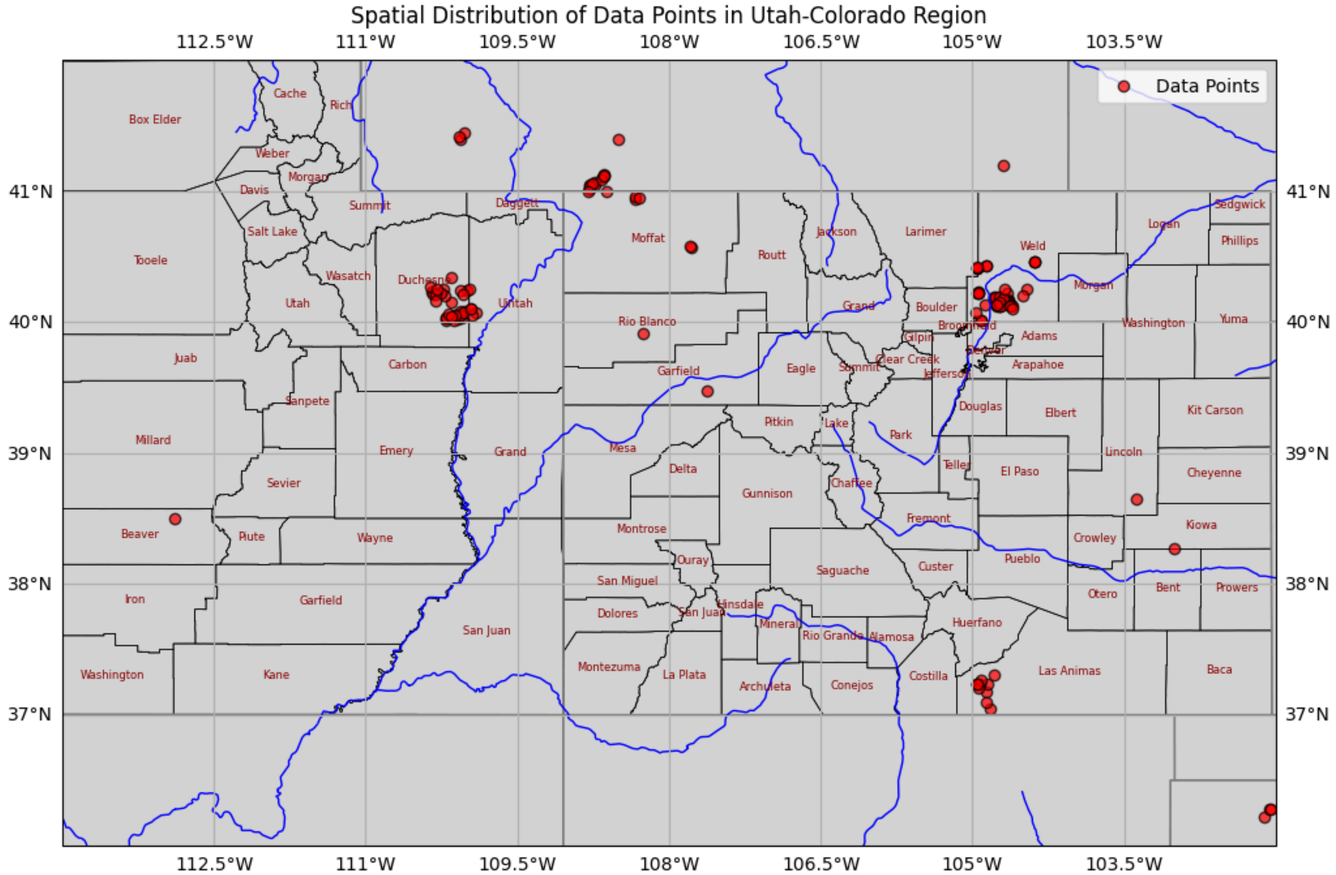
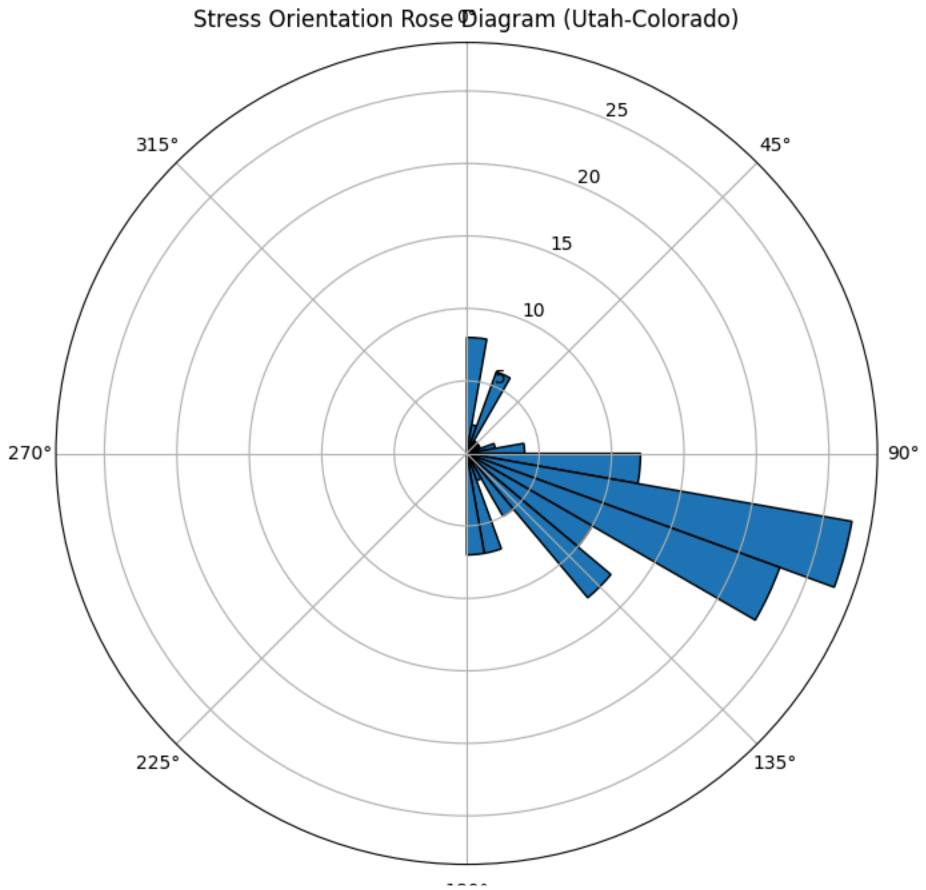
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import cartopy.crs as ccrs
import cartopy.feature as cfeature
from cartopy.io.shapereader import natural_earth, Reader
from scipy.stats import shapiro, ttest_ind, pearsonr, spearmanr
import geopandas as gpd

# Load the dataset
url = 'https://drive.google.com/uc?export=download&id=1Tgs3zrC9aOWm4Fkattsr0D4tndGHkGK-'
data = pd.read_csv(url)
print(data.head())

# Data Wrangling
shmax_or1_data = data['SHMAX_OR1'].dropna()
orientations_rad = np.radians(shmax_or1_data)

# Extract latitude and longitude data
latitudes = data['LAT_SURF']
longitudes = data['LON_SURF']

# Load state boundaries
shapefile = natural_earth(resolution='110m', category='cultural', name='admin_1_states_provinces')
states = Reader(shapefile)

# Load county boundaries from a shapefile (US counties dataset)
counties = gpd.read_file('https://www2.census.gov/geo/tiger/TIGER2021/COUNTY/tl_2021_us_county.zip')

# Filter counties for Utah and Colorado
counties = counties[counties['STATEFP'].isin(['08', '49'])]  # 08 is Colorado, 49 is Utah

# Filter data for Utah-Colorado region
utah_colorado_data = data[
    ((data['LON_SURF'] >= -114) & (data['LON_SURF'] <= -102)) &
    ((data['LAT_SURF'] >= 36) & (data['LAT_SURF'] <= 42))
]

# Spatial Distribution Plot
plt.figure(figsize=(12, 10))
ax = plt.axes(projection=ccrs.PlateCarree())
ax.add_feature(cfeature.LAND, edgecolor='black', facecolor='lightgray')
ax.add_feature(cfeature.STATES, edgecolor='gray')
ax.add_feature(cfeature.BORDERS, linestyle=':', edgecolor='black')
ax.add_feature(cfeature.RIVERS, edgecolor='blue')
ax.set_extent([-114, -102, 36, 42], crs=ccrs.PlateCarree())

# Plot county borders
counties.plot(ax=ax, edgecolor='black', facecolor='none', linewidth=0.5)

# Add county names
for _, row in counties.iterrows():
    centroid = row.geometry.centroid
    ax.text(centroid.x, centroid.y, row['NAME'], transform=ccrs.PlateCarree(), fontsize=6, ha='center', color='darkred')

plt.scatter(utah_colorado_data['LON_SURF'], utah_colorado_data['LAT_SURF'], c='red', alpha=0.7, edgecolors='k', label='Data Points')
ax.gridlines(draw_labels=True)
plt.title('Spatial Distribution of Data Points in Utah-Colorado Region')
plt.legend()
plt.show()


# Rose Diagram
fig, ax = plt.subplots(subplot_kw={'projection': 'polar'}, figsize=(8, 8))
n_bins = 36
bin_edges = np.linspace(0, 2 * np.pi, n_bins + 1)
hist, _ = np.histogram(np.radians(utah_colorado_data['SHMAX_OR1'].dropna()), bins=bin_edges)
ax.bar(bin_edges[:-1], hist, width=2 * np.pi / n_bins, edgecolor='k', align='edge')
ax.set_theta_zero_location('N')
ax.set_theta_direction(-1)
ax.set_title('Stress Orientation Rose Diagram (Utah-Colorado)', va='bottom')
plt.show()

# Depth-Dependent Stress Regimes
shmax_depth_data_uc = utah_colorado_data[['SHMAX_OR1', 'OR1_AV_DEPTH_M']].dropna()
shmax_depth_data_uc = shmax_depth_data_uc.sort_values(by='OR1_AV_DEPTH_M')
shmax_or1_uc = shmax_depth_data_uc['SHMAX_OR1']
depth_uc = shmax_depth_data_uc['OR1_AV_DEPTH_M'].astype(float)

plt.figure(figsize=(10, 6))
plt.scatter(depth_uc, shmax_or1_uc, alpha=0.7, edgecolors='k', label='Data Points')
```

```
plt.title('Shmax Orientation vs Average Depth (Utah-Colorado)')
plt.xlabel('Average Depth (m)')
plt.ylabel('Shmax Orientation (degrees)')
plt.xticks(np.arange(950, 3851, 50, dtype=int), rotation=90, fontsize=8)
plt.legend()
plt.show()

# Histogram Plot
plt.figure(figsize=(10, 6))
plt.hist(shmax_or1_uc, bins=20, color='blue', edgecolor='k', alpha=0.7)
plt.title('Distribution of Shmax Orientation (Utah-Colorado)')
plt.xlabel('Shmax Orientation (degrees)')
plt.ylabel('Frequency')
plt.grid(True)
plt.show()

# Statistical Analyses
shapiro_test = shapiro(shmax_or1_uc)
print(f"Shapiro-Wilk Test for SHMAX_OR1: Statistic = {shapiro_test.statistic:.4f}, p-value = {shapiro_test.pvalue:.4e}")

corr_pearson, p_pearson = pearsonr(shmax_or1_uc, depth_uc)
corr_spearman, p_spearman = spearmanr(shmax_or1_uc, depth_uc)
print(f"Pearson Correlation: Correlation = {corr_pearson:.2f}, p-value = {p_pearson:.4e}")
print(f"Spearman Correlation: Correlation = {corr_spearman:.2f}, p-value = {p_spearman:.4e}")

median_depth = depth_uc.median()
upper_depth = shmax_or1_uc[depth_uc > median_depth]
lower_depth = shmax_or1_uc[depth_uc <= median_depth]
t_stat, t_pvalue = ttest_ind(upper_depth, lower_depth, equal_var=False)
print(f"Two-Sample t-test: t-statistic = {t_stat:.2f}, p-value = {t_pvalue:.4e}")
```

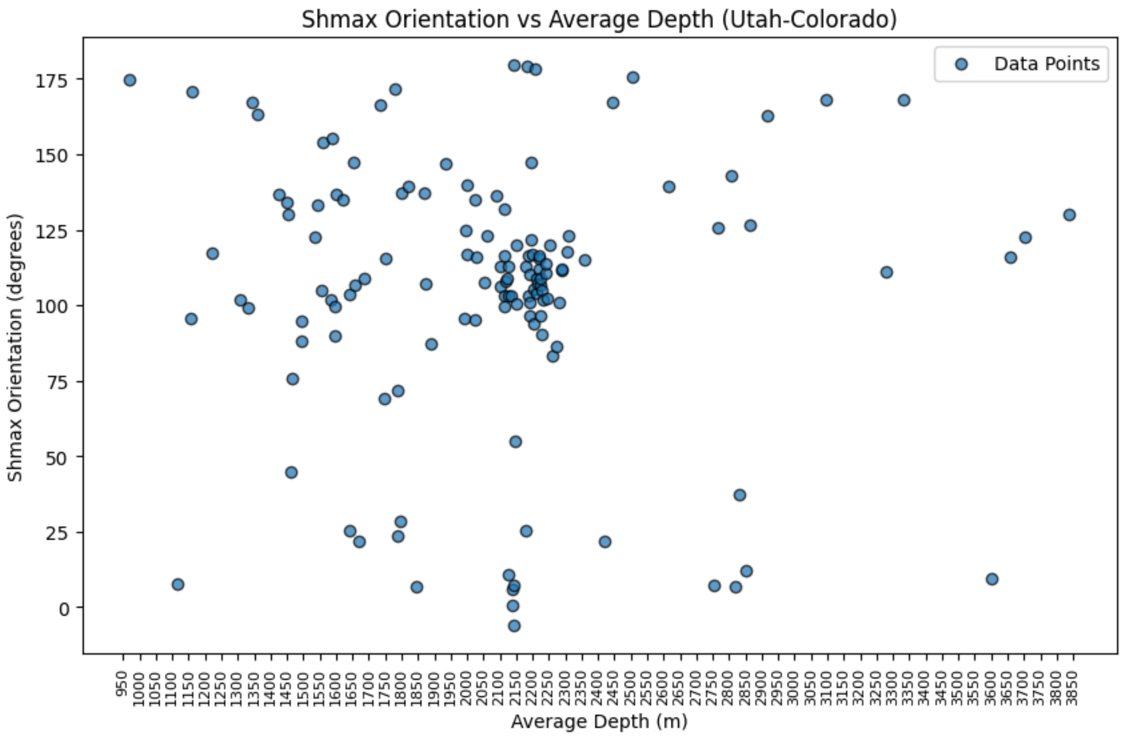
	ID	LAT_SURF	LON_SURF	SHMAX_OR1	OR1_SD	SHMIN_OR1	OR1_AV_DEPTH_M	\
0	A427	33.20	-97.20	196.000000	NaN	286.000000	NaN	
1	A428	33.20	-97.20	194.000000	NaN	284.000000	NaN	
2	A16	32.82	-97.94	47.185900	17.8099	137.185900	1316.12	
3	A108	32.67	-97.24	39.319700	5.9346	129.319700	2203.34	
4	A111	33.61	-98.03	33.111055	3.3339	123.111056	1986.89	

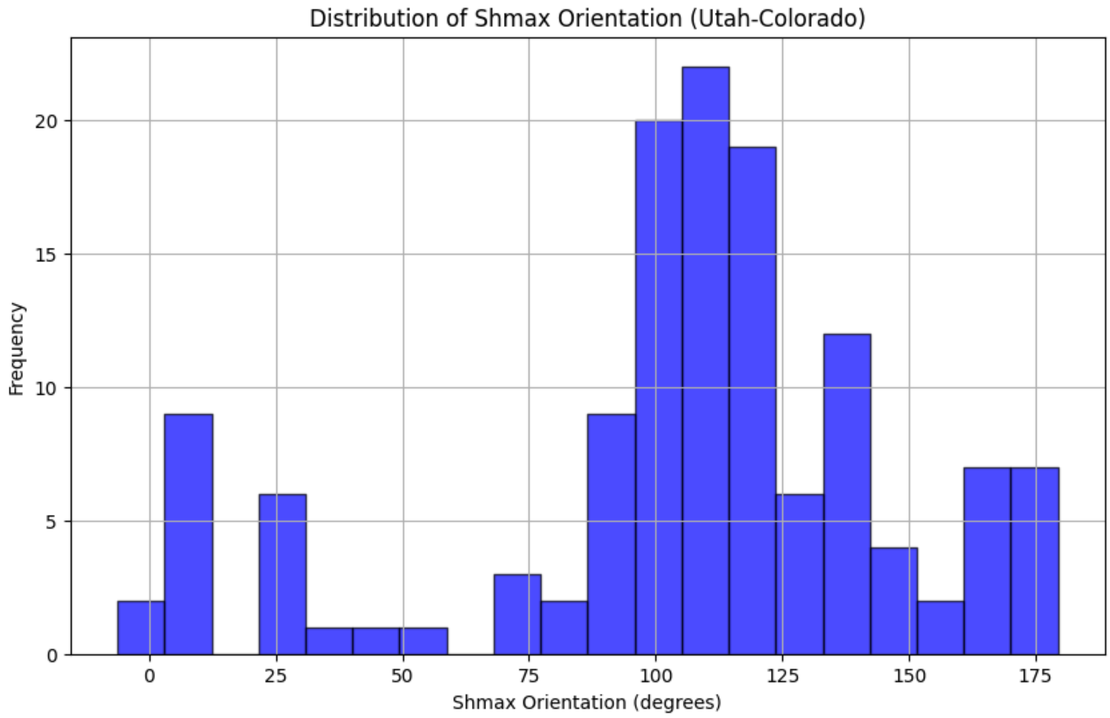
	OR1_QUAL	OR1_TYPE	OR1_METHOD	OR1_NUM_OBS	OR1_DATE_OBT	OR1_TOP_M	OR1_BOT_M
0	A	FMF	FMF	247	20170217	NaN	NaN
1	A	FMF	FMF	398	20170217	NaN	NaN
2	B	BO	FMI	7	20170818	1275.9	1492.6
3	A	DIF	FMI	72	20151003	1990.8	2330.2
4	A	DIF	FMI	36	20151003	1641.8	2086.0





180°






```
Shapiro-Wilk Test for SHMAX_OR1: Statistic = 0.9004, p-value = 6.1596e-08
Pearson Correlation: Correlation = -0.05, p-value = 5.4610e-01
Spearman Correlation: Correlation = -0.01, p-value = 8.6669e-01
Two-Sample t-test: t-statistic = -0.91, p-value = 3.6412e-01
```