

Modelling and data analysis 'Winter School'

Nick Golledge¹,
Liz Keller^{1,2},
Alex Gossart¹,
Alena Malyarenko³,
Angela Bahamondes-Dominguez³,
Mario Krapp²,
Dan Lowry²,
Stefan Jendersie¹

¹ Antarctic Research Centre, Victoria University of Wellington, New Zealand

² GNS Science, Lower Hutt, New Zealand

³ NIWA, Wellington, New Zealand



**Antarctic
Science Platform**
National Modelling Hub



**MINISTRY OF BUSINESS,
INNOVATION & EMPLOYMENT**
HĪKINA WHAKATUTUKI



NIWA
Te Hōu Nukurangi

1 | Welcome & Introduction

Day 1

10:00	Arrival & welcome	Nick
10:15	Introduction to programming	Nick
	Navigating the command line environment, scripting vs programming, pros & cons of various languages	
11:30	Introduction to models	Liz & Dan
	Climate model basics: components, types of models, internal variability. CMIP overview, climate sensitivity	
13:00	Lunch	
14:00	Time-series data – lecture	Mario
	Principal component / empirical orthogonal function analysis, calculation of correlations, anomalies, de-trending	
15:30	Afternoon tea	
15:45	Time-series data – tutorial	Mario
17:00	Wrap-up	

Day 2

09:00	Spatial data – lecture	Alex & Alena
	Understanding gridded data, map projections, data analysis and manipulations, masking, extracting vertical / horizontal sections	
10:30	Coffee	
10:45	Spatial data – tutorial	Alex & Alena
12:15	Lunch	
13:15	Document preparation in L ^A T _E X	Angela
	Learn the basics, write equations, insert figures, create your own tables, insert references	
14:45	Afternoon tea	
15:00	Work Structure & Version control	Stefan
	Defining a workflow, handling 'big data', version control for scripts/documents, best practice guidelines	

1 | Aims, Methods, & Scope

- ▶ The **aim** of the Winter School is that, by the end of the two days, participants will be able to find and download (climate model) data of interest, use simple scripts to process, analyse, and plot those data, integrate these outputs into a typeset document, and use version control software to keep track of changes.
- ▶ We will use *Python* for the majority of the work but will incorporate examples from other languages if necessary. We'll introduce you to packages like \LaTeX and tools such as *github*.
- ▶ This workshop is only intended to provide an **introduction** to working in a command-line environment, and exposure to some of the functionality available in this realm. It is not intended to be a complete course on programming, modelling, or data analysis ;-)

2 | Command-line basics (*nix)

Basic commands

command	example	description
ls	ls -ltrh	list directory contents (in long format, newest last)
cd	cd ../mydir/mysubdir	change directory (up one level, down two)
rm	rm delete-this.txt and-all-these.*	remove file(s)
mv	mv rename-this.txt to-this.txt	move (rename) file(s)
mkdir	mkdir ./new-directory	make a new (empty) directory
cp	cp this.txt ./new-dir/to-this.txt	copy file (possibly to new location)

Linux c-line tools

tool	example	description
pwd	pwd	Find out what your current personal working directory is
sed	sed -e 's/a/b/g'	stream editor, swap 'a' for 'b'
awk	awk '{print \$2, \$3}'	print fields 2 & 3 from file/stream

Other packages & utilities

package	example	description
pdflatex	pdflatex myfile.tex	compile L ^A T _E X document
git	git clone golledni/WinterSchool	Make a local copy of a github repository

2 | Simple (bash) shell scripting

- We can combine many simple commands, tools, and utilities to achieve more complex tasks

```
pwd
```

```
/home/golledni/MEGA/Work/AntSciPlat/WinterSchool
```

```
pwd | sed -e 's/\\ /g' | awk '{ print "Today my",$1,"is the",$NF}'  
Today my home is the WinterSchool
```

2 | Simple (bash) shell scripting

- But to do anything more complex than simple pipes we probably want to write a script file to contain our sequence of commands:

```
#!/bin/sh
lastupdated='head -n 1 new_papers.txt'
echo "Last updated " $lastupdated
now=$(date +%F)

echo $now > new_papers.txt
# get list of directories to loop through
list='ls -l | grep ^d | awk '{print $9}''
# find NEW papers in each of those directories
echo "\nNEW PAPERS:\n" >> new_papers.txt
for dir in $list ; do
    echo "\n***** $dir *****\n" >> new_papers.txt
    find ./$dir -type f -newermt $lastupdated -print | awk -F"/" '{print $3}' | sed -e 's/.pdf/]/g' -e 's/_/ /g' -e 's/^/\\subsubsection{/ >> new_papers.txt
done
```

tool	example
pwd	pwd
sed	sed -e 's/a/b/g'
awk	awk '{print \$2, \$3}'

Other packages & utilities

package	example
pdflatex	pdflatex myfile.tex
git	git clone gollodni/WinterSchool

Modelling and data analysis 'Winter School'

2 | Simple (bash) shell scripting

- We can combine many simple commands, tools, a
- achieve more complex tasks

2 | Control structures

- ▶ Often we want to apply a test, or series of tests, to the data we're processing, and do different things with the data depending of the results of those tests
- ▶ Control structures are what help us achieve this, and are fundamental to all languages
- ▶ The two most common generic structures are
 - ▶ if statements, and
 - ▶ for or do loops

if statement:

```
i=0
if [ $i -ge 1 ]
then
    echo "i = $i"
else
    echo "i < 1"
fi
```

do loop:

```
i=0
imax=10
while [ $i -le imax ] ; do
    echo "i = $i"
    i=`expr $i + 1`
done
```