# Solar Zenith Angle vs Altitude Analysis

## 概要

このプロジェクトは、太陽天頂角（Solar Zenith Angle, SZA）と高度の関係を統計的に解析するPythonスクリプトです。観測データにエラーバーを含む散布図の作成、加重最小二乗法（WLS）による回帰分析、および予測区間の計算を行います。

## ファイル構成

```

├── insitu\_sza\_altitude.py # メインの解析スクリプト

└── wpd\_datasets.csv # 観測データ（SZAと高度のペア）

```

## データセット形式

### `wpd\_datasets.csv`

- \*\*形式\*\*: CSV形式

- \*\*構造\*\*: 2列ペアの形式（SZA, 高度）が複数組

- \*\*行数\*\*: 3行（各測定点に対して3つの観測値）

- \*\*列\*\*: 偶数列がSZA値、奇数列が高度値

```

SZA1, Alt1, SZA2, Alt2, SZA3, Alt3, ...

値1, 値1, 値2, 値2, 値3, 値3, ...

値1, 値1, 値2, 値2, 値3, 値3, ...

値1, 値1, 値2, 値2, 値3, 値3, ...

```

## 機能

### 1. データ処理

- CSVファイルからSZAと高度データを読み込み

- 各測定点でのSZA値の平均計算

- ピーク高度の観測値、観測誤差の上限・下限の抽出

### 2. データ可視化

- エラーバー付き散布図の作成

- 回帰直線の重ね合わせ表示

- カスタマイズ可能なグラフスタイル

### 3. 統計解析

- \*\*加重最小二乗法（WLS）回帰\*\*

- エラーバーの大きさを重みとして使用

- より正確な回帰係数の推定

- \*\*予測区間の計算\*\*

- 任意のSZA値での高度予測

- 95%信頼区間の算出

## 必要なライブラリ

```python

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import statsmodels.api as sm

```

## インストール

```bash

conda install numpy pandas matplotlib statsmodels

# または

pip install numpy pandas matplotlib statsmodels

```

## 使用方法

### 基本的な実行

```bash

python insitu\_sza\_altitude.py

```

### カスタマイズ

1. \*\*データファイルの変更\*\*

```python

csx\_data\_path = "/path/to/your/data.csv"

```

2. \*\*グラフの範囲調整\*\*

```python

plt.xlim(0, 180) # SZA範囲

plt.ylim(900, 1400) # 高度範囲

```

3. \*\*予測するSZA値の変更\*\*

```python

sza\_values = [50, 70, 90, 110, 130] # 任意の値に変更可能

```

## 出力結果

### 1. グラフ出力

- \*\*エラーバー付き散布図\*\*: 観測データと不確実性を表示

- \*\*回帰直線付きグラフ\*\*: 統計的関係を視覚化

### 2. 統計結果

```

回帰係数（切片, 傾き）

95%信頼区間

予測値と予測区間

```

### 3. 予測結果例

```

mean obs\_ci\_lower obs\_ci\_upper

0 1150.23 1080.45 1220.01

1 1134.56 1064.78 1204.34

...

```

## アルゴリズム詳細

### 加重最小二乗法（WLS）

1. \*\*重みの計算\*\*: `w = 1 / σ²`

- σ = (上限 - 下限) / 2

2. \*\*回帰式\*\*: `高度 = α + β × SZA`

3. \*\*信頼区間\*\*: 統計モデルベースの区間推定

### データ処理フロー

```

CSV読み込み → データ抽出 → 平均計算 → 回帰分析 → 可視化 → 予測

```

## 注意事項

1. \*\*データ形式\*\*: CSVファイルは指定された形式に従う必要があります

2. \*\*欠損値\*\*: 数値に変換できないデータはエラーの原因となります

3. \*\*重み\*\*: エラーバーが0の場合、重みが無限大になるため注意が必要です

## トラブルシューティング

### よくあるエラー

1. \*\*ValueError: could not convert string to float\*\*

- \*\*原因\*\*: データが文字列として読み込まれている

- \*\*解決\*\*: `pd.to\_numeric()`を使用してデータを数値変換

2. \*\*ValueError: shapes not aligned\*\*

- \*\*原因\*\*: 予測時の配列の次元が不一致

- \*\*解決\*\*: `np.column\_stack()`で正しい形状の配列を作成

3. \*\*エラーバーの表示異常\*\*

- \*\*原因\*\*: 上限・下限の関係が正しくない

- \*\*解決\*\*: データの順序と意味を確認

## 拡張可能性

- \*\*多項式回帰\*\*: より複雑な関係のモデリング

- \*\*異なる重み付け\*\*: 他の重み付けスキームの実装

- \*\*クロスバリデーション\*\*: モデルの汎化性能評価

- \*\*他の統計手法\*\*: ロバスト回帰、ベイズ回帰など

## ライセンス

このプロジェクトはMITライセンスの下で公開されています。

## 更新履歴

- v1.0: 初版リリース

- 基本的なWLS回帰分析機能

- エラーバー付き可視化

- 予測区間計算

---

## 連絡先

質問があったら安田まで fielder.yasuda514@gmail.com