

20 problem statements using a **Weather Dataset**, along with the corresponding **NumPy** and **Pandas**.

1. **What is the average maximum temperature across all locations?**
`df['MaxTemp'].mean()`
2. **What is the standard deviation of rainfall in Sydney?**
`df[df['Location'] == 'Sydney']['Rainfall'].std()`
3. **Which day had the highest wind gust speed recorded?**
`df.loc[df['WindGustSpeed'].idxmax()]`
4. **What is the total number of rainy days in Melbourne?**
`df[(df['Location'] == 'Melbourne') & (df['RainToday'] == 'Yes')].shape[0]`
5. **Find the correlation between MinTemp and MaxTemp.**
`df['MinTemp'].corr(df['MaxTemp'])`
6. **How many missing values are in each column?**
`df.isnull().sum()`
7. **What is the median humidity at 3pm across all cities?**
`df['Humidity3pm'].median()`
8. **Which location had the most sunshine hours on average?**
`df.groupby('Location')['Sunshine'].mean().idxmax()`
9. **Compare average temperature at 9am and 3pm.**
`df[['Temp9am', 'Temp3pm']].mean()`
10. **Which day had the largest temperature difference (MaxTemp - MinTemp)?**
`df['TempDiff'] = df['MaxTemp'] - df['MinTemp'];
df.loc[df['TempDiff'].idxmax()]`
11. **How does wind speed at 9am compare with 3pm on rainy vs. non-rainy days?**
`df.groupby('RainToday')[['WindSpeed9am', 'WindSpeed3pm']].mean()`
12. **Filter out days with temperature above 40°C.**
`df[df['MaxTemp'] > 40]`
13. **Calculate the rolling average of MaxTemp over a 7-day window.**
`df['MaxTemp'].rolling(window=7).mean()`
14. **Get the distribution (histogram) of rainfall amounts.**
`df['Rainfall'].hist(bins=20)`
15. **Which month has the highest average rainfall?**
`df['Month'] = pd.to_datetime(df['Date']).dt.month;
df.groupby('Month')['Rainfall'].mean().idxmax()`
16. **What is the average pressure difference between 9am and 3pm?**
`(df['Pressure3pm'] - df['Pressure9am']).mean()`
17. **Which location has the most frequent "Yes" for RainTomorrow?**
`df[df['RainTomorrow'] == 'Yes']['Location'].value_counts().idxmax()`
18. **Find outliers in MaxTemp using NumPy (Z-score method).**

```
python
CopyEdit
from scipy.stats import zscore
df[np.abs(zscore(df['MaxTemp'].dropna())) > 3]
```

19. **Create a pivot table showing average temperature by location and month.**

```
python
CopyEdit
```

```
df.pivot_table(values='MaxTemp', index='Location', columns='Month',  
aggfunc='mean')
```

20. Classify days as 'Hot', 'Warm', or 'Cool' based on MaxTemp using NumPy `select`.

```
python  
CopyEdit  
conditions = [  
    df['MaxTemp'] > 35,  
    df['MaxTemp'].between(25, 35),  
    df['MaxTemp'] < 25  
]  
choices = ['Hot', 'Warm', 'Cool']  
df['TempCategory'] = np.select(conditions, choices)
```