In [24]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [25]:

```
import warnings
warnings.filterwarnings('ignore')
```

In [26]:

```
data = pd.read_csv("AQD_2019.csv")
```

In [28]:

```
data.head()
```

Out[28]:

	AQS_ID	LATITUDE	LONGITUDE	COUNTY	STATE	CBSA	PEOPLE_OF_COLOR_FRACTION
0	01-003-0010	30.497478	-87.880258	Baldwin	Alabama	Daphne-Fairhope-Foley, AL	0.13
1	01-003-0010	30.497478	-87.880258	Baldwin	Alabama	Daphne-Fairhope-Foley, AL	0.13
2	01-003-0010	30.497478	-87.880258	Baldwin	Alabama	Daphne-Fairhope-Foley, AL	0.13
3	01-003-0010	30.497478	-87.880258	Baldwin	Alabama	Daphne-Fairhope-Foley, AL	0.13
4	01-003-0010	30.497478	-87.880258	Baldwin	Alabama	Daphne-Fairhope-Foley, AL	0.13

5 rows × 22 columns

In [29]:

```
data.tail()
```

Out[29]:

	AQS_ID	LATITUDE	LONGITUDE	COUNTY	STATE	CBSA	PEOPLE_OF_COLOR_FRACTI
129465	72-021-0010	18.420089	-66.150615	Bayamon	Puerto Rico	San Juan-Carolina-Caguas, PR	N
129466	72-021-0010	18.420089	-66.150615	Bayamon	Puerto Rico	San Juan-Carolina-Caguas, PR	N
129467	72-021-0010	18.420089	-66.150615	Bayamon	Puerto Rico	San Juan-Carolina-Caguas, PR	N
129468	72-021-0010	18.420089	-66.150615	Bayamon	Puerto Rico	San Juan-Carolina-Caguas, PR	N
129469	72-021-0010	18.420089	-66.150615	Bayamon	Puerto Rico	San Juan-Carolina-Caguas, PR	N

5 rows × 22 columns

In [30]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 129470 entries, 0 to 129469
Data columns (total 22 columns):
 #   Column                           Non-Null Count   Dtype  
---  ------                           --------------   -----  
 0   AQS_ID                           129470 non-null  object 
 1   LATITUDE                         129470 non-null  float64
 2   LONGITUDE                        129470 non-null  float64
 3   COUNTY                           129470 non-null  object 
 4   STATE                            129470 non-null  object 
 5   CBSA                             117210 non-null  object 
 6   PEOPLE_OF_COLOR_FRACTION         129393 non-null  float64
 7   LOW_INCOME_FRACTION              129393 non-null  float64
 8   LINGUISTICALLY_ISOLATED_FRACTION 129393 non-null  float64
 9   LESS_THAN_HS_ED_FRACTION         129393 non-null  float64
 10  DATE                             129470 non-null  object 
 11  TEMPERATURE_CELSIUS              72703 non-null   float64
 12  RELATIVE_HUMIDITY                50670 non-null   float64
 13  WIND_SPEED_METERS_PER_SECOND     58576 non-null   float64
 14  WIND_DIRECTION                   59484 non-null   float64
 15  PM25_UG_PER_CUBIC_METER          129470 non-null  float64
 16  OZONE_PPM                        129470 non-null  float64
 17  NO2_PPB                          61395 non-null   float64
 18  CO_PPM                           39749 non-null   float64
 19  SO2_PPB                          47337 non-null   float64
 20  LEAD_UG_PER_CUBIC_METER          659 non-null     float64
 21  BENZENE_PPBC                     3307 non-null    float64
dtypes: float64(17), object(5)
memory usage: 21.7+ MB
```

In [31]:

```
data.describe()
```

Out[31]:

	LATITUDE	LONGITUDE	PEOPLE_OF_COLOR_FRACTION	LOW_INCOME_FRACTION	L
count	129470.000000	129470.000000	129393.000000	129393.000000	
mean	38.533022	-96.298816	0.383927	0.375089	
std	4.837426	17.693938	0.303357	0.215389	
min	18.420089	-158.088613	0.000000	0.000000	
25%	35.320105	-112.095767	0.110000	0.210000	
50%	39.138773	-93.512534	0.320000	0.350000	
75%	41.530011	-80.341962	0.660000	0.540000	
max	64.845690	-66.150615	1.000000	0.990000	

In [32]:

```
data.isnull().sum()
```

Out[32]:

```
AQS_ID                                 0
LATITUDE                               0
LONGITUDE                              0
COUNTY                                 0
STATE                                  0
CBSA                               12260
PEOPLE_OF_COLOR_FRACTION              77
LOW_INCOME_FRACTION                   77
LINGUISTICALLY_ISOLATED_FRACTION      77
LESS_THAN_HS_ED_FRACTION              77
DATE                                   0
TEMPERATURE_CELSIUS                56767
RELATIVE_HUMIDITY                  78800
WIND_SPEED_METERS_PER_SECOND       70894
WIND_DIRECTION                     69986
PM25_UG_PER_CUBIC_METER                0
OZONE_PPM                              0
NO2_PPB                            68075
CO_PPM                             89721
SO2_PPB                            82133
LEAD_UG_PER_CUBIC_METER           128811
BENZENE_PPBC                      126163
dtype: int64
```

In [33]:

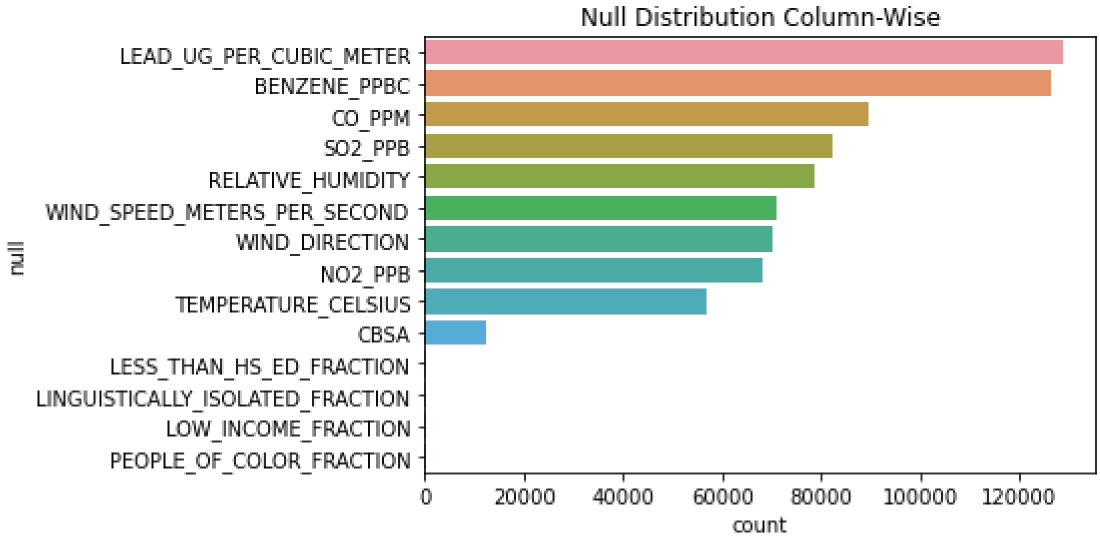
```
data.shape
```

Out[33]:

```
(129470, 22)
```

In [34]:

```
data1 = pd.DataFrame(data.isna().sum().sort_values(ascending=False))
data1['null']=data1.index
data1['count']=data1.iloc[:,:-1]
data1.reset_index(drop=True, inplace=True)
data1 = data1.drop(data1.columns[[0]],axis = 1)
plt.title('Null Distribution Column-Wise')
ax = sns.barplot(y='null',x='count',data=data1.head(14))
```

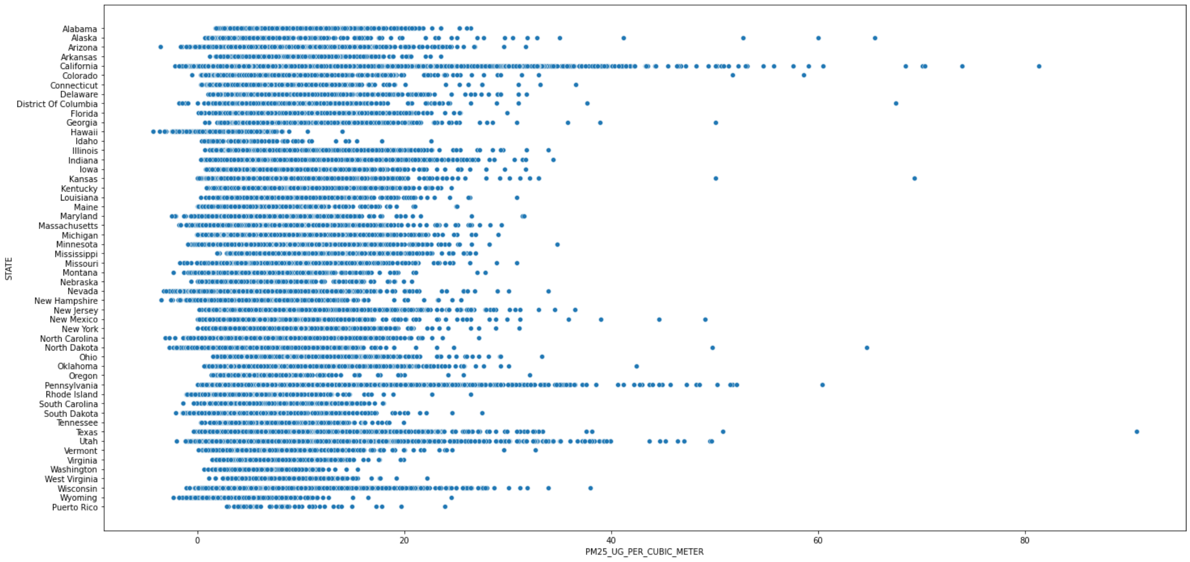


In [35]:

```
plt.figure(figsize=(24,12))
sns.scatterplot(x="PM25_UG_PER_CUBIC_METER",y="STATE",data=data)
```

Out[35]:

```
<matplotlib.axes._subplots.AxesSubplot at 0xd43751dfa0>
```



In [36]:

```
plt.figure(figsize=(12,6))
groupby=pd.DataFrame(data.groupby(['STATE']).sum())
#.plot(kind='pie', autopct='%1.0f%%',y='PEOPLE_OF_COLOR_FRACTION')
groupby['State']=groupby.index
groupby.reset_index(drop=True, inplace=True)
groupby = groupby.drop(groupby.columns[[0]],axis = 1)
groupby.head(5)
```

Out[36]:

	LONGITUDE	PEOPLE_OF_COLOR_FRACTION	LOW_INCOME_FRACTION	LINGUISTICALLY_ISOLA
0	-9.378111e+04	709.69	584.83	
1	-4.284095e+04	118.90	92.80	
2	-4.026421e+05	2265.89	2053.45	
3	-5.487731e+04	421.26	386.75	
4	-2.889200e+06	13461.82	9855.79	

```
<Figure size 864x432 with 0 Axes>
```

In [37]:

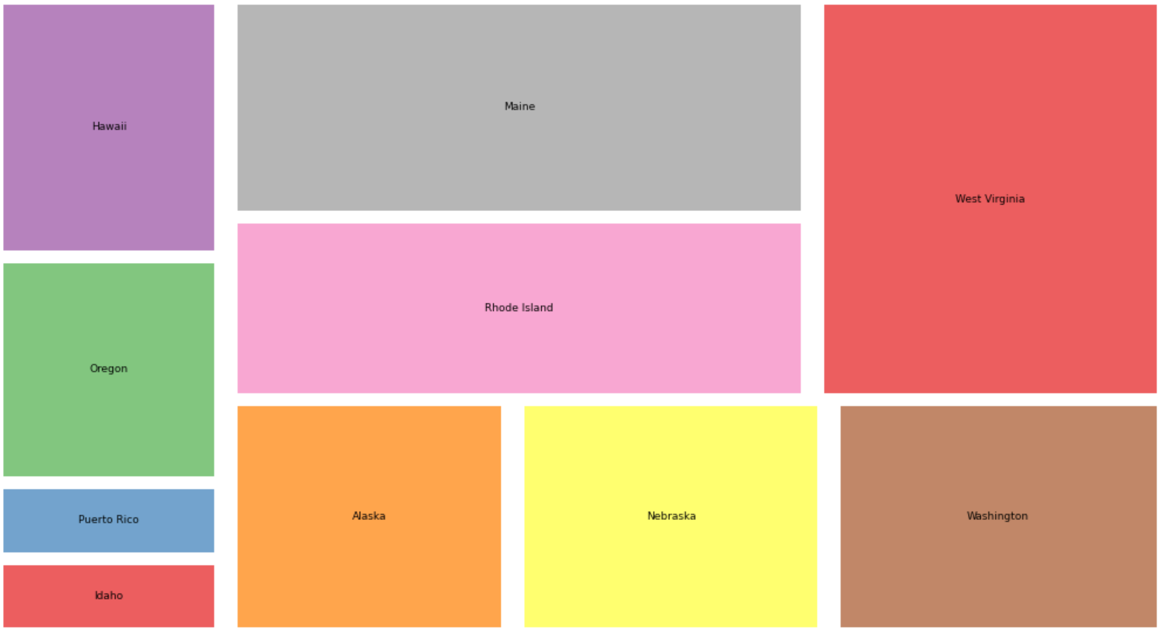
```
shapes = groupby[['PM25_UG_PER_CUBIC_METER','State']].sort_values(by='PM25_UG_PER_CUBIC_
shapes['PM25_UG_PER_CUBIC_METER'].head(10).unique()
shapes = groupby[['PM25_UG_PER_CUBIC_METER','State']].sort_values(by='PM25_UG_PER_CUBIC_
shapes['State'].head(10).unique()
```

Out[37]:

```
array(['Idaho', 'Puerto Rico', 'Oregon', 'Hawaii', 'Alaska', 'Nebraska',
       'Washington', 'Rhode Island', 'Maine', 'West Virginia'],
      dtype=object)
```

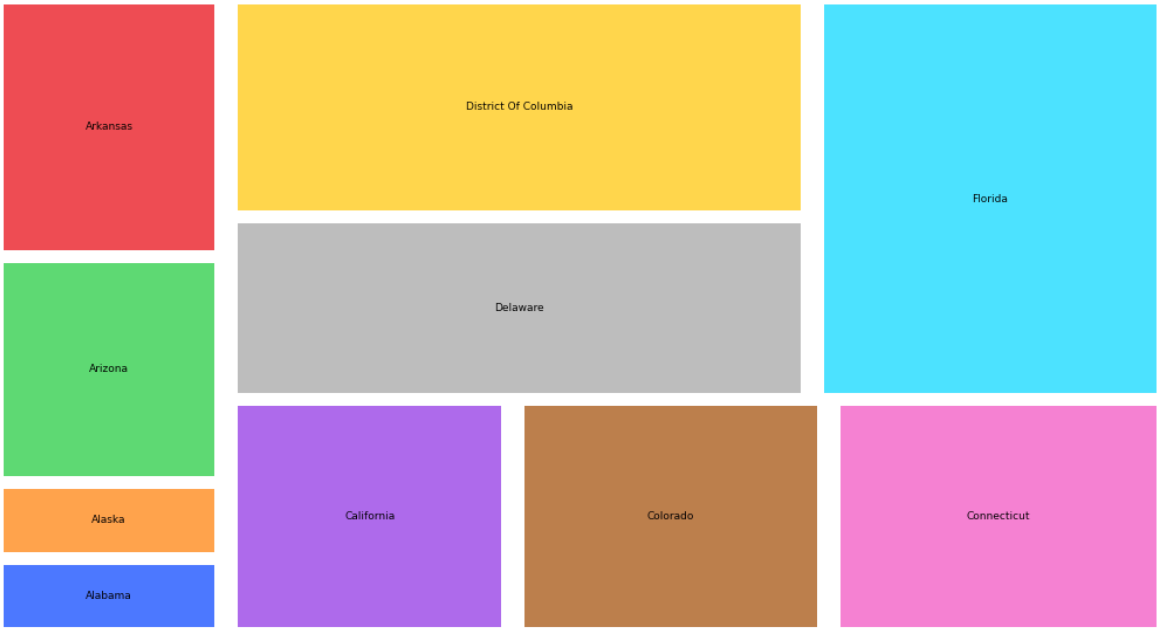
In [38]:

```
import squarify
plt.figure(figsize=(18,10))
squarify.plot(sizes=[567.7,  568.  , 1692.7, 1938.8, 2149.1, 2363.2, 2541.5, 3437.4,4114.
              label=['Idaho', 'Puerto Rico', 'Oregon', 'Hawaii', 'Alaska', 'Nebraska','W
                  pad=0.8,text_kwargs={'fontsize':9})
plt.axis('off')
plt.show()
```



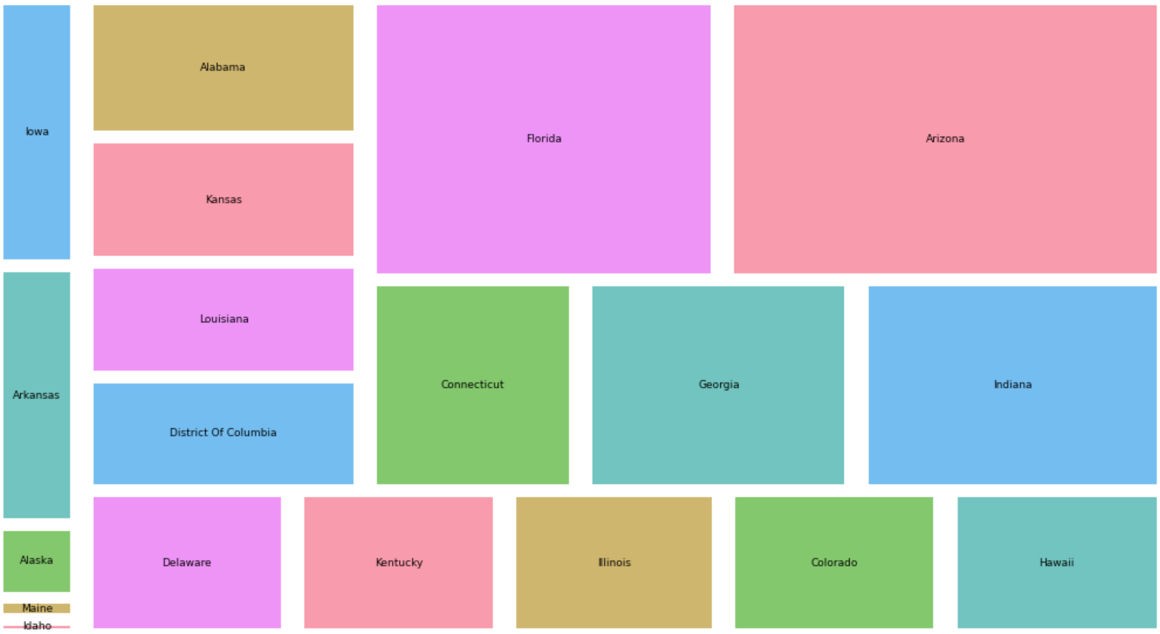
In [39]:

```
plt.figure(figsize=(18,10))
squarify.plot(sizes=[567.7,  568.  , 1692.7, 1938.8, 2149.1, 2363.2, 2541.5, 3437.4, 411
              label=['Alabama', 'Alaska', 'Arizona', 'Arkansas', 'California','Colorado
       'Florida'], alpha=.7,color = sns.color_palette('bright',10),
                  pad=0.8,text_kwargs={'fontsize':9})
plt.axis('off')
plt.show()
```



In [40]:

```
import squarify
plt.figure(figsize=(18,10))
squarify.plot(sizes=[25.92,34.56,118.9,421.26,434.97,549.02,554.63,571.27,579.36,581.04,
              label=['Idaho', 'Maine', 'Alaska', 'Arkansas', 'Iowa', 'Delaware',
       'Kentucky', 'Illinois', 'Colorado', 'Hawaii',
       'District Of Columbia', 'Louisiana', 'Kansas', 'Alabama',
       'Connecticut', 'Georgia', 'Indiana', 'Florida', 'Arizona'], alpha=.7,color = sns.
                  pad=0.8,text_kwargs={'fontsize':9})
plt.axis('off')
plt.show()
```

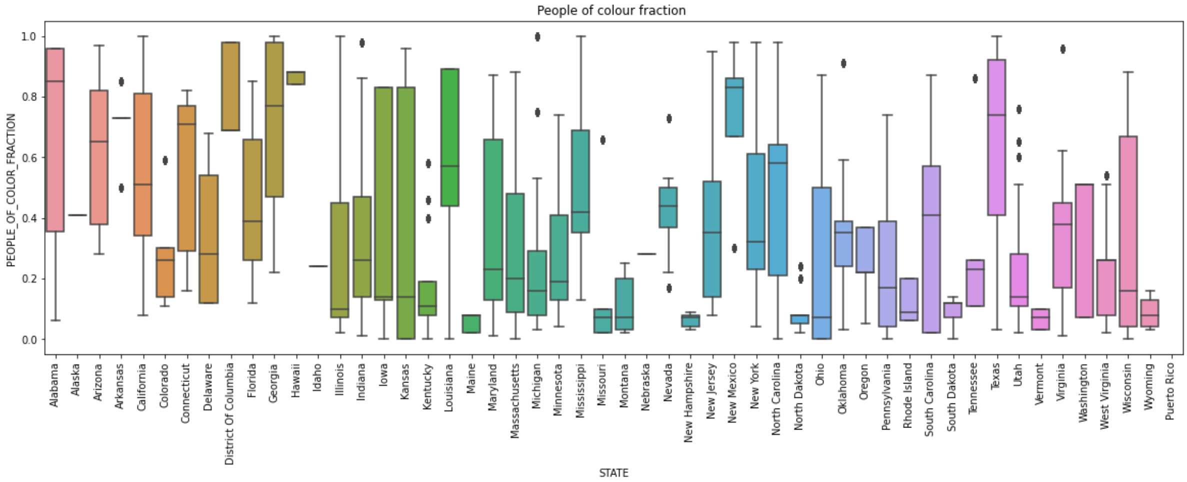


In [41]:

```
plt.figure(figsize=(20,6))
ax = sns.boxplot(x='STATE',y='PEOPLE_OF_COLOR_FRACTION',data=data)
plt.xticks(rotation=90)
ax.set_title("People of colour fraction ")
```

Out[41]:

```
Text(0.5, 1.0, 'People of colour fraction ')
```

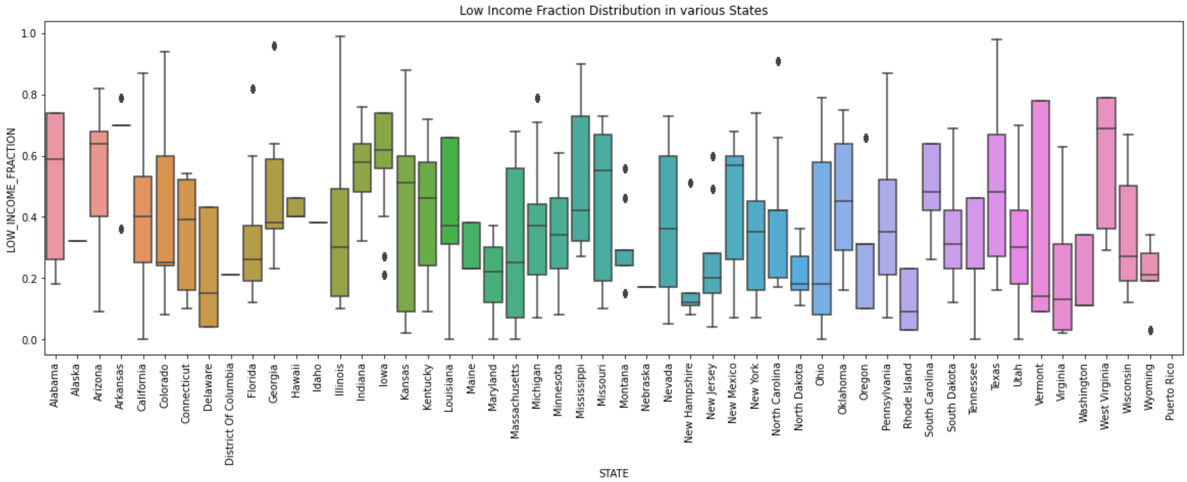


In [42]:

```
plt.figure(figsize=(20,6))
ax = sns.boxplot(x='STATE',y='LOW_INCOME_FRACTION',data=data)
plt.xticks(rotation=90)
ax.set_title("Low Income Fraction Distribution in various States")
```

Out[42]:

```
Text(0.5, 1.0, 'Low Income Fraction Distribution in various States')
```

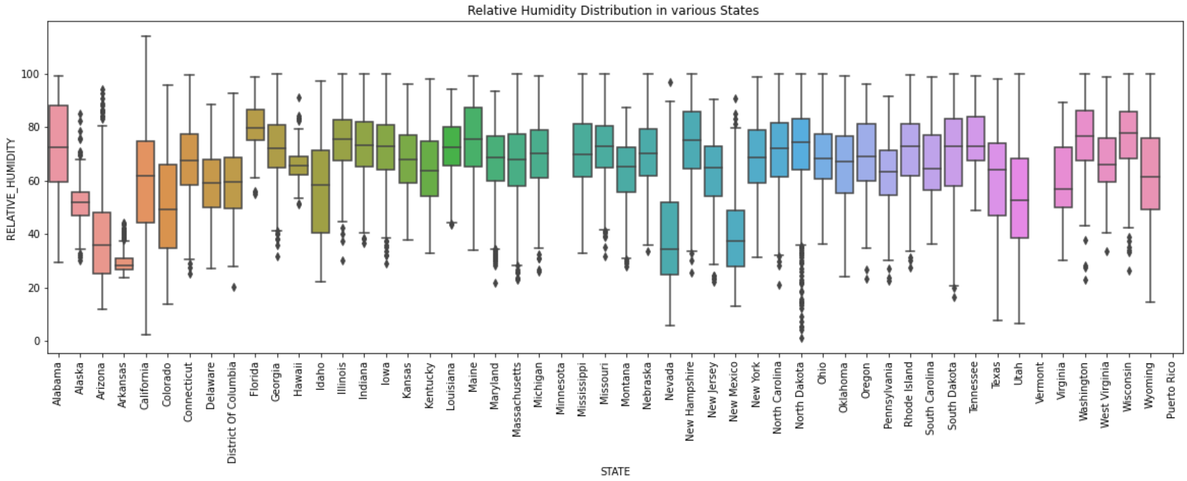


In [43]:

```
plt.figure(figsize=(20,6))
ax = sns.boxplot(x='STATE',y='RELATIVE_HUMIDITY',data=data)
plt.xticks(rotation=90)
ax.set_title("Relative Humidity Distribution in various States ")
```

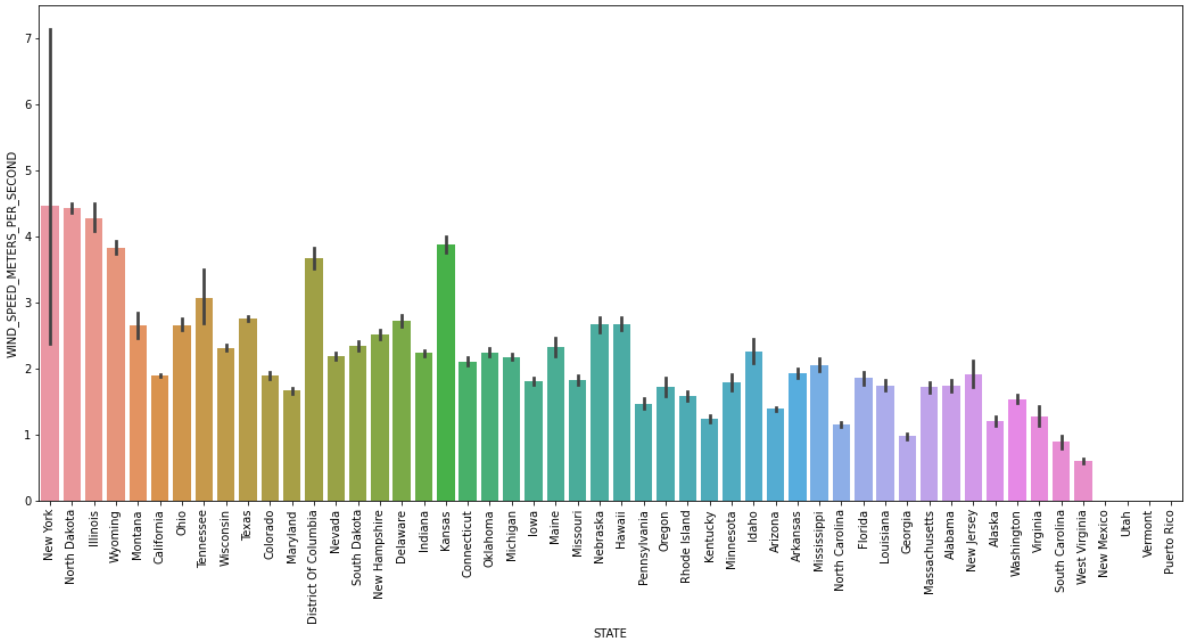
Out[43]:

```
Text(0.5, 1.0, 'Relative Humidity Distribution in various States ')
```



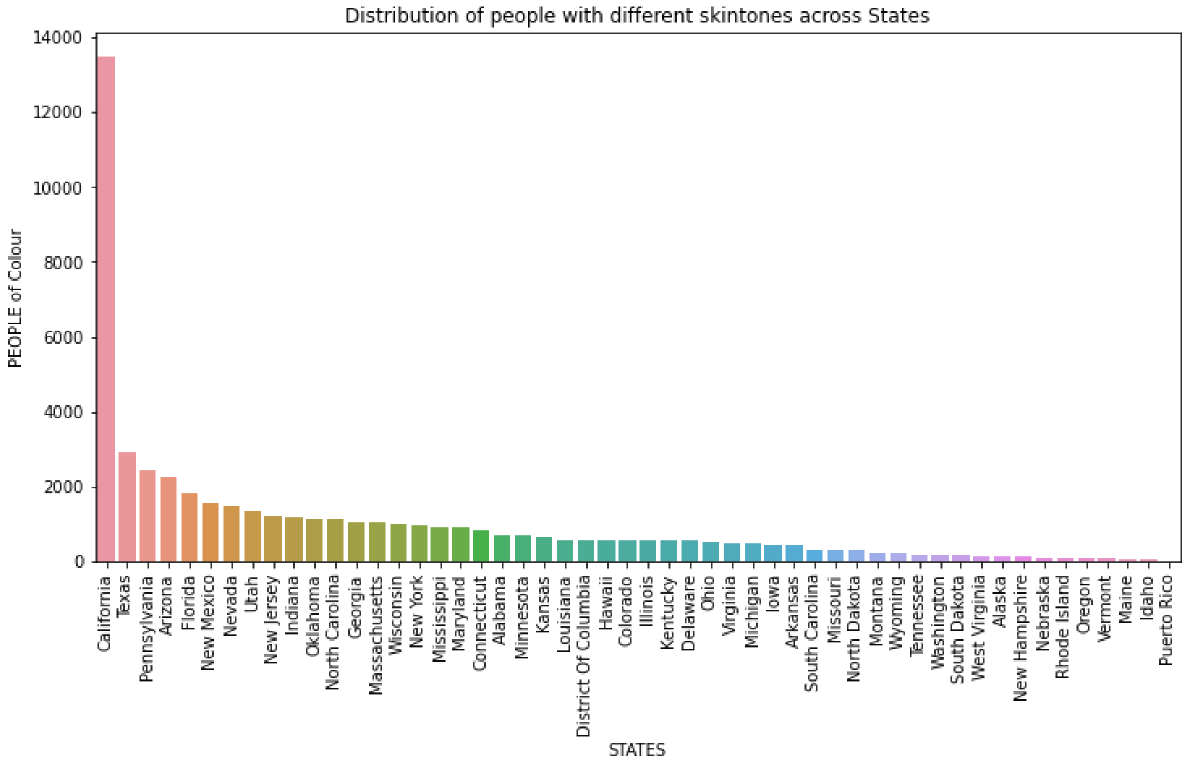
In [44]:

```
plt.figure(figsize=(18,8))
data1 = data[['WIND_SPEED_METERS_PER_SECOND','STATE']].sort_values(by='WIND_SPEED_METERS
plt.xticks(rotation=90)
sns.barplot(x='STATE',y='WIND_SPEED_METERS_PER_SECOND',data=data1)
plt.show()
```



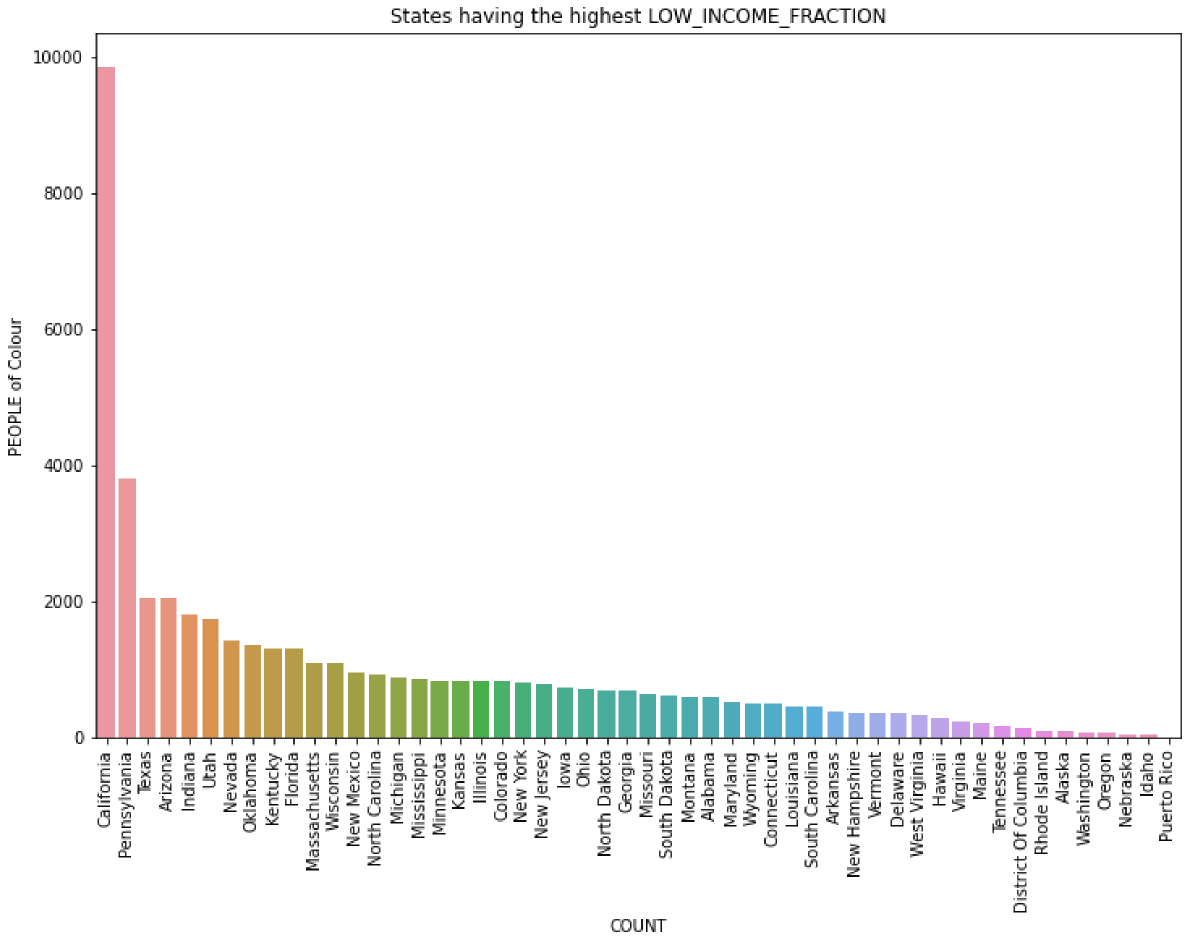
In [45]:

```
dfa=groupby.sort_values(by=['PEOPLE_OF_COLOR_FRACTION'], ascending=False)
plt.figure(figsize=(12,6))
plt.xticks(rotation=90)
sns.barplot(x='State',y='PEOPLE_OF_COLOR_FRACTION',data=dfa)
plt.xlabel('STATES')
plt.ylabel('PEOPLE of Colour')
plt.title('Distribution of people with different skintones across States')
plt.show()
```



In [46]:

```
dfa=groupby.sort_values(by=['LOW_INCOME_FRACTION'], ascending=False)
plt.figure(figsize=(12,8))
sns.barplot(x='State',y='LOW_INCOME_FRACTION',data=dfa)
plt.xticks(rotation=90)
plt.xlabel('COUNT')
plt.ylabel('PEOPLE of Colour')
plt.title('States having the highest LOW_INCOME_FRACTION')
plt.show()
```

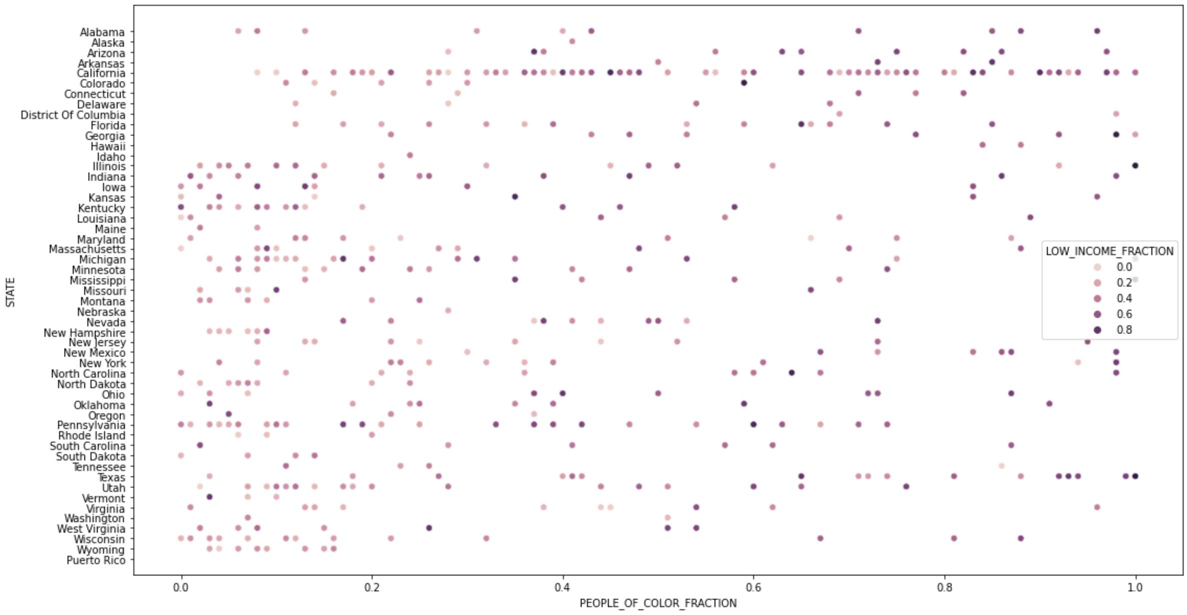


In [47]:

```
plt.figure(figsize=(18,10))
sns.scatterplot(x="PEOPLE_OF_COLOR_FRACTION",y="STATE",data=data,hue='LOW_INCOME_FRACTI
```

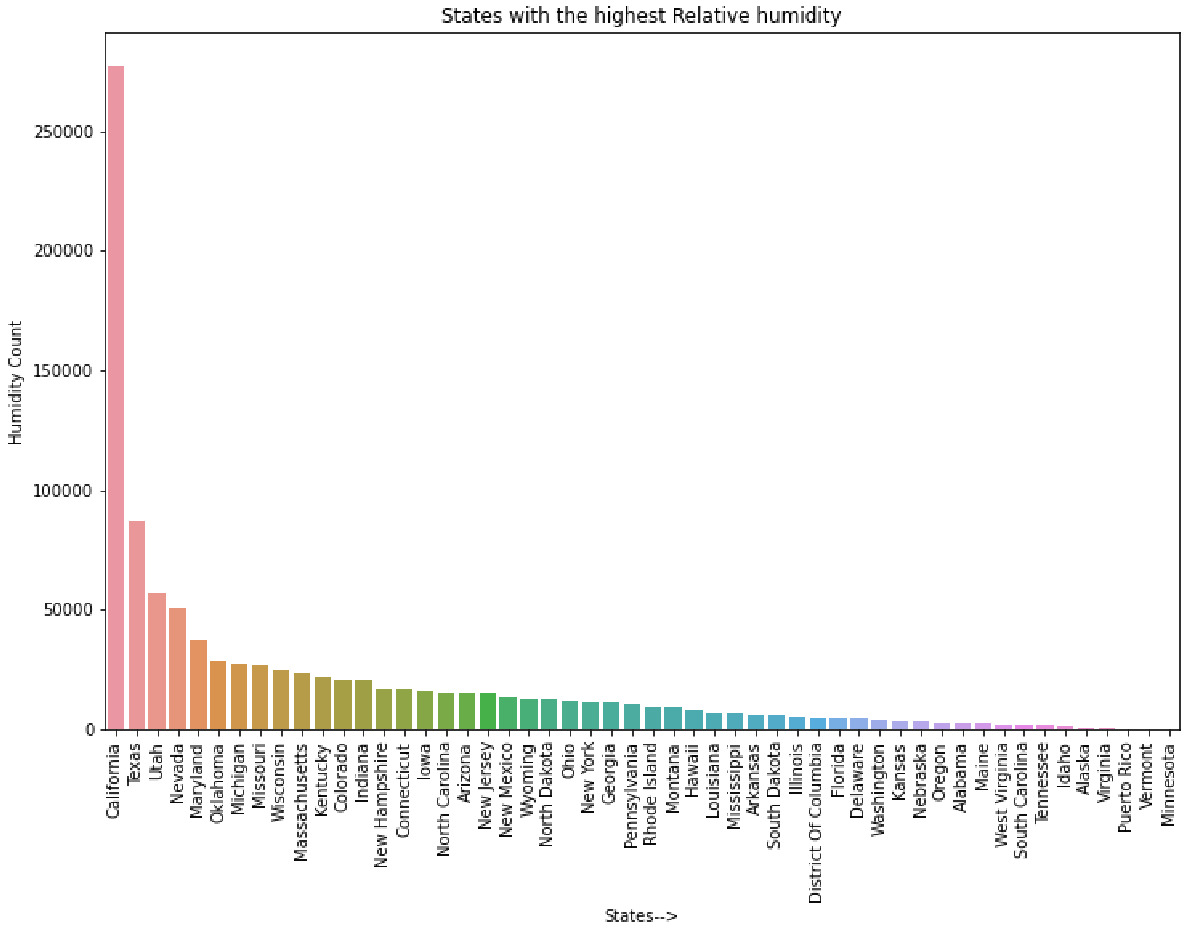
Out[47]:

```
<matplotlib.axes._subplots.AxesSubplot at 0xd43b7aa460>
```



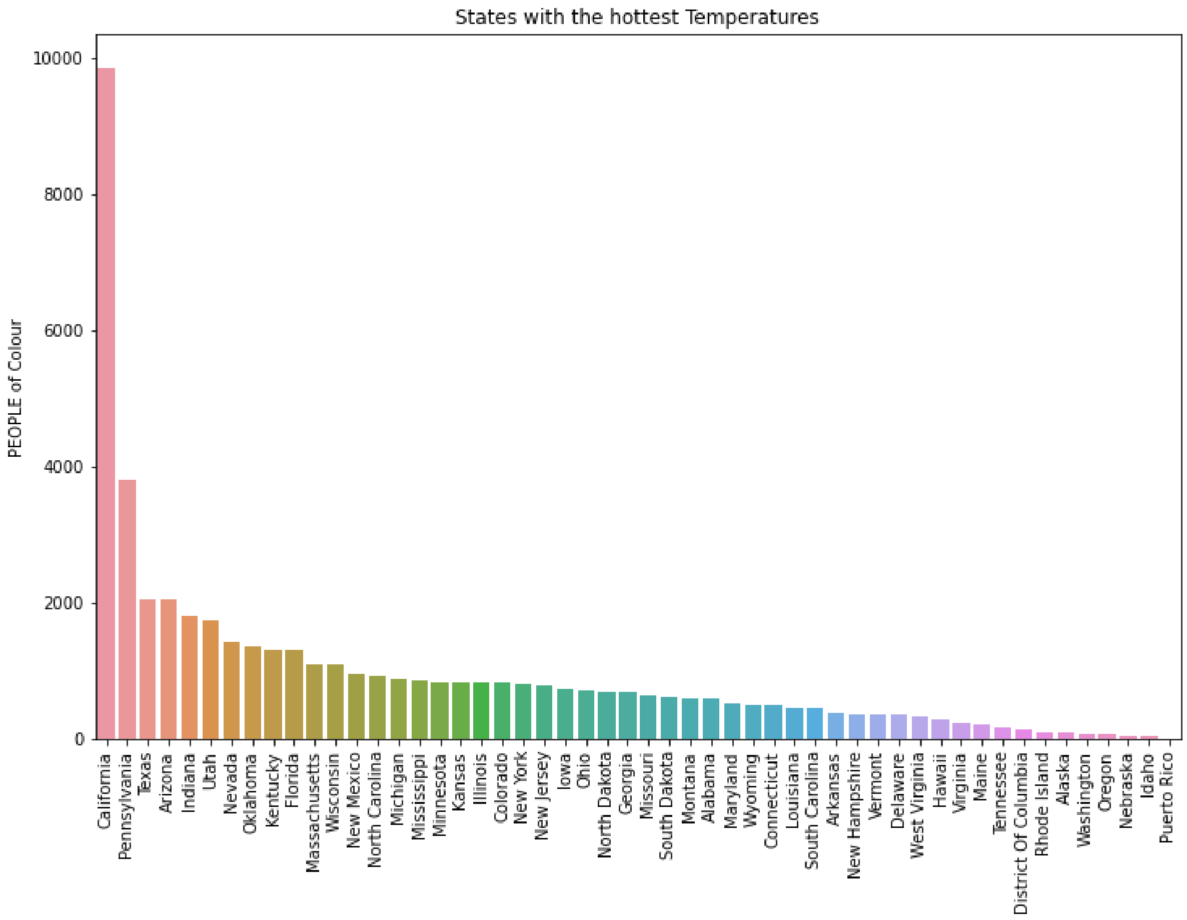
In [48]:

```
dfa=groupby.sort_values(by=['TEMPERATURE_CELSIUS'], ascending=False)
plt.figure(figsize=(12,8))
sns.barplot(x='State',y='TEMPERATURE_CELSIUS',data=dfa)
plt.xticks(rotation=90)
plt.xlabel('States-->')
plt.ylabel('Humidity Count')
plt.title('States with the highest Relative humidity')
plt.show()
```



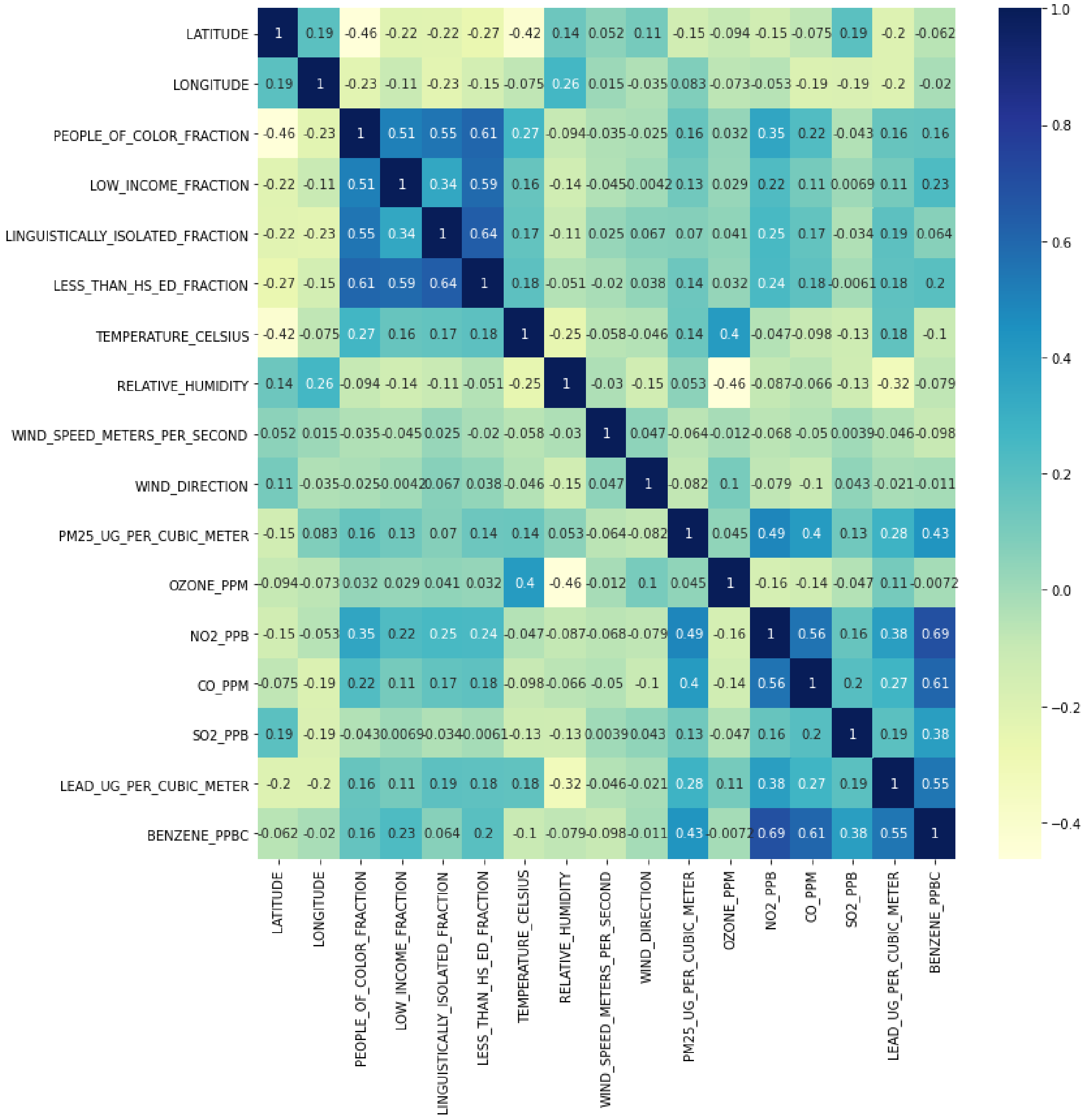
In [49]:

```
dfa=groupby.sort_values(by=['LOW_INCOME_FRACTION'], ascending=False)
plt.figure(figsize=(12,8))
sns.barplot(x='State',y='LOW_INCOME_FRACTION',data=dfa)
plt.xticks(rotation=90)
plt.xlabel('COUNT')
plt.ylabel('PEOPLE of Colour')
plt.title('States with the hottest Temperatures')
plt.show()
```



In [50]:

```
plt.figure(figsize=(12,12))
dataplot = sns.heatmap(data.corr(), cmap="YlGnBu", annot=True)
plt.show()
```



In [58]:

```
data1 = data.dropna()
```

In [59]:

```
data1.shape
```

Out[59]:

```
(110, 22)
```

In [60]:

```
data2 = data1[['RELATIVE_HUMIDITY', 'WIND_SPEED_METERS_PER_SECOND',
               'WIND_DIRECTION','OZONE_PPM','NO2_PPB','CO_PPM','SO2_PPB',
               'LEAD_UG_PER_CUBIC_METER',
               'BENZENE_PPBC']]
```

In [62]:

```
x = data2.drop(['OZONE_PPM'], axis = 1)
```

In [63]:

```
y = data2.OZONE_PPM
```

In [64]:

```
x.shape
```

Out[64]:

```
(110, 8)
```

In [65]:

```
y.shape
```

Out[65]:

```
(110,)
```

In [66]:

```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size = 0.2)
```

In [67]:

```
model= LinearRegression()
model.fit(X_train, y_train)
```

Out[67]:

```
LinearRegression()
```

In [71]:

```
y_pred = model.predict(X_test)
```

In [72]:

```
print("Training Accuracy :", model.score(X_train, y_train))
print("Testing Accuracy :", model.score(X_test, y_test))
```

```
Training Accuracy : 0.5888597175737917
Testing Accuracy : 0.5309685152061103
```