

Getting Started with PyStatsV1

Applied Statistics in Python, One Proof at a Time

Version v0.1 - 2026-06-12

Published by NEKpress.ca

A compact MVP guide for running PyStatsV1 with proof-first habits.

Contents

Getting Started with PyStatsV1

1. What PyStatsV1 is
 2. Installing Python safely
 3. Initializing the workbook
 4. Mini case study: two-sample comparison
 7. Troubleshooting
 8. Next steps and links
- MIT license note

Getting Started with PyStatsV1

Applied Statistics in Python, One Proof at a Time

Free first edition / MVP edition Version: v0.1 Release date: 2026-06-12 Publisher: NEKpress.ca

This compact first edition introduces PyStatsV1 as an MIT-licensed educational statistics project. It is designed to be downloaded, read, and used as a practical starting point before a learner commits to a longer course or textbook.

Reader promise: by the end of this short book, you should understand what PyStatsV1 is for, how to prepare a safe Python environment, how to run a small first proof, and how to read early output without overclaiming.

The goal is simple: install the tools, run a small proof, inspect the output, and learn how to make statistical work more reproducible.

1. What PyStatsV1 is

PyStatsV1 is an MIT-licensed applied statistics learning project built around Python, reproducible examples, and inspectable outputs. It is not trying to hide statistics behind a button. It is trying to help a learner see the command, the data path, the output path, and the check that proves the result was produced.

The project is educational. It is useful for learning, practice, demonstrations, and small proof-first workflows. It is not a substitute for professional review when real decisions depend on the data.

A good PyStatsV1 learning loop has four parts:

- run a small script or command;
- inspect the generated output;
- run the matching check;
- explain the result in plain language without overclaiming.

This is the same proof-first habit used throughout the LearnToProgram.ca portal. You do not just hope the tool works. You ask the tool to leave evidence.

Who this book is for

This book is for careful beginners, students, tutors, instructors, analysts, and self-directed learners who want a practical way to learn applied statistics in Python. It assumes you can use a terminal at a basic level, but it does not assume that you are already an expert programmer.

What this book is not

This book is not statistical consulting, legal advice, medical advice, financial advice, privacy advice, tax advice, or payment-compliance advice. It is an educational starting point. For real research or business decisions, review assumptions, data quality, and the consequences of error with appropriate professional care.

2. Installing Python safely

Before installing any statistics package, start by checking your Python environment. A reproducible workflow begins with a known interpreter, a known project folder, and a small set of commands that can be repeated later.

Use a project folder you control. Avoid installing experimental learning tools into a system Python environment. A virtual environment keeps the learning project separate from the rest of your computer.

A common Ubuntu or macOS workflow is:

Code

```
python3 --version
python3 -m venv .venv
source .venv/bin/activate
python -m pip install -U pip
```

On Windows, use PowerShell and a project folder you control:

Code

```
py -3 --version
py -3 -m venv .venv
.\.venv\Scripts\Activate.ps1
python -m pip install -U pip
```

Installing PyStatsV1

The learner-facing path is expected to support a workbook extra:

Code

```
python -m pip install "pystatsv1[workbook]"
pystatsv1 doctor
```

The source-inspection path is useful when you want to read the code and examples directly:

Code

```
git clone https://github.com/pystatsv1/PyStatsV1.git
cd PyStatsV1
python -m venv .venv
source .venv/bin/activate
python -m pip install -U pip
python -m pip install -e .
```

The first proof

The first proof is not a statistical result. The first proof is that your environment can run the project and report its status.

Code

```
pystatsv1 doctor
```

Save the command and the output. That small habit helps later when an installation works on one machine but fails on another.

3. Initializing the workbook

A workbook is useful because it gives the learner a stable place to run scripts, inspect outputs, and compare results. The workflow should be boring in the best possible way: initialize, run, inspect, test, explain.

A typical first pass is:

Code

```
pystatsv1 workbook init
cd pystatsv1_workbook
pytest -q
```

If tests pass, the workbook is not just present. It has produced evidence that the baseline examples are runnable in your environment.

Running the first workbook checks

After the workbook exists, pick one small script. Avoid jumping straight to a complex model. A useful first example is a descriptive-statistics or introductory hypothesis-testing script.

Code

```
python scripts/psych_ch10_problem_set.py
pytest -q
```

Read the output before changing anything. Ask three questions:

- What data or synthetic example did the script use?
- What output file or printed summary did it create?
- What test proves that the expected behavior still holds?

Reading output without overclaiming

A statistic is not a conclusion by itself. A p-value, confidence interval, regression coefficient, or diagnostic plot needs context. Before writing a claim, identify the question, the data source, the model or test, and the assumptions that could matter.

A safer sentence often looks like this:

Code

```
In this worked example, the script produced a difference consistent with the
chapter exercise. The result is evidence for the exercise workflow, not a
general claim about a population outside the example.
```

That sentence is modest on purpose. PyStatsV1 is meant to help you build reliable habits before you make broad claims.

4. Mini case study: two-sample comparison

A two-sample comparison is a good teaching example because it forces the learner to separate the question from the calculation. The statistical tool can compare groups, but the learner must still ask what the groups mean and whether the comparison is fair.

A proof-first mini workflow is:

Code

```
python scripts/psych_ch10_problem_set.py
pytest -q
```

Then write a short review note:

- What are the two groups?
- What outcome is being compared?
- What assumptions does the example rely on?
- What did the script output?
- What did the test verify?

The point is not to memorize one command. The point is to learn how a command, an output, and a test fit together.

5. Mini case study: reliability

Reliability is about whether a measurement process is stable enough to be useful. In statistics education, reliability examples help learners see that data quality comes before flashy modeling.

A proof-first reliability note should include:

- the measurement being discussed;
- the repeated items or repeated measurements;
- the summary statistic used;
- a plain-language interpretation;
- a warning about what the statistic does not prove.

If the result is based on synthetic or classroom data, say so. Synthetic data is valuable for learning because it is safe and reproducible, but it should not be presented as field evidence.

6. The proof-first workflow

For each PyStatsV1 exercise, keep this checklist nearby:

Code

Question -> Data path -> Command -> Output -> Check -> Plain-language note

That chain is the difference between a screenshot and a reproducible lesson.

7. Troubleshooting

Most early problems are environment problems, not statistics problems. Fix them one at a time and keep notes.

Python command not found

Check which command your system uses:

Code

```
python --version
python3 --version
py -3 --version
```

Use one command consistently inside a project.

Virtual environment not active

If installed packages disappear, activate the environment again:

Code

```
source .venv/bin/activate
```

On Windows PowerShell:

Code

```
.\.venv\Scripts\Activate.ps1
```

Package installation fails

Upgrade pip, then retry in a clean environment:

Code

```
python -m pip install -U pip
python -m pip install -e .
```

Tests fail

A failing test is useful evidence. Read the first failure, not the last screenful of output. Record:

- the command you ran;
- the exact first error;
- the operating system;
- the Python version;

- the current Git commit if you are using the source repository.

Statistical output surprises you

Do not force the result to match your expectation. First check the input data, grouping variable, missing values, and assumptions. Then rerun the script from a clean baseline.

8. Next steps and links

After the first proof works, continue in small steps.

Recommended next steps:

- run one descriptive-statistics example;
- run one confidence-interval example;
- run one hypothesis-testing example;
- write a one-paragraph interpretation;
- change one small input and rerun the check;
- keep evidence of what changed.

The project source is available at:

Code

<https://github.com/pystatsv1/PyStatsV1>

The LearnToProgram.ca MVP keeps the resource free while also proving a sandbox-only voluntary contribution path. Payment is not required for the download.

9. A simple learning rule

When you are unsure, return to the smallest reproducible proof. A smaller proof that you understand is more useful than a large analysis that you cannot explain.

MIT license note

PyStatsV1 software is MIT licensed. The full license text should be reviewed in the PyStatsV1 source repository.

This MVP PDF is educational material prepared for the LearnToProgram.ca portal. It summarizes a safe getting-started workflow and points learners toward the MIT-licensed PyStatsV1 software project.

A short MIT license reminder:

Code

```
Permission is granted to use, copy, modify, merge, publish, distribute,  
sublicense, and/or sell copies of the Software, subject to the conditions  
stated in the MIT License.
```

This book does not remove the need to review the actual project license, source code, documentation, and tests.

Closing reminder

Save evidence, not secrets. Do not paste API keys, payment credentials, private data, or student records into examples, screenshots, commits, or support messages.