

DOE Pipeline Setup & Installation Guide

Last Updated: 2024 Version: 2.0 Target Environment: Python 3.8+

Table of Contents

1. [Quick Start](#)
 2. [System Prerequisites](#)
 3. [Virtual Environment Setup](#)
 4. [Package Installation](#)
 5. [Verification](#)
 6. [Troubleshooting](#)
 7. [Platform-Specific Notes](#)
-

Quick Start

```
# Clone or navigate to project directory
cd /Users/vblake/doe2

# Create and activate virtual environment
python3 -m venv venv
source venv/bin/activate # macOS/Linux
# or: venv\Scripts\activate # Windows

# Install all dependencies
pip install -r requirements.txt

# Verify installation
python3 -c "import statsmodels; import plotly; print('✓ Setup complete')"

# Run the pipeline
python3 doep.py
```

System Prerequisites

Check Python Installation

```
# Verify Python 3.8+ is installed
python3 --version
```

```
# Output should be: Python 3.8.x or higher
```

If Python is not installed, follow platform-specific instructions below.

Check pip

```
# Verify pip is available
pip3 --version

# Output should be: pip X.Y.Z from /path/to/python...
```

Virtual Environment Setup

Why Virtual Environments?

A virtual environment isolates project dependencies, preventing conflicts with system packages or other projects.

Create Virtual Environment

```
# Navigate to project root
cd /Users/vblake/doe2

# Create virtual environment named 'venv'
python3 -m venv venv

# This creates the directory structure:
# /Users/vblake/doe2/
#   └── venv/
#       ├── bin/
#       │   └── activate
#       │   └── python
#       └── pip
#   └── lib/
#       └── include/
#   └── doep.py
#   └── requirements.txt
#   └── ... (other files)
```

Activate Virtual Environment

macOS/Linux:

```
source venv/bin/activate

# You should now see (venv) prefix in terminal:
# (venv) $
```

Windows:

```
venv\Scripts\activate

# You should now see (venv) prefix in terminal:
# (venv) C:\path\to\doe2>
```

Verify Activation

```
# Check which Python is being used
which python # macOS/Linux
# Output: /Users/vblake/doe2/venv/bin/python

where python # Windows
# Output: C:\path\to\doe2\venv\Scripts\python.exe

# Verify it's the venv version
python --version
```

Deactivate Virtual Environment

When finished working with the project:

```
deactivate

# Terminal prompt returns to normal (no (venv) prefix)
```

Package Installation

Method 1: Install from requirements.txt (Recommended)

```
# Ensure venv is activated
source venv/bin/activate
```

```
# Install all packages at once
pip install -r requirements.txt

# Installation takes 2-5 minutes
# You'll see: Successfully installed [package names] ...
```

Method 2: Install by Category

Data Science & Analysis Stack

```
source venv/bin/activate

pip install \
    pandas==1.3.5 \
    numpy==1.21.6 \
    scipy==1.7.3 \
    statsmodels==0.13.5 \
    scikit-learn==1.0.2

echo "✓ Data science libraries installed"
```

Visualization Libraries

```
source venv/bin/activate

pip install \
    plotly==5.4.0 \
    matplotlib==3.5.2 \
    kaleido==0.2.1

echo "✓ Visualization libraries installed"
```

Output Format Libraries

```
source venv/bin/activate

pip install \
    beautifulsoup4==4.10.0 \
    reportlab==3.6.6 \
    pillow==9.1.1 \
    pdfkit==1.0.0 \
```

```
python-pptx==0.6.21 \
lxml==4.9.1

echo "✓ Output format libraries installed"
```

Method 3: Install Incrementally (For Testing)

```
source venv/bin/activate

# Install minimum for core functionality
pip install pandas numpy scipy statsmodels

# Later, add visualization
pip install plotly kaleido

# Later, add reporting
pip install python-pptx reportlab beautifulsoup4
```

Package Categories & Purpose

 **Detailed package reference:** See [DOEP_LIB_REQS.md](#) for complete library requirements with installation instructions and compatibility information.

Data Science & Analysis (Core)

Package	Version	Purpose
pandas	1.3.5	Data loading, manipulation, CSV I/O
numpy	1.21.6	Numerical arrays, mathematical operations
scipy	1.7.3	Statistical tests, optimization, distributions
statsmodels	0.13.5	DOE modeling, ols() regression, C() categorical
scikit-learn	1.0.2	Machine learning utilities, preprocessing

Installation:

```
pip install pandas numpy scipy statsmodels scikit-learn
```

Visualization (Plotting)

Package	Version	Purpose
plotly	5.9.0	Interactive visualization library

plotly	5.4.0	Interactive HTML plots (hover, zoom, export)
matplotlib	3.5.2	Static plotting foundation
kaleido	0.2.1	PNG/SVG export from Plotly

Installation:

```
pip install plotly matplotlib kaleido
```

Output Formats (Reporting)

Package	Version	Purpose
beautifulsoup4	4.10.0	HTML parsing and manipulation
reportlab	3.6.6	PDF generation (tables, images, layouts)
pillow	9.1.1	Image processing (PIL)
pdfkit	1.0.0	Advanced PDF features (optional)
python-pptx	0.6.21	PowerPoint generation
lxml	4.9.1	XML/HTML parsing optimization

Installation:

```
pip install beautifulsoup4 reportlab pillow pdfkit python-pptx lxml
```

Verification

Method 1: Individual Package Tests

```
source venv/bin/activate

# Test each package
python3 -c "import pandas; print('✓ pandas')"
python3 -c "import numpy; print('✓ numpy')"
python3 -c "import scipy; print('✓ scipy')"
python3 -c "import statsmodels; print('✓ statsmodels')"
python3 -c "import sklearn; print('✓ scikit-learn')"
python3 -c "import plotly; print('✓ plotly')"
python3 -c "import matplotlib; print('✓ matplotlib')"
python3 -c "import kaleido; print('✓ kaleido')"
```

```
python3 -c "from bs4 import BeautifulSoup; print('✓ beautifulsoup4')"  
python3 -c "from reportlab import pdfgen; print('✓ reportlab')"  
python3 -c "from PIL import Image; print('✓ pillow')"  
python3 -c "from pptx import Presentation; print('✓ python-pptx')"  
python3 -c "import lxml; print('✓ lxml')"
```

Expected Output:

- ✓ pandas
- ✓ numpy
- ✓ scipy
- ✓ statsmodels
- ✓ scikit-learn
- ✓ plotly
- ✓ matplotlib
- ✓ kaleido
- ✓ beautifulsoup4
- ✓ reportlab
- ✓ pillow
- ✓ python-pptx
- ✓ lxml

Method 2: List Installed Packages

```
source venv/bin/activate  
  
# Show all installed packages  
pip list  
  
# Or save to file for documentation  
pip list > installed_packages.txt
```

Method 3: Run Complete Pipeline

```
source venv/bin/activate  
  
cd /Users/vblake/doe2  
  
# Run the full pipeline (takes 5-10 minutes)  
python3 doep.py
```

```
# Check outputs directory
ls -lh outputs/
```

Expected output files:

```
outputs/
├── balanced_low_df.csv
├── balanced_high_df.csv
├── doe_analysis_report.html
├── doe_analysis_reduced.html
├── doe_analysis_report.pptx
├── doe_analysis_reduced.pptx
├── doe_model_comparison.pptx
└── ... (visualization files)
```

Troubleshooting

Issue: venv not found

Symptom:

```
source: venv/bin/activate: No such file or directory
```

Solution:

```
# Create the venv directory
python3 -m venv venv

# Then activate
source venv/bin/activate
```

Issue: pip command not found

Symptom:

```
command not found: pip
```

Solution:

```
# Check venv is activated (should see (venv) prefix)
source venv/bin/activate

# Use python -m pip instead
python -m pip install pandas

# Or use pip3
pip3 install pandas
```

Issue: ModuleNotFoundError when running doep.py

Symptom:

```
ModuleNotFoundError: No module named 'statsmodels'
```

Solution:

```
# Ensure venv is activated
source venv/bin/activate

# Install requirements
pip install -r requirements.txt

# Verify
python -c "import statsmodels; print('✓ statsmodels installed')"

# Try pipeline again
python doep.py
```

Issue: kaleido fails to import

Symptom:

```
ImportError: libssl.so.1.1: cannot open shared object file
```

Solution (macOS):

```
# Install additional system dependencies
brew install openssl
```

```
# Reinstall kaleido
source venv/bin/activate
pip install --force-reinstall kaleido
```

Solution (Ubuntu/Linux):

```
# Install system dependencies
sudo apt-get install libssl-dev

# Reinstall kaleido
source venv/bin/activate
pip install --force-reinstall kaleido
```

Issue: pdfkit fails (wkhtmltopdf not found)

Symptom:

```
OSErrror: wkhtmltopdf not found on PATH
```

Solution (macOS):

```
# Install wkhtmltopdf
brew install --cask wkhtmltopdf

# Verify installation
wkhtmltopdf --version
```

Solution (Ubuntu/Linux):

```
# Install wkhtmltopdf
sudo apt-get install wkhtmltopdf

# Verify installation
wkhtmltopdf --version
```

Solution (Windows):

- Download installer from: <https://wkhtmltopdf.org/downloads.html>
 - Or use Chocolatey: choco install wkhtmltopdf
-

Issue: Permission denied when running script

Symptom:

```
-bash: ./doep.py: Permission denied
```

Solution:

```
# Make file executable
chmod +x doep.py

# Or run with Python explicitly
python doep.py
```

Issue: Python 2.7 being used instead of Python 3

Symptom:

```
python --version
# Output: Python 2.7.18
```

Solution:

```
# Use python3 explicitly
python3 --version
# Output: Python 3.8.x or higher

# Create venv with python3
python3 -m venv venv

# Activate and verify
source venv/bin/activate
python --version # Should now show 3.8+
```

Issue: requirements.txt file location error

Symptom:

```
ERROR: Could not open requirements file: [Errno 2] No such file or c
```

Solution:

```
# Ensure you're in project root directory
cd /Users/vblake/doe2

# Verify requirements.txt exists
ls requirements.txt

# Try installation again
pip install -r requirements.txt
```

Platform-Specific Notes

macOS

Additional Dependencies:

```
# If using M1/M2 chip, you may need architecture-specific packages
# Most packages now have ARM64 support

# Install Homebrew (if not already installed)
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"

# Install Python3
brew install python3

# Install additional system libraries (if needed)
brew install openssl libffi
```

Example Setup Script:

```
#!/bin/bash
cd /Users/vblake/doe2

# Install Python3 via Homebrew
brew install python3

# Create and activate venv
python3 -m venv venv
source venv/bin/activate

# Install requirements
```

```
pip install --upgrade pip setuptools wheel  
pip install -r requirements.txt  
  
echo "✓ macOS setup complete!"
```

Linux (Ubuntu/Debian)

System Prerequisites:

```
# Update package manager  
sudo apt-get update  
  
# Install Python3 and pip  
sudo apt-get install python3 python3-pip python3-venv  
  
# Install build tools (for compiling some packages)  
sudo apt-get install build-essential  
  
# Install system libraries (for kaleido, etc.)  
sudo apt-get install libssl-dev libffi-dev
```

Setup Steps:

```
cd /Users/vblake/doe2  
  
# Create and activate venv  
python3 -m venv venv  
source venv/bin/activate  
  
# Install requirements  
pip install --upgrade pip setuptools wheel  
pip install -r requirements.txt  
  
echo "✓ Linux setup complete!"
```

Windows

Python Installation:

- Download from <https://www.python.org/downloads/>
- Ensure "Add Python to PATH" is checked during installation

- Verify: `python --version` in Command Prompt/PowerShell

Setup Steps:

```
cd C:\path\to\doe2

# Create virtual environment
python -m venv venv

# Activate (PowerShell)
venv\Scripts\Activate.ps1

# Or activate (Command Prompt)
venv\Scripts\activate.bat

# Install requirements
pip install -r requirements.txt

echo ✓ Windows setup complete!
```

PowerShell Execution Policy (if needed):

```
# If scripts are blocked, allow execution for current user
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser

# Then activate venv
venv\Scripts\Activate.ps1
```

Requirements File Reference

The `requirements.txt` file contains all package versions needed for reproducibility:

```
# Data Science & Analysis
pandas==1.3.5
numpy==1.21.6
scipy==1.7.3
statsmodels==0.13.5
scikit-learn==1.0.2

# Visualization
plotly==5.4.0
matplotlib==3.5.2
```

```
kaleido==0.2.1

# HTML/XML Processing
beautifulsoup4==4.10.0
lxml==4.9.1

# PDF Generation
reportlab==3.6.6
pillow==9.1.1
pdfkit==1.0.0

# PowerPoint Generation
python-pptx==0.6.21
```

Version Pinning

Each package has a specific version (e.g., `pandas==1.3.5`) to ensure:

- **Reproducibility:** Same code produces same results across machines
- **Stability:** Known-working versions without breaking changes
- **Compatibility:** Packages tested to work together

Updating Packages

```
source venv/bin/activate

# Update all packages to latest compatible versions
pip install --upgrade -r requirements.txt

# Update specific package
pip install --upgrade pandas

# Generate new requirements file
pip freeze > requirements_updated.txt
```

Verification Checklist

Before running the pipeline, verify:

- Python 3.8+ installed: `python3 --version`
- pip available: `pip3 --version`
- Virtual environment created: `ls -d venv/`
- Virtual environment activated: (venv) prefix visible in terminal

- requirements.txt exists: ls requirements.txt
 - All packages installed: pip list | grep pandas (should show)
 - Packages importable: python -c "import statsmodels; import plotly"
 - Input data available: ls outputs/fan_*.csv (or outputs/ directory exists)
 - Output directory writable: touch outputs/test.txt && rm outputs/test.txt
-

Next Steps

1. Verify Installation:

```
python3 -c "import statsmodels; import plotly; print('✓ Ready!')
```

2. Run Pipeline:

```
python3 doep.py
```

3. Review Output:

```
ls -lh outputs/
```

4. Examine Results:

- Open outputs/doe_analysis_report.html in browser
 - Open outputs/doe_analysis_report.pptx in PowerPoint
-

Additional Resources

DOE Pipeline Documentation

- **DOEP_LIB_REQS.md** - Complete library requirements reference
- **DOEP_README.md** - Pipeline documentation and reference
- **DOCSTRING_PLAN.md** - Architecture and planning documentation

External Resources

- **Python Virtual Environments:** <https://docs.python.org/3/tutorial/venv.html>
 - **pip Documentation:** <https://pip.pypa.io/en/latest/>
 - **statsmodels Docs:** <https://www.statsmodels.org/>
 - **Plotly Documentation:** <https://plotly.com/python/>
 - **pandas User Guide:** https://pandas.pydata.org/docs/user_guide/
-

Support

For issues or questions:

1. Check [Troubleshooting](#) section
 2. Review [Platform-Specific Notes](#)
 3. Examine error messages carefully (they often indicate missing packages)
 4. Verify venv is activated before running any Python code
-

Setup Guide Version: 1.0

Last Updated: 2024

Maintained By: DOE Pipeline Development Team