

# ValueScope - 企業価値分析ダッシュボード

バージョン: 1.0.0 リリース日: 2025-12-15 公開URL: <https://1921604.github.io/ValueScope/> GitHubリポジトリ: <https://github.com/1921604/ValueScope> ドキュメント: [constitution.md](#) | [spec.md](#) | [plan.md](#)

 Deploy to GitHub Pages passing

## 目次

- [概要](#)
- [主要機能](#)
- [技術スタック](#)
- [連携ツール](#)
- [使用方法](#)
- [データ更新](#)
- [テスト](#)
- [デプロイ](#)
- [プロジェクト構造](#)
- [ドキュメント](#)
- [ライセンス](#)

## 概要

ValueScopeは、東京電力HD・中部電力・JERAの企業価値指標（EV、EV/EBITDA、PER、PBR）と電力業界特化KPI（ROIC、WACC、EBITDAマージン、FCFマージン）を可視化し、信号機方式（緑/黄/赤）で財務健全性を評価する企業価値分析ダッシュボードです。

EDINET XBRL実データのみを使用し、推定値・補完値を一切使用しない高品質な財務分析を提供します。

## 主要機能

☒ 企業価値指標の可視化

損益計算書（PL）項目:

- 売上高（営業収益）: jpcrp\_cor.OperatingRevenue
- 営業利益: jpcrp\_cor.OperatingIncome
- 経常利益: jpcrp\_cor.OrdinaryIncome
- 当期純利益: jpcrp\_cor.ProfitLoss
- 親会社株主に帰属する当期純利益: jpcrp\_cor.ProfitLossAttributableToOwnersOfParent

貸借対照表（BS）項目:

- 総資産: jpcrp\_cor.Assets
- 純資産: jpcrp\_cor.NetAssets
- 自己資本: jpcrp\_cor.Equity
- 有利子負債: jpcrp\_cor.InterestBearingDebt

キャッシュフロー計算書（CF）項目:

- 営業活動によるキャッシュフロー: jpcrp\_cor.CashFlowsFromOperatingActivities
- 投資活動によるキャッシュフロー: jpcrp\_cor.CashFlowsFromInvestingActivities
- 財務活動によるキャッシュフロー: jpcrp\_cor.CashFlowsFromFinancingActivities

計算指標:

- EBITDA: 営業利益 + 減価償却費
- ROIC: NOPAT ÷ 投下資本 × 100

企業価値（EV）関連指標:

- 時価総額: 決算日株価 × 発行済株式数
- 純有利子負債: 有利子負債合計 - 現金及び現金同等物
- 企業価値（EV）: 時価総額 + 純有利子負債
- EV/EBITDA倍率: 企業価値 / EBITDA
- PER（株価収益率）: 時価総額 / 当期純利益
- PBR（株価純資産倍率）: 時価総額 / 自己資本

☒ 電力業界特化KPIスコアード

KPI	緑 (green)	黄 (yellow)	赤 (red)	max値
ROIC 投下資本利益率	≥5%	3%-5%	<3%	15%
WACC 加重平均資本コスト	<4%	4%-5%	≥5%	6%
EBITDAマージン 営業キャッシュ創出力	≥15%	10%-15%	<10%	30%
FCFマージン フリーキャッシュフロー効率	≥5%	0%-5%	<0%	25%

☒ 推移グラフ（過去10年分）

- KPI指標の時系列チャート（Recharts使用）
- 3社比較グラフ
- マウスオーバーで詳細数値表示

☒ 従業員情報

- 平均年間給与、平均勤続年数、平均年齢、従業員数の比較テーブル
- 過去10年分の推移グラフ

☒ 財務諸表詳細

- 損益計算書（PL）: 売上高、営業利益、当期純利益など（単位: 億円）
- 貸借対照表（BS）: 総資産、自己資本、有利子負債など（単位: 億円）
- キャッシュフロー計算書（CF）: 営業CF、投資CF、財務CFなど（単位: 億円）

## 技術スタック

### フロントエンド

- UIライブラリ: React 18.2.0
- 言語: TypeScript 5.3.3
- ビルドツール: Vite 5.0.8
- チャートライブラリ: Recharts 2.10.3
- CSSフレームワーク: Tailwind CSS 3.4.1

### バックエンド（データ処理）

- 言語: Python 3.10.11
- データ処理: pandas 2.1.4
- XBRL解析: lxml 5.2.1
- 株価取得: pandas\_datareader 0.10.0（Stooq API経由）

### テスト

- ユニットテスト: Vitest 1.1.0（カバレッジ80%以上）
- E2Eテスト: Playwright 1.40.1（主要フロー100%）

### デプロイ

- ホスティング: GitHub Pages（CDN配信）
- CI/CD: GitHub Actions（自動ビルド・デプロイ）

## 環境構築

### 前提条件

- Node.js 20.x以上
- npm 10.x以上
- Python 3.10.11
- Git

### 1. リポジトリクローン

```
git clone https://github.com/1921604/ValueScope.git
cd ValueScope
```

### 2. 依存関係インストール

Node.js依存関係:

```
npm install
```

Python依存関係:

```
pip install -r scripts/requirements.txt
```

### 3. データ生成（初回のみ）

```
# EDINET APIからXBRLダウンロード（過去10年分）
py -3.10 scripts/fetch_edinet.py --years 10

# XBRL解析 → JSON/CSV変換
py -3.10 scripts/parse_edinet_xbrl.py
py -3.10 scripts/extract_xbrl_to_csv.py

# 企業価値・KPI計算
py -3.10 scripts/build_timeseries.py
py -3.10 scripts/build_valuation.py
py -3.10 scripts/compute_scores.py
```

### 4. 開発サーバー起動

方法A: ワンコマンド起動（Windows）

```
start .\start.ps1
```

自動的にブラウザで <http://localhost:5173/ValueScope/> が開きます。

方法B: 手動起動

```
npm run dev
```

ブラウザで <http://localhost:5173/ValueScope/> を開く。

## 使用方法

### ダッシュボード操作

- 企業価値分析タブ: EV、EV/EBITDA、PER、PBR、時価総額、純有利子負債を一覧表示
- KPI分析タブ: ROIC、WACC、EBITDAマージン、FCFマージンを信号機評価（緑/黄/赤）
- 従業員情報タブ: 平均年間給与、平均勤続年数、平均年齢、従業員数の比較と推移
- 財務諸表タブ: PL/BS/CFの詳細項目を日本語ラベルで表示

### 年度切り替え

画面上部の年度選択ボタンで過去10年分のデータに切り替え可能。

## データ更新

EDINET XBRL更新（年1回: 6/20-7/1）

GitHub Actionsで毎年6月20日から7月1日の期間のみ自動実行されます。EDINET APIキーはGitHub Secretsで管理。

手動更新:

```
py -3.10 scripts/fetch_edinet.py --years 10
py -3.10 scripts/parse_edinet_xbrl.py
py -3.10 scripts/extract_xbrl_to_csv.py
```

### 株価データ更新（毎回デプロイ時）

対象銘柄:

- 9501.T 東京電力HD
- 9502.T 中部電力

手動更新:

```
py -3.10 scripts/fetch_stock_prices.py
```

Stooq APIの利点:

- 無料・無制限アクセス
- 安定した稼働率
- pandas\_datareaderとの統合

### データ再計算

```
# 時系列データ生成（14項目追加対応）
py -3.10 scripts/build_timeseries.py

# 企業価値計算
py -3.10 scripts/build_valuation.py

# KPIスコア計算
py -3.10 scripts/compute_scores.py
```

### XBRLタグマップ自動生成

財務3表CSV全項目から488項目のXBRLタグマップを自動生成:

```
py -3.10 scripts/generate_xbrl_map.py
```

生成ファイル:

- `src/components/xbrlTagMap.ts`: 488項目のXBRLマッピング
- `scripts/xbrl_fields_list.json`: 検証用フィールドリスト

## テスト

### ユニットテスト

```
npm run test
```

カバレッジ確認:

```
npm run test:coverage
```

目標: 80%以上（現在82%達成）

### E2Eテスト

```
npm run test:e2e
```

対象フロー:

- 企業価値指標表示（valuation-display.spects）
- KPIスコアード表示（scorecard-display.spects）
- 推移グラフ表示（trend-display.spects）
- 従業員情報表示（employee-info.spects）
- 財務諸表表示（financial-statements.spect.s）

## デプロイ

### GitHub Pagesへの自動デプロイ

mainブランチへのpushで自動実行されます。

```
git checkout main
git pull origin main
git add .
git commit -m "deploy: update ValueScope"
git push origin main
```

2-4分後に <https://1921604.github.io/ValueScope/> が更新されます。

### ローカルビルド

```
npm run build
npm run preview
```

<http://localhost:4173/ValueScope/> でプレビュー確認。

## プロジェクト構造

```
ValueScope/
├── .github/
│   └── workflows/
│       └── deploy-pages.yml      # CI/CDパイプライン
├── src/
│   ├── App.tsx                  # メインコンポーネント
│   ├── components/              # Reactコンポーネント
│   ├── hooks/                   # カスタムフック
│   ├── types/                   # TypeScript型定義
│   └── utils/                    # ユーティリティ関数
├── scripts/
│   ├── fetch_edinet.py          # EDINET APIデータ取得
│   ├── parse_edinet_xbrl.py     # XBRL解析
│   ├── build_timeseries.py       # 時系列KPI計算
│   ├── build_valuation.py       # 企業価値計算
│   ├── compute_scores.py        # KPIスコアリング
│   └── requirements.txt         # Python依存関係
├── data/
│   ├── kpi_targets.json         # KPI閾値定義
│   ├── edinet_parsed/          # EDINET解析結果
│   └── prices/                  # 株価データ
├── public/
│   ├── data/                    # ビルド入力用JSONファイル
│   ├── tests/
│   │   └── e2e/                 # Playwright E2Eテスト
│   └── docs/
│       ├── DEPLOY_GUIDE.md      # デプロイ手順書
│       └── 完全仕様書.md        # 完全実装仕様書
├── .specify/
│   └── memory/
│       └── constitution.md       # 開発憲法
├── package.json
├── vite.config.ts               # Vite設定
└── README.md                    # トップページ
```

## ドキュメント

### 仕様書

- [機能仕様書 \(spec.md\)](#)
- [実装計画書 \(plan.md\)](#)
- [タスクリスト \(tasks.md\)](#)

### 技術ドキュメント

- [完全実装仕様書](#)
- [デプロイガイド](#)
- [開発憲法](#)

## パフォーマンス

### 目標値

メトリクス	目標	現在値	ステータス
LCP（Largest Contentful Paint）	< 2.5秒	1.8秒	達成
TTI（Time to Interactive）	< 2.0秒	1.5秒	達成
初期バンドルサイズ（gzip）	< 200KB	174KB	達成
チャート再描画	< 200ms	150ms	達成
Lighthouseスコア	≥ 90	92	達成
ユニットテストカバレッジ	≥ 80%	82%	達成
E2E主要フロー	100%	100%	達成

## 開発憲法

このプロジェクトは[開発憲法 \(constitution.md\)](#)で定義された7つのコア原則に準拠します。

### 7つのコア原則

- テスト駆動開発（TDD）の徹底
- データ品質とXBRL実データ原則
- セキュリティファースト
- パフォーマンス要件の定量化
- 電力業界特化KPIの厳格な閾値管理
- バージョン管理とドキュメント整合性
- CI/CD（継続的インテグレーション/デプロイメント）

## トラブルシューティング

問題1: データが表示されない

原因: `public/data/*.json` が存在しないか、形式が不正

解決:

```
py -3.10 scripts/build_timeseries.py
py -3.10 scripts/build_valuation.py
py -3.10 scripts/compute_scores.py
```

問題2: ビルドエラー

原因: TypeScriptの型エラー

解決:

```
npx tsc --noEmit
```

問題3: E2Eテスト失敗

原因: Playwrightプラグインが未インストール

解決:

```
npx playwright install
```

## ライセンス

このプロジェクトは個人利用のみを目的としています。商用利用は禁止されています。

## 作成者

J1921604 GitHub: <https://github.com/1921604>

## 更新履歴

バージョン	日付	変更内容
1.0.0	2025-12-15	初版リリース

## 関連リンク

- 公開URL: <https://1921604.github.io/ValueScope/>
- GitHubリポジトリ: <https://github.com/1921604/ValueScope>
- GitHub Actions: <https://github.com/1921604/ValueScope/actions>
- Issues: <https://github.com/1921604/ValueScope/issues>