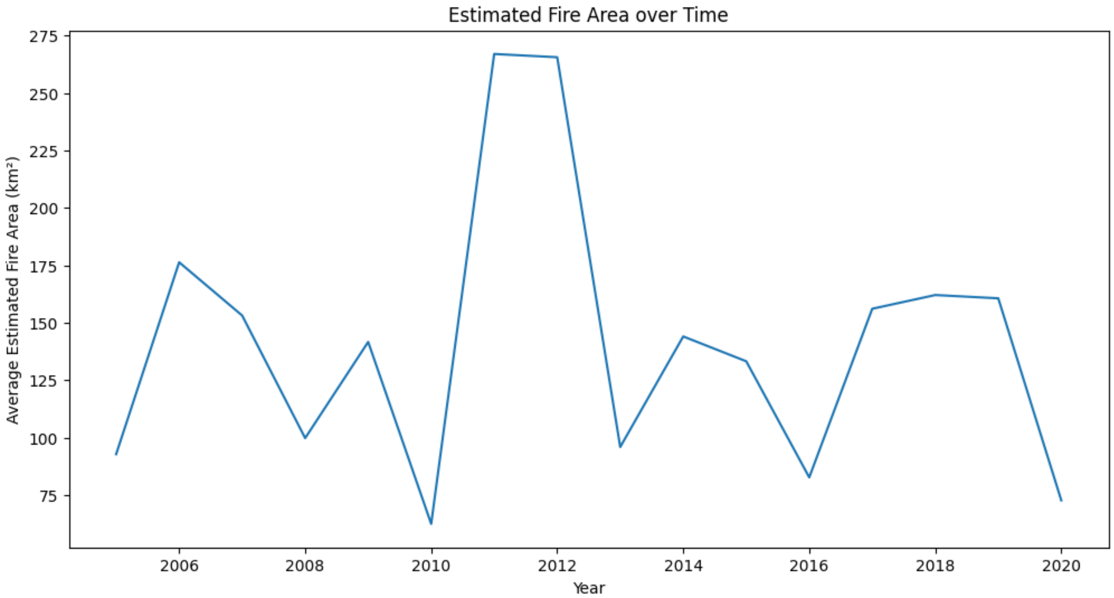
[11]:
```
df.dtypes
```

[11]:
```
Region                                 object
Date                                   object
Estimated_fire_area                   float64
Mean_estimated_fire_brightness        float64
Mean_estimated_fire_radiative_power   float64
Mean_confidence                       float64
Std_confidence                        float64
Var_confidence                        float64
Count                                   int64
Replaced                               object
Year                                    int32
Month                                   int32
dtype: object
```

2 Practice Tasks

2.1 TASK 1.1: Let's try to understand the change in average estimated fire area over time (use pandas to plot)

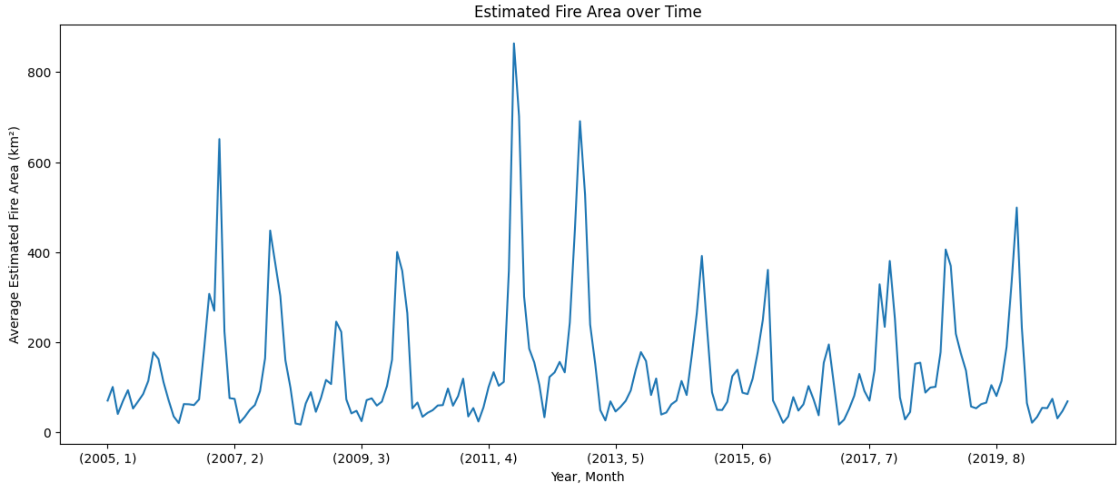
[12]:
```
plt.figure (figsize=(12, 6))
#grouping the data by "Year" and calculating the mean of 'Estimated_fire_area'
df_new= df.groupby("Year")['Estimated_fire_area'].mean()
#plotting the data
df_new.plot(x=df_new.index, y=df_new.values)
plt.xlabel('Year')
plt.ylabel('Average Estimated Fire Area (km²)')
plt.title('Estimated Fire Area over Time')
plt.show()
```



2.2 TASK 1.2: You can notice the peak in the plot between 2010 to 2013. Let's narrow down our finding, by plotting the estimated fire area for year grouped together with month.

[16]:

```
plt.figure (figsize=(15, 6))
# Grouping the data by both 'Year' and 'Month', and calculating the mean of␣
↪'Estimated_fire_area'
df_new = df.groupby(['Year','Month'])['Estimated_fire_area'].mean()
# Plotting the data
df_new.plot(x=df_new.index, y=df_new.values)
plt.xlabel('Year, Month')
plt.ylabel('Average Estimated Fire Area (km²)')
plt.title('Estimated Fire Area over Time')
plt.show()
```

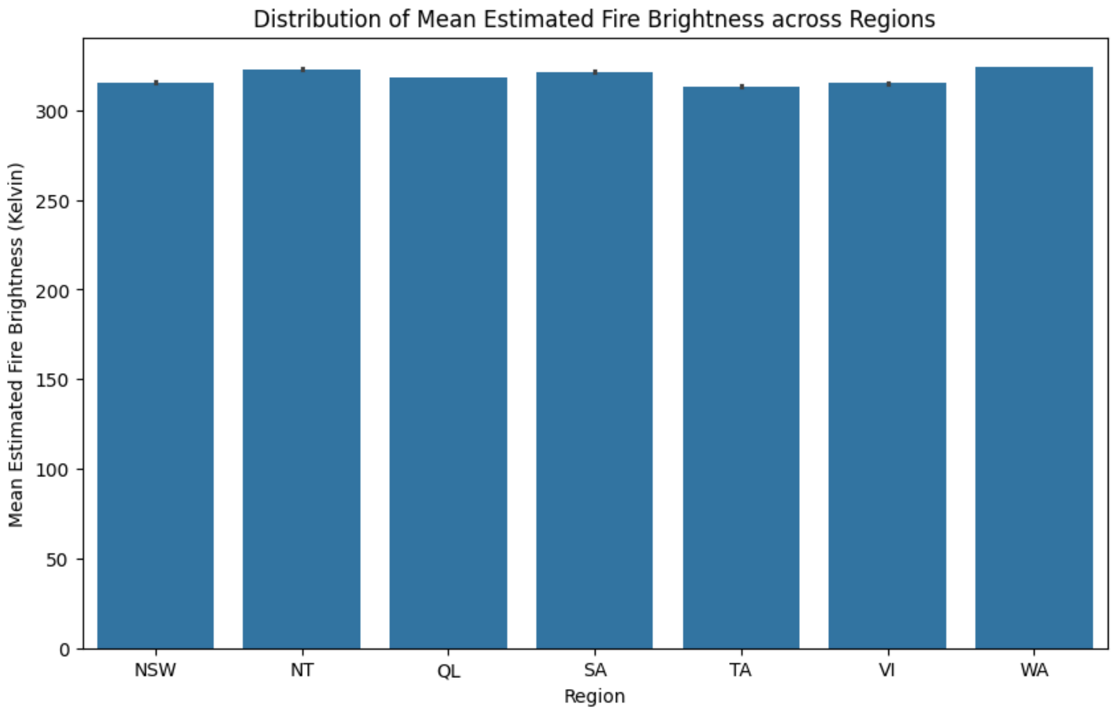


2.3 TASK 1.3: Let's have an insight on the distribution of mean estimated fire brightness across the regions use the functionality of seaborn to develop a barplot

[17]:
```
df['Region'].unique()
```

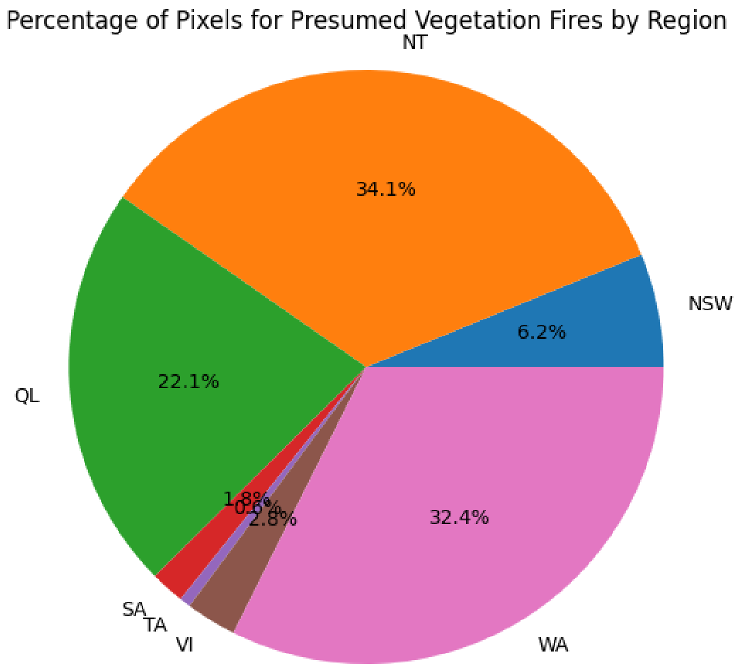
[17]:
```
array(['NSW', 'NT', 'QL', 'SA', 'TA', 'VI', 'WA'], dtype=object)
```

[18]:
```
# Creating a bar plot using seaborn to visualize the distribution of mean␣
↪estimated fire brightness across regions
plt.figure(figsize=(10, 6))
# Using seaborn's barplot function to create the plot
sns.barplot(data=df, x='Region', y='Mean_estimated_fire_brightness')
plt.xlabel('Region')
plt.ylabel('Mean Estimated Fire Brightness (Kelvin)')
plt.title('Distribution of Mean Estimated Fire Brightness across Regions')
plt.show()
```



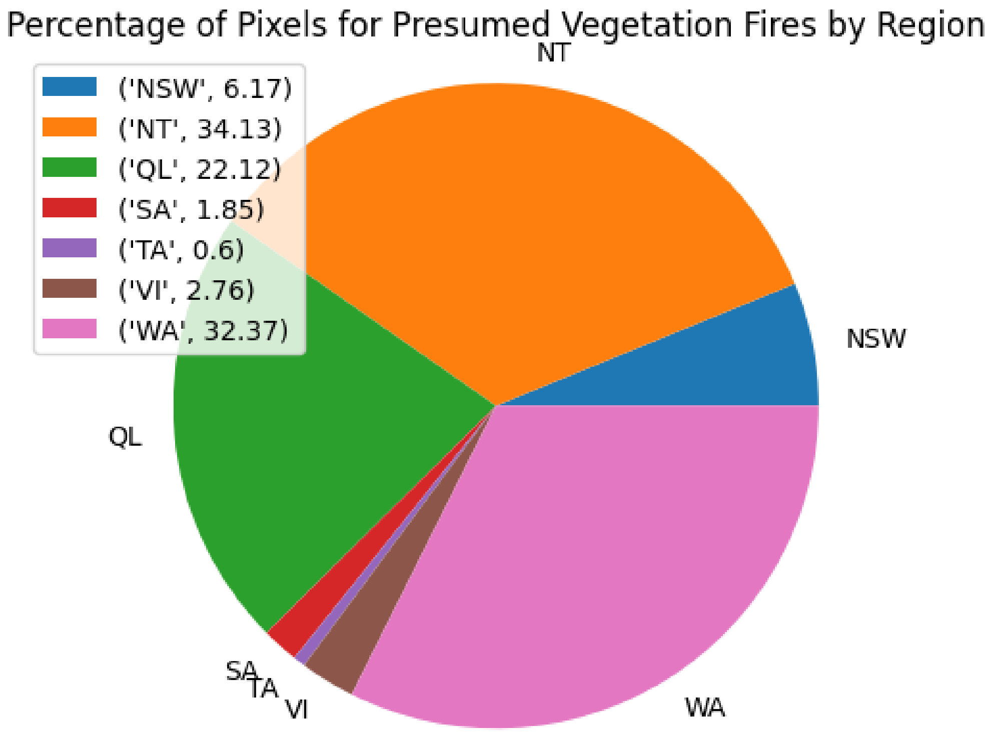
2.4 TASK 1.4: Let's find the portion of count of pixels for presumed vegetation fires vary across regions we will develop a pie chart for this

[19]:
```
# Creating a pie chart to visualize the portion of count of pixels for presumed␣
↪vegetation fires across regions
plt.figure(figsize=(10, 6))
# Grouping the data by region and summing the counts
region_counts = df.groupby('Region')['Count'].sum()
# Creating the pie chart using plt.pie function
# Labels are set to the region names, and autopct is used to display percentage
plt.pie(region_counts, labels=region_counts.index, autopct='%1.1f%%')
plt.title('Percentage of Pixels for Presumed Vegetation Fires by Region')
plt.axis('equal')
plt.show()
```



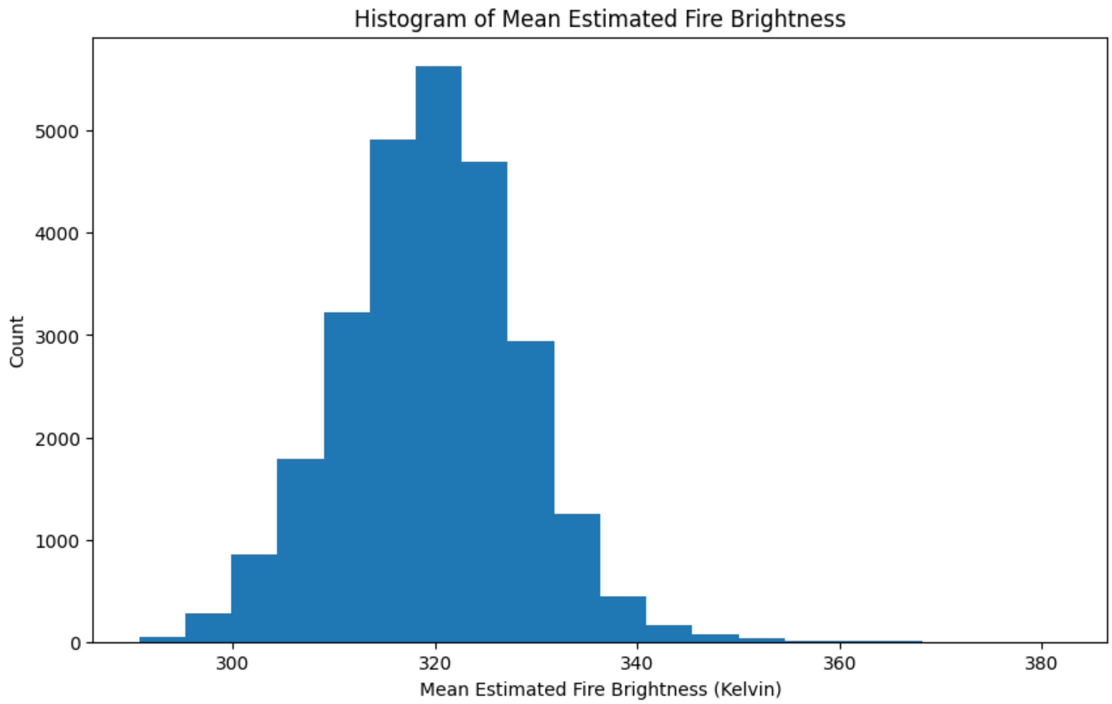
2.5 TASK 1.5: See the percentage on the pie is not looking so good as it is overlaped for Region SA, TA, VI

[20]:
```
plt.pie(region_counts, labels=region_counts.index)
plt.title('Percentage of Pixels for Presumed Vegetation Fires by Region')
plt.legend([(i,round(k/region_counts.sum()*100,2)) for i,k in zip(region_counts.
  ↪index, region_counts)])
plt.axis('equal')
plt.show()
```



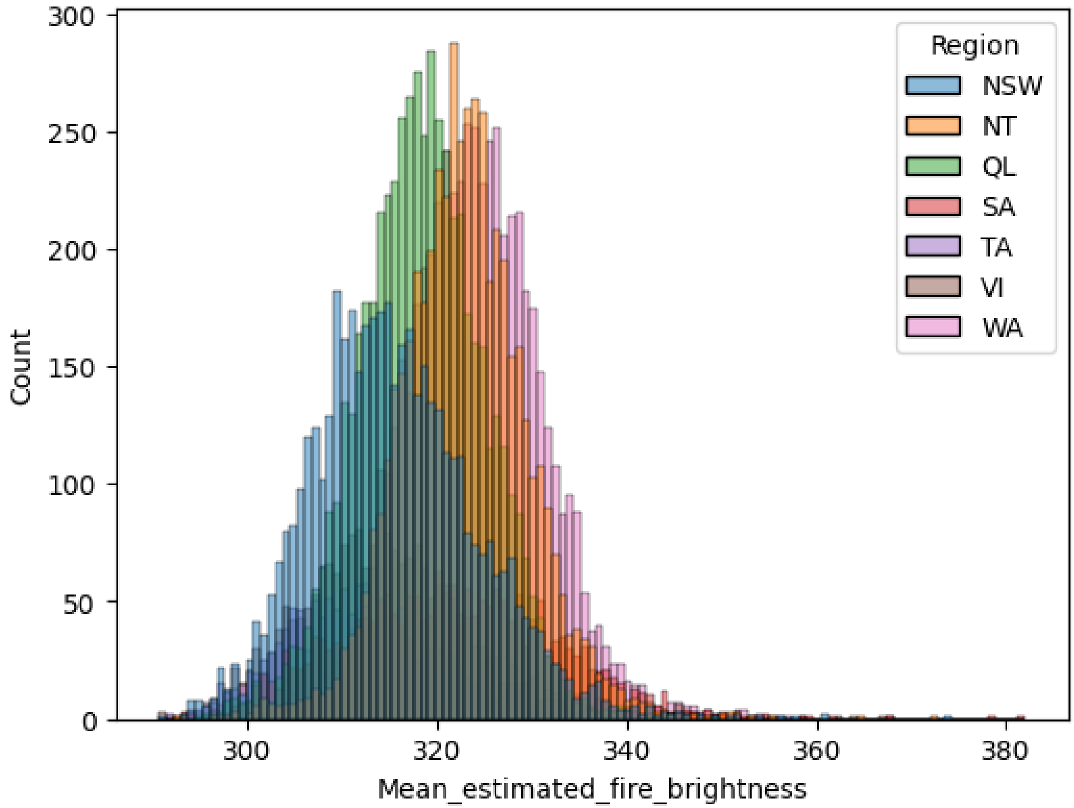
2.6 TASK 1.6: Let's try to develop a histogram of the mean estimated fire brightness Using Matplotlib to create the histogram

[21]:
```
# Creating a histogram to visualize the distribution of mean estimated fire␣
↪brightness
plt.figure(figsize=(10, 6))
# Using plt.hist to create the histogram
# Setting the number of bins to 20 for better visualization
plt.hist(x=df['Mean_estimated_fire_brightness'], bins=20)
plt.xlabel('Mean Estimated Fire Brightness (Kelvin)')
plt.ylabel('Count')
plt.title('Histogram of Mean Estimated Fire Brightness')
plt.show()
```

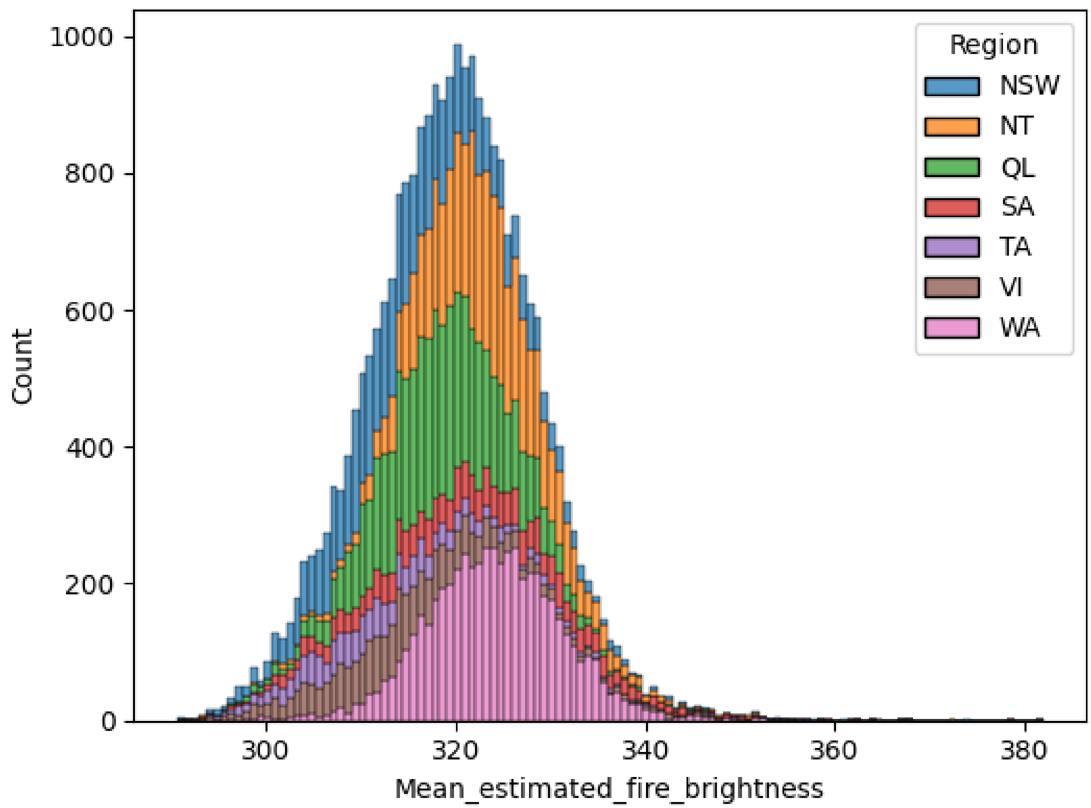


2.7 TASK 1.7: What if we need to understand the distribution of estimated fire brightness across regions? Let's use the functionality of seaborn and pass region as hue

[22]:
```
sns.histplot(data=df, x='Mean_estimated_fire_brightness', hue='Region')
plt.show()
```

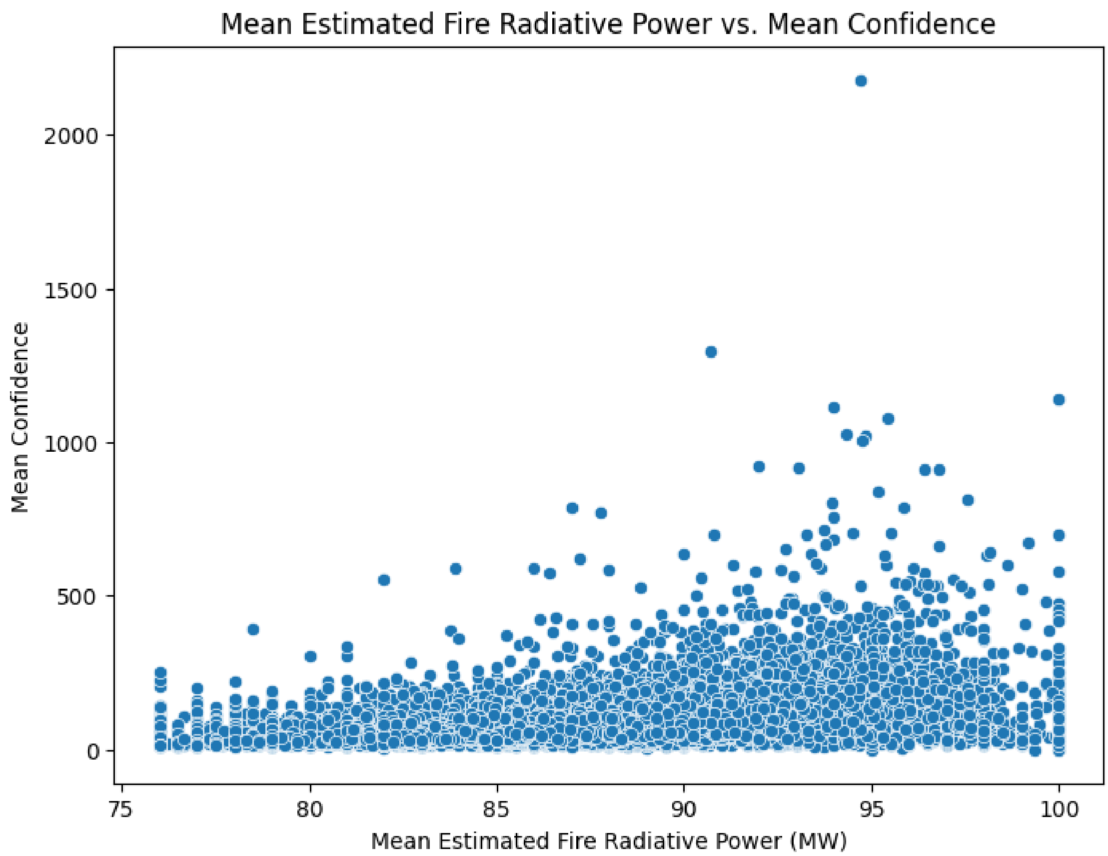


[23]:
```
sns.histplot(data=df, x='Mean_estimated_fire_brightness', hue='Region',␣
    ↪multiple='stack')
plt.show()
```



2.8 TASK 1.8: Let's try to find if there is any correlation between mean estimated fire radiative power and mean confidence level?

[24]:
```
plt.figure(figsize=(8, 6))
sns.scatterplot(data=df, x='Mean_confidence',␣
↪y='Mean_estimated_fire_radiative_power')
plt.xlabel('Mean Estimated Fire Radiative Power (MW)')
plt.ylabel('Mean Confidence')
plt.title('Mean Estimated Fire Radiative Power vs. Mean Confidence')
plt.show()
```



2.9 TASK 1.9: Let's mark these seven regions on the Map of Australia using Folium

[25]:
```
region_data = {'region':['NSW','QL','SA','TA','VI','WA','NT'], 'Lat':[-31.
↪8759835,-22.1646782,-30.5343665,-42.035067,-36.5986096,-25.2303005,-19.
↪491411],
              'Lon':[147.2869493,144.5844903,135.6301212,146.6366887,144.
↪6780052,121.0187246,132.550964]}
reg=pd.DataFrame(region_data)
reg
```

[25]:

	region	Lat	Lon
0	NSW	-31.875984	147.286949
1	QL	-22.164678	144.584490
2	SA	-30.534367	135.630121
3	TA	-42.035067	146.636689
4	VI	-36.598610	144.678005
5	WA	-25.230300	121.018725
6	NT	-19.491411	132.550964

[26]:

```python
# instantiate a feature group
aus_reg = folium.map.FeatureGroup()

# Create a Folium map centered on Australia
Aus_map = folium.Map(location=[-25, 135], zoom_start=4)

# loop through the region and add to feature group
for lat, lng, lab in zip(reg.Lat, reg.Lon, reg.region):
    aus_reg.add_child(
        folium.features.CircleMarker(
            [lat, lng],
            popup=lab,
            radius=5, # define how big you want the circle markers to be
            color='red',
            fill=True,
            fill_color='blue',
            fill_opacity=0.6
        )
    )

# add incidents to map
Aus_map.add_child(aus_reg)
```

[26]:

```
<folium.folium.Map at 0x238a71b51c0>
```

[ ]:

```

```