

# CWRU DSCI351-351M-451: Week 01b-p Bash Git BitBucket Agile Knuth

Profs: R. H. French, L. S. Bruckman, P. Leu, K. Davis, S. Cirlos

TAs: W. Oltjen, K. Hernandez, M. Li, M. Li, D. Colvin

06 September, 2022

## Contents

1.2.2.1	Class Readings, Assignments, Syllabus Topics . . . . .	2
1.2.2.1.1	Reading, Lab Exercises, SemProjects . . . . .	2
1.2.2.1.2	Textbooks . . . . .	2
1.2.2.1.3	Syllabus . . . . .	2
1.2.2.2	R Learning Resources . . . . .	2
1.2.2.2.1	SDLE Teatime Learning . . . . .	2
1.2.2.2.2	SDLE TeaTime Learnings Materials are available Online . . . . .	4
1.2.2.3	Punchlist: What we need setup for class work (repeat list) . . . . .	4
1.2.2.4	Bash: The language of the Linux Console . . . . .	5
1.2.2.4.1	On our Rstudio Server 4.2.1 on Markov we use Git in the terminal . . . . .	5
1.2.2.4.2	On our Win10 ODS Desktop's we use Git Bash to work with Git . . . . .	5
1.2.2.4.3	Lets see some Bash Commands we'll be using . . . . .	5
1.2.2.4.4	A good resource for Bash Commands and Man pages . . . . .	5
1.2.2.5	Now lets start working with our local Git Server . . . . .	5
1.2.2.5.1	Check your Git Server Configuration . . . . .	5
1.2.2.5.2	Essential <code>git config --global's</code> , Set your user info . . . . .	6
1.2.2.5.3	First we need to go up to Bitbucket and "Fork" the Prof. Repo . . . . .	6
1.2.2.5.4	Now you want to open Git Bash on Windows . . . . .	6
1.2.2.5.5	Important Note: <b>Windows ignores case, Linux and BSD-Unix (Mac) respect case</b> . . . . .	6
1.2.2.6	Now lets Clone your personal class repo . . . . .	7
1.2.2.6.1	For class repos . . . . .	7
1.2.2.6.2	Now lets pull and push changes from to your repo . . . . .	7
1.2.2.6.3	Making local changes, Adding, Committing and Pushing . . . . .	7
1.2.2.7	Some notes on using the ODS Tool Chain . . . . .	7
1.2.2.8	Markov & the SDLE Rstudio Server (rxfl31), vs. the ODS Desktop . . . . .	8
1.2.2.9	Agile Software Development . . . . .	9
1.2.2.9.1	The Agile Manifesto in 2001 . . . . .	9
1.2.2.9.2	The Rise and Evolution of Agile Software Development . . . . .	10
1.2.2.9.3	Git code versioning . . . . .	10
1.2.2.9.4	Slack, a component of Agile Software Development . . . . .	10
1.2.2.10	Donald Knuth, A computer scientist and Case Alumnus . . . . .	10
1.2.2.10.1	The Art of Computer Programming, by Donald Knuth . . . . .	10
1.2.2.10.2	LaTeX, by Donald Knuth . . . . .	11
1.2.2.11	Your Open Data Science Tool Chain . . . . .	11
1.2.2.11.1	Its all about a Data Science Tool Chain . . . . .	11
1.2.2.11.2	Pilot your DSCI studies using available data . . . . .	12

1.2.2.11.3 Use Git repositories . . . . .	12
1.2.2.12 Links . . . . .	12

## 1.2.2.1 Class Readings, Assignments, Syllabus Topics

### 1.2.2.1.1 Reading, Lab Exercises, SemProjects

- Readings
  - For today: PRP35-64
  - For next class: OIS 1,2
- Laboratory Exercises:
  - LE1: **due Tuesday Sept. 13th**
  - LE1 & LE2 are worth 7 points
  - 1 point/day deducted for late submission
- Office Hours: (Class Canvas Calendar for Zoom Link)
  - Wednesday @ 4:00 PM to 5:00 PM, Will Oltjen
  - Saturday @ 3:00 PM to 4:00 PM, Kristen Hernandez
  - **Office Hours are on Zoom, and recorded**
- Exams
  - MidTerm: Tuesday October 18th, in class or remote, 11:30 - 12:45 PM
  - Final: Monday December 19, 2022, 12:00PM - 3:00PM, Nord 356 or remote

### 1.2.2.1.2 Textbooks Introduction to R and Data Science

- For R, Coding, Inferential Statistics
  - Peng: R Programming for Data Science
  - Peng: Exploratory Data Analysis with R

Textbooks for this class

- OIS = Diez, Barr, Çetinkaya-Runde: Open Intro Stat v4
- R4DS = Wickham, Grolemund: R for Data Science

Textbooks for DSCI353/353M/453, And in your Repo now

- ISLR = James, Witten, Hastie, Tibshirani: Intro to Statistical Learning with R, 2nd Edition
- ESL = Trevor Hastie, Tibshirani, Friedman: Elements of Statistical Learning
- DLwR = Chollet, Allaire: Deep Learning with R

Magazine Articles about Deep Learning

- DL1 to DL13 are “Deep Learning” articles in 3-readings/2-articles/

### 1.2.2.1.3 Syllabus

### 1.2.2.2 R Learning Resources

- Peng: R Programming for Data Science (Book, in 3-readings)
- Roger Peng’s [Youtube Playlist for 4 weeks of Coursera R Programming](#)
- [LinkedIn Learning through CWRU](#)
  - [Lynda.com R Courses](#)

### 1.2.2.2.1 SDLE Teatime Learning

- 2016 year was intro to datascience, R, Python, Git, LaTeX
- 2017 was more advanced topics including Hadoop and Spark and SparklyR
- 2018 continued with more advanced topics and review

Day:Date	Foundation	Practicum	Reading	Due
w01a:Tu:8/30/22	ODS Tool Chain	R, Rstudio, Git		
w01b:Th:9/1/22	Setup ODS Tool Chain	Bash, Git, Slack, Agile	PRP4-33	LE1
w02a:Tu:9/6/22	What is Data Science	OIS:Intro2R, Git	PRP35-64	
w02b:Th:9/8/22	Summarizing Data	Intro2R	OIS1,2	
w02Pr:Fr:9/9/22			PRP65-93	<b>451 Update1</b>
w03a:Tu:9/13/22	Summarizing Data	Git, Rmds, Loops,	PRP94-116	LE2 LE1 Due
w03b:Th:9/15/22	Rand. Var. Normal Dist.	Data Analytic Style	OIS4	
w04a:Tu:9/20/22	Tidy Check Explore	Tidy GapMinder	EDA1-31	
w04b:Th:9/22/22	Inference, DSCI Process	Other Distrib. 7 ways	R4DS1-3	LE3 LE2 Due
w04Pr:Fr:9/23/22			EDA32-58	<b>451 Update2</b>
w05a:Tu:9/27/22	OIS4 Rand. Var.	EDA of PET Degr.	OIS5	
w05b:Th:9/29/22	OIS5 Found. of Infer.	Multivar Corr. Plot	R4DS4-6	
w05Pr:Fr:9/30/22				<b>451 RepOut1</b>
w06a:Tu:10/4/22	Pred., Algorithm, Model	Anscombe's Quartets	R4DS7-8	
w06b:Th:10/6/22	EDA stats, vis	Summ. Stats & Vis.	R4DS9-16	LE4 LE3 Due
w06Pr:Fr:10/7/22	Corr. Coeff. Pairs Plots			<b>451 Update3</b>
w07a:Tu:10/11/22	Confidence Intervals	Penguins	OIS6.1-2	PeerRv1 Due
w07b:Th:10/13/22	Midterm Rev.	Hypo.Test, Sampl. Dist.		
w08a:Tu:10/18/22	<b>MIDTERM</b>	<b>EXAM</b>		
w08b:Th:10/20/22	Programming & Coding	Coding Expect.		LE4 Due
w08Pr:Fr:10/21/22				<b>451 Update4</b>
Tu:10/24,25	<b>CWRU</b>	<b>FALL BREAK</b>	R4DS17-21	
w09b:Th:10/27/22	Cat. Inf. 1 & 2 propor.	Indep. Test, 2-way tables	OIS6.3-4	LE5
w09Pr:Fr:10/28/22				<b>451 RepOut2</b>
w10a:Tu:11/1/22	Goodness of Fit, $\chi^2$ test	t-tests 1&2 means	OIS7.1-4	
w10b:Th:11/3/22	Num. Infer, Cont. Tables	Stat. Power		
w10Pr:Fr:11/4/22				<b>451 Update5</b>
w11a:Tu:11/8/22	Sample & Effect Size	Stat. Power GGmap	OIS8	PeerRv2 Due
w11b:Th:11/10/22	Inf. 4 Regr, Test & Train	Curse of Dimen.	ISLR1,2.1,2	LE6 LE5 Due
w12a:Tu:11/15/22	Lin. Regr. Part 1	Residuals	OIS9	
w12b:Th:11/17/22	Lin. Regr. Part 2	Regr. Diagnostics		
w12Pr:Fr:11/18/22				<b>451 Update6</b>
w13a:Tu:11/22/22	Mult. Lin. Regr.	Var. & Mod. Selec.,	ISLR3.1	LE7 LE6 due
w13b:Th:11/24/22	Log. Regr.	GIS Trends	ISLR3.2	
w13Pr:Fr:11/25/22				<b>451 RepOut3</b>
w14a:Tu:11/23/22	Classificat., Sup. Lrning	Caret, Broom 4 modeling	ISLR4.1-3	
Th,Fr:11/24,25	<b>THANKSGIVING</b>	<b>Vacation</b>		
w15a:Tu:11/29/22		Clustering		PeerRv3 Due
w15b:Th:12/1/22	Big Data Analytics	Dist. Comp., Hadoop		
w15SPr:Fr:12/2/22		Read Article by	Mirletz,2015	
w16a:Tu:12/6/22	Final Exam Review			
w15b:Th:12/8/22				<b>LE7 due</b>
<b>Friday 12/12</b>	<b>SemProj</b>	<b>Final Report</b>		<b>SemProj4 due</b>
<b>Monday 12/19</b>	<b>FINAL EXAM</b>	<b>12:00-3:00pm</b>	Nord 356	or remote

Figure 1: IT Fundamentals: Applied Data Science with R, Syllabus

#### 1.2.2.2.2 SDLE TeaTime Learnings Materials are available Online

- [2018 SDLE Teatime Repo](#)
  - 2018 contains the prior years code
  - [2016 SDLE Teatime Repo](#)
  - [2017SDLE Teatime Repo](#)
- [SDLE Teatime Youtube Videos and Playlists](#)

#### 1.2.2.3 Punchlist: What we need setup for class work (repeat list)

- Setup our Markov and Open Data Science (ODS) Computers
  1. For **Markov Data Science Cluster**
    - login to <http://ondemand.case.edu> with your CaseID account
    - Launch the SDLE Rstudio Server (rxf131)
    - Check your “Library Paths”
      - \* in the R console
      - \* run `.libPaths()`
      - \* And the first directory MUST be
      - \* `“/home/rxf131/ondemand/ubuntu2004/r4” “/usr/local/lib/R/site-library”`
    - otherwise refer to the file in the root directory of your repo
      - \* named `FixRstudioServer-R-libPaths.txt`
      - \* and run the `“source(‘/home/rxf131/ondemand/share/config/r-lib-path-fix.R’)“`
      - \* In the R console
      - \* then check your `.libPaths()` again
    - On Markov, launch **LXDE Desktop** (rxf131)
      - \* make a Git folder under `/home/CaseID/`
        - \* Login to DSCI Slack in your firefox browser on LXDE desktop
  2. For the **Open Data Science (ODS) Desktop**
    - login to <http://myapps.case.edu> with your CaseID account
    - Drag icons of to your desktop
      - \* for R, Rstudio, Git Bash, VScodium, PyCharm, Jupyter Notebook, Slack
  3. Setup Git
    - make `/home/caseID/Git` folder on Markov
      - \* git config your name and email of your git server
    - make `H:\Git` folder on ODS Desktop
      - \* git config your name and email of your git server
  4. Git Fork the Class “Prof” Repo
    - In your Bitbucket Account
  5. Git Clone your Fork of the Class Repo
  6. When in Rstudio (on Markov or ODS)
    - Its ESSENTIAL that you open the `.Rproj` file in the upper right corner
    - this tells Rstudio where your root directory of your project is.
  7. Setup Bitbucket account
  8. Setup [DSCI Slack Account](#)
  9. Setup StackExchange account

#### 1.2.2.4 Bash: The language of the Linux Console

- Bash is the command line processor of the Linux Console
- R has its own command line processor for the R Console
- Bash is the default Console for both Linux and for Mac
  - Mac's are based on BSD-Unix OS
  - A close variant of Linux, only differentiated by the licensing
- Windows uses the DOS command line processor in its 'Command Prompt'

##### 1.2.2.4.1 On our Rstudio Server 4.2.1 on Markov we use Git in the terminal

- Terminal, like Konsole in KDE,
  - Uses the standard Bash commands
  - And tools like vim (the visual text editor)
- You can do this in the Rstudio terminal (not the R console)
  - Or you can start a "Markov Desktop KDE" In [ondemand.case.edu](http://ondemand.case.edu)

##### 1.2.2.4.2 On our Win10 ODS Desktop's we use Git Bash to work with Git

- MinGW64 is a little Linux OS running inside Windows
  - It has the standard Bash commands
  - And tools like vim (the visual text editor)

##### 1.2.2.4.3 Lets see some Bash Commands we'll be using

- `ls` is the "list" command,
  - to get a directory of files and folders
- `pwd` is the "present working directory" command,
  - to know where you are
- `cd` is "change directory"
- `..` refers to the directory one up from where you are
  - so `cd ..` moves you up one directory
- and `cd Downloads` would move you down
  - into Downloads directory (if it exists)
- To copy a file use `cp`
- To move a file use `mv`
- To make a new directory use `mkdir`

##### 1.2.2.4.4 A good resource for Bash Commands and Man pages

- Is [An A-Z Index of the Bash command line for Linux](#)
- There are many other resources too

#### 1.2.2.5 Now lets start working with our local Git Server

- Using Git Bash to talk to it
  - Git is also a linux program
- All Git commands are entered at the Bash Prompt
- All Git commands start with `git`
  - So that the Bash prompt know who to send the subsequent command to

##### 1.2.2.5.1 Check your Git Server Configuration

- `git config --list`

#### 1.2.2.5.2 Essential git config --global's, Set your user info

- `git config --global user.name "[name]"`
- `git config --global user.email "[email address]"`
- `git config --global color.ui auto`

#### 1.2.2.5.3 First we need to go up to Bitbucket and “Fork” the Prof. Repo

- Bitbucket will require you make an **Bitbucket App Password**
  - This is different from your password to login to Bitbucket website
  - And they show it to you only once
  - So record it
- Best is to store it in [Bitwarden](#)
  - An open-source Password Manager
- Now you need to **\*\*Fork the Prof Course Repo**
  - And change “-prof” to “-yourCaseID”
- This will give you a copy of Prof. Repo
  - In your personal account area on BitBucket website

We have the “Fa22-dsci51” group in the CWRU-DSCI bitbucket team

- So this group has access to our class prof repo
- [21f-dsci351-351m-451-e1451-e2451-prof](#)
- To Fork this repo, go to the repo’s page on bitbucket
  - On the repo page
  - Look at the plus on the left side bar
  - And choose “Fork this Repo”
- On the next Forking page
  - You want to remove “prof” from the repo name,
  - and put you caseID, as the end of your repo name
- Once you have forked the prof repo
  - You will want to clone it to your computer
  - Using http protocol, if you are on Windows
  - Or using ssh protocol, on Linux or Mac

#### 1.2.2.5.4 Now you want to open Git Bash on Windows

- You need to save your Repos on your H: drive, NOT C drive
  - C Drive is restricted
  - H Drive is your personal area that follows your caseID login
- So in Git Bash
  - `pwd` will tell you your present working directory
  - `cd ..` moves up a directory
  - `pwd` to see where you have moved
  - Now change to H: `cd /h`
  - `pwd` see where you are
  - `ls` see what files are there
  - `mkdir Git` this will make a new directory at H:Git
    - \* So you’ll keep all your repositories under H:Git

#### 1.2.2.5.5 Important Note: Windows ignores case, Linux and BSD-Unix (Mac) respect case

- So Git and git are the same on windows for a folder
- They are totally different on Linux or Mac
- Best practice Use capitals sparingly
- About only useful place is in CamelBack filenames

- Since I said, no spaces in filenames
- To make things readable, you can do CamelBack
- Example: 1906ITFun-ADSwR-ThisIsMyReport-Name.rmd

### 1.2.2.6 Now lets Clone your personal class repo

- Now you want to Clone your personal class repo
  - This is a one time operation
  - To copy all the files and folders down to your local computer
- In Git Bash, you want to be at H:Git or h:Git Check with `pwd`
- Now go to your personal class repo on Bitbucket
  - And find the clone command
  - **For windows choose https protocol (Not ssh)**
- Copy the command
  - Its something like this
  - `git clone https://vuvlab@bitbucket.org/cwrudsci/21f-dsci351-351m-451-e1451-e2451-prof.git`
- Now that that is on your clipboard
- Go to you Git Bash, and use “Shift-Insert” (Not “Cntrl-v”)
  - To copy it onto the Bash Command line
  - Hit enter, and watch a full copy of your repo being copied locally

#### 1.2.2.6.1 For class repos

- Before each class, or whenever you want
- Up on Bitbucket
  - You should sync your fork
  - With my Prof repo
  - To get the latest file version and new files
  - After syncing
  - Now `git pull` to bring the updated files to your local git server

#### 1.2.2.6.2 Now lets pull and push changes from to your repo

- `cd` into your repo’s top folder
  - This can be done with tab completion
  - `cd 19su-t` and hit tab, it auto completes
- Now type `git pull` To see if there are any changes up on bitbucket And to pull these down and merge them in

#### 1.2.2.6.3 Making local changes, Adding, Commmitting and Pushing

- Now change a local file by adding something into it
- Now you add this changed file to be tracked by Git
  - `git add --all :/`
- Now commit your changes
  - `git commit -m 'I have changed the readme.md'`
- Now push your changes up to Bitbucket, to your personal repo
  - `git push`

### 1.2.2.7 Some notes on using the ODS Tool Chain

- So if someone installs R programming language, and Rstudio IDE,
  - then they can run R commands and R scripts.

But in the computers we configure and use in the DSCI classes

- the SDLE Rstudio Server-4.2.1 on Markov datascience cluster
  - you access using `ondemand.case.edu`,
- or the ODS Desktop
  - on the Win10 Citrix server
  - that you access using `myapps.case.edu`,
- or if you follow section 11 in the course syllabus
  - and configure you own notebook that runs linux, mac or windows

We don't just have R & Rstudio.

- We install the following things that don't come with just R/Rstudio.
- git for code versioning and collaboration
- A LaTeX distribution
  - (either TexLive on Linux, MikTeX on windows, or MacTeX on Mac)
- A gui LaTeX Editor called TexStudio
- R version 4.2.1 (have to also add Rtool4 to add compilers to windows)
- Rstudio latest version “2022.07.1 Build 554”
- A set of about 500 R packages,
  - which give us all the data analysis tools we will use in the class.

So if just having installed R and Rstudio,

- you won't have setup for the work in this course.
  - You won't have git,
  - you won't have LaTeX
    - \* so you can't compile your “R markdown” (`filename.Rmd`) scripts
    - \* to produce a pdf report,
  - and you won't have all the modeling and graphic packages
    - \* we use routinely in our coding.
- We refer to this as the Open Data Science Tool Chain,
  - and both SDLE Rstudio Server (`rxfl31`) on Markov, and the ODS Desktop on `myapps` are already setup.
- Since it takes time for us to get everyone up and running
  - with accounts,
  - and configurations,
  - forks and clones,
- We suggest leaving your “setting up my personal notebook computer
  - for use as an Open Data Science computer,
  - for a month or two.

#### 1.2.2.8 Markov & the SDLE Rstudio Server (`rxfl31`), vs. the ODS Desktop

- So on the ODS Desktop (i.e. Win10)
  - your personal files are kept on `H:`,
  - so making `H:\Git\` folder
  - and then use the command `cd /h/Git`
    - will make your “present working directory” (`pwd`) be that Git folder.
- On Markov, You can launch a “Markov Desktop (LXDE)” session.
  - And then you have a graphical file manager, and terminal.
- On Markov, you can also do the steps in the terminal (not the R console)
  - of the “SDLE Rstudio Server (`rxfl31`)”
  - that you will see in the bottom left quadrant of Rstudio.
- In this case this is a linux terminal,
  - but you don't have a graphical desktop environment for the computer.



- And the on Markov,
  - you want to make a Git folder under your “home” directory.
  - Your “home” directory on Markov is `/home/caseID/`
- So you want to put your Git repos on Markov
  - in the folder `“/home/caseID/Git”`
- And the first time you want to copy your personal fork of the “Prof” repo,
  - you need to clone it.
- So when you Clone your personal class repo.
  - You want to use the https (normal web) protocol.
  - The ssh protocol is more complex to setup,
    - \* so the default ssh protocol when you first clone won’t work,
    - \* because you dont have any configured ssh keys.
- So Clone the repo,
  - by being in the directory Git folder, on H in the ODS Desktop,
  - or being in the directory `/home/caseID/Git` folder on Markov.

So the order of doing things

- Fork the prof repo
  - and remove the “prof” and
  - replace with your caseID in the repo name.
- Clone the repo,
  - selecting the http protocol in the upper right of the Clone dialog box,
  - to get and copy the Clone command.
- And now using that command,
  - go into a Git folder in a sensible location
    - \* of your Markov cloud computer,
    - \* or ODS Desktop cloud computer,
  - or to have a copy of the repo on your personal notebook,
    - \* into a Git folder on your notebook computer.
- Run the git clone command, which will look like this:
  - `git clone https://vuvlab@bitbucket.org/cwrudsci/22f-dsci354-454-CASEID.git`
- After you have cloned your personal repo,
  - (probably to these 3 different computers),
  - you have a full copy of the repo on each.
- After the first time you clone it.
  - Everything after that will be `git pull ... , git push ...`
    - \* to pull down new files or push up new files to and from bitbucket.

### 1.2.2.9 Agile Software Development

#### 1.2.2.9.1 The Agile Manifesto in 2001

- <http://agilemanifesto.org/>
  - [Twelve Principles of Agile Software](#)
- Has changed how software is developed
  - From large, monolithic programs, with complex “specifications”
  - To small, package-based, codes that integrate for applications
- e.g. How google updates apps in an ongoing basis
  - Numerous small updates and changes
  - Instead of large multi-year releases
- Linux is a package-based operating system
  - Windows used to be monolithic
  - And is trying to become more agile

- Agile Development, has now been generalized for more than software
  - Many companies use Agile Processes in their businesses

#### 1.2.2.9.2 The Rise and Evolution of Agile Software Development

- Article by Hoda, Salleh, Grundy in your repo 3-readings/2-articles
  - Agile software development has dominated the second half of the past 50 years of software engineering.
  - Retrospectives, one of the most common agile practices, enables reflection on past performance, discussion of current progress, and charting forth directions for future improvement.
  - Because of agile's burgeoning popularity as the software development model of choice and a significant research subdomain of software engineering, it demands a retrospective of its own.
  - This article provides a historical overview of agile's main focus areas and a holistic synthesis of its trends, their evolution over the past two decades, agile's current status, and, forecast from these, agile's likely future.
  - This article is part of a theme issue on software engineering's 50th anniversary.

#### 1.2.2.9.3 Git code versioning

- Enables large teams to work on software development
- [For example the Linux Kernel](#)

#### 1.2.2.9.4 Slack, a component of Agile Software Development

- An intrinsic part of agile software development
  - There is Slack app for phones
  - And client for computers, its on vdi.
- an online collaboration tool

CWRU Data Science Slack

- We have a CWRU Data Science Slack
- You can join by signing up at <https://cwr-dsci.slack.com>
  - Sign up using your case.edu email address
- There is a specific class channel,
  - along with general,
  - applied data science
  - and job opportunities channels.

The class Slack channel

- is a good place for discussing questions
- from class and/or assignments.

Class communications will be via Slack

#### 1.2.2.10 Donald Knuth, A computer scientist and Case Alumnus [Donald Knuth](#)

- Alma mater
  - Case Institute of Technology (B.S., M.S.) 1960
  - California Institute of Technology (Ph.D.)
- [“Why I choose to go to Case Institute of Technology](#)

#### 1.2.2.10.1 The Art of Computer Programming, by Donald Knuth

- Major things he's done in Computing

[The Art of Computer Programming](#)

- (sometimes known by its initials TAOCP)
- covers many kinds of programming algorithms and their analysis.

[Literate Programming](#), was another of his goals

- Literate programming is a programming paradigm introduced by Donald Knuth
  - in which a program is given as an explanation of the program logic in a natural language, such as English,
  - interspersed with snippets of macros and traditional source code, from which a compilable source code can be generated.

The literate programming paradigm, as conceived by Knuth,

- represents a move away from writing programs
  - in the manner and order imposed by the computer,
  - and instead enables programmers to develop programs in the order demanded by the logic and flow of their thoughts.
- Literate programs are written as an uninterrupted exposition of logic
  - in an ordinary human language, much like the text of an essay,
  - in which macros are included to hide abstractions and traditional source code.
- Literate programming (LP) tools are used to obtain two representations from a literate source file:
  - one suitable for further compilation or execution by a computer, the “tangled” code,
  - and another for viewing as formatted documentation, which is said to be “woven” from the literate source.
- While the first generation of literate programming tools
  - were computer language-specific,
  - the later ones are language-agnostic
  - and exist above the programming languages.

[Now adays one can integrate R and Python code in a common shared environmet, as can be done with Rstudio v1.2 and the reticulate package. We use this in our data analytics in the SDLE Research Center at CWRU.]

### 1.2.2.10.2 LaTeX, by Donald Knuth

- [LaTeX publishing system](#)
  - A typesetting system designed & written by Donald Knuth, released in 1978.
  - TeX was designed with two main goals in mind:
    - \* to allow anybody to produce high-quality books using minimal effort, and
    - \* to provide a system that would give exactly the same results on all computers, at any point in time.[2]
  - TeX is free software, which made it accessible to a wide range of users

### 1.2.2.11 Your Open Data Science Tool Chain

#### 1.2.2.11.1 Its all about a Data Science Tool Chain

- Use R and build on the communities foundation
- Use Rstudio as a comfy environment
- Share your Open Data and Open Source Code
- Produce Reproducible Science with Rmarkdown
  - Use [Creative Commons Licenses](#)
  - Or other [Open Source Licenses](#)
  - Such as the [Gnu Public License: GPL](#)

#### 1.2.2.11.2 Pilot your DSCI studies using available data

- Find available data sets
- Before starting the costly process of making data

#### 1.2.2.11.3 Use Git repositories

- For version control
- For Collaboration
- For Open Science sharing

#### 1.2.2.12 Links

- <https://www.r-project.org>
- <https://help.ubuntu.com/community/UsingTheTerminal>
- <http://www.r-project.org>
- Rory Winston, for the [Learning R Intro](#)
- StackExchange <http://stackexchange.com/sites>
- Twitter <http://twitter.com>
- Agile Manifesto <http://agilemanifesto.org/principles.html>
- Slack <http://slack.com>
- [emelineliu.com](http://emelineliu.com)
- [Github Pages](#)
- [Jekyll](#)
- [Poole](#)
- R. Hoda, N. Salleh, J. Grundy, [The Rise and Evolution of Agile Software Development](#), IEEE Software. 35 (2018) 58–63.