New Zealand's Climate Data in R — An Introduction to clifro

Blake M. Seers & Nick T. Shears*

March 25, 2015

Abstract

The National Climate Database contains data collected from around 6,500 weather stations throughout New Zealand, providing a necessary repository for any study involving New Zealand's weather or climate. Although data from this database can be accessed via a web portal (CliFlo), the vast amount of information is not readily consumable. The `clifro` R package provides the necessary functions to easily integrate the National Climate Database with R, and provides a variety of elegant plotting methods for data exploration. Also, by supplying functions to access the database via CliFlo programmatically, `clifro` allows for transparent and reproducible research when using these data.

1 Introduction

Publicly available and accessible weather and climate data is a necessity for transparent, reproducible and credible research not only within the atmospheric sciences, but any discipline requiring the use of such data. While many public repositories exist online making the data accessible to the public, the *information* is not readily consumable. For researchers and enthusiasts with experience in converting raw data to information, this process takes time, and for endusers unfamiliar with these tasks, this conversion can be daunting.

The R software is a cross-platform, open-source language and environment for statistical computing (R Core Team, 2014) and is becoming very popular among geoscientists due to it's geospatial data handling and analysis capabilities, alongside it's attraction for reproducible research (Pebesma et al., 2012). There are a few add-on packages created in R for accessing global and regional, weather and climate data from online databases including real-time and historic, modelled and reanalysis data hosted by NOAA (Bowman and Lees, 2015; Edmund et al., 2014; Kemp et al., 2012) and observational data from weather underground (Narasimhan, 2014), although a rich coverage of New Zealand's weather and climate is limited.

*Jointly, the Department of Statistics and Leigh Marine Lab, University of Auckland

New Zealand's National Climate Database stores weather data obtained from around 6,500 stations located all around New Zealand for various time spans dating as far back as 1850. These data can be obtained via the online portal, provided by the National Institute of Water and Atmospheric Research (NIWA), known as CliFlo (`http://cliflo.niwa.co.nz/`). Due to the rich spatial extent these data span in New Zealand they have been used for; interpolating daily potential evapotranspiration (Tait and Woods, 2007) and rainfall (Tait et al., 2006, 2012), modelling the interannual snow climate (Clark et al., 2009), exploring the climatic drivers of coastal turbidity (Seers and Shears, 2015) along with many others conducted by local groups, councils and institutions.

The `clifro` R package has been designed to minimise the burden of obtaining, cleaning, formatting and visualising data from CliFlo, before any further analysis takes place. This short article demonstrates the basic usage of `clifro`. The remaining sections of this article are set out in order of the intended, typical workflow when using `clifro`:

1. Create a `clifro` user
2. Select datatypes
3. Select weather stations
4. Retrieve the data
5. Plot the data

The R code in the following sections are differentiated from the text by being contained within the light-grey boxes and can be run interactively in R, assuming the clifro package has first been downloaded from CRAN and loaded into the current session. The output from R is distinguished from the user input by the output lines starting with '`#R>`', there is no prefix for the input R code to make copy-pasting easier.

```
# Download the latest version from CRAN
install.packages("clifro")

# Load into the current R session
library(clifro)
```

This package uses `ggplot2` (Wickham, 2009) for all the plotting methods, along with `RColorBrewer` (Neuwirth, 2014) for the colours. `XML` (Lang, 2013), `selectr` (Potter, 2012) and `RCurl` (Lang, 2014) are used for sending and retrieving data from the web including authentication and XML parsing, `reshape2` (Wickham, 2007), `lubridate` (Grolemund and Wickham, 2011) and `scales` (Wickham, 2014) are also imported to aid in data manipulation and date conversion. The documentation for `clifro` was written using `roxygen2` (Wickham et al., 2014) and `knitr` (Xie, 2014).

2 Create a clifro user

To download data from the National Climate Database, the user needs to have subscribed to CliFlo (free of charge) to set up an account (`http://cliflo.niwa.co.nz/pls/niwp/wsubform.intro`). The Reefton electronic weather station (EWS) is the only CliFlo station where data is available to anyone without a subscription, therefore the user may forgo subscribing if only data from this station is required. For reproducibility purposes, this article demonstrates the usability of `clifro` using only the data from the Reefton EWS station, however subscribers will have access to data from all the CliFlo stations.

The `clifro` user is the first requirement for building a query and one is easily created using the `cf_user` function, where the user supplies their CliFlo username and password.

```
# Create a clifro user
me_cfuser = cf_user(username = "public", password = "")
```

3 Select datatypes

Choosing datatypes using the CliFlo web portal is done by selecting the appropriate options from a range of menus, checkboxes and combo boxes using the datatye selection menu (`http://cliflo.niwa.co.nz/pls/niwp/wgenf.choose_datatype?cat=cat1`). In `clifro`, datatype selection uses exactly the same menus and options, but can be performed using the single `cf_datatype` function. This function can be called without any arguments that takes the user through interactive menus and options, otherwise the datatypes may be chosen programatically if the corresponding menu options are known in advance. The following two subsections demonstrate choosing a single datatype (9am surface wind) using both the interactive and programmatic approaches. Section 3.3 generalises this approach of programmatically choosing a single datatype to situations involving more than one datatype that may each contain more than one option.

3.1 Interactive datatype selection

Selecting datatypes in `clifro` is made easy with the use of interactive menus to help the user choose the datatype of interest.

```
surface_wind_dt = cf_datatype()
```

```
#R> Daily and Hourly Observations
#R>
#R> 1: Combined Observations
```

```
#R> 2: Wind
#R> 3: Precipitation
#R> 4: Temperature and Humidity
#R> 5: Sunshine and Radiation
#R> 6: Weather
#R> 7: Pressure
#R> 8: Clouds
#R> 9: Evaporation / soil moisture
```

The first menu that appears is the 'Daily and Hourly Observations' menu and in this case we are interested in 'Wind' (**2**).

```
#R> Wind
#R>
#R> 1: Surface wind
#R> 2: Max Gust
```

The next menu prompts us for the type of wind we are interested in, for which we choose 'Surface wind' (**1**).

```
#R> Surface wind options
#R>
#R> 1: WindRun
#R> 2: HlyWind
#R> 3: 3HlyWind
#R> 4: 9amWind
```

The next menu is the options for the chosen datatype. We are interested in daily surface wind therefore we choose '9amWind' (**4**). The user may choose more than one option for a given datatype although these must be chosen one at a time. This is made possible by a menu prompting whether or not to select another datatype, every time an option is chosen.

```
#R> Units
#R>
#R> 1: m/s
#R> 2: km/hr
#R> 3: knots
```

The final options menu is typically associated with the units of the datatype (although not always) and is sometimes not necessary, depending on the datatype. For this example there is a fourth menu that prompts for the units of the daily surface wind, which is 'knots' (**3**). The options associated with the 9am surface wind (knots) datatype that are needed for the query are now saved in R as '`surface_wind_dt`'.

```
surface_wind_dt

#R>     dt.name      dt.type dt.options dt.combo
#R> dt1    Wind Surface wind  [9amWind]    knots
```

3.2 Programmatically

The previous example uses the menus interactively to choose a datatype, however repeating this for any given datatype can be cumbersome, and since it requires user input, is undesirable for batch processing and reproducible research. Once the menu options are known, the appropriate datatype can be chosen by passing the relevant menu choices as arguments to the `cf_datatype` function.

Recall the menu choices above for daily surface wind (knots) were 'Wind' (**2**) → 'Surface wind' (**1**) → '9amWind' (**4**) → 'Wind' (**3**). In genreal, if we know the selections needed for each of the four menus then we can choose any datatype without using the menus.

```
surface_wind_dt = cf_datatype(2, 1, 4, 3)
surface_wind_dt

#R>     dt.name      dt.type dt.options dt.combo
#R> dt1    Wind Surface wind  [9amWind]    knots
```

3.3 Selecting multiple datatypes

Most applications involving the environmental data contained within the National Climate Database will require selection of more than one option for more than one datatype. Consider an application where hourly and 9am surface wind are required, along with hourly and daily rainfall, hourly counts of lightning flashes and daily and hourly grass temperature extremes.

There are a few ways to choose all of these datatypes in `clifro`. Firstly, one could go through the menu options one by one, selecting the corresponding datatypes and options and saving the resulting datatypes as different R objects, as done in section 3.1. A less laborious alternative is to create each of these datatypes without the menus, which assumes the selections at each branch of the datatype selection menus are known.

```
# Hourly and 9am surface wind (knots)
surface_wind_dt = cf_datatype(2, 1, c(2, 4), 3)

# Hourly and daily rainfall
rainfall_dt = cf_datatype(3, 1, c(1, 2))
```

```r
# Hourly counts of lightning flashes
lightning_dt = cf_datatype(6, 1, 1)

# Daily and hourly grass temperature extremes
temperature_dt = cf_datatype(4, 2, c(5, 6))
```

If all of these datatypes are to be used within the same query, only one call to the function is required. Table 1 displays the menu and option choices for each of the four datatypes and is in a convenient format for use with the `cf_datatype` function.

	Surface wind	Rainfall	Lightning	Temperature
First selection	2	3	6	4
Second selection	1	1	1	2
Third selection(s)	(2, 4)	(1, 2)	1	(5, 6)
Combo box option	3	NA	NA	NA

Table 1: Main menu and option selection numbers for; 9am and hourly surface wind (knots), daily and hourly rainfall, lightning, and daily and hourly grass temperature extremes.

```r
# Create a single R object containing all these datatypes

#                       Wind     Rainfall Lightning Temperature
all_dt = cf_datatype(c(2,       3,       6,         4),
                     c(1,       1,       1,         2),
                list(c(2, 4), c(1, 2),   1,      c(5, 6)),
                     c(3,       NA,      NA,        NA))
```

```r
all_dt

#R>                       dt.name              dt.type
#R> dt1                      Wind         Surface wind
#R> dt2             Precipitation Rain (fixed periods)
#R> dt3                   Weather            Lightning
#R> dt4 Temperature and Humidity         Max_min_temp
#R>               dt.options dt.combo
#R> dt1   [HlyWind,9amWind]    knots
#R> dt2     [Daily ,Hourly]
#R> dt3              [Ltng]
#R> dt4 [DlyGrass,HlyGrass]
```

4 Select weather stations

There are around 6,500 current and historic weather stations that have data stored within the National Climate Database, hence choosing which stations to return data from can be difficult. Each weather station has a unique 'agent' ID, and if these are known a priori then it is a simple task of providing the comma separated agent numbers to the `cf_station` function.

```
# Reefton EWS (3925)
reefton_st = cf_station(3925)
reefton_st

#R>          name network agent      start        end open
#R> 1) Reefton Ews  F21182  3925 1960-08-01 2015-03-26 TRUE
#R>    distance     lat    lon
#R> 1)        0 -42.117 171.86
```

Note that although only the agent number is required, the resulting object contains all other relevant station information including the status of the station (open or closed), latitude and longitude, date range etc. This station information is not held within R therefore each time this function is called, the relevant station information is scraped from the web, resulting in up-to-date station information. Plotting the location of this station on a map is easily done by calling the `cf_save_kml` function for export to a KML file, or plotted directly in R (figure 1) using the `ggmap` package (Kahle and Wickham, 2013).

```
ggmap(get_map("Reefton", zoom = 7)) +
  geom_point(data = as(reefton_st, "data.frame"),
             shape = "X", size = 5)
```

4.1 Searching for weather stations

The agent numbers are generally not known and therefore need to be searched. Searching for stations in `clifro` is equivalent to searching for stations using the CliFlo web portal, however `clifro` provides important additional features currently unavailable with CliFlo such as simultaneous searches and visualising stations on a map.

To search for all stations within the greater Auckland region, for example, and visualise these stations by outputting to a KML file is easily done in `clifro`, with the use of only two functions. The resultant KML file can be opened with any GIS software supporting the Keyhole Markup Language, and contains the unique agent number, network ID, and the start and end dates for each site to make choosing appropriate weather stations much easier (figure 2).

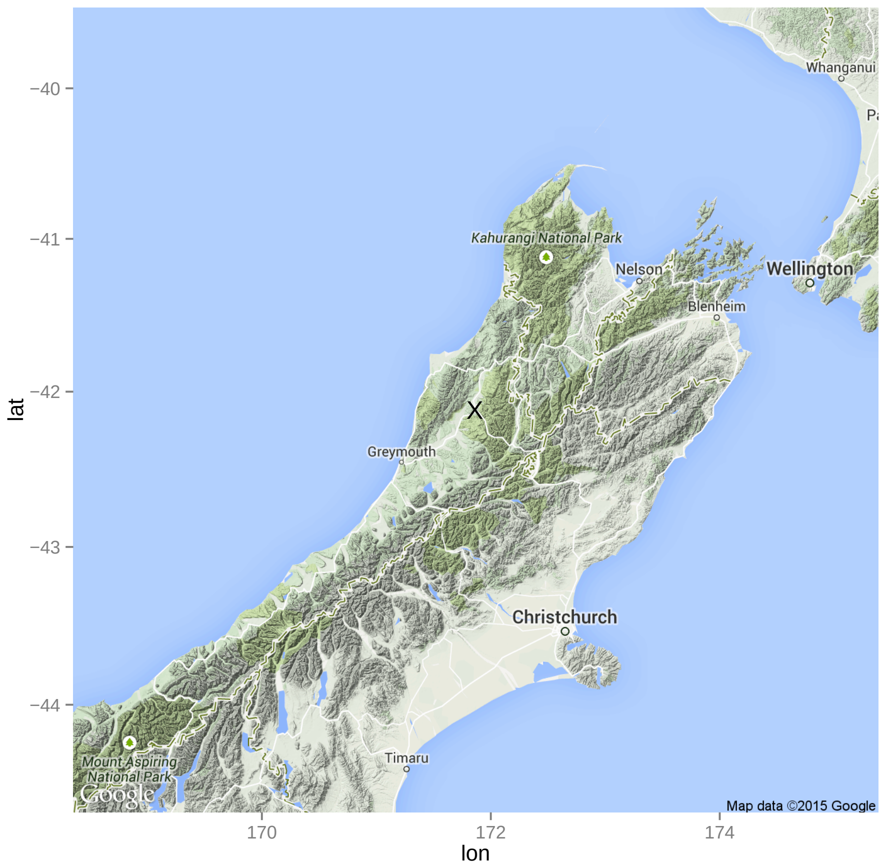

Figure 1: Position of Reefton EWS indicated by an 'X', plotted within R using the `ggmap` package.

```r
all_auckland_st = cf_find_station("Auckland", search = "region",
                                  status = "all")
cf_save_kml(all_auckland_st,
            file_name = "allAucklandWeatherStations")
```

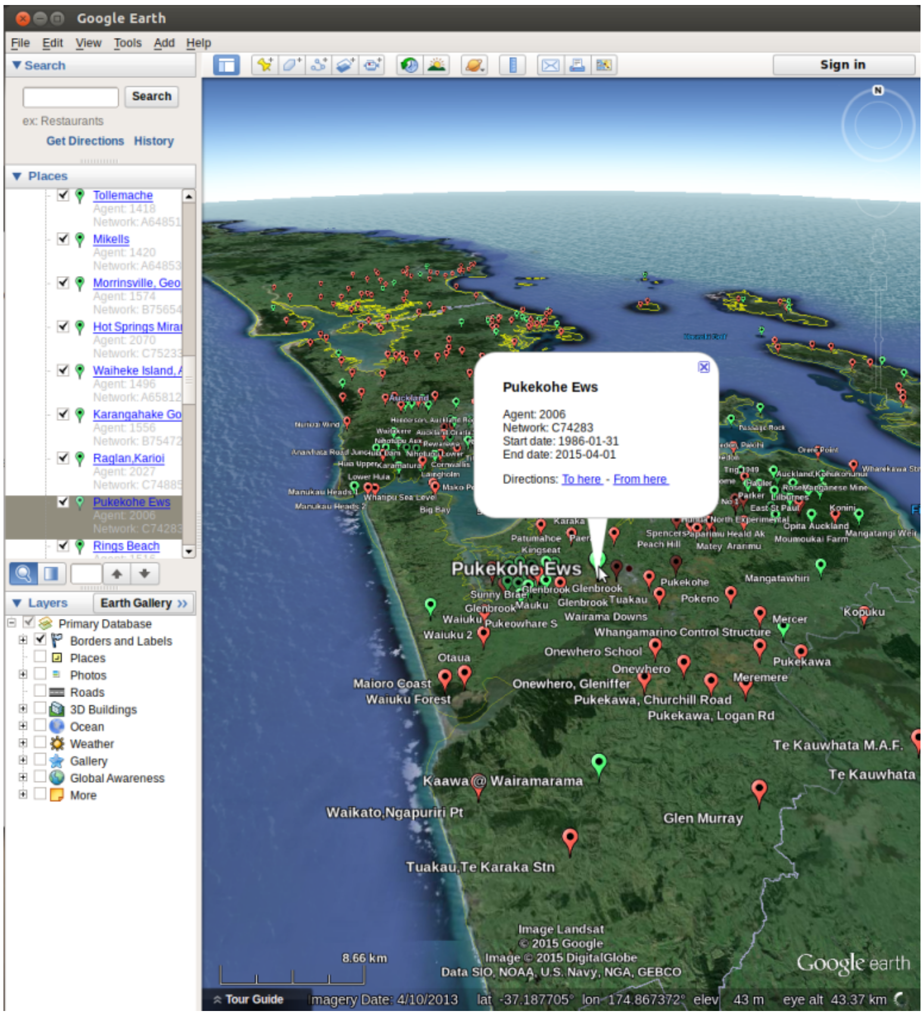

Figure 2: Map showing the locations of all the weather stations in the greater Auckland region. This is a screenshot of the resultant KML file from the cf_save_kml function after opening in Google Earth™ and left-clicking on the Pukekohe electronic weather station. The open and closed weather stations are saved in different sub-folders, and are indicated by green and red markers respectively.

There are 29 regions around New Zealand that are part of the 'region' search and the user can also choose the region interactively by omitting `"Auckland"` in the function call, i.e. `cf_find_station(search = "region", status = "all")`.

5 Retrieve the data

A query to the National Climate Database via **clifro** (and CliFlo) requires a user (section 2), the relevant datatypes (section 3), the locations (section 4), and the date range. All of these are then combined into the `cf_query` function that posts the relevant data to CliFlo that then queries the database and retrieves the data. The data retrieved is then automatically imported back into R and made available to the user as either a single **clifro** data (`cfData`) object if data for one datatype is returned, or a list of **clifro** data objects (`cfDataList`) if data for more than one datatype is returned.

The following R code queries the National Climate Database for the multiple daily datatypes selected in section 3 at Reefton EWS for the year 2014. Note only the daily datatypes are queried as public users have a query limit of 1,000 output rows.

```
# Select only the daily options for the datatypes
all_daily_dt = cf_datatype(c(2, 3, 6, 4),
                           c(1, 1, 1, 2),
                           list(4, 1, 1, 5),
                           c(3, NA, NA, NA))
```

```
# Send the query to retrieve the weather data
my_query = cf_query(user = me_cfuser,
                    datatype = all_daily_dt,
                    station = reefton_st,
                    start_date = "2014-01-01 00",
                    end_date = "2015-01-01 00")
my_query

#R> List containing clifro data frames:
#R>                data      type              start
#R> df 1) Surface Wind  9am only (2014-01-01  9:00)
#R> df 2)         Rain     Daily (2014-01-01  9:00)
#R> df 3)      Max_min     Daily (2014-01-01  9:00)
#R>                           end rows
#R> df 1) (2014-12-31  9:00)  365
#R> df 2) (2014-12-31  9:00)  361
#R> df 3) (2014-10-02  9:00)  274
```

All the 1,000 output rows are saved in R in the object '`my_query`', and are conveniently separated by datatype. If the user mistakenly did not assign the query to an R object, the data can be retrieved by calling `cf_last_query()` rather than unnecessarily depleting subscription rows. The daily lightning datatype is clearly unavailable at Reefton EWS as it was part of the query but is not included in the output data. The user can search for stations using datatypes to ensure the data are available before sending a query by simply including the `clifro` datatype object into the `cf_find_station` function. The interested reader is referred to the appriopriate vignette (`vignette("choose-datatype")`).

6 Plot the data

Each dataframe returned by the query is assigned a relevant class to make generic plotting of these objects possible, and easy for the enduser. The method for default plotting of `clifro` data is therefore automatically determined based on the class of the object, resulting in efficient and easy exploration of the weather data (figure 3).

```
# Default plot for the first dataframe in the list
# (9am surface wind)
plot(my_query)

# Default plot for the second dataframe in the list
# (daily rainfall)
plot(my_query, 2)

# Remove the soil runoff and deficit from this plot
plot(my_query, 2, include_runoff = FALSE, ggtheme = "bw")

# Default plot for the third dataframe in the list
# (temperature extremes)
plot(my_query, 3, ggtheme = "classic")
```

Wind data returned from a `clifro` query has two other generic plotting functions to further explore the data. Although the windrose is the default plot for wind data, this is not very informative if the temporal aspect of wind speed or direction is of interest. The following R code shows how to easily plot the wind speed and directions through time (figure 4), and then demonstrates how to produce a seasonal windrose (figure 5).

```
speed_plot(my_query, ggtheme = "bw")
direction_plot(my_query, ggtheme = "bw")
```

These plots allow the user to explore the temporal wind patterns in speed and direction. The wind at Reefton appears to increase in speed around October to February (warmer months in New Zealand), and also the predominant south-easterlies appear to become predominantly north-westerlies throughout this period. We can further explore the seasonality of the wind data in `clifro` by creating a seasonal windrose, which also shows a similar, seasonal pattern (figure 5).

```
# Create seasons from dates
reefton_months = months(my_query[1]$Date, abbreviate = TRUE)
reefton_season = factor(rep("Summer", length(reefton_months)),
                        levels = c("Summer", "Autumn",
                                   "Winter", "Spring"))
reefton_season[reefton_months %in% unique(reefton_months)[3:5]] =
  "Autumn"
reefton_season[reefton_months %in% unique(reefton_months)[6:8]] =
  "Winter"
reefton_season[reefton_months %in% unique(reefton_months)[9:11]] =
  "Spring"

# Plot seasonal windrose
windrose(speed = my_query[1][, 4],
         direction =my_query[1][, 3],
         facet = reefton_season,
         n_col = 2,
         legend_title = "Reefton wind \nspeed (knots)")
```

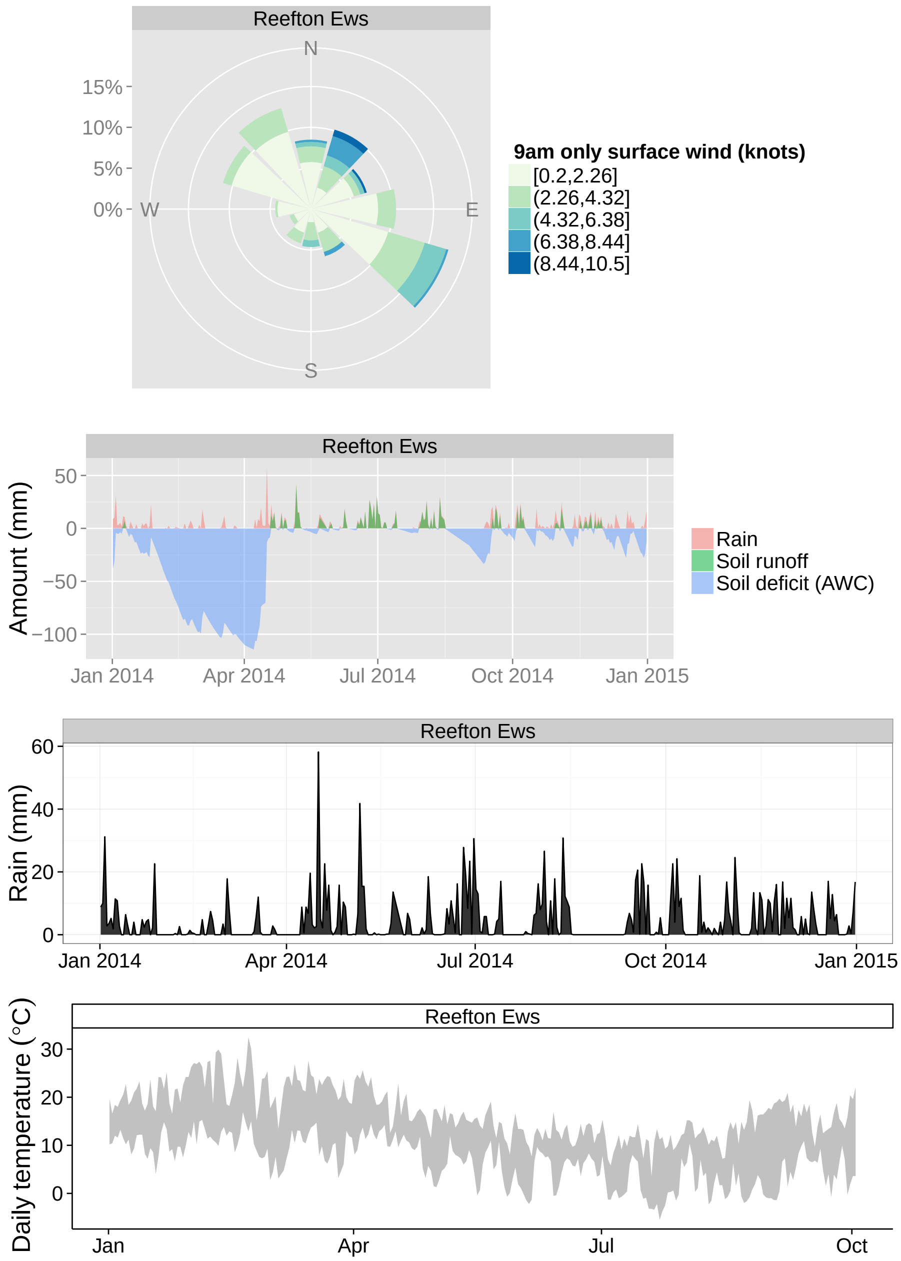

Figure 3: Default plots for the wind, rain and temperature data returned from the National Climate Database. Starting from the top: 1) Wind rose for daily surface wind with all default arguments, 2) Rainfall, runoff and soil deficit plotted through time with all default arguments, 3) Timeseries plot of rainfall using the black-and-white theme, 4) Ribbon plot showing the temperature extremes through time using the classic theme.

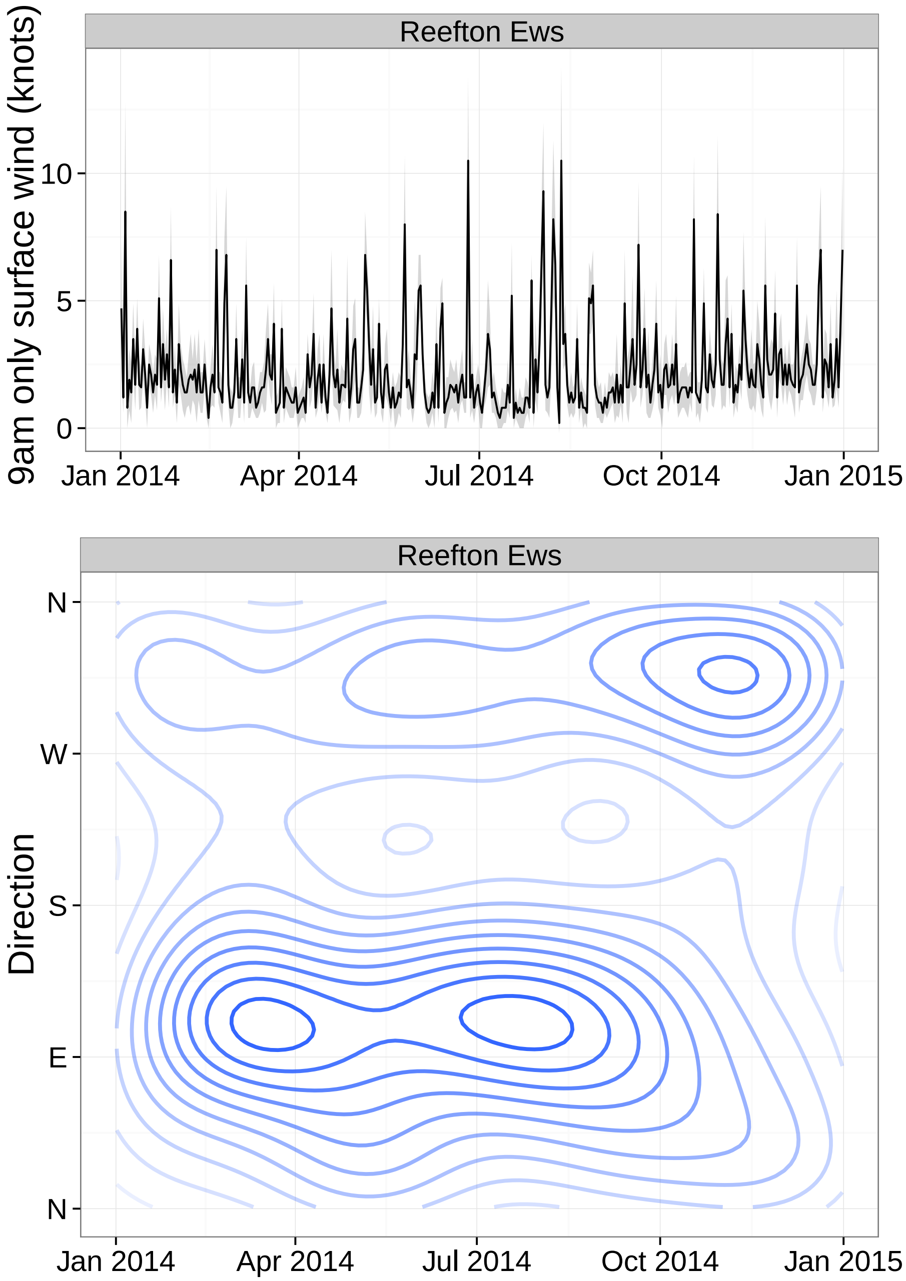

Figure 4: Further generic plots for wind data using the black-and-white theme. The top graph shows the wind speeds (±1 sd) through time , and the bottom graph is a contour plot of wind directions through time. Both of these generic plots aid in the exploration of temporal wind patterns

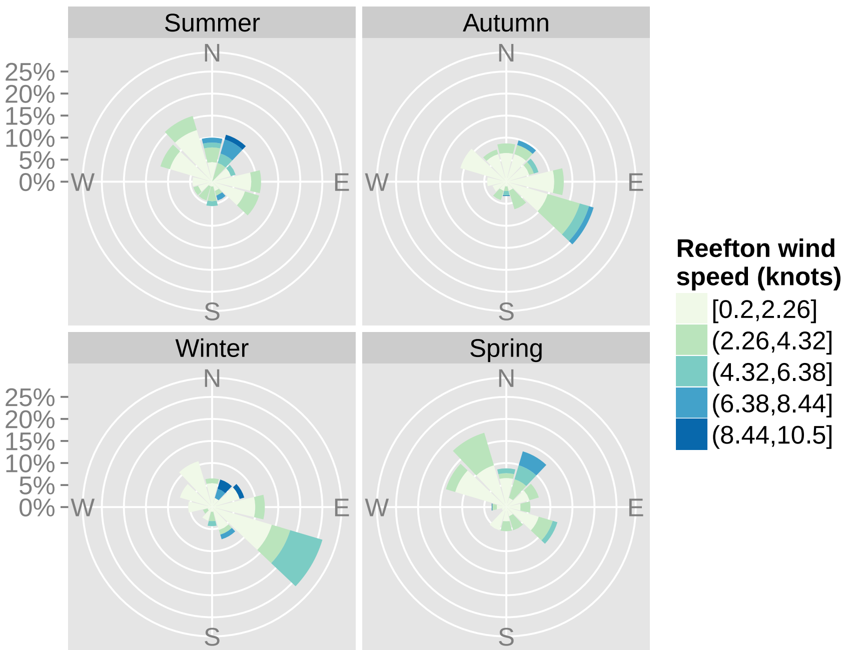

Figure 5: Windroses of surface wind speed at Reefton EWS at each of the four seasons for the year 2013.

7 Conclusion

This introduction to the `clifro` package has shown the intended workflow and basic usage of the primary functions for creating a user, choosing datatypes and stations, querying the National Climate Database and exploring the data, with an emphasis on visualisation techniques throughout. The `clifro` package is intended to make research involving the use of climate data from New Zealand's National Climate Database easier, transparent and reproducible. The examples contained within this document are fairly general and the user is referred to the many examples in the help files of the functions, and the vignettes for more details and extended examples.

References

Daniel C. Bowman and Jonathan M. Lees. Near real time weather and ocean model data access with rNOMADS. *Computers & Geosciences*, 78:88–95, 2015. doi: 10.1016/j.cageo.2015.02.013.

M. Clark, E.. Hreinsson, G. Martinez, A. Tait, A. Slater, J. Hendrikx, I. Owens, H. Gupta, J. Schmidt, and R. Woods. Simulations of seasonal snow for the south Island, New Zealand. *Journal of Hydrology New Zealand*, 48(2):41–58, 2009.

Hart Edmund, Scott Chamberlain, and Karthik Ram. *rnoaa: NOAA Climate Data from R*, 2014. URL `http://CRAN.R-project.org/package=rnoaa`. R package version 0.3.3.

Garrett Grolemund and Hadley Wickham. Dates and times made easy with lubridate. *Journal of Statistical Software*, 40(3):1–25, 2011. URL `http://www.jstatsoft.org/v40/i03/`.

D. Kahle and H. Wickham. ggmap: Spatial visualization with ggplot2. *R Journal*, 5(1):144–161, 2013.

Michael U. Kemp, E. Emiel van Loon, Judy Shamoun-Baranes, and Willem Bouten. RNCEP: global weather and climate data at your fingertips. *Methods in Ecology and Evolution*, 3(1):65–70, 2012. doi: 10.1111/j.2041-210X.2011.00138.x.

Duncan Temple Lang. *XML: Tools for parsing and generating XML within R and S-Plus.*, 2013. URL `http://CRAN.R-project.org/package=XML`. R package version 3.98-1.1.

Duncan Temple Lang. *RCurl: General network (HTTP/FTP/...) client interface for R*, 2014. URL `http://CRAN.R-project.org/package=RCurl`. R package version 1.95-4.3.

Ram Narasimhan. *weatherData: Get Weather Data from the Web*, 2014. URL `http://CRAN.R-project.org/package=weatherData`. R package version 0.4.1.

Erich Neuwirth. *RColorBrewer: ColorBrewer Palettes*, 2014. URL `http://CRAN.R-project.org/package=RColorBrewer`. R package version 1.1-2.

Edzer Pebesma, Daniel Nüst, and Roger Bivand. The R software environment in reproducible geoscientific research. *Eos, Transactions American Geophysical Union*, 93(16):163–163, 2012. doi: 10.1029/2012EO160003.

Simon Potter. Introducing the selectr package. Technical report, The University of Auckland, Auckland, New Zealand, 2012. URL `http://stattech.wordpress.fos.auckland.ac.nz/2012-10-introducing-the-selectr-package/`.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2014. URL `http://www.R-project.org/`.

B.M. Seers and N.T. Shears. Spatio-temporal patterns in coastal turbidity — Long-term trends and drivers of variation across an estuarine-open coast gradient. *Estuarine, Coastal and Shelf Science*, 154:137–151, 2015. doi: 10.1016/j.ecss.2014.12.018.

A. Tait and R. Woods. Spatial interpolation of daily potential evapotranspiration for New Zealand using a spline model. *Journal of Hydrometeorology*, 8 (3):430–438, 2007. doi: 10.1175/JHM572.1.

A. Tait, R. Henderson, R. Turner, and X. Zheng. Thin plate smoothing spline interpolation of daily rainfall for New Zealand using a climatological rainfall surface. *International Journal of Climatology*, 26(14):2097–2115, 2006. doi: 10.1002/joc.1350.

A. Tait, J. Sturman, and M. Clark. An assessment of the accuracy of interpolated daily rainfall for New Zealand. *Journal of Hydrology New Zealand*, 51 (1):25–44, 2012.

Hadley Wickham. Reshaping data with the reshape package. *Journal of Statistical Software*, 21(12):1–20, 2007. URL `http://www.jstatsoft.org/v21/i12/`.

Hadley Wickham. *ggplot2: elegant graphics for data analysis*. Springer New York, 2009. ISBN 978-0-387-98140-6. URL `http://had.co.nz/ggplot2/book`.

Hadley Wickham. *scales: Scale functions for graphics.*, 2014. URL `http://CRAN.R-project.org/package=scales`. R package version 0.2.4.

Hadley Wickham, Peter Danenberg, and Manuel Eugster. *roxygen2: In-source Documentation for R*, 2014. URL `http://CRAN.R-project.org/package=roxygen2`. R package version 4.1.0.

Yihui Xie. *knitr: A General-Purpose Package for Dynamic Report Generation in R*, 2014. URL `http://CRAN.R-project.org/package=knitr`. R package version 1.8.