# PyForecast

## Data Validation & Error Handling Guide

Comprehensive Guide to Production Data Quality Assurance

*January 2026*

# Table of Contents

# 1. Executive Summary

PyForecast includes a comprehensive data validation and error handling system designed to catch data quality issues before they impact decline curve analysis.

This system provides:

- Automatic detection of common production data problems
- Structured error codes for easy issue identification
- Configurable thresholds to match your operational parameters
- Actionable guidance for resolving each issue type
- Integrated reporting in batch processing workflows

The validation system operates at three levels:

| Level | Description |
|---|---|
| Input Validation | Verifies data format and value ranges |
| Data Quality | Detects gaps, outliers, and anomalies |
| Fitting Validation | Ensures fitting prerequisites and quality |

# 2. Getting Started

## Basic Usage

Validation runs automatically during batch processing:

```
pyforecast process production_data.csv -o output/
```

After processing, you'll see a validation summary showing wells with errors, warnings, and issues by category.

## Configuration

Generate a configuration file with validation settings:

```
pyforecast init -o pyforecast.yaml
```

Key validation parameters you can customize:

- max_oil_rate: Maximum expected oil rate (bbl/mo)
- max_gas_rate: Maximum expected gas rate (mcf/mo)
- gap_threshold_months: Minimum gap size to flag
- outlier_sigma: Standard deviations for outlier detection
- min_r_squared: Minimum acceptable R-squared value

# 3. Validation Categories

## Overview

All validation issues are categorized for easy filtering and reporting:

| Category | Description | When Checked |
|---|---|---|
| DATA_FORMAT | Column, date, value format issues | Before fitting |
| DATA_QUALITY | Gaps, outliers, shut-ins | Before fitting |
| FITTING_PREREQ | Pre-fit requirements not met | Before fitting |
| FITTING_RESULT | Post-fit quality concerns | After fitting |

## Severity Levels

Each issue has a severity level indicating required action:

| Severity | Meaning | Action Required |
|---|---|---|
| ERROR | Cannot proceed safely | Must resolve before trusting results |
| WARNING | Can proceed with caution | Review recommended |
| INFO | Informational only | No action required |

# 4. Input Validation (IV Codes)

Input validation checks that production data is properly formatted and within expected ranges.

### IV001: Negative Production Values

**ERROR**

Production values must be non-negative. Negative values indicate data entry errors or unit conversion problems.

Resolution: Check source data for typos, verify unit conversions, replace negative values with zero or interpolated values.

### IV002: Values Exceed Threshold

**WARNING**

Production values exceed configured maximum rates. Very high values may indicate unit conversion errors (e.g., daily rates instead of monthly).

Default thresholds: Oil 50,000 bbl/mo, Gas 500,000 mcf/mo, Water 100,000 bbl/mo

### IV003: Date Parsing Failed

**ERROR**

Production dates could not be parsed. This typically indicates an unsupported date format.

Supported formats: YYYY-MM-DD, MM/DD/YYYY, DD-Mon-YYYY, Excel serial numbers

### IV004: Future Dates in Data

**WARNING**

Production dates are in the future, which may indicate data entry errors or placeholder values.

# 5. Data Quality Validation (DQ Codes)

Data quality validation detects patterns that may affect fitting accuracy.

### DQ001: Data Gaps Detected

**WARNING**

Consecutive months of zero or near-zero production surrounded by non-zero production. Gaps may represent missing data or operational shut-ins.

Default threshold: 2+ months. Resolution: Determine if gaps are operational or data issues; regime detection handles shut-ins automatically.

### DQ002: Outliers Detected

**WARNING**

Values significantly different from typical production pattern. Uses Modified Z-Score with Median Absolute Deviation (MAD) for robust detection.

Default threshold: 3.0 sigma. Resolution: Investigate outliers, correct errors, or adjust outlier_sigma sensitivity.

### DQ003: Shut-in Periods Detected

**INFO**

Periods where production drops to near-zero then resumes. These trigger regime detection, which fits only the most recent decline period.

Default threshold: < 1.0 bbl/month. Usually no action required.

### DQ004: Low Data Variability

**WARNING**

Production data has very low variability (near-constant values). May indicate synthetic data, smoothed data, or allocation issues.

Default threshold: CV < 0.05. Resolution: Verify data is metered production.

# 6. Fitting Prerequisite Validation (FP Codes)

Pre-fit validation ensures data is suitable for decline curve analysis.

### FP001: Insufficient Data Points

**ERROR**

Not enough data points to perform reliable curve fitting.

Default minimum: 6 months. Resolution: Obtain more history or skip fitting.

### FP002: Increasing Trend

**WARNING**

Production shows an increasing trend rather than decline. Decline curve analysis assumes production is declining.

Resolution: Verify well is in decline phase, check for recent workover, wait for stabilization, or use regime detection.

### FP003: Flat Trend

**WARNING**

Production shows minimal decline, which hyperbolic models may not fit well.

Resolution: Verify production pattern, consider if decline forecasting is appropriate.

# 7. Fitting Result Validation (FR Codes)

Post-fit validation assesses the quality and reasonableness of fitted parameters.

### FR001: Poor Fit Quality

**WARNING**

The fitted curve does not match the production data well. WARNING for R-squared 0.3-0.5, ERROR for R-squared < 0.3.

Default threshold: R-squared < 0.5. Resolution: Review data quality, check regime detection, consider alternative models.

### FR003: B-Factor at Lower Bound

**INFO**

The fitted b-factor is at the configured lower bound (default 0.01), suggesting near-exponential decline.

Usually acceptable - exponential decline is a valid pattern.

### FR004: B-Factor at Upper Bound

**WARNING**

The fitted b-factor is at the configured upper bound (default 1.5). Very high b-factors may indicate transient flow or data quality issues.

Resolution: Review production plot, check for early-time transient behavior.

### FR005: Very High Decline Rate

**WARNING**

The fitted initial decline rate exceeds 100% per year, which is unusual and may indicate fitting issues.

Resolution: Verify data is monthly, check early months quality. High declines may be valid for tight oil/gas plays.

# 8. Error Code Quick Reference

| Code | Severity | Category | Description |
|---|---|---|---|
| IV001 | ERROR | DATA_FORMAT | Negative production values |
| IV002 | WARNING | DATA_FORMAT | Values exceed threshold |
| IV003 | ERROR | DATA_FORMAT | Date parsing failed |
| IV004 | WARNING | DATA_FORMAT | Future dates in data |
| DQ001 | WARNING | DATA_QUALITY | Data gaps detected |
| DQ002 | WARNING | DATA_QUALITY | Outliers detected |
| DQ003 | INFO | DATA_QUALITY | Shut-in periods detected |
| DQ004 | WARNING | DATA_QUALITY | Low data variability |
| FP001 | ERROR | FITTING_PREREQ | Insufficient data points |
| FP002 | WARNING | FITTING_PREREQ | Increasing trend |
| FP003 | WARNING | FITTING_PREREQ | Flat trend |
| FR001 | WARN/ERR | FITTING_RESULT | Poor fit (R-squared < threshold) |
| FR003 | INFO | FITTING_RESULT | B-factor at lower bound |
| FR004 | WARNING | FITTING_RESULT | B-factor at upper bound |
| FR005 | WARNING | FITTING_RESULT | Very high decline rate |

# 9. Programmatic Usage

## Basic Validation

```
from pyforecast.validation import (
    InputValidator,
    DataQualityValidator,
    FittingValidator,
)

# Create validators
input_validator = InputValidator(max_oil_rate=75000)
quality_validator = DataQualityValidator(outlier_sigma=2.5)
fitting_validator = FittingValidator(min_r_squared=0.6)

# Validate a well
result = input_validator.validate(well)
if result.has_errors:
    for error in result.errors():
        print(f"{error.code}: {error.message}")
```

## Working with Results

```
from pyforecast.validation import merge_results, IssueCategory

# Filter by category
quality_issues = result.by_category(IssueCategory.DATA_QUALITY)

# Merge multiple results
combined = merge_results([input_result, quality_result])

# Check status
print(f"Valid: {combined.is_valid}")
print(f"Errors: {combined.error_count}")
```

# 10. Configuration Reference

## Complete Validation Configuration

```
validation:
  max_oil_rate: 50000      # bbl/month - IV002
  max_gas_rate: 500000     # mcf/month - IV002
  max_water_rate: 100000   # bbl/month - IV002
  gap_threshold_months: 2  # Months - DQ001
  outlier_sigma: 3.0       # Std deviations - DQ002
  shutin_threshold: 1.0    # Rate threshold - DQ003
  min_cv: 0.05             # Coefficient of variation - DQ004
  min_r_squared: 0.5       # R-squared threshold - FR001
  max_annual_decline: 1.0  # Annual fraction - FR005
  strict_mode: false       # Treat warnings as errors
```

## Parameter Summary

| Parameter | Default | Description |
|---|---|---|
| max_oil_rate | 50,000 | Max expected oil rate (bbl/mo) |
| max_gas_rate | 500,000 | Max expected gas rate (mcf/mo) |
| gap_threshold_months | 2 | Min consecutive zero months to flag |
| outlier_sigma | 3.0 | Modified Z-score threshold |
| min_cv | 0.05 | Minimum coefficient of variation |
| min_r_squared | 0.5 | Minimum acceptable R-squared |
| max_annual_decline | 1.0 | Max annual decline (1.0 = 100%) |
| strict_mode | false | Treat warnings as errors |

# 11. Best Practices

## Data Preparation Checklist

Before running PyForecast:

- Verify all dates are in a consistent format
- Confirm production values are monthly (not daily/annual)
- Check for negative values and correct
- Remove forecast/projected data from historical records
- Verify units match expected (bbl, mcf)

## Inter

- Address ERRORs first - These prevent reliable analysis
- Review WARNINGs - Understand the cause before accepting
- Note INFOs - Generally informational, no action needed

## Adju

| Scenario | Adjustment |
|---|---|
| Prolific wells | Increase max_oil_rate and max_gas_rate |
| Tight formations | Increase max_annual_decline |
| Noisy data | Increase outlier_sigma to reduce false positives |
| Strict QC | Set strict_mode: true |

## Common Workflows

Initial data load:

```
pyforecast process data.csv -o output/ --no-plots
cat output/validation_report.txt
```

Production processing with custom config:

```
pyforecast process data.csv -o output/ --config basin_config.yaml
```