

# Course and Platform Introduction

UBCO Master of Data Science – DATA 530

*I respectfully acknowledge that UBC Okanagan is situated on the traditional, unceded and ancestral territory of the Syilx Okanagan Nation*



# Objectives

---

- Understand MDS structure and goals
- List the different software used and their purpose
- Define command line and list some of its uses
- Know how to open the command line window on Mac OS and Windows
- Define: file system, folder, file
- Explain the difference between an absolute and relative path
- Define Git/GitHub and explain why they are useful for analysts
- Access the web interface of GitHub
- Using command line clone repository, add files, commit changes, and push to origin using git commands
- View and edit text files directly on GitHub

# Introductions

---

Instructor: Dr. Ifeoma Adaji

- Assistant Professor, Computer Science since July 2021
- Ph.D., Computer Science, University of Saskatchewan, Canada, 2020
- M.Sc., Computer Science, University of Aberdeen, Scotland, UK
- Research area: Data science, social network analysis, user modeling, machine learning
- Canadian Industry experience: Data scientist at Insightrix Research

TA: Yining Zhou

# Setup iClicker

---

<https://lthub.ubc.ca/guides/iclicker-cloud-student-guide/>

**Search for Data 530-2023**

# iClicker - Anonymous

---

Where are you from?

- A. British Columbia, Canada
- B. Other provinces, Canada
- C. North America
- D. Outside North America

# iClicker - Anonymous

---

Why are you here?

- A. I'm changing careers from non-tech
- B. I'm changing careers from a tech related career to data science
- C. I'm a self-taught data scientist but want formal training
- D. I'm fresh out of school; I want a second degree
- E. Not sure

# iClicker - Anonymous

---

How will you measure your performance/success at the MDS program?

- A. If I score high grades
- B. If I get a job after the program
- C. If I get the job I want after the program

# UBC Okanagan Master of Data Science

---

The overall goal is to:

**Master data analytical skills, tools, and techniques desired by employers to obtain employment**

It is essential that you are dedicated full-time to learn these skills as you will get a job based on what you can do.

A UBC degree will help you get an interview, but you must demonstrate your abilities to get the job.



# Course Objectives

---

- 1) Install and setup a variety of software tools and programs used by data analysts
- 2) Perform basic and advanced data analysis and visualization in Excel
- 3) Setup IDEs and write small programs in Python and R
- 4) Understand the pros and cons of each tool/software package and criteria to select the best tool for the job

# UBC Okanagan Master of Data Science Evaluation

---

This course will consist of 8 lectures and 4 labs.

Marks are assigned for labs, quizzes, and iClickers exercises.

- Not all courses have the same marking breakdown and criteria

A grade below 60% is a fail. If you have more than 2 courses with grades below 68%, continuation in the program is not guaranteed.

# GitHub and Canvas

---

GitHub will be used extensively for distributing materials and collecting assignments.

- Personal information (except for your name) and student numbers should not be put on GitHub for privacy reasons.

Instructions on how to use GitHub will be provided in the lab and lectures.

UBC learning system called Canvas will be used to distribute grades.

# Academic Dishonesty

---

Cheating is strictly prohibited and is taken very seriously by UBC.

A guideline to what constitutes cheating:

- Labs
  - Submitting code produced by others.
  - Working in groups to solve questions and/or comparing answers to questions once they have been solved (except for group assignments).
  - Discussing HOW to solve a particular question instead of WHAT the question involves.
- Quizzes
  - Only materials permitted by instructor should be in the quizzes.

Academic dishonesty may result in a "F" for the course and removal from the MDS program.

# How to Excel in This Course

---

Attend **every** class:

- Read notes **before** class as preparation and try the questions.
- Participate in class exercises and questions.

Attend and complete all labs:

- Labs practice the fundamental employable skills as well as being for marks.

Practice on your own. Practice makes *improvement*.

- Do more questions than in the labs.
- Read the additional reference material and perform practice questions.

# Systems and Tools

---

Course material is on GitHub.

Marks are distributed on Canvas.

Your laptop will be used to install all software and run programs.

# The Lab Assignments

---

Weekly lab assignments are worth **30%** of your overall grade.

Lab assignments may take more than the two hours lab time.

You have until the deadline to complete each lab.

- No late labs will be accepted.
- A lab may be handed in any time before the due date and may be marked immediately by the TA in the lab.

Lab assignments are done individually.

The lab assignments are critical to learning the material and are designed both to prepare you for the exams and build up your skills!

# The In-Class iClicker Quizzes

---

To promote understanding, 10% of your overall grade is allocated to answering in-class questions.

These questions are answered electronically using the iClicker.

Instructions to set up an iClicker account are in this week's lab

- <https://lthub.ubc.ca/guides/iclicker-cloud-student-guide/>
- **Search for Data 530-2023**

There will be at least 50 questions throughout the semester. Each question is worth 1 mark, and you need at least 40 right answers to get the full 10%.

- That is, if you answer 30 questions right, you get 30/40 or 75%.



# Tools: Have you used Git/GitHub?

---

A) Yes

B) No

# Tools: Have you used Excel?

---

A) Yes

B) No

# Tools: Have you used Python?

---

A) Yes

B) No

# Tools: Have you used R?

---

A) Yes

B) No

# What Grade are You Expecting to Get?

---

A) A

B) B

C) C

D) D

E) F

# Why This Course is Important

---

One of the hardest and most time-consuming parts of data analysis is installing and configuring your development environment.

Understanding a variety of tools and how to deploy them provides independence and ability to work without support (or engage support when needed).

Future courses will build upon this knowledge and use these tools extensively.

# Key Data Analysis Tools Used

Software	Purpose	Reasons
Python (Jupyter)	General purpose programming	Libraries for data analysis, stats, machine learning, visualizations. Standard IDE.
R and RStudio	Programming for analysis/stats	Popular open source language. RStudio is easy to use IDE. Popular for stats analysis.
Git and GitHub	Version/change management code/data distribution	Industry standard skills used in technology domain. Most popular GUI for Git.
Command line	File management, installation, scripts	Automation and control not possible with GUI
Excel	Rapid analysis and sharing	Widespread tool for analysis. Speed/flexibility
Text editors	Edit code and data files	Allows editing of any type of file.

# Why learn the Command Line?

---

The *command line* is the text interface to the computer.

Understanding the command line allows you to interact with the computer in ways that you often cannot with the user interface.

The command line is commonly used for scripting and automation of tasks and when accessing remote systems.





# What is the Command Line?

---

The **command line** is the text interface to the computer that accepts commands that the computer will execute. These commands include:

- starting programs
- navigating directories and manipulating files
- searching, sorting, and editing text files
- system and environment configuration

The command line is part of the **operating system**, which is software that manages your computer including all devices and programs.

- Common operating systems include Windows, Mac OS, and Linux/Unix.

# Windows Command Line

---

The command line on Windows dates back to the original Microsoft operating system called **DOS (Disk Operating System)** in 1981.

This command line interface is still part of all modern Windows operating systems and is accessible as the "Command Prompt".



It is commonly used for system administration and scripting.

# Command Line - Windows

```

Command Prompt
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\rlawrenc>cd
C:\Users\rlawrenc

C:\Users\rlawrenc>echo Hello
Hello

C:\Users\rlawrenc>mkdir 530

C:\Users\rlawrenc>cd 530

C:\Users\rlawrenc\530>notepad test.txt

C:\Users\rlawrenc\530>dir
Volume in drive C has no label.
Volume Serial Number is 4457-AA3B

Directory of C:\Users\rlawrenc\530

2018-09-04  01:32 PM    <DIR>          .
2018-09-04  01:32 PM    <DIR>          ..
2018-09-04  01:32 PM                5 test.txt
               1 File(s)                5 bytes
               2 Dir(s) 140,533,850,112 bytes free

C:\Users\rlawrenc\530>more test.txt
Hello

C:\Users\rlawrenc\530>del test.txt

C:\Users\rlawrenc\530>cd ..

C:\Users\rlawrenc>rmdir 530

C:\Users\rlawrenc>

```

# Mac OS Command Line

The command line for Mac OS uses the same commands as Linux. It can be opened using `Finder` then `Utilities` then `Terminal`.

```
A4003869:~ rlawrenc$ pwd
/Users/rlawrenc
A4003869:~ rlawrenc$ echo Hello
Hello
A4003869:~ rlawrenc$ mkdir 530
A4003869:~ rlawrenc$ cd 530
A4003869:530 rlawrenc$ atom test.txt
A4003869:530 rlawrenc$ ls
test.txt
A4003869:530 rlawrenc$ cat test.txt
This is a test!
A4003869:530 rlawrenc$ rm test.txt
A4003869:530 rlawrenc$ cd ..
A4003869:~ rlawrenc$ rmdir 530
A4003869:~ rlawrenc$
```

# Entering a Command

Enter a **command** at a **prompt**.

- The prompt may be a `>` or a `$` or customized by the user.

Press ENTER to execute the command.

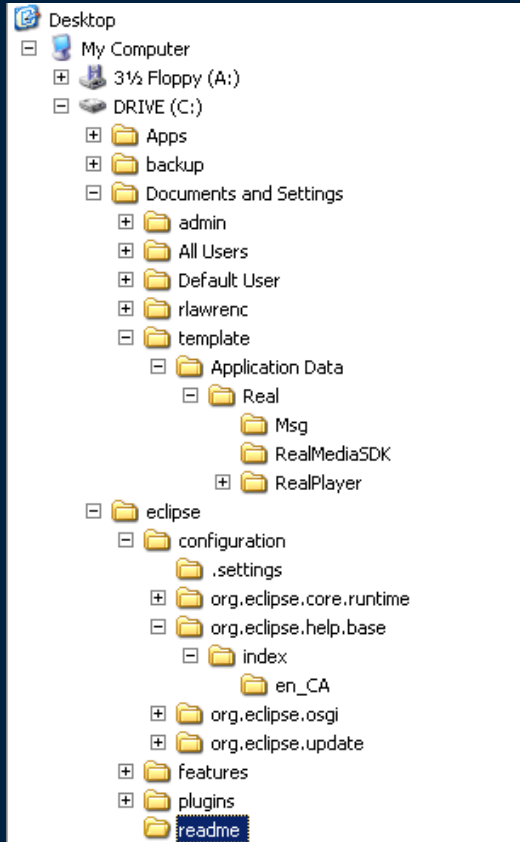
On Windows, commands are mostly case-insensitive while on Mac/Linux they are case-sensitive.

```

rlawrenc — -bash — 56x18

A4003869:~ rlawrenc$ pwd
/Users/rlawrenc
A4003869:~ rlawrenc$ echo Hello
Hello
A4003869:~ rlawrenc$ mkdir 530
A4003869:~ rlawrenc$ cd 530
A4003869:530 rlawrenc$ atom test.txt
A4003869:530 rlawrenc$ ls
test.txt
A4003869:530 rlawrenc$ cat test.txt
This is a test!
A4003869:530 rlawrenc$ rm test.txt
A4003869:530 rlawrenc$ cd ..
A4003869:~ rlawrenc$ rmdir 530
A4003869:~ rlawrenc$
  
```

# File System



The **file system** organizes data on a device as a hierarchy of directories and files.

Each **folder** (directory) has a name and can contain any number of files or subdirectories.

Each **file** has a name.

The user can change (navigate) directories in the hierarchy.

# Absolute versus Relative Paths

The **root** of the file system is the directory `" / "`.

- There is only one root of a directory hierarchy.

A path to a new location (from your current location) can be specified as an **absolute path** from the root:

```
cd /Users/iadaji/Documents/Data 530/folder
```

or a **relative path** from your current location:

```
cd Data 530/folder
```

To back up one directory level, use the command: `cd ..`

# Navigating relative paths

---

. is the short-form for the *current directory*

.. signifies the *parent directory* of the *current directory*

For example, to navigate (i.e. change directories) to the parent directory of the current directory, use the command: `cd ..`

Note that this command is dependent on your current directory (i.e. the folder you are currently in).

To print your current working directory type `pwd/cd` (Mac/Windows) then ENTER

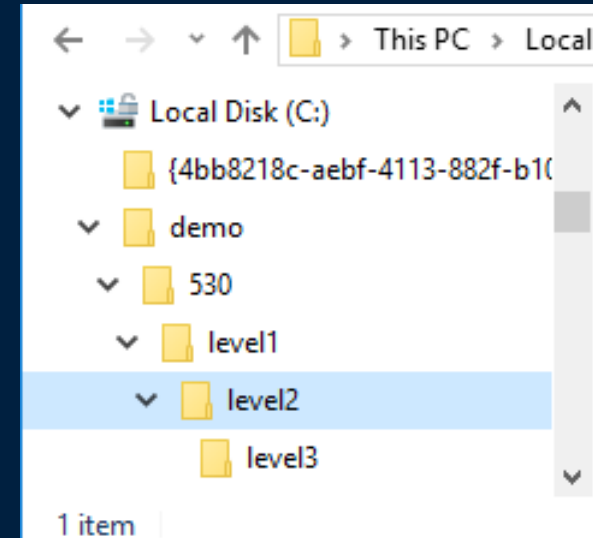


# Absolute versus Relative Path Question

**Question:** Given this directory hierarchy and that the user is currently in the directory `level2` and `level1` directory contains a file `test.txt`. How many of the following statements are **TRUE**?

- 1) A relative path to change to directory `530` is `..`
- 2) Absolute path to `test.txt` is `/demo/530/level1/test.txt`
- 3) Relative path to `test.txt` is `../test.txt`
- 4) Relative path to `test.txt` is different if user was currently in `level3` directory.
- 5) There is only one root of the directory hierarchy.

**A) 0    B) 1    C) 2    D) 3    E) 4**



# Commonly Used File Navigation Commands

	Windows	Mac OS and Linux
List contents of directory	<code>dir</code>	<code>ls</code>
Change directory	<code>cd 530</code>	<code>cd 530</code>
Print working directory	<code>cd</code>	<code>pwd</code>
Make a directory	<code>mkdir 530</code>	<code>mkdir 530</code>
Remove a directory	<code>rmdir 530</code>	<code>rmdir 530</code>
Rename a file	<code>ren old.txt new.txt</code>	<code>mv old.txt new.txt</code>
Remove a file	<code>del file.txt</code>	<code>rm file.txt</code>
Copy a file	<code>copy src.txt dest.txt</code>	<code>cp src.txt dest.txt</code>

# Commonly Used Text Related Commands

	Windows	Mac OS and Linux
Open a text editor	notepad	nano or atom
Echo output	echo <i>Hello</i>	echo <i>Hello</i>
Output contents of a file	more <i>file.txt</i>	cat <i>file.txt</i>
Search text files	find	grep
Sort text files	sort	sort

# Try it: Navigating Directories with Commands

---

**Question:** Using a terminal window on your computer, perform the following actions:

- 1) Create a directory called 530.
- 2) Change into the directory 530.
- 3) Echo I am awesome!
- 4) Show your current directory (print working directory).
- 5) Create a text file called `message.txt` with a message in it.
- 6) List the contents of your directory.
- 7) Rename the file `message.txt` to `test.txt`. Verify the name change.
- 8) Delete the `test.txt` file.
- 9) Change directory to directory above 530.
- 10) Delete directory 530.

# Why learn Git and GitHub?

---

**Git** is a version control system for managing files.

- Used by developers and analysts to store, exchange, and track files.
- Files are grouped into a **repository**.

**GitHub** is one of the most popular hosting services for Git with an easy-to-use interface.

- Git/GitHub provides version control, storage, and communication and coordination between collaborators.

GitHub will be used to distribute materials and collect assignments.

ubco-mds-2022 / Data-530

Type to search

<> Code

Issues

Pull requests

Actions

Projects

Security

Insights

Settings

Data-530

Private

Watch 0

Fork 0

Star 0

main

1 branch

0 tags

Go to file

Add file

<> Code

iadaji Update README.md

d6a2463 on Sep 25, 2022 34 commits

Labs

Lab4

last year

Lectures

Lectures

last year

README.md

Update README.md

last year

README.md

## Data-530: Computing Platforms for Data Science

Introduction to software and tools for Data Science. Setup process.

### Schedule

**Lecture:** Monday/Wednesday 9:30am to 11:00am in SCI 396  
**Lab:** Tuesday 1:30pm to 3:30pm in SCI 396

Date	Topic	Reading
2022-09-07	Course and Platform Introduction	GitHub:Cloning
2022-09-		

### About

Computing Platforms for Data Science

- Readme
- Activity
- 0 stars
- 0 watching
- 0 forks

### Releases

No releases published  
[Create a new release](#)

### Packages

No packages published  
[Publish your first package](#)

### Contributors 2

- iadaji
- Christel0315 Christel Deas

# Git Clone

Find repository on website and click on "Clone" button.

The screenshot shows the GitHub repository page for 'Data-530' (Private). The 'Code' button is highlighted, and the dropdown menu is open, showing the 'Clone' option. The repository is owned by 'iadaji' and has 1 branch and 0 tags. The repository description is 'Computing Platforms for Data Science'. The 'About' section lists 'Readme', 'Activity', '0 stars', '0 watching', and '0 forks'. The 'Releases' section shows 'No releases published' and a link to 'Create a new release'. The 'Packages' section shows 'No packages published' and a link to 'Publish your first package'. The 'Contributors' section lists 'iadaji' and 'Christel0315 Christel Deas'.

**Data-530: Computing Platforms**

Introduction to software and tools for Data Science. Setup process.

**Schedule**

**Lecture:** Monday/Wednesday 9:30am to 11:00am in SCI 396  
**Lab:** Tuesday 1:30pm to 3:30pm in SCI 396

Date	Topic	Reading
2022-09-07	Course and Platform Introduction	GitHub: Cloning
2022-09-		

# Installing Git Bash

---

To use Git on command line, follow these install instructions:

## Mac Users

- Open Terminal and run the command: `$ xcode-select --install`
- This will install git and many other very useful applications as well.

## Windows Users

- Go to <https://git-scm.com/downloads>.
- Click on Windows download link and accept all defaults in the install process.
  - Note that when selecting the default editor during the install, you may want to select Notepad++, if you are a Windows user and prefer to use Notepad++ over Atom.



# Git Bash

```

MINGW64/c/Users/r1awrenc/gitdemo/data-530-lab-1-r1awrenc
r1awrenc@A4010049 MINGW64 ~
$ mkdir gitdemo

r1awrenc@A4010049 MINGW64 ~
$ cd gitdemo

r1awrenc@A4010049 MINGW64 ~/gitdemo
$ git clone https://github.com/ubco-mds-2019-labs/data-530-lab-1-r1awrenc
Cloning into 'data-530-lab-1-r1awrenc'...
warning: You appear to have cloned an empty repository.

r1awrenc@A4010049 MINGW64 ~/gitdemo
$ dir
data-530-lab-1-r1awrenc

r1awrenc@A4010049 MINGW64 ~/gitdemo
$ cd data-530-lab-1-r1awrenc/

r1awrenc@A4010049 MINGW64 ~/gitdemo/data-530-lab-1-r1awrenc (master)
$ git add .

r1awrenc@A4010049 MINGW64 ~/gitdemo/data-530-lab-1-r1awrenc (master)
$ git commit -m "Python screenshot"
[master (root-commit) 62e9de1] Python screenshot
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 python_screenshot.jpg

r1awrenc@A4010049 MINGW64 ~/gitdemo/data-530-lab-1-r1awrenc (master)
$ git push
Counting objects: 3, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 113.58 KiB | 22.72 MiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/ubco-mds-2019-labs/data-530-lab-1-r1awrenc
 * [new branch]      master -> master

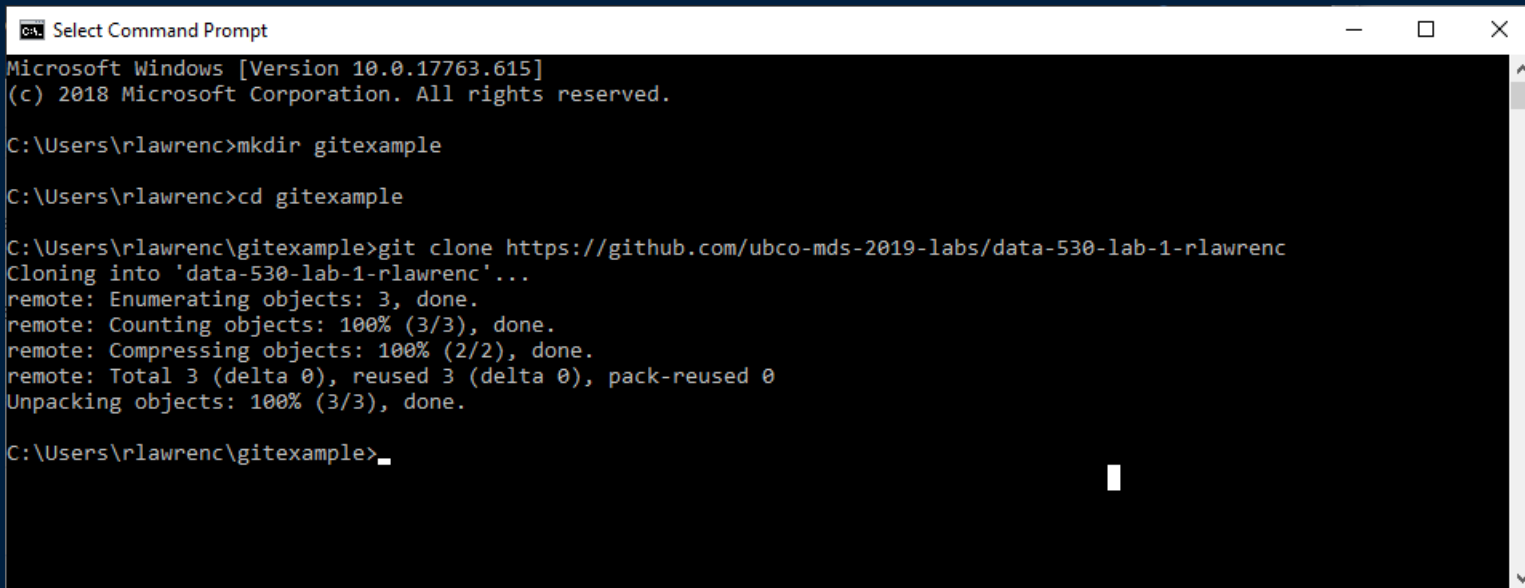
r1awrenc@A4010049 MINGW64 ~/gitdemo/data-530-lab-1-r1awrenc (master)
$

```

# Git Example: Clone

Use command-line to clone repository to local folder:

```
git clone <repo>
```



```
Select Command Prompt
Microsoft Windows [Version 10.0.17763.615]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\rlawrenc>mkdir gitexample

C:\Users\rlawrenc>cd gitexample

C:\Users\rlawrenc\gitexample>git clone https://github.com/ubco-mds-2019-labs/data-530-lab-1-rlawrenc
Cloning into 'data-530-lab-1-rlawrenc'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.

C:\Users\rlawrenc\gitexample>_
```

# Git Example: Add and Commit

---

Create and modify files in repository.

To add them to the repository run:

```
git add <files>
```

```
git add .
```

To commit to repository run:

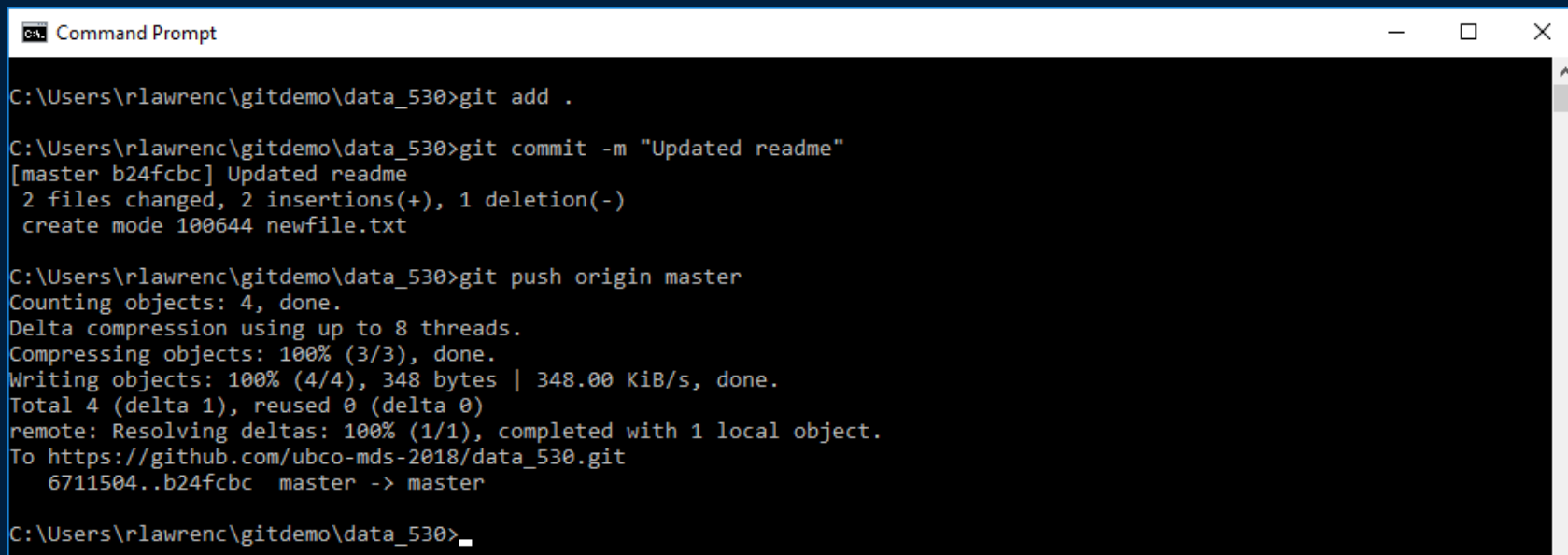
```
git commit -m "Commit message on changes"
```

# Git Example: Push

To push them to remote repository:

```
git push origin master
```

```
git push
```



```

C:\Users\rllawrenc\gitdemo\data_530>git add .

C:\Users\rllawrenc\gitdemo\data_530>git commit -m "Updated readme"
[master b24fcabc] Updated readme
 2 files changed, 2 insertions(+), 1 deletion(-)
 create mode 100644 newfile.txt

C:\Users\rllawrenc\gitdemo\data_530>git push origin master
Counting objects: 4, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 348 bytes | 348.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/ubco-mds-2018/data_530.git
 6711504..b24fcabc  master -> master

C:\Users\rllawrenc\gitdemo\data_530>_
  
```

# Try it: Clone DATA\_530 Repository

---

**Question:** Interact with GitHub using web interface and command line:

- 1) Navigate on [GitHub.com](https://github.com/ubco-mds-2023/Data-530) to `ubco-mds-2023/Data-530` repository.
- 2) Click `Clone` or `download` button to get URL.
- 3) Using command line to clone repository.
- 4) Edit the `README.md` file and add a new `test.txt` file.
- 5) Run `git add .`
- 6) Commit changes locally.
- 7) Try to push changes to GitHub. What happens?
- 8) Create your own repository on GitHub and repeat steps.

# Conclusion

---

The Master of Data Science is a professional program designed to train you on skills demanded by employers for data analysts.

- Requires dedicated, continual effort and practice.

The **command line** is the text interface to the computer that accepts commands that the computer will execute including running programs, manipulating files, and running scripts.

- The command line allows for automation and more control than may be available in the user interface.

**Git** is a version control system. **GitHub** is a hosted web service for Git. Version control allows for easy tracking, sharing, and communicating of data and files.



THE UNIVERSITY OF BRITISH COLUMBIA

