The Journal of Open Source Education

offlinedatasci: A Python Package for Managing Data Science Software Installers when Limited Access to the Internet is Anticipated

Virnaliz Cruz[1], **Colin Sauze**[3], **Abhishek Dasgupta**[4], **Jannetta S. Steyn**[2], **Heather L Turner**[5], and **Ethan P. White**[1]

1 University of Florida **2** Newcastle University **3** National Oceanography Centre **4** University of Oxford **5** University of Warwick

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Submitted: 30 May 2024
Published: unpublished

Summary

Teaching, learning, and conducting data science often rely on internet connections for accessing and distributing data, software, and educational materials. As a result, it can be challenging to run data science training and conduct data science work in locations with limited or no internet access. We developed the offlinedatasci package to help address this challenge, as part of a broader set of tools and instructional materials developed by CarpentriesOffline to facilitate teaching and practicing data science in internet-limited environments.

The offlinedatasci package automates downloading or updating a bank of materials for running workshops and conducting offline data science work more broadly. These materials include open source statistical and graphing software (R (R Core Team, 2024) and Python (Van Rossum & Drake, 2009)), the associated integrated development environments (IDEs; RStudio (Posit team, 2024) and Jupyter Notebooks (Kluyver et al., 2016)), data science focused partial mirrors of the associated package repositories (CRAN, PyPI, and lesson materials structured for local use via the browser. The package provides both Python and command-line interfaces and is designed for maintaining local servers for instructors to use in teaching or for individual learners and data science practitioners to create a local repository of essential resources.

Introduction and Statement of Need

The practice of data science has become more accessible with increased data generation, more open data sharing practices, and improvements in computational power and storage capacity (Kelleher & Tierney, 2018). In response, there has been an increase in the development of software for manipulating, visualizing, and analyzing data, as well as instructional materials to make it easier to learn these important skills and tools. The resulting data, software, and educational materials are typically distributed online. As a result, these improvements in access to data science tools and skills are not homogeneously distributed. The median percentage of population with internet access across all countries is only 60.1% [cia2021internetusers]. This includes a connection from any device with varying degrees of consistency ranging from continuously, to several times a week, to once every few months. In the US, some of the factors that are associated with limited internet access are race and ethnicity, geography, and most importantly income (Swenson & Ghertner, 2021). Low-income US households are less likely to have access to broadband and more likely to have no internet access at all (Swenson & Ghertner, 2021). Although the increase in internet access worldwide is undeniable, the rate at which access increases and the quality of that access remains unequally distributed.

Most online data science tools and teaching materials make two basic assumptions about the users' resources: 1) access to computers; and 2) a stable internet connection to download data, install software, and view teaching materials while learning or working. While access to a computer is an unavoidable requirement for most stages of data science, the need for regular internet access can be mitigated by obtaining the necessary data, software, and lesson materials when and where internet access is available. Once these materials are downloaded, much of the associated training and data science work can be accomplished without internet access. However, the knowledge necessary to accomplish this is often not available to beginning data scientists. This makes limited internet access particularly challenging in teaching environments, where students often learn how to download and install data science tools during classes and workshops. Workshops may have to be run in venues without reliable internet access and many of the students may not have sufficient, affordable internet access prior to the workshop, leading to problems in acquiring hundreds of megabytes worth of software applications and their dependencies for workshop participants. Simplifying the downloading and offline use of data science components that have internet requirements could ameliorate some of the challenges that students and data scientists face due to unequal accessibility to the internet.

The offlinedatasci package is part of a growing set of tools and instructional materials developed by CarpentriesOffline to facilitate teaching and practicing data science in internet-limited environments. The larger ecosystem allows local computers and low power devices such as the Raspberry Pi to be used as isolated servers that provide a wireless network to workshop participants, so that they can acquire the necessary materials during workshops even when there is no internet access. The offlinedatasci package automates downloading or updating a bank of materials for running workshops or practicing data science offline, by providing: 1) open source statistical and graphing software (R and Python), 2) integrated development environments (IDEs) for working with this software (RStudio and Jupyter), 3) up-to-date mirrors of the package repositories used to install data science packages (CRAN, PyPI), and 4) online lesson materials configured for local viewing (currently a selection of Carpentries workshop lessons with their respective practice data sets).

Software Design (Methods)

This package is designed for two use cases. The original design focused on instructors teaching data science in internet limited environments using a Raspberry Pi, or a local computer capable of serving content over WiFi, that would provide students with access to data, installers, package repositories, and lesson material. This local server would serve as a replacement for a connection to the internet. The offlinedatasci package was designed to make creating and updating the content on this local teaching server easier. To make the software more broadly useful it has been designed to be helpful to both individual learners outside of a workshop and for individuals working in data science who anticipate unreliable or no access to the internet. It downloads a selection of software installers, configures partial mirrors of package repositories, and downloads lesson content for later use on the internet limited computer. This means that when an internet connection is available, a single command can be executed to download, update, and configure all necessary material for later use.

User knowledge assumptions

The package assumes that the user: 1) has an understanding of paths for storing and accessing files; 2) is capable of either using a basic command line interface (including flags) or running functions with arguments from a Python package; and 3) knows how to use pip to install Python packages.

Core design and backend

The offlinedatasci package automatically downloads the most recent versions of installers for essential tools including R, Python, and Rstudio. Obtaining up-to-date installers for all systems, that students are likely to use, requires automating the download of the most recent version for each operating system. We accomplish this by parsing the HTML from the relevant installer download pages, for R (https://cran.r-project.org/), Python (https://www.python.org/downloads/), and RStudio (https://posit.co/download/rstudio-desktop/) to determine the most recent versions and download the corresponding installers for both Windows and macOS. In cases where multiple installers are available for different architectures (e.g., M1/M2 macs and Intel-based macs) we download all available installers to support the widest range of possible user architectures (1.36 GB total as of 2023-08-15). By leveraging Python's capabilities to parse web pages and extract version information, we eliminate the need for manual checks for updates and facilitate instructors, researchers, and data scientists having the latest software readily available for future use. To avoid unnecessary downloads in internet limited environments, the update mechanism checks if the most recent version of the required components is already available locally (based on the filenames of the installers which include the version number) and if the local version is up-to-date it is not redownloaded. This approach avoids unnecessary data use while ensuring that the latest version of the software is available.

Offlinedatasci also creates partial local mirrors of the R and Python package repositories, containing data science packages for data manipulation, visualization, and analysis. It also allows users to add other packages to these mirrors. Installing packages is a common activity in data science workshops and research. Creating local mirrors of these package repositories can be complicated because 1) packages typically depend on other packages and therefore require not only downloading the package of interest but also its entire dependency tree; and 2) package repositories must follow specific file structures with appropriate metadata. To address this issue, we leverage software packages designed to create partial mirrors of the CRAN and PyPI package repositories. We use miniCRAN (Vries et al., 2022) for mirroring CRAN and pypi-mirror (montag451, 2023) for mirroring PyPI. These packages automate the download of packages including their full dependency trees and set up the local repository file structures. These local mirrors can then be used by pointing to a local teaching server with the repository mirror or by individual users pointing to the mirrored repository on their own machine. The latter use case is facilitated by offlinedatasci commands that can be used to configure R and Python to perform installs from a specific local mirror. By default users can access a pre-selected curated selection of packages and add more packages as needed without worrying about dependency management and file structures. We focus on partial mirrors containing the essential packages needed for data science tasks, rather than full mirrors, to save time, bandwidth, and storage since the full mirrors can be hundreds of gigabytes. Both miniCRAN and pypi-mirror check versions and only download packages that are either not present or for which a new release is available. This allows package repository install and update commands to be run regularly to ensure that the most up-to-date versions of packages are always available.

Offlinedatasci downloads lesson material to facilitate workshop instruction and individual learning. The lesson materials currently included are the Software Carpentry, Data Carpentry, and Library Carpentry lessons. These open lesson materials serve as the foundation for a global teaching effort, run by The Carpentries (https://carpentries.org/), that involves instruction in a number of regions with limited internet. The software is also designed to allow the easy addition of any online teaching material. Lesson material is written in a variety of different formats and using a range of build systems that frequently rely on external dependencies for rendering the lesson material into websites. Therefore offlinedatasci downloads rendered content directly from lesson websites to avoid the complexity and fragility associated with upstream changes when building lessons from multiple sources. Our approach uses Wget (Foundation, 2024), a software package that enables retrieving files using common internet protocols. We use Wget to manage this process, leveraging it's capabilities to: 1) recursively mirror directories; automating the process of finding all of the web pages associated with multiple page lessons; 2) convert absolute links in downloaded documents to relative links, allowing local links between pages to work in the local copies of the lessons; 3) automate downloading all of the external resources ensuring inclusion of things like images and CSS that are crucial for the proper presentation of materials; 4) only download lesson pages that have been updated since the last download; and 5) resume aborted downloads, minimizing data use in cases of interruptions to internet access. The lessons are presented on a single unified landing page, so that users can open a single index.html file with their browser of choice and smoothly navigate to all local lessons just as if they were connected to the world wide web.

Offlinedatasci uses the following R and Python packages for unmentioned processes: airium (Kaczmarczyk, 2023), requests (Reitz, 2024), beautifulsoup4 (Richardson, 2023), importlib-resources (Warsaw, 2024), remotes (Csárdi, 2024) and multiple packages that are distributed as part of Python 3: (argparse, os, pathlib, re, secrets, shutil, subprocess, sys, warnings; (Van Rossum & Drake, 2009)).

Installation

The package can be installed via the Python Package Index (PyPI) using pip:

```
pip install offlinedatasci
```

The development version can be installed directly from the associated GitHub repository (https://github.com/carpentriesoffline/offlinedatasci/):

```
pip install git+https://git@github.com/carpentriesoffline/offlinedatasci.git
```

User interface

The package has two interfaces, a command line interface and a Python interface.

Command line interface

For workshop instructors, the standard approach to using offlinedatasci will be to install all components for use on their local teaching server. This is done using:

```
offlinedatasci install all <path>
```

where <path> is replaced with the path where offlinedatasci should create its storage directory. This will download software for both macOS and Windows, set up repository mirrors for both Python and R packages, and download and set up the default instructional material for viewing from a local web browser.

More granular control for installing individual components is also available to facilitate personal use and customizing content for specific workshops. For example:

- Install Python: `offlinedatasci install python <path>`
- Install R and RStudio: `offlinedatasci install r rstudio <path>`
- Install lessons: `offlinedatasci install lessons <path>`
- Install R and Python package mirrors: `offlinedatasci install r-packages python-packages <path>`
- Add additional R packages: `offlinedatasci add r-packages <packagename> <packagename> <path>`
- Add additional Python packages:`offlinedatasci add python-packages <packagename> <packagename> <path>`

Python interface

The Python interface follows a similar structure but calling Python functions directly rather than through the CLI. The default installation command for workshop instructors that installs/updates all of the software and lesson material is:

```
import offlinedatasci as ods
```

```
ods.download_all("<path>")
```

The more granular functions follow a similar structure to those in the CLI. For example:

- Install Python: `ods.download_python("<path>")`
- Install lesson material: `ods.download_lessons("<path>")`
- Install R packages: `ods.download_r_packages("<path>")`
- Install custom R packages: `ods.download_r_packages("<path>", [<packagename>, <packagename>])`

Documentation

Documentation for offlinedatasci is built automatically on each commit to the GitHub repository using Sphinx (Brandl, 2010) and Read The Docs (https://about.readthedocs.com/?ref=readthedocs.org). The documentation is available at https://offlinedatasci.readthedocs.io.

Acknowledgements

We would like to acknowledge The Carpentries for their support through regular interactions with their core staff, invaluable feedback their team members working in internet limited environments, and for infrastructure support including communications channels and incorporation into their incubator program.

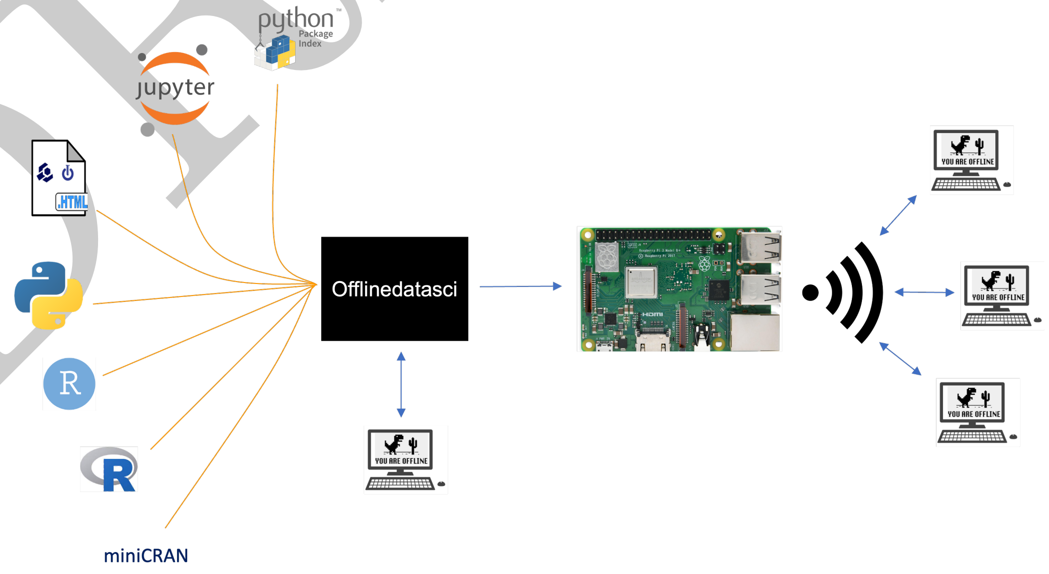

Figure 1: figure1

Figure 1. Visualization of how offlinedatasci works in the context of the larger Carpentries Offline system. The offlinedatasci package handles downloading and configuring software and lessons. This can be done on a local teaching server, like a Raspberry Pi, that can then be used to serve materials to learners taking classes or workshops. It can also be used by individual learners or data science practitioners by installing it on their personal computers.

References

Brandl, G. (2010). Sphinx documentation. *URL Http://Sphinx-Doc. Org/Sphinx. Pdf*.

Csárdi, G. (2024). *Remotes: R package installation from remote repositories, including 'GitHub'* (Version 2.5.0) [Computer software]. https://cran.r-project.org/web/packages/remotes/index.html

Foundation, F. S. (2024). *GNU wget* (Version 1.24.5). http://www.gnu.org/software/wget/

Kaczmarczyk, M. (2023). *Airium* (Version 0.2.6) [Computer software]. https://pypi.org/project/airium/

Kelleher, J. D., & Tierney, B. (2018). *Data science*. MIT Press.

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S., Willing, C., & team, J. development. (2016). Jupyter notebooks ? A publishing format for reproducible computational workflows. In F. Loizides & B. Scmidt (Eds.), *Positioning and power in academic publishing: Players, agents and agendas* (pp. 87–90). IOS Press. https://eprints.soton.ac.uk/403913/

montag451. (2023). *Python-pypi-mirror* (Version 5.2.1) [Computer software]. https://pypi.org/project/python-pypi-mirror/

Posit team. (2024). *RStudio: Integrated development environment for r*. Posit Software, PBC. http://www.posit.co/

R Core Team. (2024). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. https://www.R-project.org/

Reitz, K. (2024). *Requests* (Version 2.31.0) [Computer software]. https://pypi.org/project/requests/

Richardson, L. (2023). *beautifulsoup4* (Version 4.12.3) [Computer software]. https://pypi.org/project/beautifulsoup4/

Swenson, K., & Ghertner, R. (2021). *People in low-income households have less access to internet services*. https://aspe.hhs.gov/reports/low-income-internet-access

Van Rossum, G., & Drake, F. L. (2009). *Python 3 reference manual*. CreateSpace. ISBN: 1441412697

Vries, A. de, Chubaty, A., & Microsoft. (2022). *miniCRAN: Create a mini version of CRAN containing only selected packages* (Version 0.2.16) [Computer software]. https://cran.r-project.org/web/packages/miniCRAN/index.html

Warsaw, B. (2024). *Importlib-resources* (Version 6.4.0) [Computer software]. https://pypi.org/project/importlib-resources/