Exercise 1

In [ ]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
import math
```

In [ ]:
```
def generate_samples(M, N):
    samples = np.random.choice([-1, 1], size=(M, N))
    X_N_samples = np.sum(samples, axis=1) / np.sqrt(N)
    return X_N_samples
```

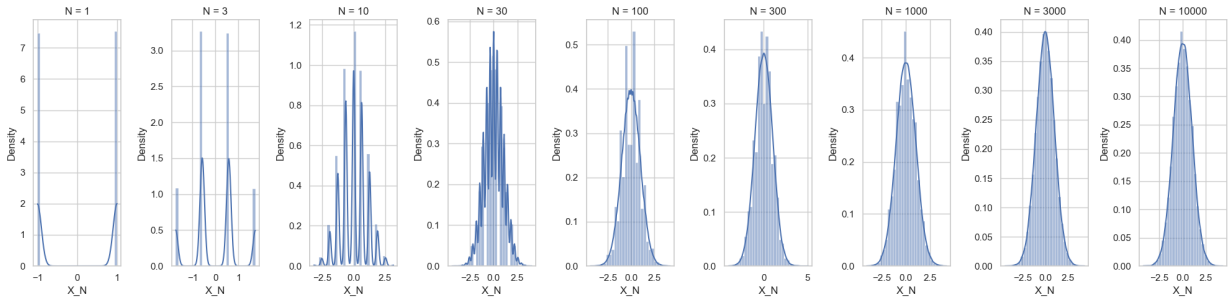
Added 300, 1000, 3000, 10000 to the value of ***N*** for better visualisation

In [ ]:
```
M = 100000
N_values = [1, 3, 10, 30, 100, 300, 1000, 3000, 10000]

fig, axes = plt.subplots(1, len(N_values), figsize=(20, 5))
sns.set(style='whitegrid')

for i, N in enumerate(N_values):
    samples = generate_samples(M, N)
    sns.histplot(samples, kde=True, ax=axes[i], stat="density", bins=30)
    axes[i].set_title(f'N = {N}')
    axes[i].set_xlabel('X_N')
    axes[i].set_ylabel('Density')

plt.tight_layout()
plt.show()
```



We can see as the value of **N → ∞** the the historgram becomes more like a gaussian distribution hence central limit theorm holds

Exercise 2

In [ ]:
```
df = pd.read_csv('temperature_data_processed.csv')
```

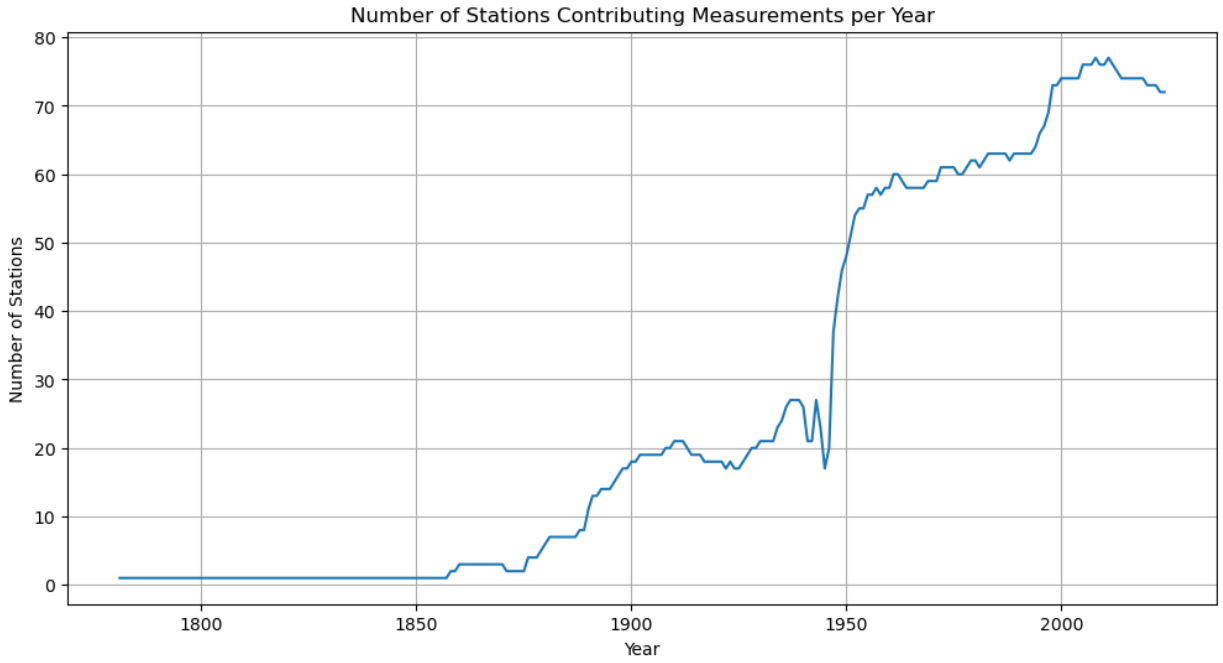
In [ ]:
```
df['date'] = df['date'].astype(str)

df['year'] = df['date'].str[:4].astype(int)
df['month'] = df['date'].str[4:6].astype(int)
df['day'] = df['date'].str[6:8].astype(int)

df_cleaned = df[df['temp'] != -999]

station_count_per_year = df_cleaned.groupby('year')['stationid'].nunique(

station_count_per_year_plot = station_count_per_year.plot(kind='line', ti
plt.grid(True)
plt.show()
```



In [ ]:
```
stations_1900 = set(df_cleaned[df_cleaned['year'] == 1900]['stationid'])
stations_2020 = set(df_cleaned[df_cleaned['year'] == 2020]['stationid'])

reference_stations = stations_1900 & stations_2020
```

Out[ ]:
```
{691, 722, 880, 1358, 1684, 1757, 2290, 3126, 3631, 3987, 4104, 4625, 579
2}
```

In [ ]:
```
filtered_df = df_cleaned[(df_cleaned['year'] > 1900) & (df_cleaned['year'
            & (df_cleaned['month'].isin([6, 7, 8]))
            & (df_cleaned['time'].isin([12,14])))]
```

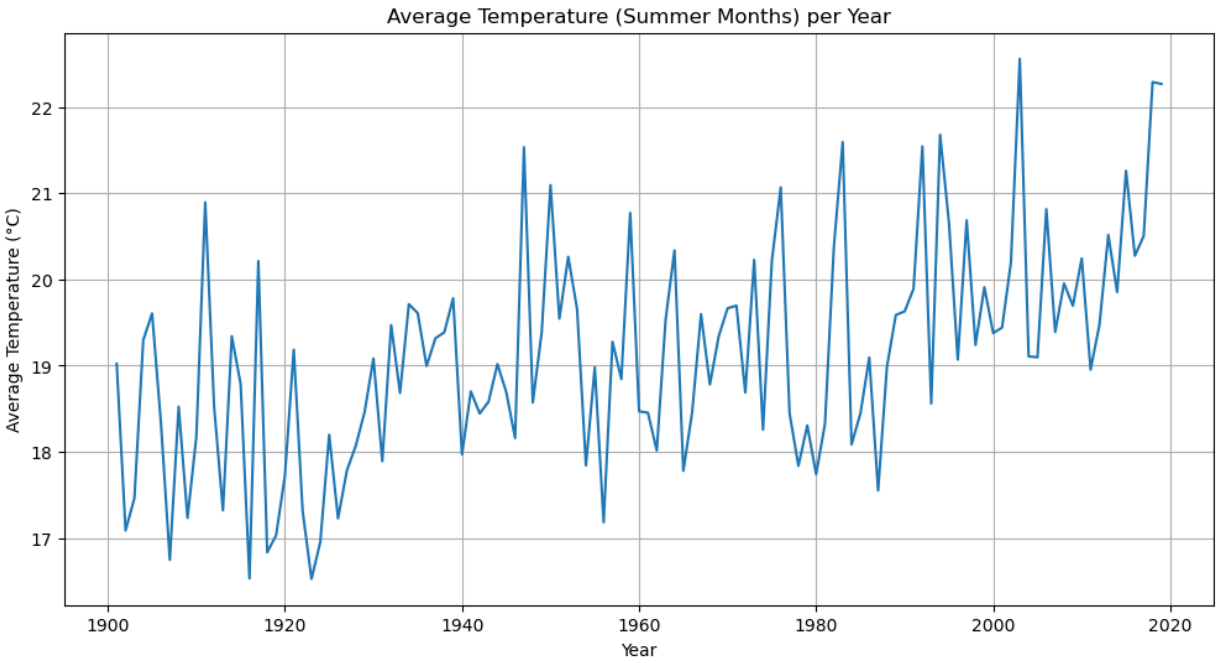
In [ ]:
```
import numpy as np
#Group by year and calculate the average temperature

avg_temp_per_year = filtered_df.groupby('year')['temp'].mean()

# Plot the result
avg_temp_per_year_plot = avg_temp_per_year.plot(kind='line', title='Avera

plt.grid(True)
plt.show()
```

```
[[0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]]
```



In [ ]: