In [58]:
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sb
```

In [84]:
```python
G12 = pd.read_csv("g2.csv")
G13 = pd.read_csv("g3.csv")
G14 = pd.read_csv("g4.csv")
G15 = pd.read_csv("g5.csv")
G16 = pd.read_csv("g6.csv")
G17 = pd.read_csv("g7.csv")
G18 = pd.read_csv("g8.csv")
G19 = pd.read_csv("g9.csv")
G21 = pd.read_csv("g21.csv")
tds = pd.read_csv("total mlt data.csv",encoding="latin1")
```

In [85]:
```python
df = pd.concat([G12,G13,G14,G15,G16,G17,G18,G19,G21,tds], ignore_index = True)
```

In [86]:
```python
df['STATE'] = df['STATE'].str.upper()
```

In [87]:
```python
display(df)
```

	Station Code	Station Name	STATE	Temperature Min	Temperature Max	pH Min	pH Max	Conductivity (µmhos/cm) Min	Conductivity (µmhos/cm) Max	Year	Source.Name	TDS
0	15.0	WELL AT KUYYURA, A.P.	ANDHRA PRADESH	23.0	28.0	6.82	7.85	195.0	226.0	2012.0	NaN	NaN
1	16.0	WELL AT TADAVAI A.P.	ANDHRA PRADESH	27.0	28.0	7.21	7.72	663.0	1121.0	2012.0	NaN	NaN
2	26.0	WELL AT VIJAYWADA, A.P.	ANDHRA PRADESH	26.0	32.0	7.19	7.39	1601.0	1661.0	2012.0	NaN	NaN
3	27.0	WELL AT PEDDAVOORA, A.P.	ANDHRA PRADESH	26.0	27.0	7.82	8.80	1420.0	1668.0	2012.0	NaN	NaN
4	1513.0	B W. - KRISHNA MURTHY, D.NO. 48-16-43 AUTONAGA...	ANDHRA PRADESH	25.0	30.0	6.80	7.11	1454.0	1575.0	2012.0	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...
17126	NaN	WEST BENGAL	26	26.0	7.4	7.40	365.00	365.0	2019.0	NaN	g9.csv	381.393536
17127	4737.0	GROUND WATER AT SHYAMPUR	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	g9.csv	305.558109
17128	NaN	WEST BENGAL	28	28.0	7.7	7.70	7439.00	7439.0	2019.0	NaN	g9.csv	482.267280
17129	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	155.313718
17130	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	239.432459

17131 rows × 12 columns

In [88]:
```python
df.sort_values(by = "Year", inplace = True, ascending = False, ignore_index = 'True')
df
```

Out[88]:

	Station Code	Station Name	STATE	Temperature Min	Temperature Max	pH Min	pH Max	Conductivity (µmhos/cm) Min	Conductivity (µmhos/cm) Max	Year	Source.Name	TDS
0	NaN		JODHPUR"	NaN	29.0	30.0	8.0	8.3	15400.0	23000.0	g6.csv	376.575062
1	NaN	VILLAGE-NANDWAN	JODHPUR"	NaN	28.1	32.3	6.9	7.3	12000.0	13100.0	g6.csv	182.920215
2	NaN	NADESARI	VADODARA"	NaN	23.0	29.0	6.9	7.0	10180.0	12350.0	g9.csv	92.799589
3	NaN	BALOTRA	RAJASTHAN"	NaN	24.0	32.0	8.3	8.6	4900.0	12000.0	g7.csv	77.540880
4	NaN	NADESARI	VADODARA"	NaN	29.0	29.0	7.0	7.1	8980.0	11360.0	g8.csv	189.376592
...	...	...	...	...	...	...	...	...	...	...	...	...
17126	NaN	WEST BENGAL	26	26.0	7.4	7.4	365.0	365.0	2019.0	NaN	g9.csv	381.393536
17127	4737.0	GROUND WATER AT SHYAMPUR	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	g9.csv	305.558109
17128	NaN	WEST BENGAL	28	28.0	7.7	7.7	7439.0	7439.0	2019.0	NaN	g9.csv	482.267280
17129	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	155.313718
17130	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	239.432459

17131 rows × 12 columns

In [89]:
```python
df.dropna(axis = 0,inplace = True)
```

In [90]:
```python
result = df.isnull().sum()
result
```

Out[90]:
```
Station Code                   0
Station Name                   0
STATE                          0
Temperature Min                0
Temperature Max                0
pH Min                         0
pH Max                         0
Conductivity (µmhos/cm) Min    0
Conductivity (µmhos/cm) Max    0
Year                           0
Source.Name                    0
TDS                            0
dtype: int64
```

In [91]:
```python
df.reset_index(inplace=True, drop=True)
df.head(10)
```

Out[91]:

	Station Code	Station Name	STATE	Temperature Min	Temperature Max	pH Min	pH Max	Conductivity (µmhos/cm) Min	Conductivity (µmhos/cm) Max	Year	Source.Name	TDS
0	2530.0	HOT SPRING AT BAKRESHWAR	WEST BENGAL	61.0	67.0	9.1	9.1	487.0	523.0	2021.0	g21.csv	208.411231
1	2538.0	MOREGRAM CROSSING	WEST BENGAL	29.0	29.0	7.3	8.0	217.0	546.0	2021.0	g21.csv	209.029463
2	1519.0	BOREWELL AT NAGIRI	ANDHRA PRADESH	28.0	29.0	7.1	7.3	1872.0	2741.0	2021.0	g21.csv	344.153439
3	4008.0	GROUND WATER FROM GELEKY	ASSAM	23.0	25.0	7.0	7.2	216.0	287.0	2021.0	g21.csv	188.990083
4	1522.0	OPEN WELL PEDDANUVVI -VIZIANAGARAM	ANDHRA PRADESH	27.0	28.0	7.4	7.9	1120.0	1190.0	2021.0	g21.csv	83.137937
5	3155.0	GROUND WATER FROM SUPUAL	BIHAR	25.0	27.0	7.5	8.0	643.0	839.0	2021.0	g21.csv	276.252515
6	2597.0	GROUND WATER FROM SIWAN	BIHAR	24.0	24.0	7.3	7.3	448.0	448.0	2021.0	g21.csv	484.473936
7	3160.0	GROUND WATER FROM RAXAUL	BIHAR	25.0	25.0	7.9	7.9	269.0	269.0	2021.0	g21.csv	408.328651
8	3150.0	GROUND WATER FROM MANER	BIHAR	24.0	24.0	7.2	7.2	773.0	775.0	2021.0	g21.csv	391.426090
9	3149.0	GROUND WATER FROM KUDRA	BIHAR	22.0	22.0	7.9	7.9	571.0	571.0	2021.0	g21.csv	180.271734

In [92]:
```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2970 entries, 0 to 2969
Data columns (total 12 columns):
 #   Column                       Non-Null Count  Dtype  
---  ------                       --------------  -----  
 0   Station Code                 2970 non-null   float64
 1   Station Name                 2970 non-null   object 
 2   STATE                        2970 non-null   object 
 3   Temperature Min              2970 non-null   float64
 4   Temperature Max              2970 non-null   float64
 5   pH Min                       2970 non-null   float64
 6   pH Max                       2970 non-null   float64
 7   Conductivity (µmhos/cm) Min  2970 non-null   float64
 8   Conductivity (µmhos/cm) Max  2970 non-null   float64
 9   Year                         2970 non-null   float64
 10  Source.Name                  2970 non-null   object 
 11  TDS                          2970 non-null   float64
dtypes: float64(9), object(3)
memory usage: 278.6+ KB
```

In [93]:
```python
df['Temperature_Mean']=(df['Temperature Min']+df['Temperature Max'])/2
df['pH_Mean']=(df['pH Min']+df['pH Max'])/2
df['Conductivity_Mean']=(df['Conductivity (µmhos/cm) Min']+df['Conductivity (µmhos/cm) Max'])/2
```

In [69]:
```python
df.head()
```

Out[69]:

	Station Code	Station Name	STATE	Temperature Min	Temperature Max	pH Min	pH Max	Conductivity (µmhos/cm) Min	Conductivity (µmhos/cm) Max	Year	Temperature_Mean	pH_Mean	Conductivity_Mean
0	2548.0	HIDCO OFFICE, RAJARHAT	WEST BENGAL	26.0	30.0	7.4	7.5	253.0	1264.0	2021	28.0	7.45	758.5
1	2535.0	VISVA BHARATI	WEST BENGAL	30.0	31.0	7.9	8.2	370.0	437.0	2021	30.5	8.05	403.5
2	2546.0	ULUBERIA COLLEGE AT HOWRAH	WEST BENGAL	26.0	30.0	7.2	7.6	909.0	1090.0	2021	28.0	7.40	999.5
3	1774.0	TOPSIA CALCUTTA, WEST\nBENGAL	WEST BENGAL	29.0	31.0	7.3	7.3	1705.0	1921.0	2021	30.0	7.30	1813.0
4	1773.0	TANGRA, CALCUTTA , WEST\nBENGAL	WEST BENGAL	28.0	31.0	7.3	7.4	1435.0	1730.0	2021	29.5	7.35	1582.5

In [70]:
```python
df=df[['Station Code', 'Station Name','STATE', 'Temperature Min','Temperature Max', 'Temperature_Mean', 'pH Min','pH Max', 'pH_Mean', 'Conductivity (µmhos/cm) Min', 'Conductivity (µmhos/cm) Max', 'Conductivity_Mean','Year']]
```

In [71]:
```python
unique_state = df['STATE'].unique()
unique_state
```

Out[71]:
```
array(['WEST BENGAL', 'ANDHRA PRADESH', 'UTTARAKHAND', 'UTTAR PRADESH',
       'TRIPURA', 'TELANGANA', 'TAMIL NADU', 'RAJASTHAN', 'PUNJAB',
       'PUDUCHERRY', 'ODISHA', 'MIZORAM', 'NAGALAND', 'MEGHALAYA',
       'MAHARASHTRA', 'MANIPUR', 'MADHYA PRADESH', 'LAKSHADWEEP',
       'KERALA', 'HIMACHAL PRADESH', 'JAMMU & KASHMIR', 'JHARKHAND',
       'KARNATAKA', 'HARYANA', 'GUJARAT', 'DADRA & NAGAR HAVELI',
       'DAMAN & DIU', 'GOA', 'BIHAR', 'CHANDIGARH', 'CHHATTISGARH',
       'ASSAM', 'PONDICHERRY'], dtype=object)
```

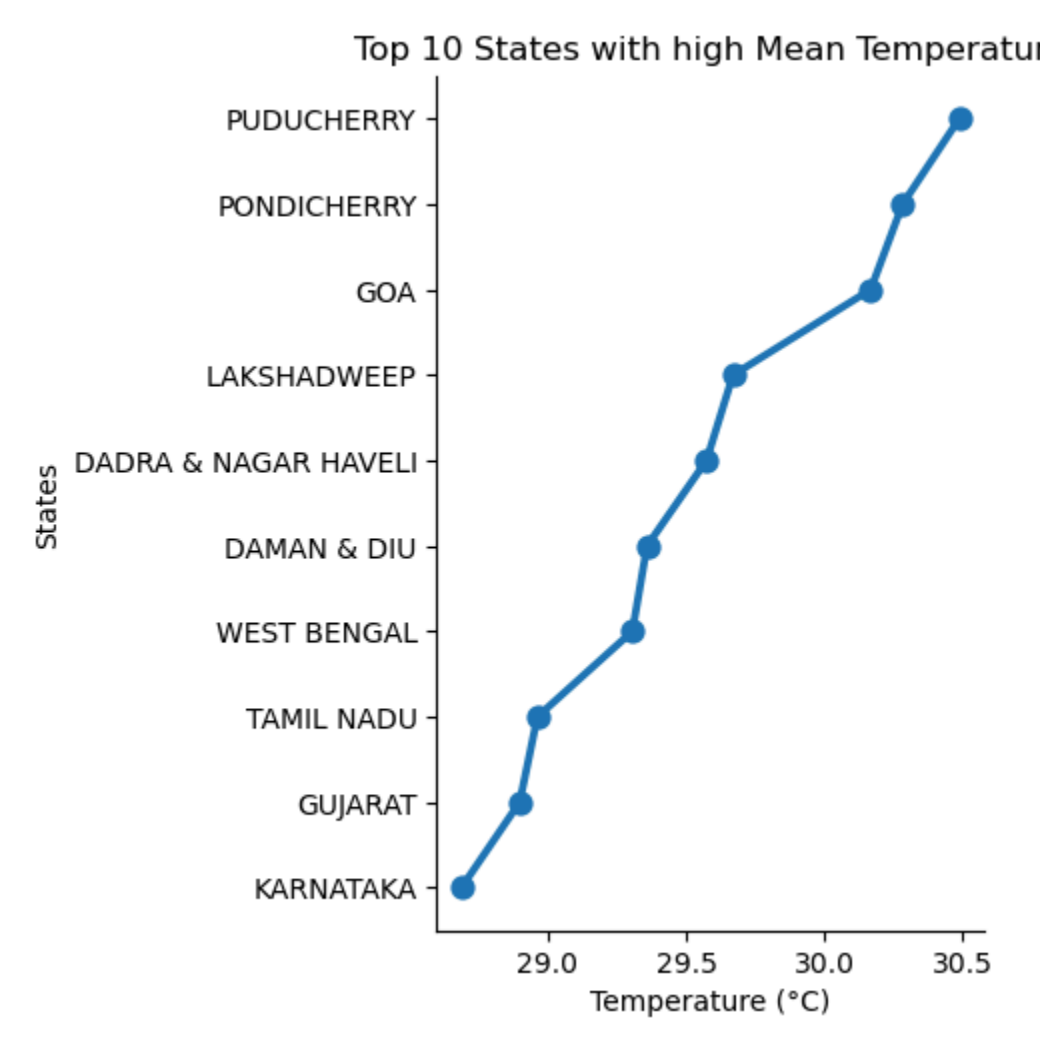
In [72]:
```python
sb.color_palette('husl',8)
plt.rcParams['figure.figsize']=50,10
```

In [73]:
```python
top_states = df.groupby('STATE')['Temperature_Mean'].mean().sort_values(ascending = False).to_frame().reset_index()
```

In [74]:
```python
top_10_states = top_states.head(10)
```

In [75]:
```python
sb.catplot(data=top_10_states, x='Temperature_Mean', y='STATE', kind='point')
plt. title( 'Top 10 States with high Mean Temperature')
plt.ylabel( 'States')
plt.xlabel('Temperature (°C)')
```

Out[75]: Text(0.5, 28.999999999999986, 'Temperature (°C)')



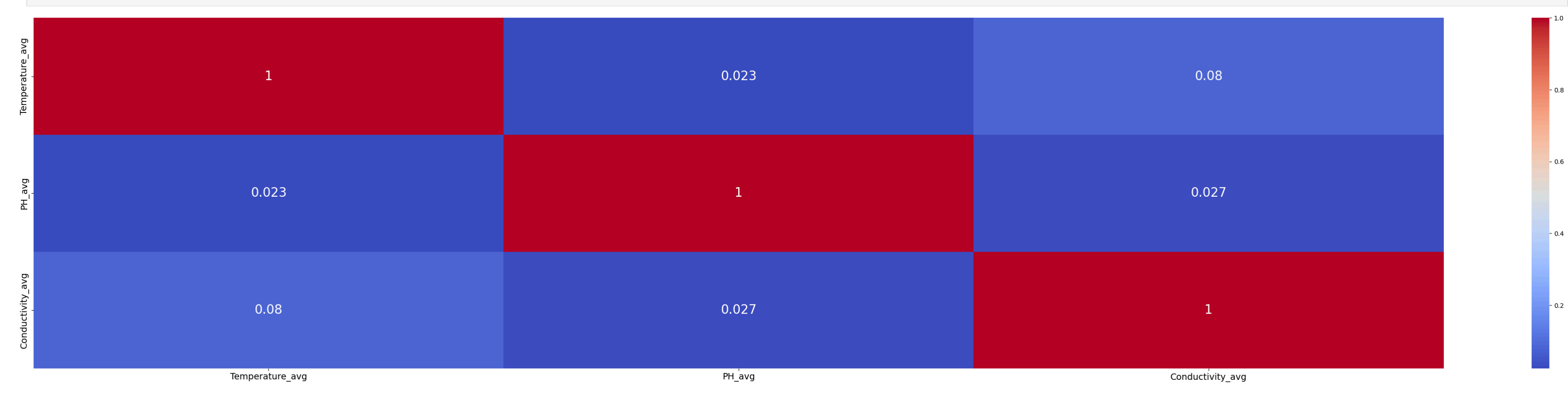
In [76]:
```python
df['Temperature_avg'] = (df['Temperature Min'] + df['Temperature Max']) / 2
df['PH_avg'] = (df['pH Min'] + df['pH Max']) / 2
df['Conductivity_avg'] = (df['Conductivity (µmhos/cm) Min'] + df['Conductivity (µmhos/cm) Max']) / 2
```

```
C:\Users\shaik\AppData\Local\Temp\ipykernel_30696\3563509002.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df['Temperature_avg'] = (df['Temperature Min'] + df['Temperature Max']) / 2
C:\Users\shaik\AppData\Local\Temp\ipykernel_30696\3563509002.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df['PH_avg'] = (df['pH Min'] + df['pH Max']) / 2
C:\Users\shaik\AppData\Local\Temp\ipykernel_30696\3563509002.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df['Conductivity_avg'] = (df['Conductivity (µmhos/cm) Min'] + df['Conductivity (µmhos/cm) Max']) / 2
```

In [77]:
```python
df['PH_range'] = df['pH Max'] - df['pH Min']
df['Temp_range'] = df['Temperature Max'] - df['Temperature Min']
```

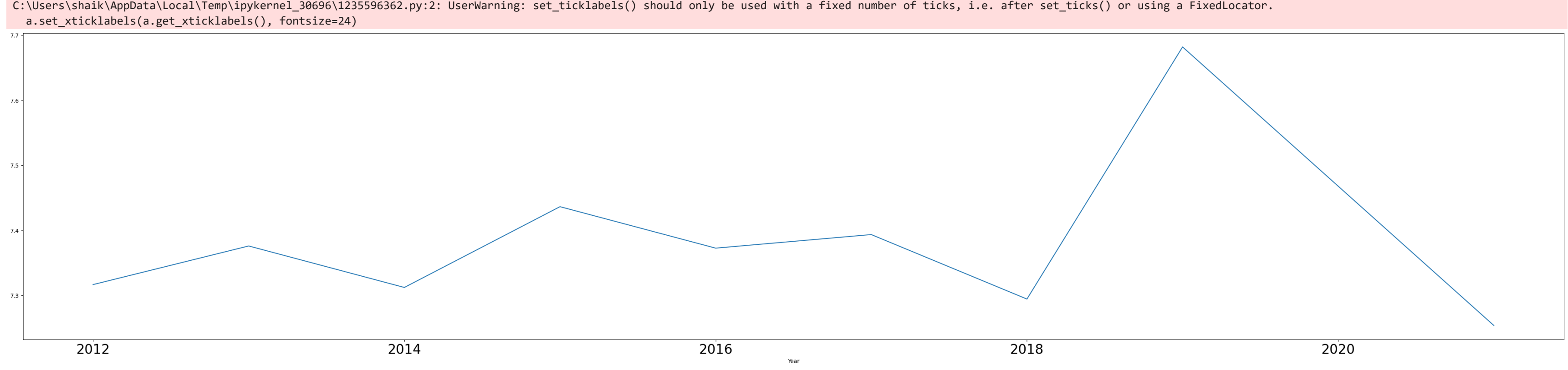
```
C:\Users\shaik\AppData\Local\Temp\ipykernel_30696\1645673603.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df['PH_range'] = df['pH Max'] - df['pH Min']
C:\Users\shaik\AppData\Local\Temp\ipykernel_30696\1645673603.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df['Temp_range'] = df['Temperature Max'] - df['Temperature Min']
```

In [78]:
```python
import seaborn as sns
import matplotlib.pyplot as plt
ax = sns.heatmap(df[['Temperature_avg','PH_avg','Conductivity_avg']].corr(),
            annot=True,
            cmap="coolwarm",
            annot_kws={"size": 20}
           )
ax.set_xticklabels(ax.get_xticklabels(), fontsize=14)
ax.set_yticklabels(ax.get_yticklabels(), fontsize=14)

plt.show()
```

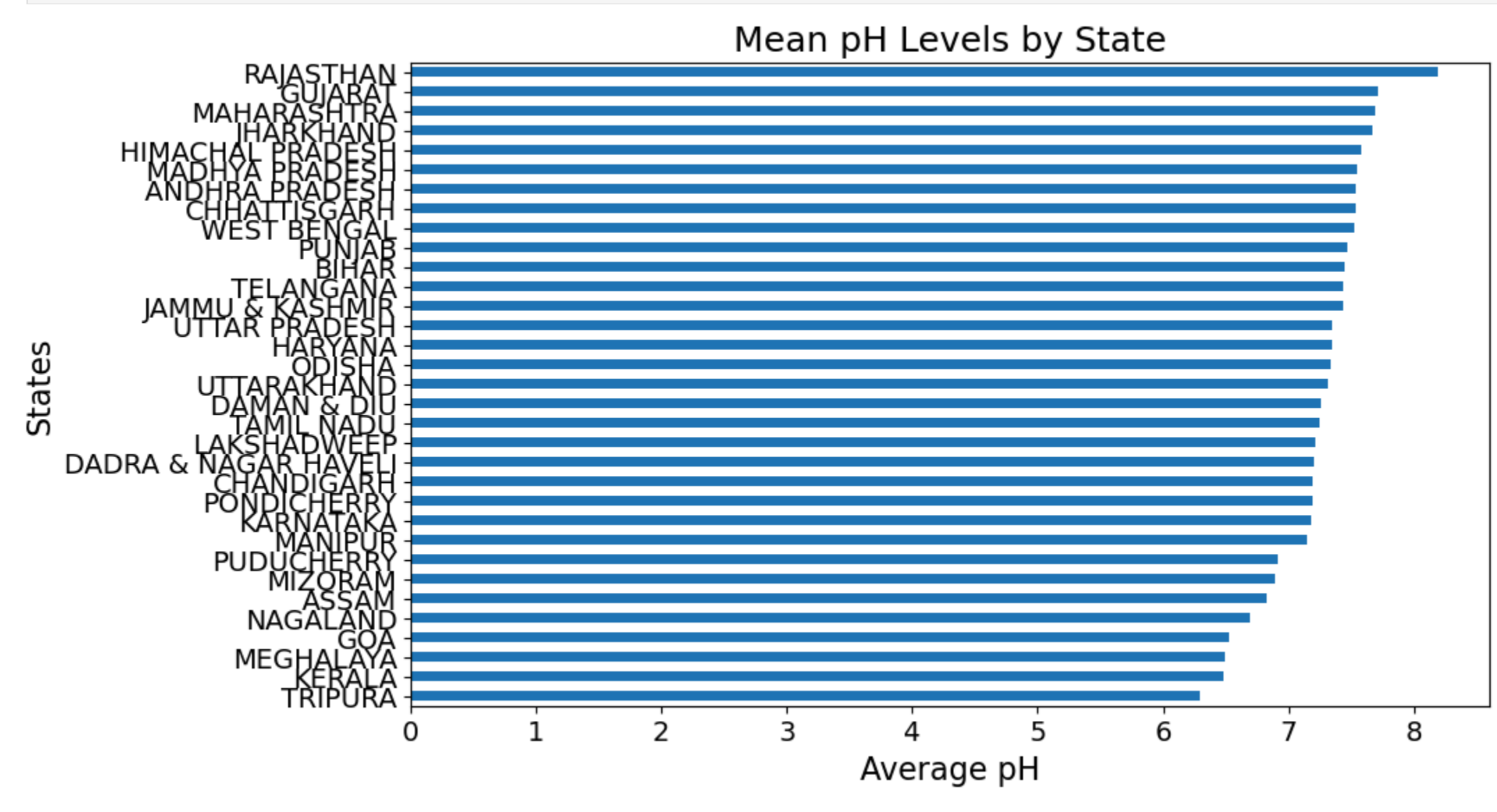


In [79]:
```python
a = df.groupby('Year')['PH_avg'].mean().plot(kind='line')
a.set_xticklabels(a.get_xticklabels(), fontsize=24)
plt.show()
```

```
C:\Users\shaik\AppData\Local\Temp\ipykernel_30696\1235596362.py:2: UserWarning: set_ticklabels() should only be used with a fixed number of ticks, i.e. after set_ticks() or using a FixedLocator.
  a.set_xticklabels(a.get_xticklabels(), fontsize=24)
```



In [80]:
```python
ax = df.groupby('STATE')['PH_avg'].mean().sort_values().plot(kind='barh', figsize=(10,6))

# Increase font size of axis values
plt.xticks(fontsize=14)
plt.yticks(fontsize=14)

# Add labels and title (optional)
plt.xlabel("Average pH", fontsize=16)
plt.ylabel("States", fontsize=16)
plt.title("Mean pH Levels by State", fontsize=18)

plt.show()
```



In [95]:
```python
print("TDS Statistics:")
print(tds["TDS"].describe())
```

```
TDS Statistics:
count    10815.000000
mean       276.682265
std        129.472495
min         50.001924
25%        163.889742
50%        276.294565
75%        390.881880
max        499.969975
Name: TDS, dtype: float64
```

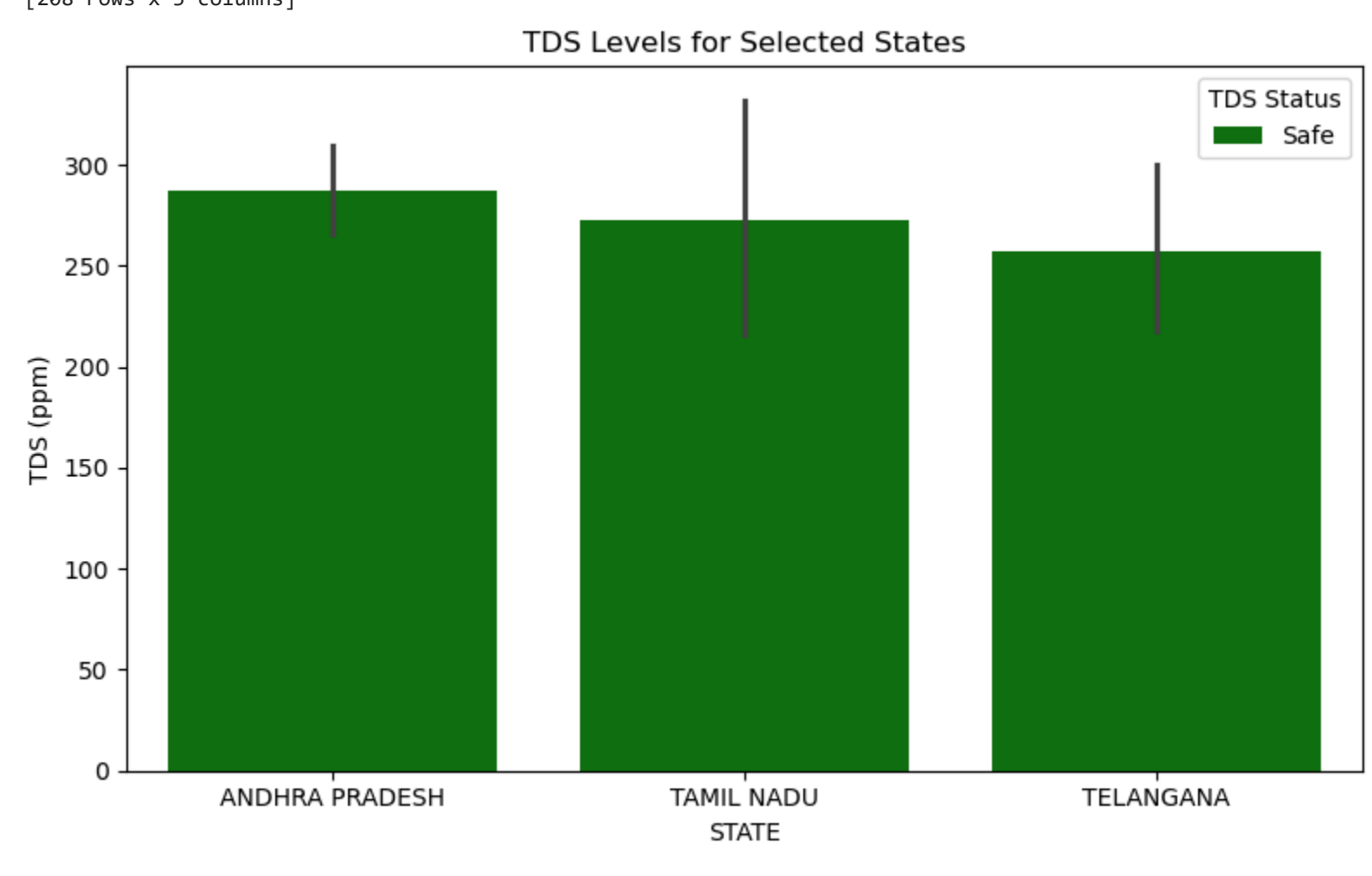
In [123…
```python
def categorize_tds(value):
    if value <= 150:
        return "Excellent"
    elif value <= 300:
        return "Good"
    elif value <= 500:
        return "Fair"
    elif value <= 1000:
        return "Poor"
    else:
        return "Unsafe"

tds["TDS_Category"] = tds["TDS"].apply(categorize_tds)
state_name = "ANDHRA PRADESH"
filtered = tds[tds["STATE"] == state_name]

#print(filtered[["STATE", "TDS", "TDS_Category"]])
filtered.head()
```

Out[123…

	Source.Name	Station Code	Station Name	STATE	Temperature Min	Temperature Max	pH Min	pH Max	Conductivity (µmhos/cm) Min	Conductivity (µmhos/cm) Max	Year	TDS	TDS_Category	TDS_Flag
0	g2.csv	15.0	WELL AT KUYYURA, A.P.	ANDHRA PRADESH	23.0	28.0	6.82	7.85	195.0	226.0	2012.0	68.870547	Excellent	Normal
1	g2.csv	16.0	WELL AT TADAVAI A.P.	ANDHRA PRADESH	27.0	28.0	7.21	7.72	663.0	1121.0	2012.0	338.572454	Fair	Normal
2	g2.csv	26.0	WELL AT VIJAYWADA, A.P.	ANDHRA PRADESH	26.0	32.0	7.19	7.39	1601.0	1661.0	2012.0	424.281766	Fair	Normal
3	g2.csv	27.0	WELL AT PEDDAVOORA, A.P.	ANDHRA PRADESH	26.0	27.0	7.82	8.80	1420.0	1668.0	2012.0	413.007667	Fair	Normal
4	g2.csv	1513.0	B W. - KRISHNA MURTHY, D.NO. 48-16-43 AUTONAGA...	ANDHRA PRADESH	25.0	30.0	6.80	7.11	1454.0	1575.0	2012.0	465.079573	Fair	Normal

In [136…
```python
tds["TDS_Status"] = tds["TDS"].apply(lambda x: "Safe" if x <= 500 else "Unsafe")

# Select specific states
selected_states = ["ANDHRA PRADESH", "TELANGANA", "TAMIL NADU"]
filtered = tds[tds["STATE"].isin(selected_states)]

# --- Display filtered data ---
print(filtered[["STATE", "TDS", "TDS_Status"]])

# --- Plot bar chart for selected states ---
plt.figure(figsize=(8,5))
sns.barplot(
    x="STATE",
    y="TDS",
    data=filtered,
    hue="TDS_Status",
    dodge=False,
    palette={"Safe":"green","Unsafe":"red"}
)

plt.title("TDS Levels for Selected States")
plt.ylabel("TDS (ppm)")
plt.xlabel("STATE")
plt.legend(title="TDS Status")
plt.tight_layout()
plt.show()
```

```
                STATE         TDS TDS_Status
0      ANDHRA PRADESH   68.870547       Safe
1      ANDHRA PRADESH  338.572454       Safe
2      ANDHRA PRADESH  424.281766       Safe
3      ANDHRA PRADESH  413.007667       Safe
4      ANDHRA PRADESH  465.079573       Safe
...               ...         ...        ...
10427       TELANGANA  130.086373       Safe
10467       TELANGANA   60.233453       Safe
10468       TELANGANA  149.078521       Safe
10475       TELANGANA  427.334757       Safe
10483       TELANGANA  425.398434       Safe

[208 rows x 3 columns]
```



In [ ]:

In [ ]:

In [106…