



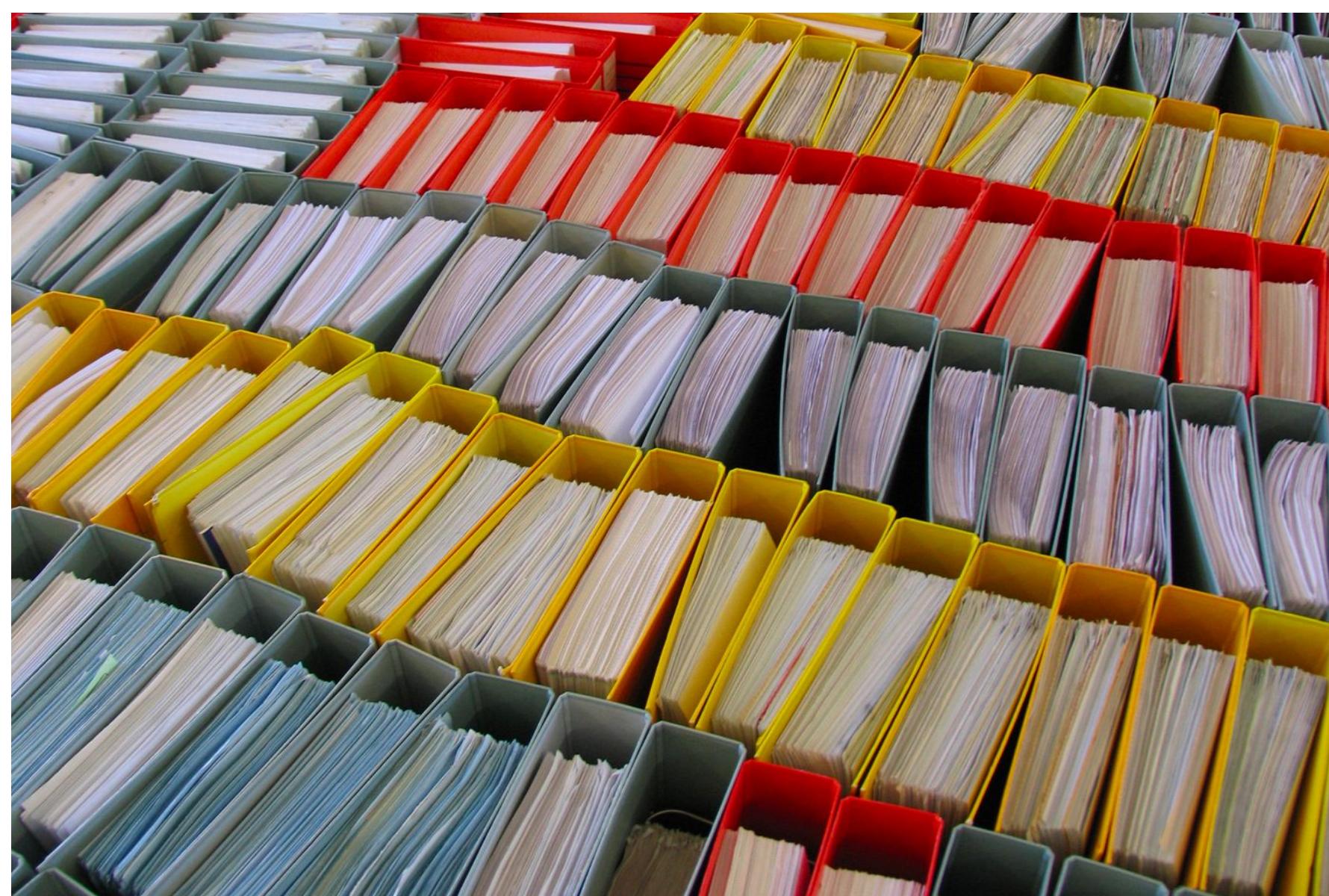
Research Data Management Techniques

⌚ 2.5 hours

🔧 No Prerequisites

📍 At your campus

\$ No cost to members



Why do this course?

Are you drowning in research data? Do you want to know where you should be storing your data? Are you required to comply with funding body data management requirements, but don't know how?

This workshop is ideal for researchers who want to know how research data management can support project success and are interested in research data management services and support available at their institution. Combining slide-based background material, discussions, and case studies this workshop will equip participants with best practices for managing their valuable research data.

You'll learn about:

- How to manage research data according to legal, statutory, ethical, funding body and university requirements.
- Approaches to planning, collecting, organising, managing, storing, backing up, preserving, and sharing your data.
- Services supporting research data at your institution.

Preparation

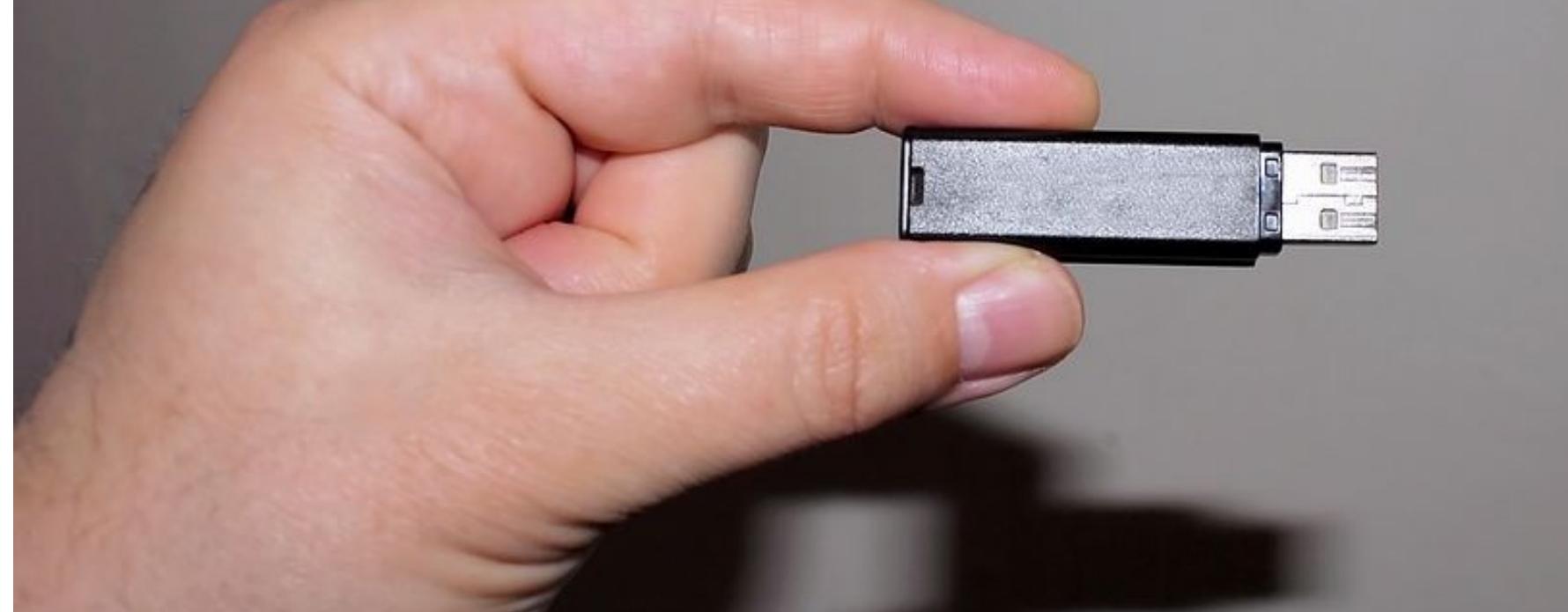
Bring along your laptop and any questions you may have on research data management.

The Intersect approach to training

At Intersect, we work closely with our member universities to develop and deliver training that targets the day-to-day software and technology problems that researchers face. We deliver hands-on courses in a relaxed setting with knowledgeable, helpful trainers who are themselves researchers and who know how researchers work.

Questions always welcome.

For more information visit
Learn.intersect.org.au



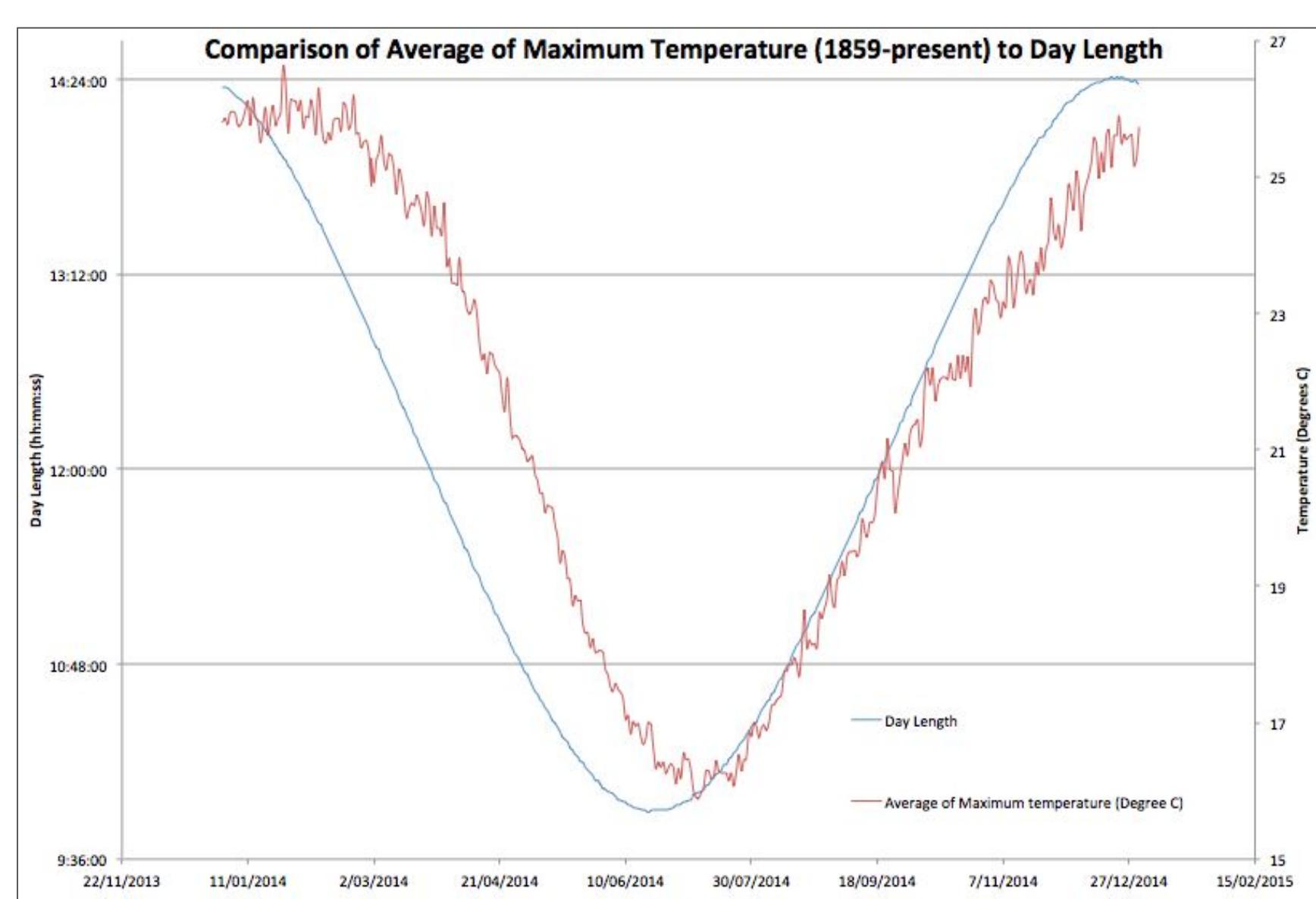
Excel For Researchers

⌚ 1 day

🔧 No Prerequisites

📍 At your campus

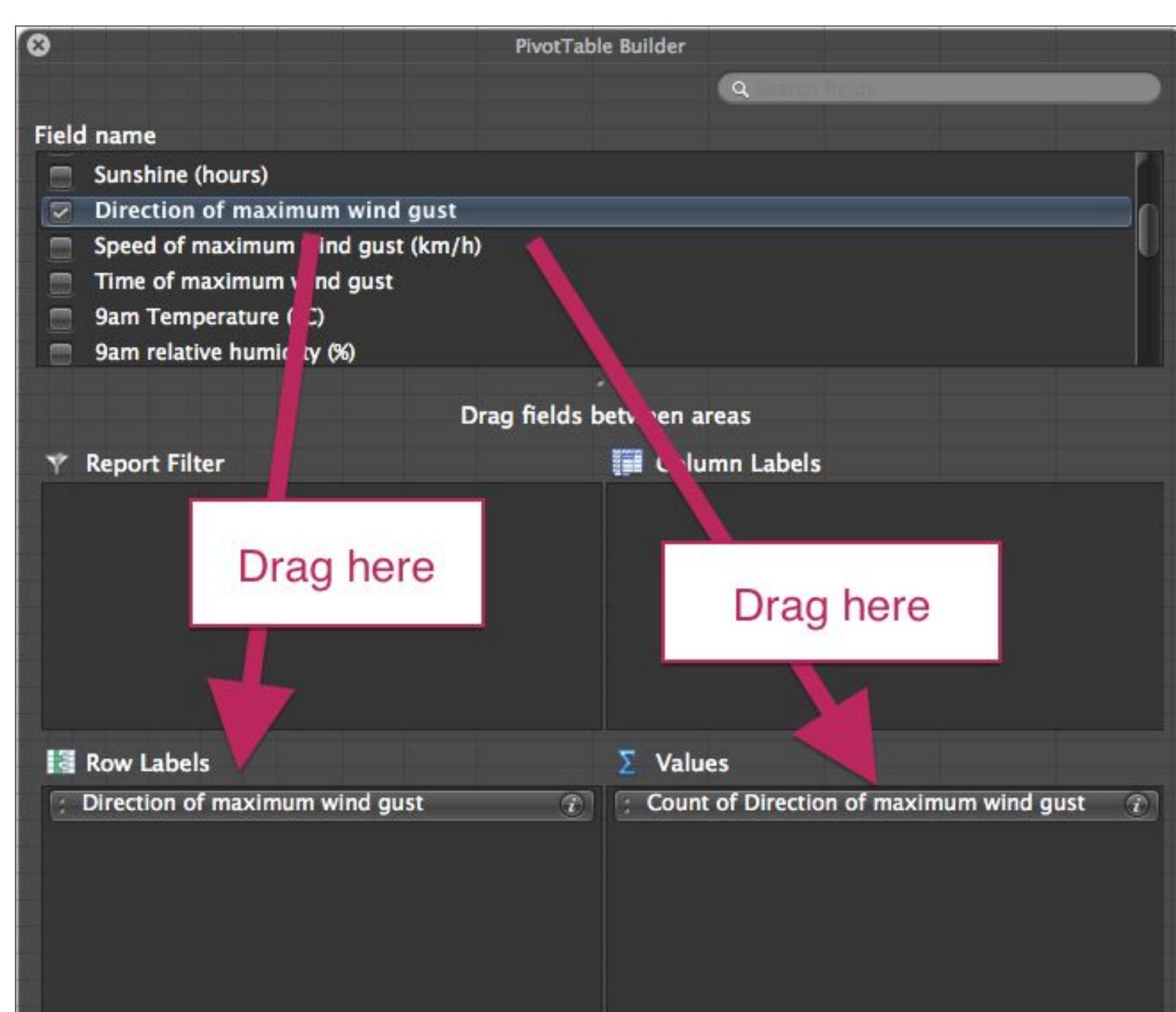
\$ No cost to members



Why do this course?

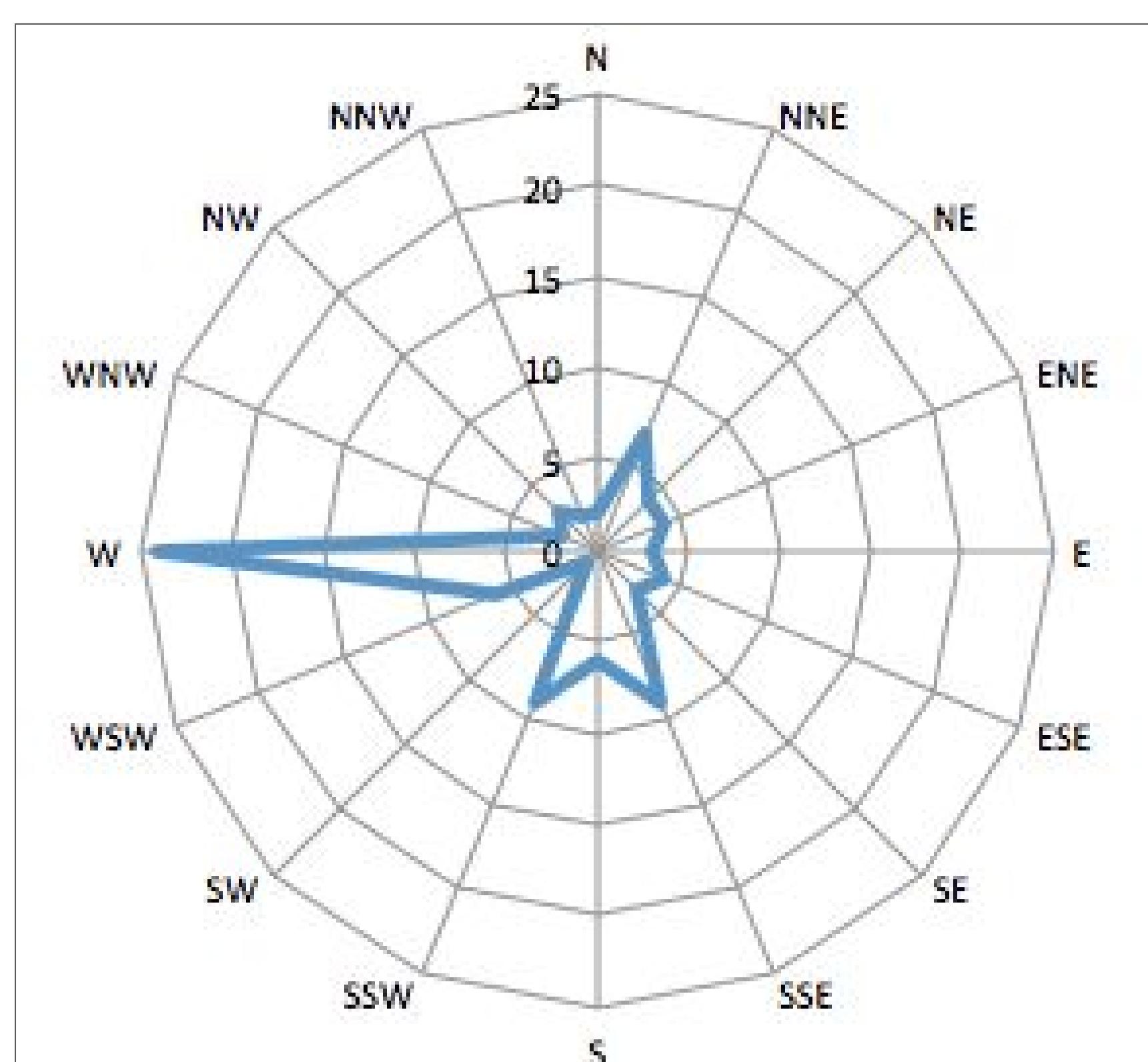
Data rarely comes in the form you require. Often it is messy. Sometimes it is incomplete. And sometimes there's too much of it. Frequently, it has errors. We'll use one of the most widespread data wrangling tools, Microsoft Excel, to import, sort, filter, copy, protect, transform, summarise, merge, and visualise research data.

While aimed at novice Excel users, most attendees will walk away with new tricks to work more efficiently with their research data.



You'll learn how to:

- Find, import and ‘clean up’ messy research data
- Organise, format and name your data in Excel
- Analyse your data (answer research questions)
- Interpret your data (filtering, conditional formatting)
- Perform calculations on your data (max, min, average)
- Extract significant findings from your data (pivot tables)
- Manipulate your data (data conversion, dates and times)
- Create graphs and charts to visualise your data



The Intersect approach to training

At Intersect, we work closely with our member universities to develop and deliver training that targets the day-to-day software and technology problems that researchers face. We deliver hands-on courses in a relaxed setting with knowledgeable, helpful trainers who are themselves researchers and who know how researchers work.

Questions always welcome.

For more information visit
Learn.intersect.org.au



Learn
.intersect.org.au



Data Visualisation with Google Fusion Tables

2.5 hours

No Prerequisites

At your campus

No cost to members

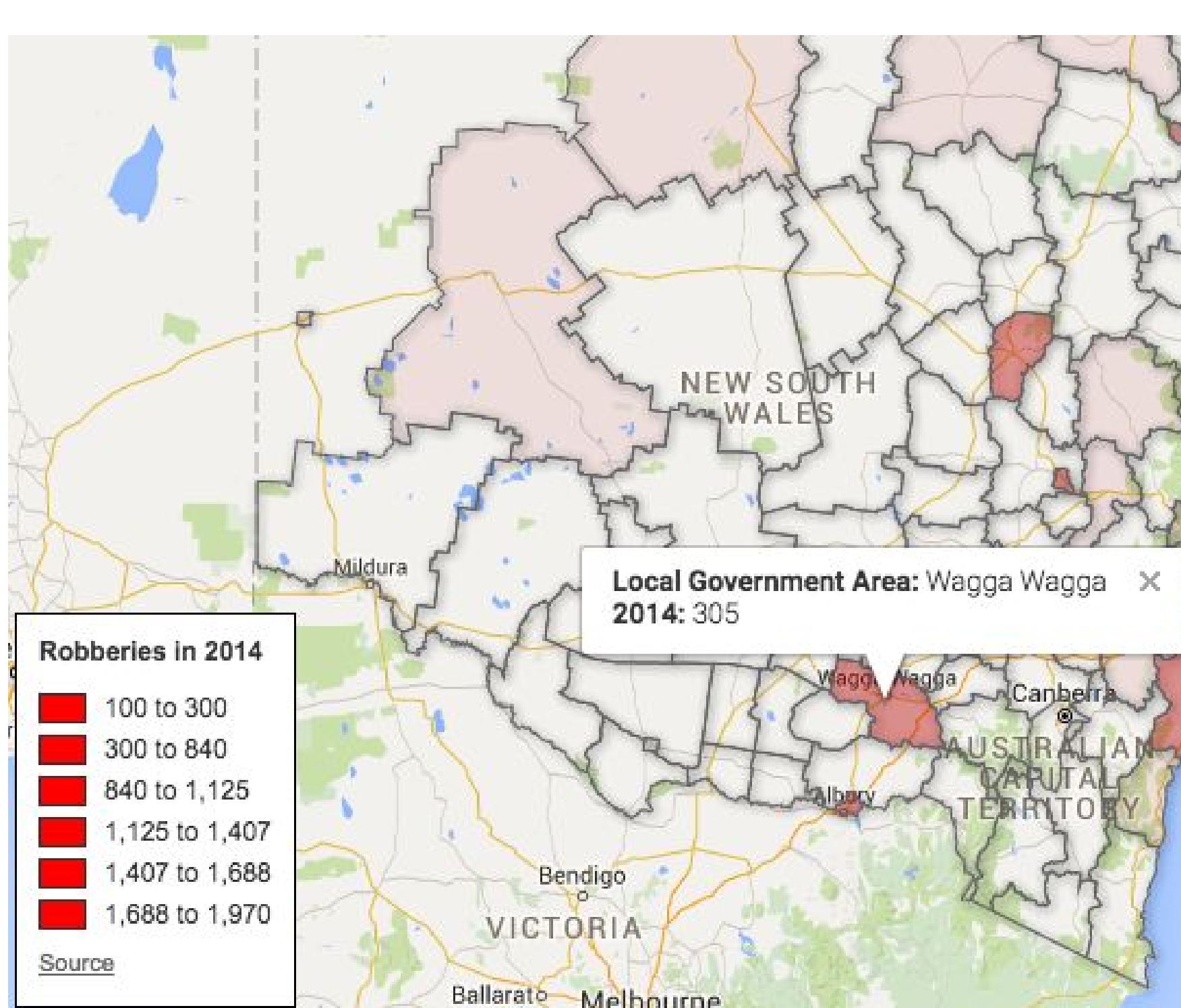
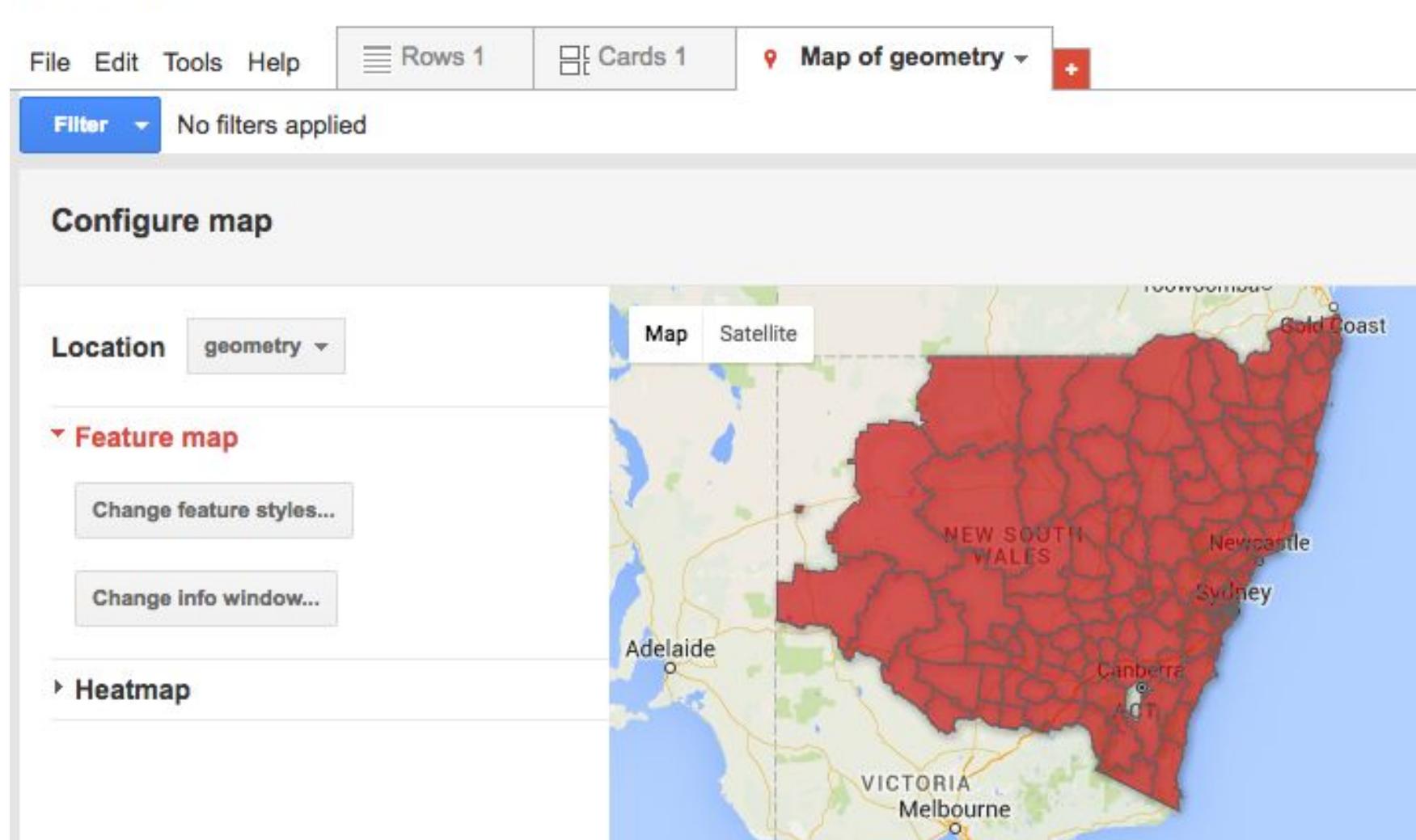
SpecificCrimeResults-2016-557658-3

Imported at Tue Jun 07 21:03:16 PDT 2016 from SpecificCrimeResults-2016-557658-3.c
Edited at 14:03

Local Government Area	1999	2000	2001	2002	2003	2004
Albury	349	571	637	568	466	496
Armidale Dumaresq	412	323	432	381	471	299
Ashfield	645	610	647	584	528	466
Auburn	983	674	743	620	503	574
Ballina	346	225	232	286	305	392
Balranald	21	25	19	17	18	24

AUS-NSW-LGA-2012

Imported at Wed Jun 10 20:49:18 PDT 2015 from AUS-NSW-LGA-2012.kml.
Edited on 2015 June 11



Why do this course?

Have you got data in different forms that you want to compare directly? Do you want to show your data on a map to improve your research message?

This workshop is an introduction to data visualisation techniques using hands-on activities in Google Fusion Tables. The course is ideal for participants who work with large data sets and want to convey their research outcomes clearly and persuasively in a visual manner. By merging geospatial data and crime statistics to create a heatmap, participants will gain skills in visualisation that they can apply to their research.

You'll learn how to:

- Import data into Google Fusion Tables
- Merge multiple data sets in Fusion Tables
- Create interactive heatmaps on a Google Map
- Vary the way the visualisation is presented in Fusion Tables

Preparation

Google Fusion Tables is an experimental Google Drive app. A personal Gmail account is required to participate in this course. Institutional accounts, such as those at universities, will not work.

The Intersect approach to training

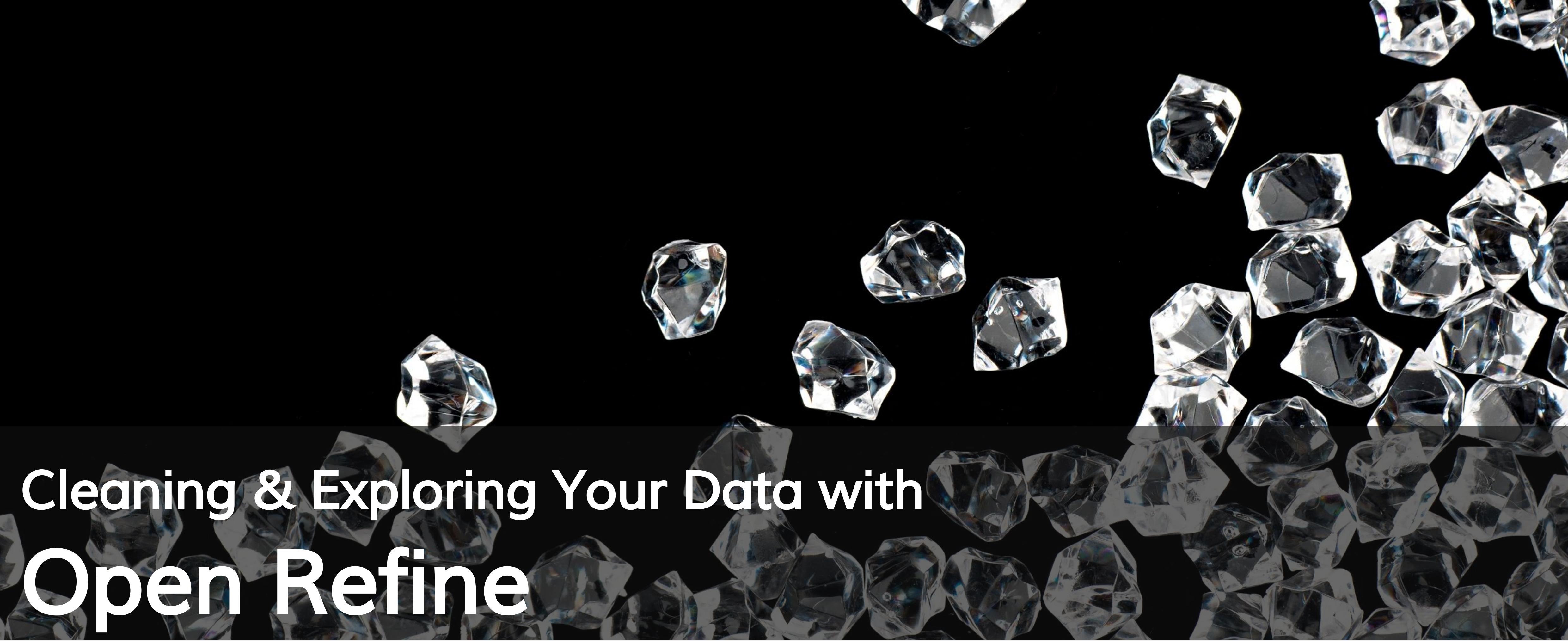
At Intersect, we work closely with our member universities to develop and deliver training that targets the day-to-day software and technology problems that researchers face. We deliver hands-on courses in a relaxed setting with knowledgeable, helpful trainers who are themselves researchers and who know how researchers work.

Questions always welcome.

For more information visit
Learn.intersect.org.au



Learn
.intersect.org.au



Cleaning & Exploring Your Data with Open Refine

⌚ 3 hours

🔧 No Prerequisites

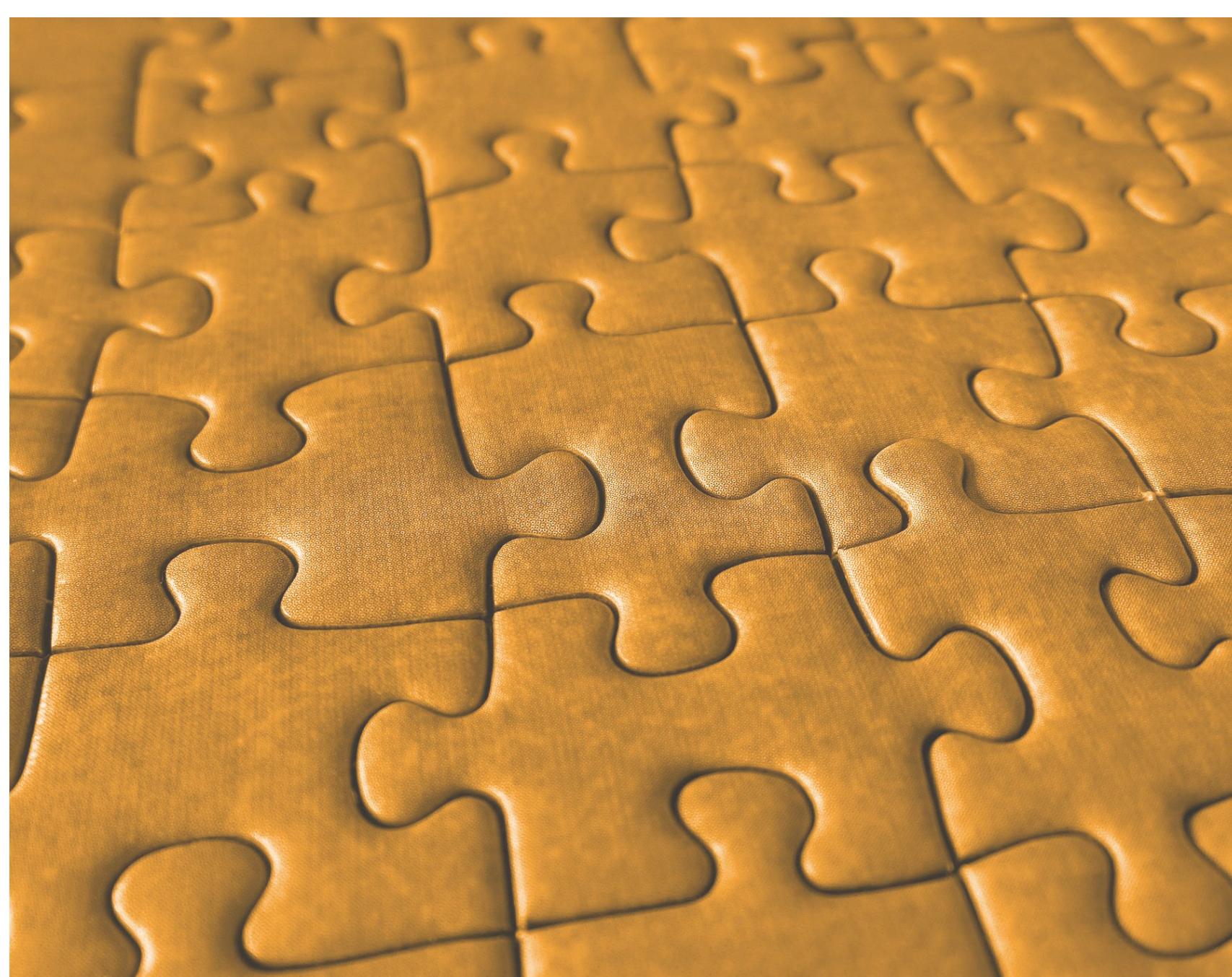
📍 At your campus

\$ No cost to members



Records Show: 5 10 25 50 rows

	norm	ty
et	4A YARN DYEING	account
t filter	4S PARK STYLE	account
:cells	Split into several columns...	
:column	Add column based on this column	
:response	Add column by fetching URLs...	
:column statistics	Add columns from Freebase ...	
...	Add columns from DBpedia ...	
N	Extract entities from text (Zemant...	



Why do this course?

Do you have messy data from multiple inconsistent sources, or open-responses to questionnaires? Do you want to improve the quality of your data by refining it and using the power of the internet?

Open Refine is the perfect partner to Excel. It is a powerful, free tool for exploring, normalising and cleaning datasets, and extending data by accessing the internet through APIs. In this course we'll work through the various features of Refine, including importing data, facetting, clustering, and calling remote APIs, by working on a fictional but plausible humanities research project.

You'll learn how to:

- Download, install and run Open Refine
- Import data from csv, text or online sources and create projects
- Navigate data using the Open Refine interface
- Explore data by using facets
- Clean data using clustering
- Parse data using GREL syntax
- Extend data using Application Programming Interfaces (APIs)
- Export project for use in other applications

The Intersect approach to training

At Intersect, we work closely with our member universities to develop and deliver training that targets the day-to-day software and technology problems that researchers face. We deliver hands-on courses in a relaxed setting with knowledgeable, helpful trainers who are themselves researchers and who know how researchers work.

Questions always welcome.

For more information visit
Learn.intersect.org.au



Learn
.intersect.org.au



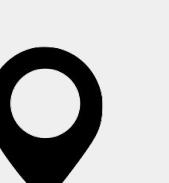
Powerful Text Searching & Matching with Regular Expressions



2.5 hours



No Prerequisites



At your campus



No cost to members

Tools Replace List Details

Date: \$4-\$3-\$2

Date: 1513-27-3
Date: 1524-17-4
Date: 1590-15-8
Date: 1607-14-5

(?:\|\\.\|\-|,\s)[0-9]{4}/g

for the first time, on 3.27,
he Atlantic coast of North Ameri-
cated, on 8/15/1590
settlement, on 5/14, 1607
tories of New Netherland, on 1.
ded on 3-1-1620

Character classes

- . any character except newline
- \w \d \s word, digit, whitespace
- \W \D \S not word, digit, whitespace
- [abc] any of a, b, or c
- [^abc] not a, b, or c
- [a-g] character between a & g

Anchors

- ^abc\$ start / end of the string
- \b word boundary

Why do this course?

Have you ever wanted to extract phone numbers out of a block of unstructured text? Or email addresses. Or find all the words that start with “e” and end with “ed”, no matter their length? Or search through DNA sequences for a pattern? Or extract coordinates from GPS data?

Regular Expressions (regexes) are a powerful way to handle a multitude of different types of data. They can be used to find patterns in text and make sophisticated replacements. Think of them as **find and replace on steroids**. Come along to this workshop to learn what they can do and how to apply them to your research.

You'll learn how to:

- Comprehend and apply the syntax of regular expressions
- Use the <http://regextester.com> tool to test a regular expression against some text
- Construct simple regular expressions to find capitalised words; all numbers; all words that start with a specific set of letters, etc. in a block of text
- Craft and test a progressively more complex regular expression
- Find helpful resources covering regular expressions on the web

The Intersect approach to training

At Intersect, we work closely with our member universities to develop and deliver training that targets the day-to-day software and technology problems that researchers face. We deliver hands-on courses in a relaxed setting with knowledgeable, helpful trainers who are themselves researchers and who know how researchers work.

Questions always welcome.

For more information visit
Learn.intersect.org.au



Learn
.intersect.org.au



Regular Expressions on Command

⌚ 3 hours

✖ Prerequisites (see below)

📍 At your campus

\$ No cost to members

```
$GPGGA,000157.000,3508.2382,S,14841.19
$GPRMC,000157.000,A,3508.2382,S,14841.
$GPGGA,000340.000,3508.3105,S,14841.10
$GPRMC,000340.000,A,3508.3105,S,14841.
$GPGGA,000519.000,3508.3764,S,14840.77
$GPRMC,000519.000,A,3508.3764,S,14840.
$GPGGA,000700.000,3508.3982,S,14840.75
$GPRMC,000700.000,A,3508.3982,S,14840.
$GPGGA,000839.000,3508.6913,S,14840.65
$GPRMC,000839.000,A,3508.6913,S,14840.
```

```
#!/usr/bin/awk -f
function to_deg(coord, dir)
{
    match(coord, /([0-9]+)([0-9]{2})/);
    minutes = arr[2] ". " arr[3];
    fractional_degrees = minutes / 60;
    degrees = arr[1] + fractional_degrees;

    if (dir == "S" || dir == "W")
        degrees = -degrees;

    return degrees
}
```

Why do this course?

Would you like to use regular expressions with the classic command line utilities `find`, `grep`, `sed` and `awk`? These venerable Unix utilities allow you to search, filter and transform large amounts of text (including many common data formats) efficiently and repeatably.

You'll learn how to use:

- `find` to locate files and directories matching regexes.
- `grep` to filter lines in files based on pattern matches.
- `sed` to find and replace using regular expressions and captures.
- `awk` to work with row- and column-oriented data.

Prerequisites

This course assumes prior knowledge of the basic syntax of regular expressions. If you're new to regular expressions or would like a refresher, take our **Powerful Text Searching & Matching with Regular Expressions** course first.

This course also assumes basic familiarity with the Bash command line environment found on GNU/Linux and other Unix-like environments. Take our **Introduction to Unix** course to get up to speed quickly.

The Intersect approach to training

At Intersect, we work closely with our member universities to develop and deliver training that targets the day-to-day software and technology problems that researchers face. We deliver hands-on courses in a relaxed setting with knowledgeable, helpful trainers who are themselves researchers and who know how researchers work.

Questions always welcome.

For more information visit
Learn.intersect.org.au



Learn
.intersect.org.au

PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
1	0.0	0.1	34892	4748	?	Ss	2015	4:40	/sbin/init
2	0.0	0.0	0	0	?	S	2015	0:02	[kthreadd]
3	0.0	0.0	0	0	?	S	2015	3:47	[ksoftirqd/0]
5	0.0	0.0	0	0	?	S<	2015	0:00	[kworker/0:0]
7	0.0	0.0	0	0	?	S	2015	167:16	[rcu_sched]
8	0.0	0.0	0	0	?	S	2015	0:00	[rcu_bh]
9	0.0	0.0	0	0	?	S	2015	227:34	[rcuos/0]
10	0.0	0.0	0	0	?	S	2015	0:00	[rcuob/0]
11	0.0	0.0	0	0	?	S	2015	0:00	[migration/0]
12	0.0	0.0	0	0	?	S	2015	2:43	[watchdog/0]
13	0.0	0.0	0	0	?	S<	2015	0:00	[khelper]
14	0.0	0.0	0	0	?	S	2015	0:00	[kdevtmpfs]
15	0.0	0.0	0	0	?	S	2015	0:00	[netns]
16	0.0	0.0	0	0	?	S	2015	0:00	[perf]
17	0.0	0.0	0	0	?	S<	2015	0:00	[khungtaskd]
18	0.0	0.0	0	0	?	S	2015	0:00	[writeback]
19	0.0	0.0	0	0	?	S	2015	0:00	[flusher]
20	0.0	0.0	0	0	?	S	2015	0:00	[lru_kicker]
21	0.0	0.0	0	0	?	S	2015	0:00	[kswapd0]
22	0.0	0.0	0	0	?	S	2015	0:00	[kblockd]
23	0.0	0.0	0	0	?	S	2015	0:00	[klogd]
24	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:1]
25	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:2]
26	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:3]
27	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:4]
28	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:5]
29	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:6]
30	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:7]
31	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:8]
32	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:9]
33	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:10]
34	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:11]
35	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:12]
36	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:13]
37	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:14]
38	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:15]
39	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:16]
40	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:17]
41	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:18]
42	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:19]
43	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:20]
44	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:21]
45	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:22]
46	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:23]
47	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:24]
48	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:25]
49	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:26]
50	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:27]
51	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:28]
52	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:29]
53	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:30]
54	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:31]
55	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:32]
56	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:33]
57	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:34]
58	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:35]
59	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:36]
60	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:37]
61	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:38]
62	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:39]
63	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:40]
64	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:41]
65	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:42]
66	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:43]
67	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:44]
68	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:45]
69	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:46]
70	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:47]
71	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:48]
72	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:49]
73	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:50]
74	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:51]
75	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:52]
76	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:53]
77	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:54]
78	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:55]
79	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:56]
80	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:57]
81	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:58]
82	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:59]
83	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:60]
84	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:61]
85	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:62]
86	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:63]
87	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:64]
88	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:65]
89	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:66]
90	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:67]
91	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:68]
92	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:69]
93	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:70]
94	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:71]
95	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:72]
96	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:73]
97	0.0	0.0	0	0	?	S	2015	0:00	[kworker/0:74]
9									

Introduction to Unix



3 hours



No Prerequisites



At your campus



No cost to members

```
nelle@training: ~$ cowsay "Learn Unix today!"  
  
< Learn Unix today! >  
-----  
 \ ^__^  
  (oo)\_____  
   (__)\       )\/\  
     ||----w ||  
 nelle@training: ~$ █
```

```
nelle@training: ~$ cat list.txt
Apple
Tomato
Pineapple
Capsicum
Onion
Custard apple
Banana
nelle@training: ~$ sort list.txt
Apple
Banana
Capsicum
Custard apple
Onion
Pineapple
Tomato
nelle@training: ~$ grep -i apple list.txt
Apple
Pineapple
Custard apple
nelle@training: ~$
```

Why do this course?

The Unix environment is incredibly powerful but quite daunting to the newcomer. Command line confidence unlocks powerful computing resources beyond the desktop, including virtual machines and High Performance Computing. It enables repetitive tasks to be automated. And it comes with a swag of handy tools that can be combined in powerful ways. Getting started is the hardest part, but our helpful instructors are there to demystify Unix as you get to work running programs and writing scripts on the command line.

Every attendee is given a dedicated training environment for the duration of the workshop, with all software and data fully loaded and ready to run.

We teach this course within a **GNU/Linux** environment. This is best characterised as a **Unix-like** environment. We teach how to run commands within the **Bash Shell**. The skills you'll learn at this course are generally transferable to other Unix environments.

You'll learn how to:

- navigate and work with files and directories (folders).
 - use a selection of essential tools.
 - combine data and tools to build a processing workflow.
 - automate repetitive analysis using the command line.

The Intersect approach to training

At Intersect, we work closely with our member universities to develop and deliver training that targets the day-to-day software and technology problems that researchers face. We deliver hands-on courses in a relaxed setting with knowledgeable, helpful trainers who are themselves researchers and who know how researchers work.

Questions always welcome

For more information visit
Learn.intersect.org.au



Learn



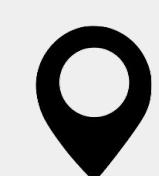
Introduction to Programming in Python



1 day



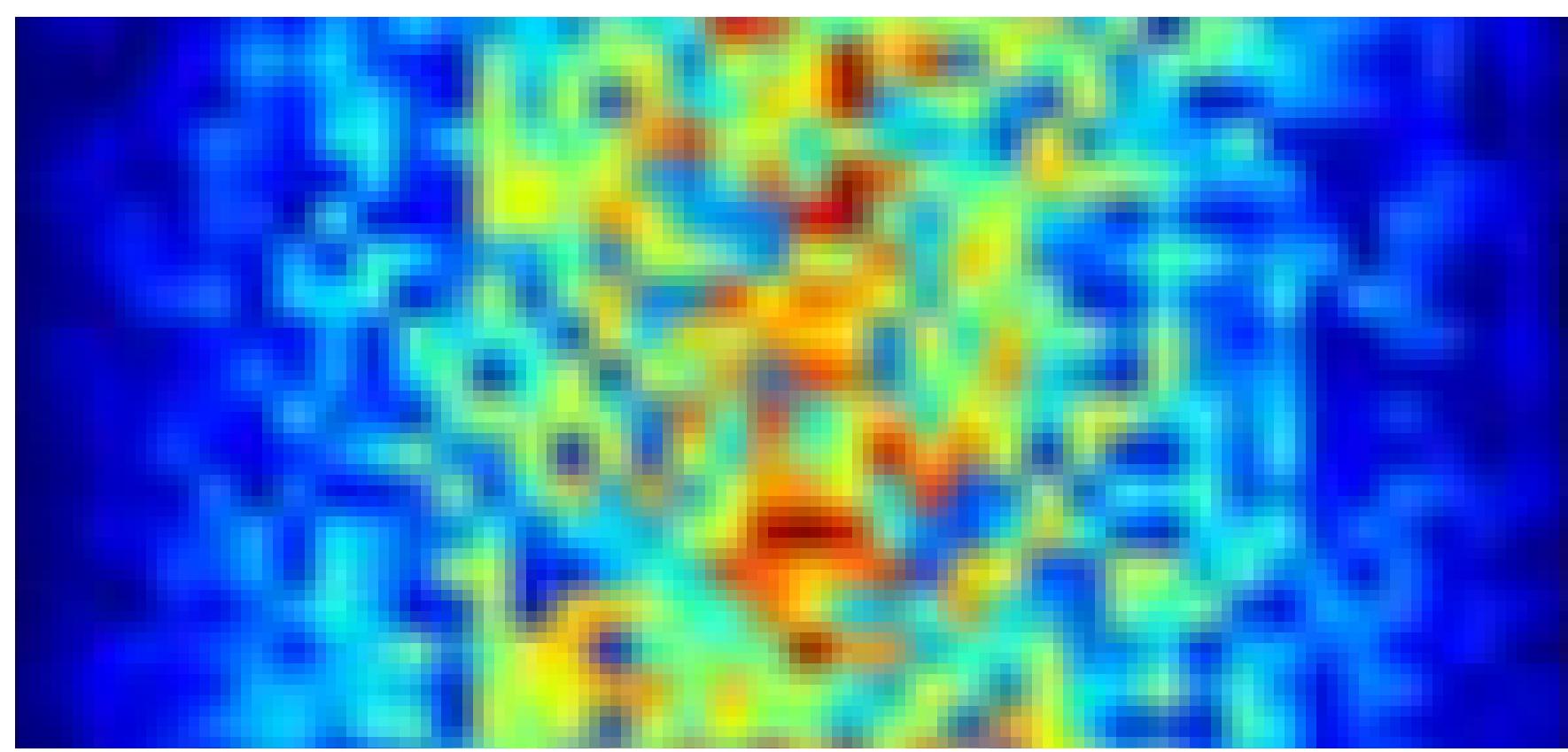
No Prerequisites



At your campus



No cost to members



Intersect Software Courses

Introduction to Unix

Introduction to Programming in Python

Introduction to Programming in R

Introduction to Programming in MATLAB

Introduction to Version Control with Git

Software Carpentry Workshop

This programming course can be taken on its own or as part of an Intersect organised **Software Carpentry Workshop**

Why do this course?

Python has deservedly become a popular language for scientific computing. It has all the friendly features and conveniences you'd expect of a modern programming language. And it boasts a rich set of libraries for working with data.

We teach using Jupyter notebooks, which allow program code, results, visualisations and documentation to be blended seamlessly. Perfect for sharing insights with others while producing reproducible research.

Join us for this live coding workshop where we write programs that produce results, using the researcher-focused training modules from the highly regarded Software Carpentry Foundation (software-carpentry.org).

You'll learn:

- Programming concepts and techniques.
- Basic syntax, control structures and data types in Python.
- How to import powerful libraries that support numerical analysis (NumPy) and visualisation (Matplotlib)
- Approaches to debugging, testing and defensive programming.
- How to blend code, output and documentation with Jupyter notebooks

The Intersect approach to training

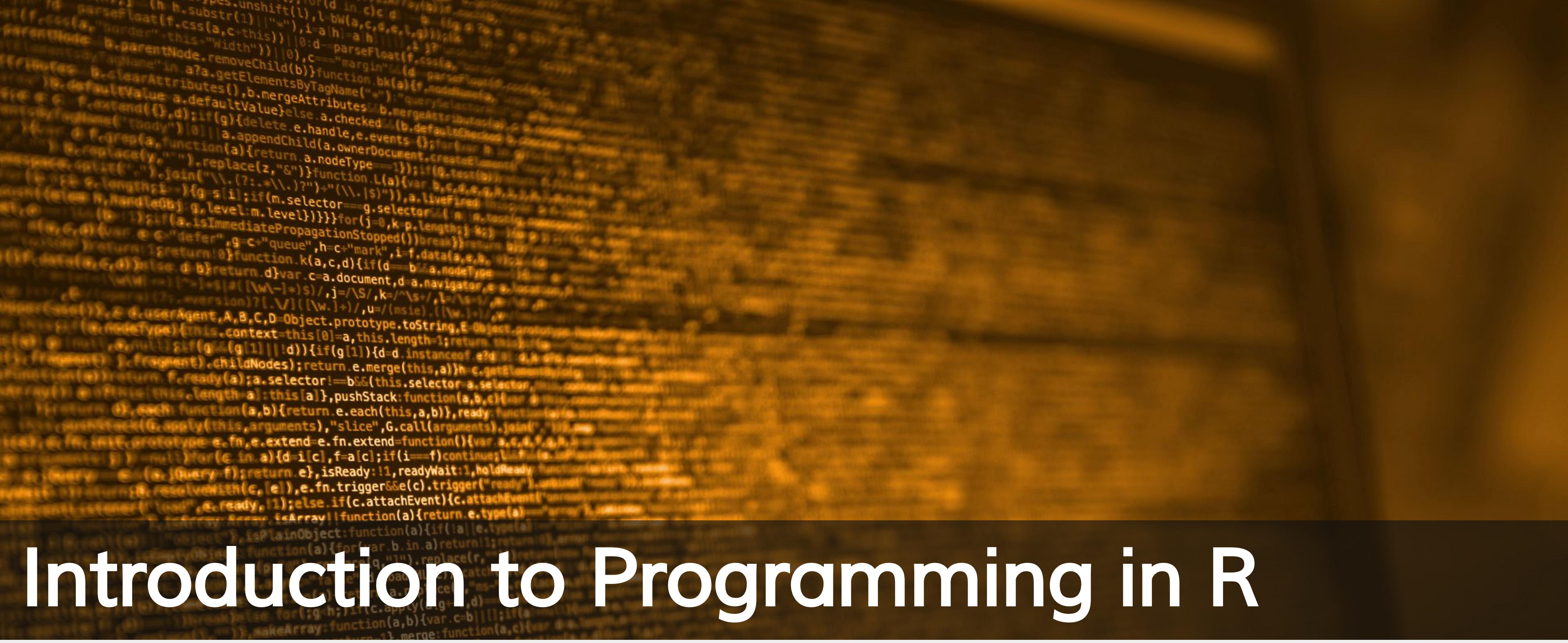
At Intersect, we work closely with our member universities to develop and deliver training that targets the day-to-day software and technology problems that researchers face. We deliver hands-on courses in a relaxed setting with knowledgeable, helpful trainers who are themselves researchers and who know how researchers work.

Questions always welcome.

For more information visit
Learn.intersect.org.au



Learn
intersect.org.au



Introduction to Programming in R



1 day



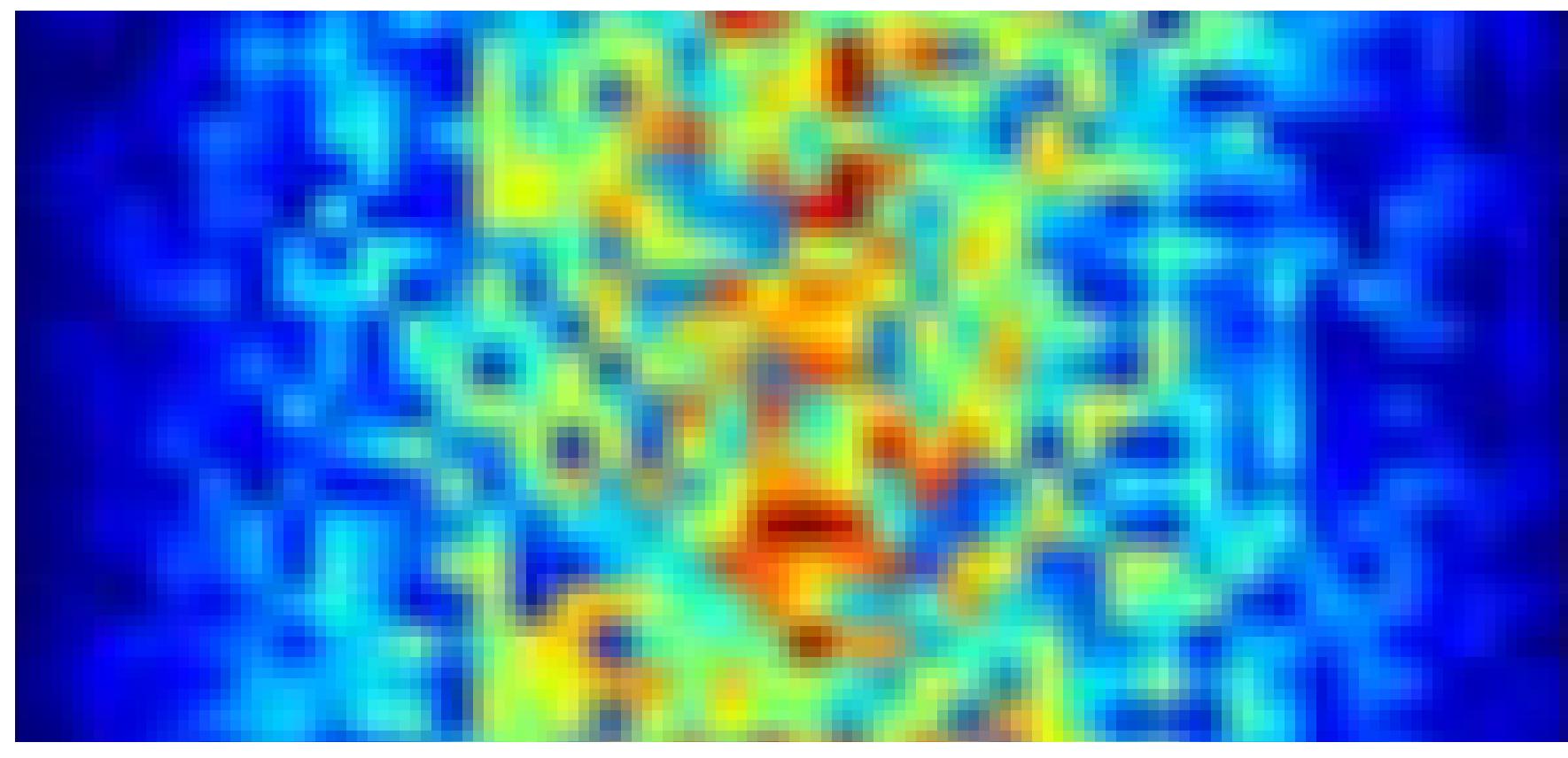
No Prerequisites



At your campus



No cost to members



Intersect Software Courses

Introduction to Unix

Introduction to Programming in Python

Introduction to Programming in R

Introduction to Programming in MATLAB

Introduction to Version Control with Git

Software Carpentry Workshop

This programming course can be taken on its own or as part of an Intersect organised **Software Carpentry Workshop**

Why do this course?

R is quickly gaining popularity as a programming language of choice for statisticians, data scientists and researchers. It has an excellent ecosystem including the powerful RStudio development environment and the Shiny web application framework.

But getting started with R can be challenging, particularly if you've never programmed before. That's where this introductory course comes in.

Join us for a live coding workshop where we write programs that produce results, using the researcher-focused training modules from the highly regarded Software Carpentry Foundation (software-carpentry.org).

You'll learn:

- Programming concepts and techniques.
- Basic syntax, control structures and data types in R.
- How to load external data into R.
- Ways to visualise data.
- How to use factors for category data.
- Best practices for writing code in R.

The Intersect approach to training

At Intersect, we work closely with our member universities to develop and deliver training that targets the day-to-day software and technology problems that researchers face. We deliver hands-on courses in a relaxed setting with knowledgeable, helpful trainers who are themselves researchers and who know how researchers work.

Questions always welcome.

For more information visit
Learn.intersect.org.au



Learn
.intersect.org.au

Introduction to Programming in MATLAB



1 day



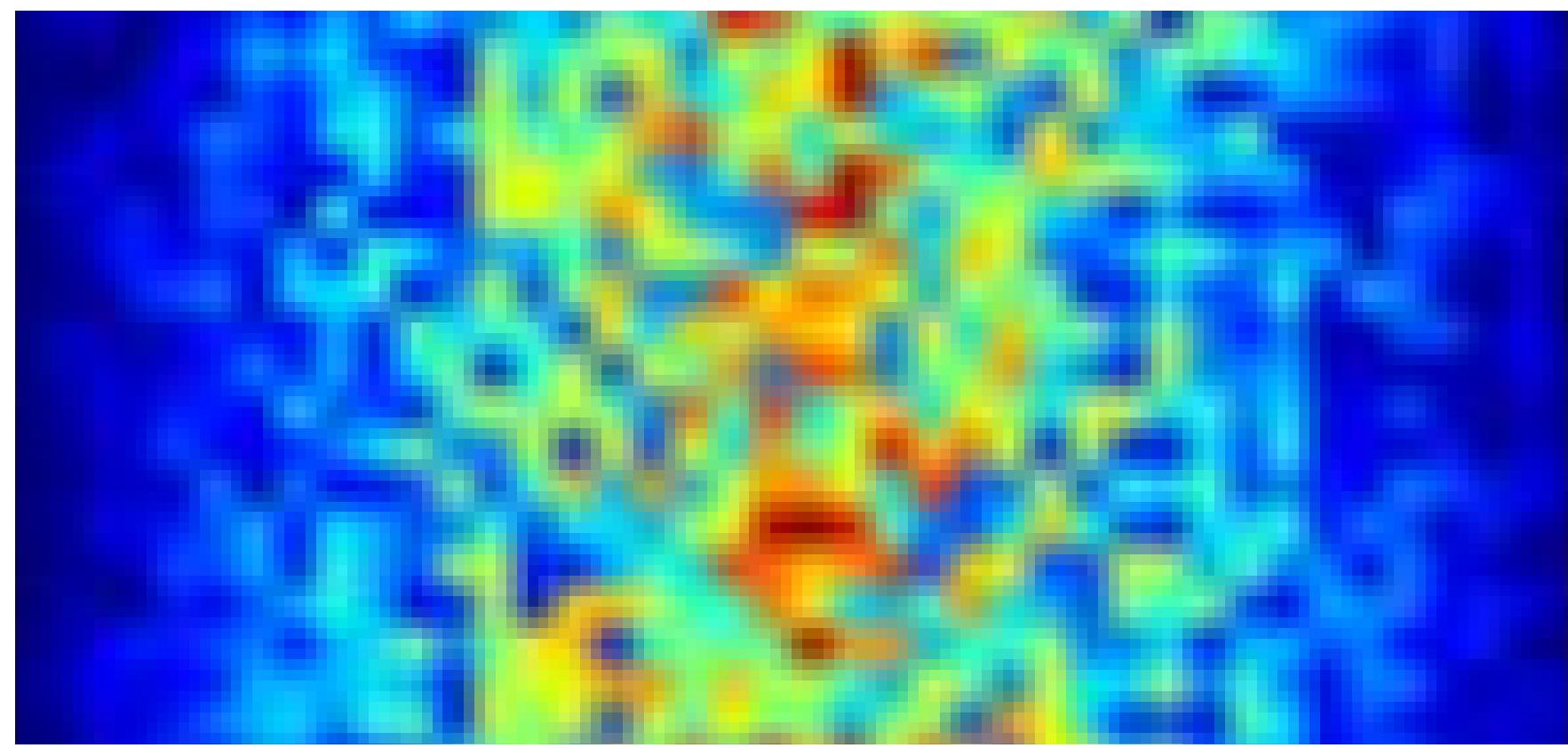
No Prerequisites



At your campus



No cost to members



Intersect Software Courses

Software Carpentry Workshop

Introduction to Unix

Introduction to Programming in Python

Introduction to Programming in R

Introduction to Programming in MATLAB

Introduction to Version Control with Git

This programming course can be taken on its own or as part of an Intersect organised **Software Carpentry Workshop**

Why do this course?

MATLAB is an incredibly powerful programming environment with a rich set of analysis toolkits. But what if you're just getting started – with MATLAB and, more generally, with programming?

Nothing beats a hands-on, face-to-face training session to get you past the inevitable syntax errors!

So join us for this live coding workshop where we write programs that produce results, using the researcher-focused training modules from the highly regarded Software Carpentry Foundation (software-carpentry.org).

You'll learn:

- Programming concepts and techniques.
- Basic syntax, control structures and data types in MATLAB.
- How to load external data into MATLAB.
- Ways to visualise data.
- Defensive programming techniques for avoiding errors.

The Intersect approach to training

At Intersect, we work closely with our member universities to develop and deliver training that targets the day-to-day software and technology problems that researchers face. We deliver hands-on courses in a relaxed setting with knowledgeable, helpful trainers who are themselves researchers and who know how researchers work.

Questions always welcome.

For more information visit
Learn.intersect.org.au



Learn
.intersect.org.au



Introduction to Version Control with Git



3 hours



Prerequisites (see below)



At your campus



No cost to members

```

nelle@training:~/repository$ git init
Initialized empty Git repository in /home/nelle/repo/
nelle@training:~/repository$ touch hello.txt
nelle@training:~/repository$ git add hello.txt
nelle@training:~/repository$ git commit -m "first commit"
[master (root-commit) 705e172] first commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 hello.txt
nelle@training:~/repository$ echo "a change" >> hello.txt
nelle@training:~/repository$ git add hello.txt
nelle@training:~/repository$ git commit -m "a small change"
[master ea15efc] a small change
 1 file changed, 1 insertion(+)
nelle@training:~/repository$ git log
commit ea15efcddd2e87accd71418983f2eee51644a996
Author: Nelle <nelle@intersect.org.au>
Date:   Thu Aug 4 04:37:10 2016 +0000

    a small change

commit 705e172aa79b5ffd94598f91221008d07f794ee0
Author: Nelle <nelle@intersect.org.au>
Date:   Thu Aug 4 04:36:33 2016 +0000

  first commit
nelle@training:~/repository$ 

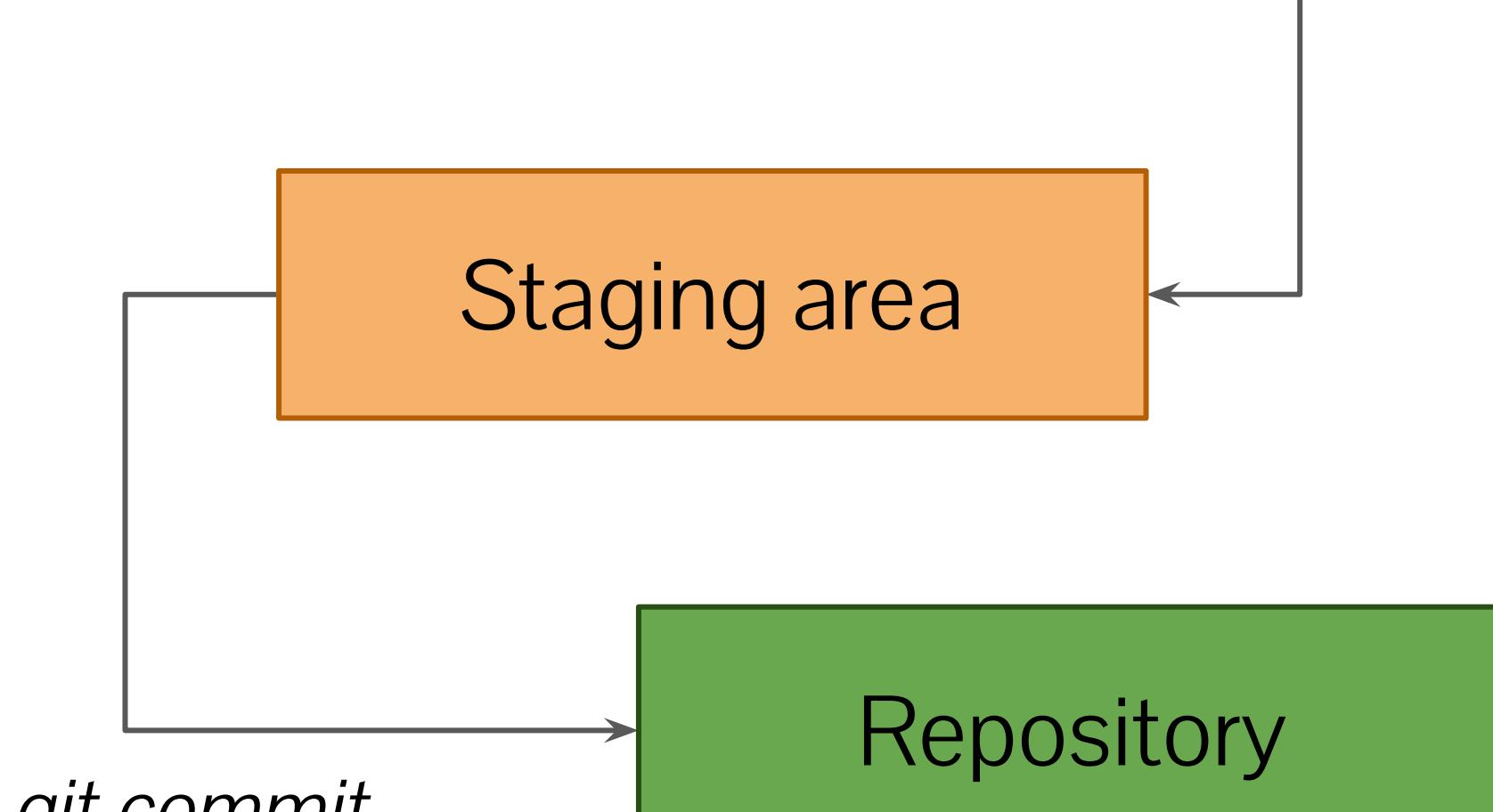
```

Working directory

git add

Staging area

- keep versions of data, scripts, and other files
- examine commit logs to find which files were changed when
- restore earlier versions of files
- compare changes between versions of a file
- push your versioned files to a remote location, for backup and to facilitate collaboration



Prerequisites

This course assumes basic familiarity with the Bash command line environment found on GNU/Linux and other Unix-like environments. Take our **Introduction to Unix** course to get up to speed quickly.

The Intersect approach to training

At Intersect, we work closely with our member universities to develop and deliver training that targets the day-to-day software and technology problems that researchers face. We deliver hands-on courses in a relaxed setting with knowledgeable, helpful trainers who are themselves researchers and who know how researchers work.

Questions always welcome.

For more information visit
Learn.intersect.org.au



Learn
[.intersect.org.au](http://intersect.org.au)



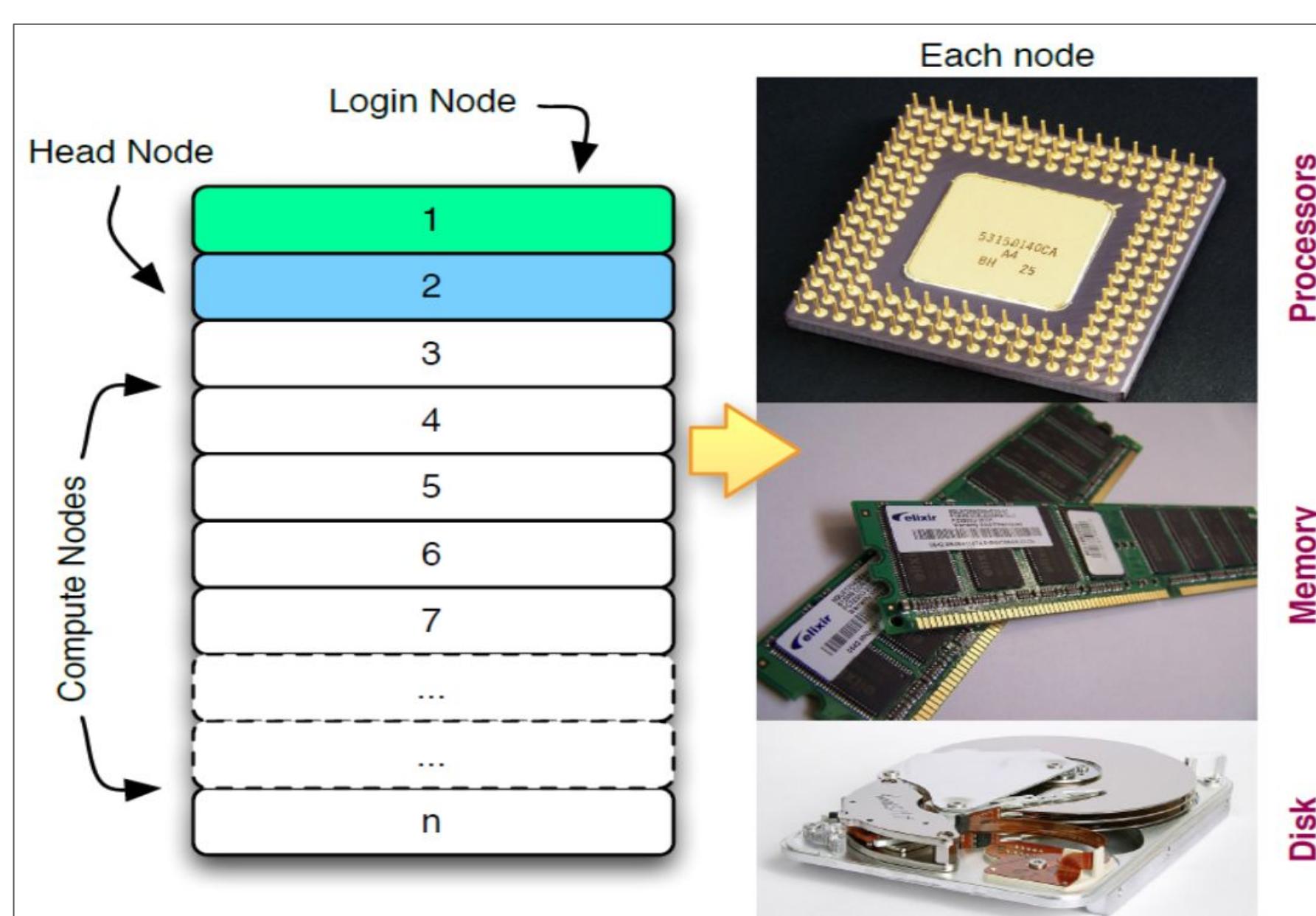
Unix for HPC

⌚ 2 days

🔧 No Prerequisites

📍 At your campus

\$ No cost to members



```
1 #!/bin/bash
2 # Request resources
3 # * 10 minutes wall time to run
4 #PBS -l walltime=00:10:00
5 # * 1 node, 1 processor
6 # * 100 megabytes physical memory allocated to each node
7 #PBS -l select=1:ncpus=1:mem=100mb
8 # Specify a project code (for accounting)
9 #PBS -P a40
10 # Set email address
11 #PBS -M nobody@intersect.org.au
12 # Send an email when jobs
13 # begins (b), gets aborted (a)
14 # and ends (e)
15 #PBS -m abe
16 # Move to directory job was submitted in
17 cd $PBS_O_WORKDIR
18 # Specify the job to be done
19 date
20 sleep 60
21 date
```

Why do this course?

Is your computer's limited power throttling your research ambitions? Are your analysis scripts pushing your laptop's processor to its limits? Is your software crashing because you've run out of memory? Would you like to unleash the power of the Unix command line to automate your analysis and then apply those skills to run your analysis on supercomputers that you can access for free?

High-Performance Computing (HPC) allows you to accomplish your analysis faster by using many parallel CPUs and huge amounts of memory simultaneously. This 2-day course will introduce you to the Unix environment and show you how to transfer your data onto, and run software on HPC infrastructure.

You'll learn how to:

- Use the Unix command line to work with HPC
- Get your data onto a supercomputer
- Run analysis on a supercomputer using batch jobs
- Access the facilities available to you as a researcher

Even if you're not ready to use HPC yet, the course makes a great primer for anyone wanting to start using GNU/Linux and other Unix-like systems

📅 Day 1

Get a solid grounding in the Unix command line

📅 Day 2

Transfer data to a supercomputer
Write and run scripts to analyse data
Learn how to apply for ongoing access

The Intersect approach to training

At Intersect, we work closely with our member universities to develop and deliver training that targets the day-to-day software and technology problems that researchers face. We deliver hands-on courses in a relaxed setting with knowledgeable, helpful trainers who are themselves researchers and who know how researchers work.

For more information visit
Learn.intersect.org.au



Learn
.intersect.org.au

Intermediate HPC

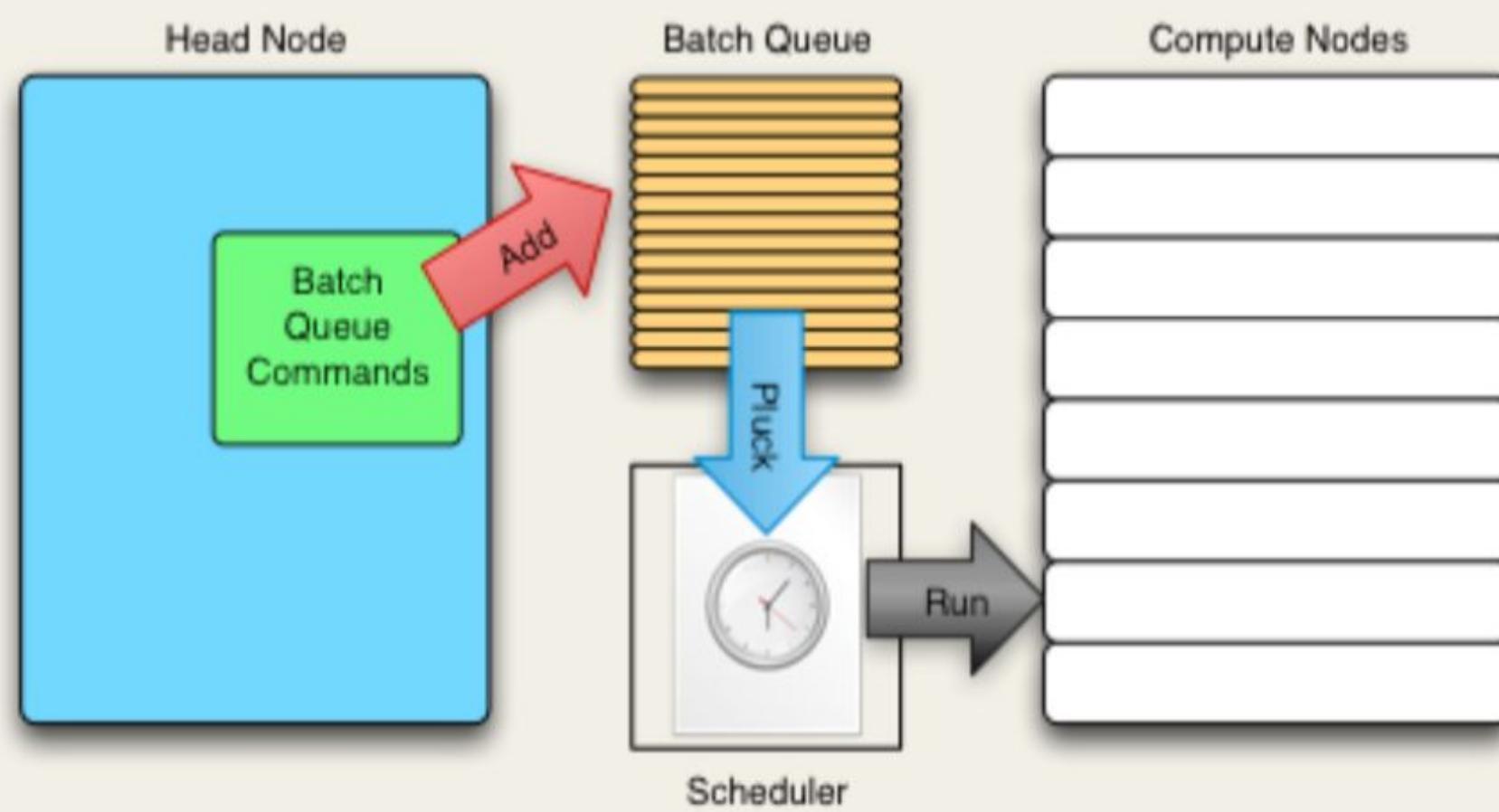
⌚ 3 hours

🔧 Prerequisites (see below)

📍 At your campus

\$ No cost to members

The Batch Queuing System



Intersect HPC courses

Introduction to Unix (Intersect)

Unix for HPC

Data Transfer from PC to HPC

Intermediate HPC

Parallel Programming for HPC

"Intermediate HPC" can be taken as its own course or as part of the comprehensive "Unix for HPC" course. The choice is yours!

Why do this course?

High-Performance Computing (HPC) allows you to accomplish your analysis faster by using many parallel CPUs and huge amounts of memory simultaneously. This short course provides a hands on introduction to running software on HPC infrastructure.

You'll learn how:

- HPC systems differ from traditional Unix environments
- to apply for ongoing access of an HPC machine
- the scheduler and batch systems work in HPC
- to submit a job and check its progress to completion
- to retrieve the results of your program

Prerequisites

This course assumes basic familiarity with the Bash command line environment found on GNU/Linux and other Unix-like environments. It also assumes knowledge of how to transfer files between computers using ssh and sftp.

If you lack these skills, consider taking "Unix for HPC" which blends "Intermediate HPC" with a primer on Unix and moving data around.

The Intersect approach to training

At Intersect, we work closely with our member universities to develop and deliver training that targets the day-to-day software and technology problems that researchers face. We deliver hands-on courses in a relaxed setting with knowledgeable, helpful trainers who are themselves researchers and who know how researchers work.

Questions always welcome.

For more information visit
Learn.intersect.org.au



Learn
.intersect.org.au



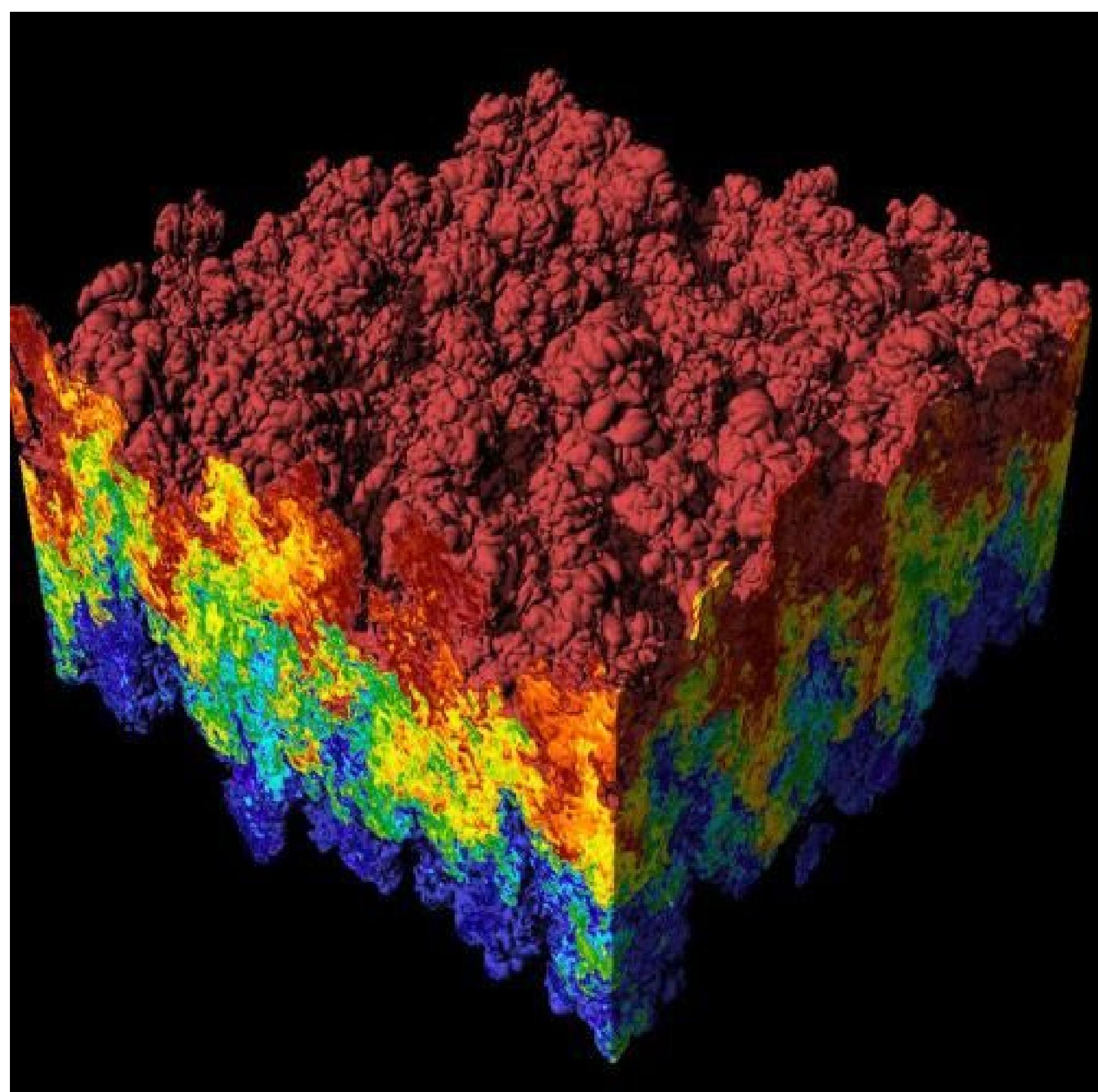
Parallel Programming for HPC

6 hours

Prerequisites (see below)

At your campus

No cost to members



Intersect HPC courses

Introduction to Unix (Intersect)

Unix for HPC

Data Transfer from PC to HPC

Intermediate HPC

Parallel Programming for HPC

Why do this course?

You have written, compiled and run functioning programs in C and/or Fortran. You know how HPC works and you've submitted batch jobs.

Now you want to move from writing single-threaded programs into the parallel programming paradigm, so you can truly harness the full power of High Performance Computing.

You'll learn how to program with:

- **OpenMP** (Open Multi-Processing): a widespread method for shared memory programming
- **MPI** (Message Passing Interface): a leading distributed memory programming model

Prerequisites

To do this course you need to have:

- a good working knowledge of HPC. Consider taking our **Unix for HPC** course to come up to speed beforehand.
- prior experience of writing programs in either C or Fortran.

The Intersect approach to training

At Intersect, we work closely with our member universities to develop and deliver training that targets the day-to-day software and technology problems that researchers face. We deliver hands-on courses in a relaxed setting with knowledgeable, helpful trainers who are themselves researchers and who know how researchers work.

Questions always welcome.

For more information visit
Learn.intersect.org.au



Learn
.intersect.org.au