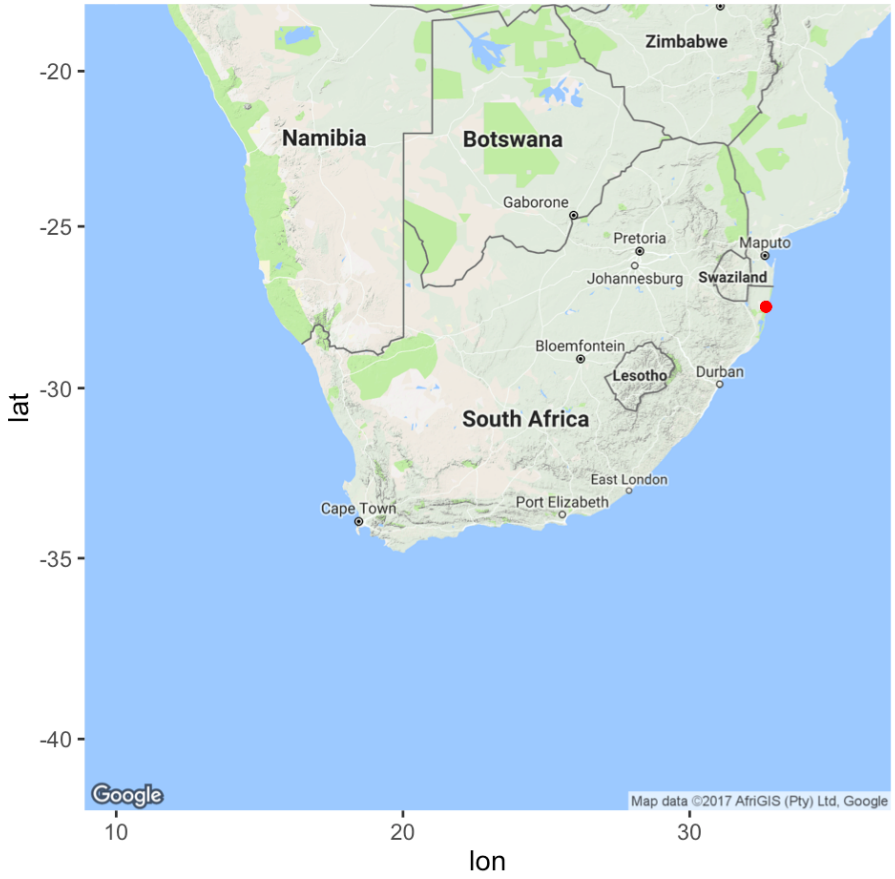
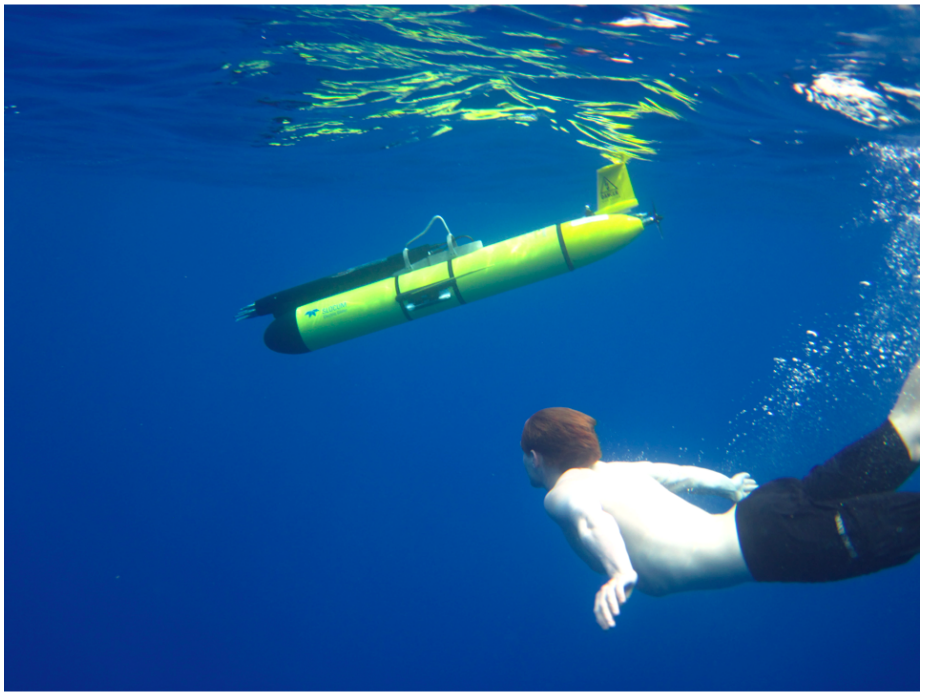
Using R for oceanography 2

Katrin Tirok, UKZN

2/18/2017

glider mission





glider data

- raw data in binary format –> bash script to read binary –> matlab package (https://github.com/kerfoot/spt) to read 'dbd' format and export data –> R
- use gsw (gibbs sea water library) to calculate seawater characteristics such as salinity, density … from conductivity and temperature

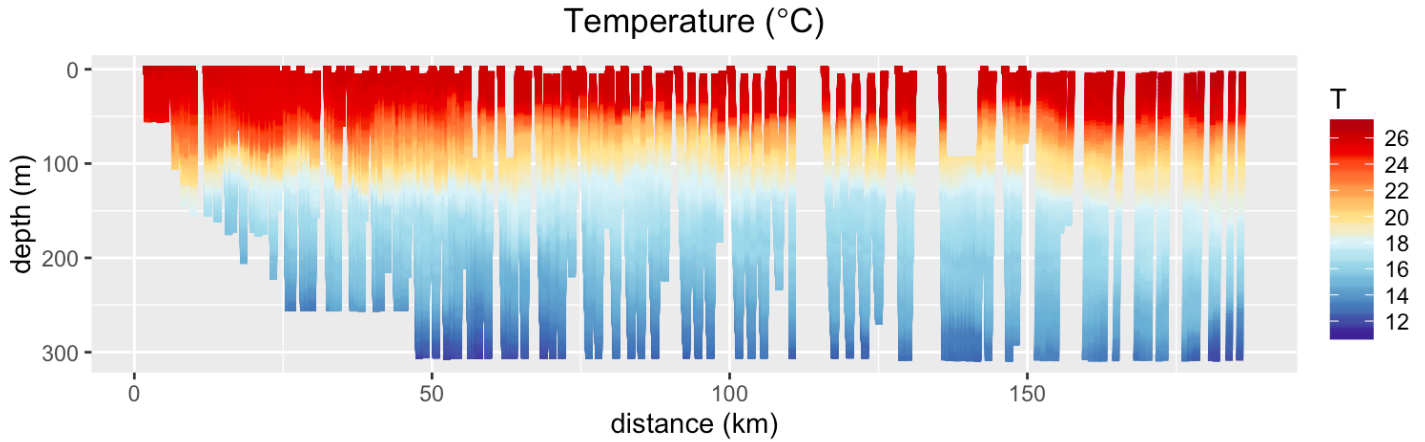
```
# practical salinity
SP = gsw_SP_from_C(C, t, p)

# in-situ density
rho = gsw_rho(SA, CT, p)
```

- where to go from here?

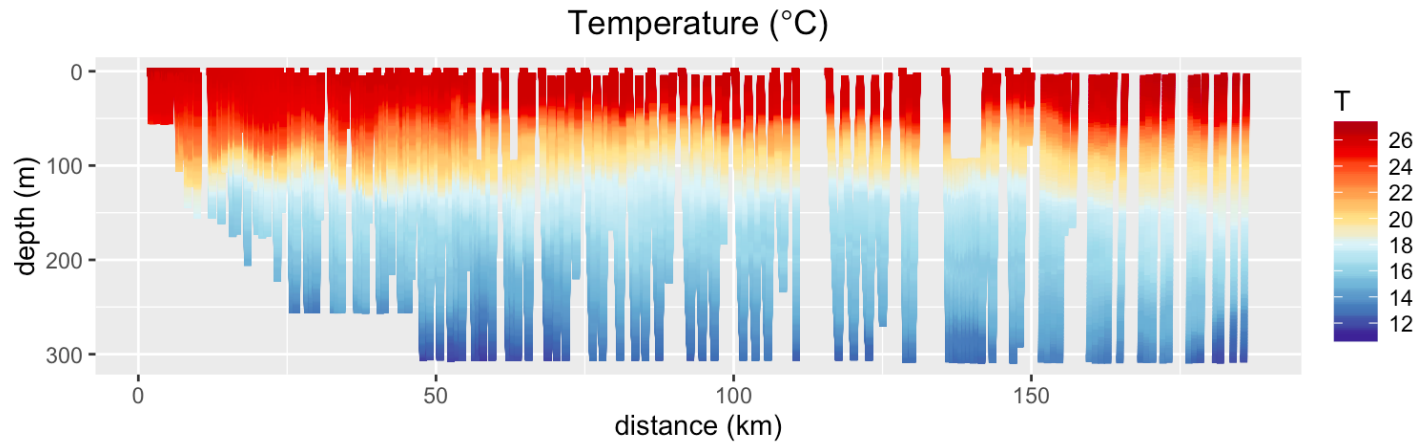
plot of temperature over depth versus distance

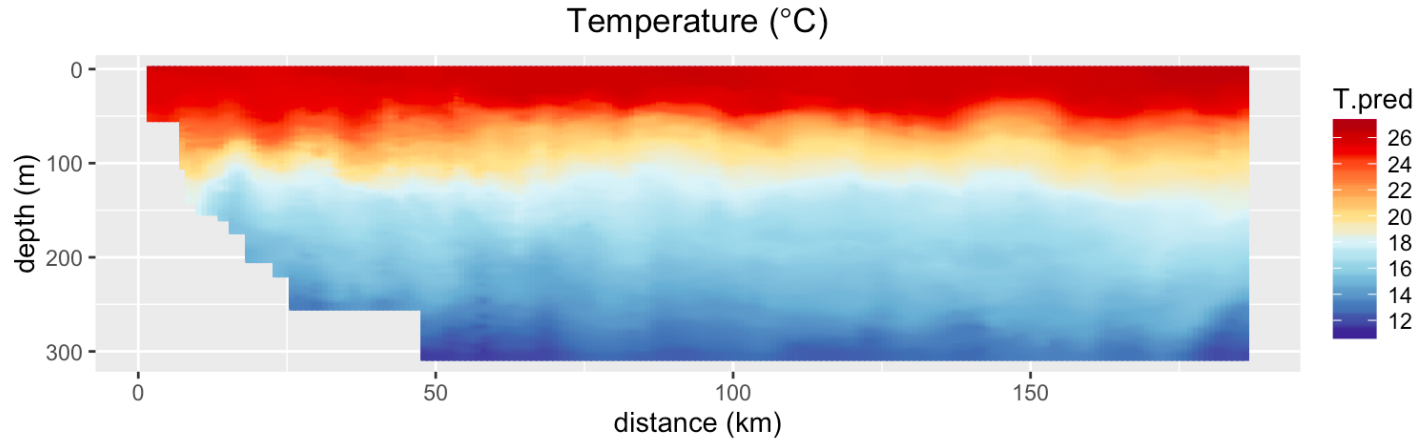
- irregular data points –> use scatterplot with colourscale (RColorBrewer) to plot

```
library(RColorBrewer)
library(ggplot2)
g <- ggplot(data, aes(x = dist, y = depth)) + geom_point(aes(col = T), size = 1,
    shape = 15) + scale_colour_gradientn(colours = rev(brewer.pal(10, "RdYlBu")),
    breaks = seq(10, 28, 2), limits = c(11, 27)) + scale_y_reverse() + labs(x = "distance (km)'
    y = "depth (m)", title = expression(paste("Temperature (", degree, "C)")))
```



Plot of temperature over depth versus distance



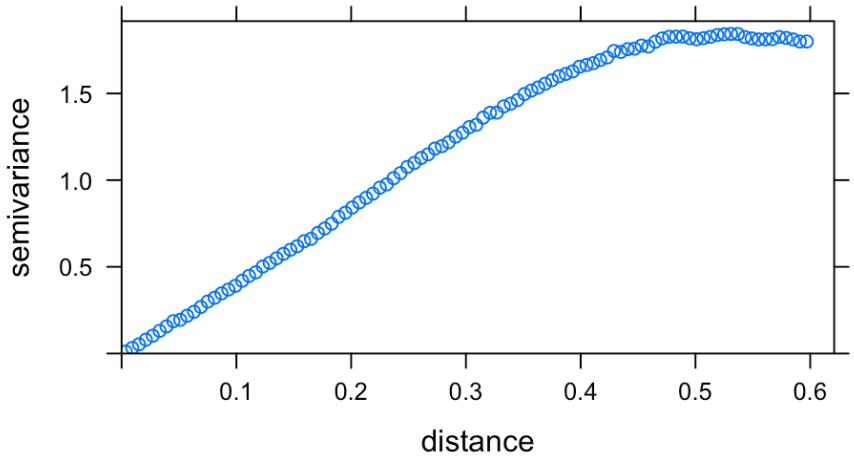


kriging

- geostatistical method that considers distance and degree of variation between known data points
- basis is a spatial correlation model –> variogram
- packages I use:
  - sp: Classes and Methods for Spatial Data
  - gstat: Spatial and Spatio-Temporal Geostatistical Modelling, Prediction and Simulation

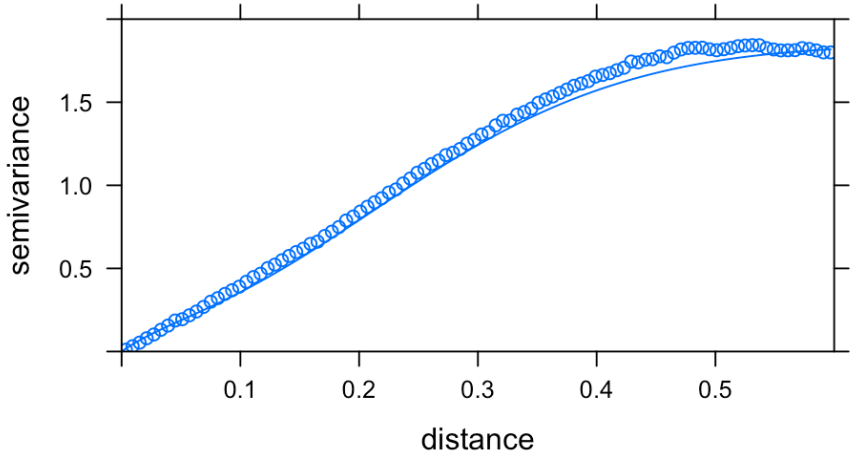
sample variogram with 'variogram'

```
library(gstat)
library(sp)
coordinates(mydata) <- ~dist.s + depth.s
vg <- variogram(temp ~ dist.s + depth.s, mydata, cutoff = 0.6, width = 0.6/100)
```



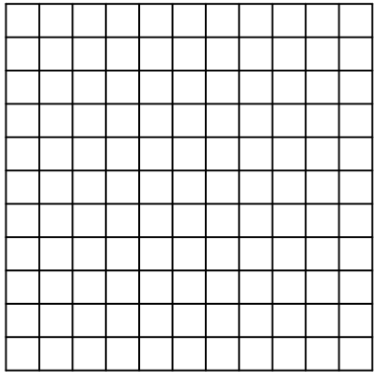
fit model to variogram with 'vgm'

```
# model for T --> combination of exponential (first bit) and gaussian
vgmT <- vgm(psill = 1.65, "Gau", range = 0.3, nugget = 0, add.to = vgm(psill = 0.2,
    "Exp", range = 0.04))
```



define new spatial grid with 'expand.grid'

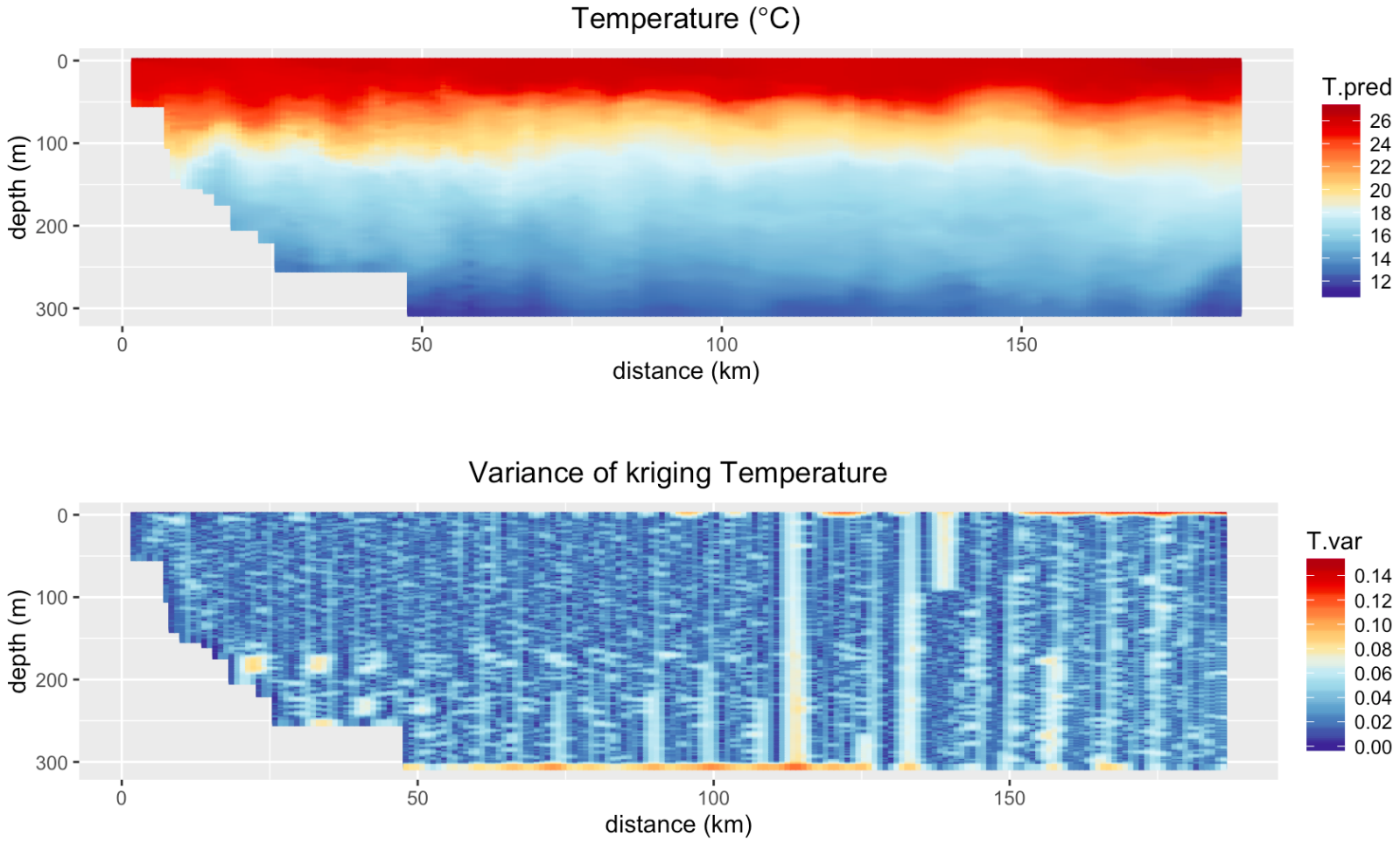
```
x.range <- as.integer(range(mydata$dist.s))
y.range <- as.integer(range(mydata$depth.s))
data.grd <- expand.grid(x = seq(from = x.range[1], to = x.range[2], by = 0.1),
    y = seq(from = y.range[1], to = y.range[2], by = 0.1))
coordinates(data.grd) <- ~x + y
gridded(data.grd) <- TRUE
```



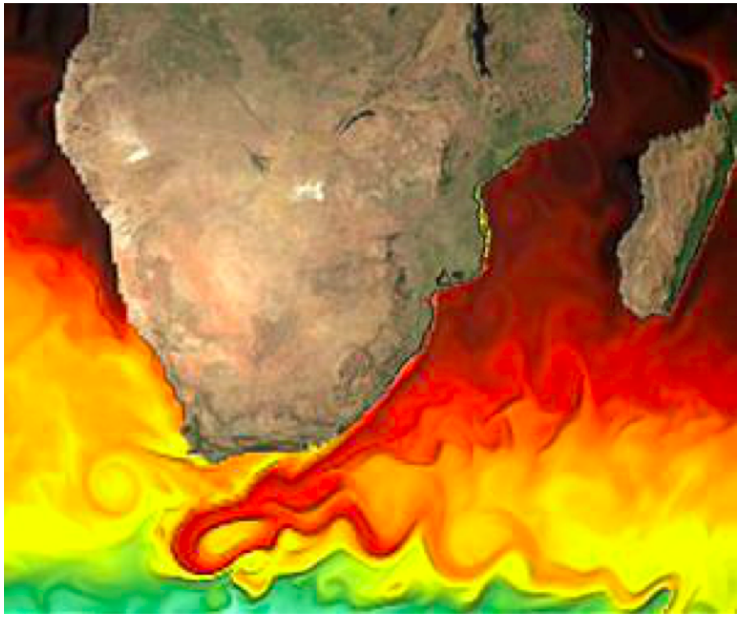
universal local kriging using 'krige'

```
# universal local kriging maximum distance --> 20km (ca 31.5 m for depth)
coordinates(mydata) <- ~dist.s + depth.s
maxd <- (20 - min(mydata$dist))/(max(mydata$dist) - min(mydata$dist))

kr.l <- krige(temp ~ dist.s + depth.s, mydata, newdata = data.grd, model = vgmT,
    maxdist = maxd, nmin = 10)
```

kriging results



average current velocities



- velocity values –> how to show current velocity on map?

average current velocities

- velocities given as x and y component of vector, can be transformed to speed and direction

```
a <- read.csv("velocities.csv", header = T, sep = ",")
```

```
##       m_final_water_vx m_final_water_vy m_gps_lat m_gps_lon
## 35479        -0.341266        -0.367326 -2743.987  3242.502
## 35496        -0.175817        -0.212771 -2744.030  3242.489
## 37053        -0.175817        -0.212771 -2744.414  3242.643
## 37069        -0.206427        -0.291396 -2744.456  3242.639
## 37100        -0.206427        -0.291396 -2744.463  3242.638
## 37104        -0.206427        -0.291396 -2744.463  3242.638
```

current velocities

- velocities as x(u) and y(v) component –> convert current to direction and speed using 'metvurst' and uv2wdws
- METeorological Visualisation Utilities using R for Science and Teaching
- calculate max and min speed

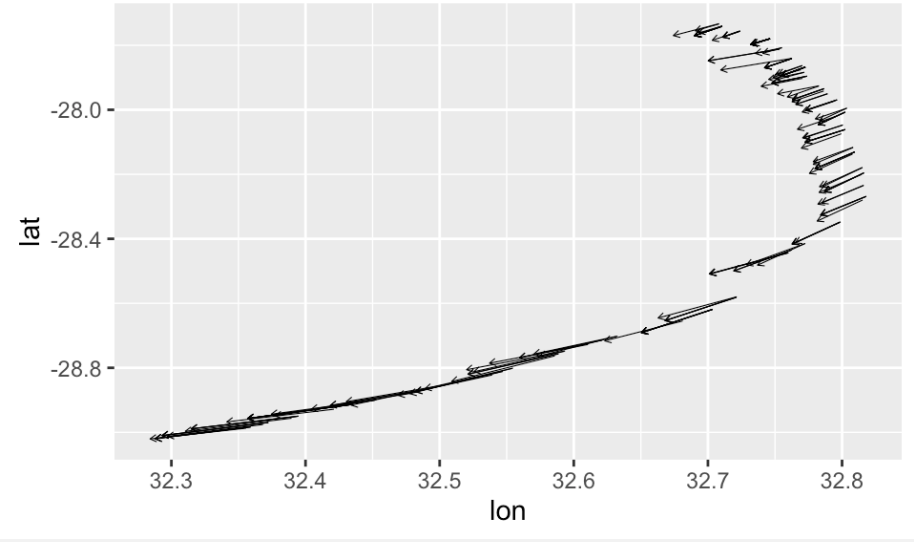
```
u = c$m_final_water_vx
v = c$m_final_water_vy
library(metvurst)
cdcs <- uv2wdws(u, v)

print(paste("max=", max(cdcs$ws), ", min=", min(cdcs$ws), " m/s", sep = ""))
```

```
## [1] "max=1.11, min=0.23 m/s"
```

plot velocities

- ggplot + geom_segment (draws a straight line between points)

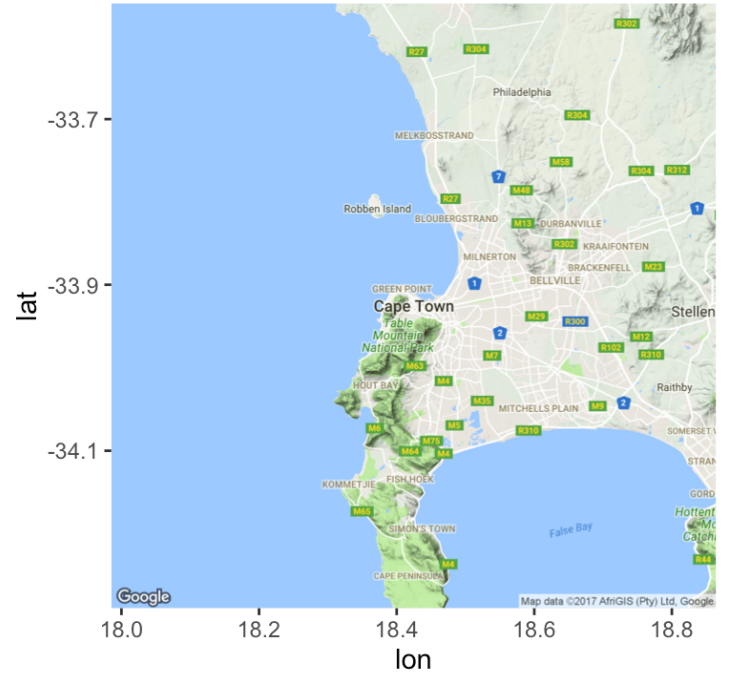
```
library(ggplot2)
scale <- 0.1
mycoord <- data.frame(lon, lat, u = u * scale, v = v * scale)
g <- ggplot() + geom_segment(data = mycoord, mapping = aes(x = lon, y = lat,
    xend = lon + u, yend = lat + v), arrow = arrow(length = unit(0.1, "cm")),
    size = 0.2, color = "black")
```



plot map

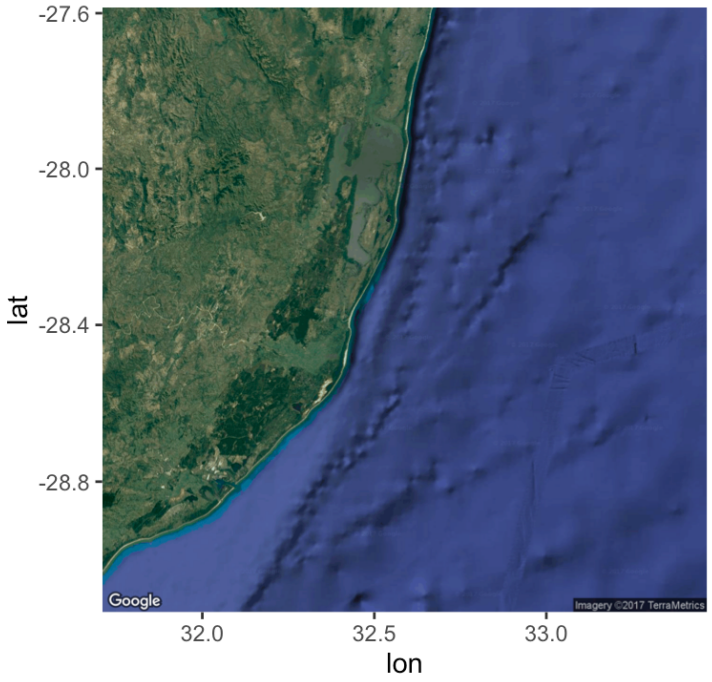
- using package ggmap: spatial visualisation with ggplot2
- function 'get_map'

```
library(ggmap)
m <- ggmap(get_map(location = "cape town"))
```



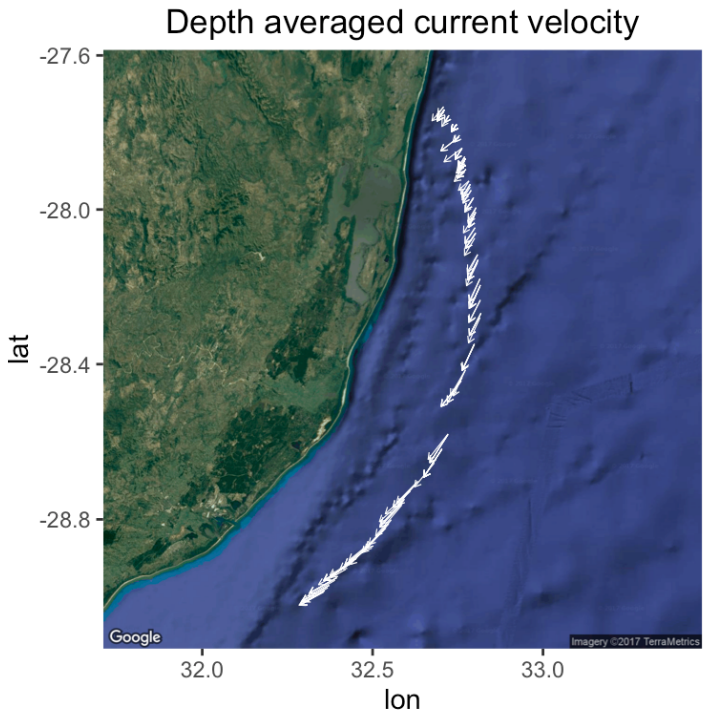
plot map

```
map <- get_map(location = c(min(mycoord$lon) - 0.2, min(mycoord$lat) - 2, max(mycoord$lon) +
    0.2, max(mycoord$lat) + 2), zoom = 9, maptype = "satellite")
m <- ggmap(map)
```



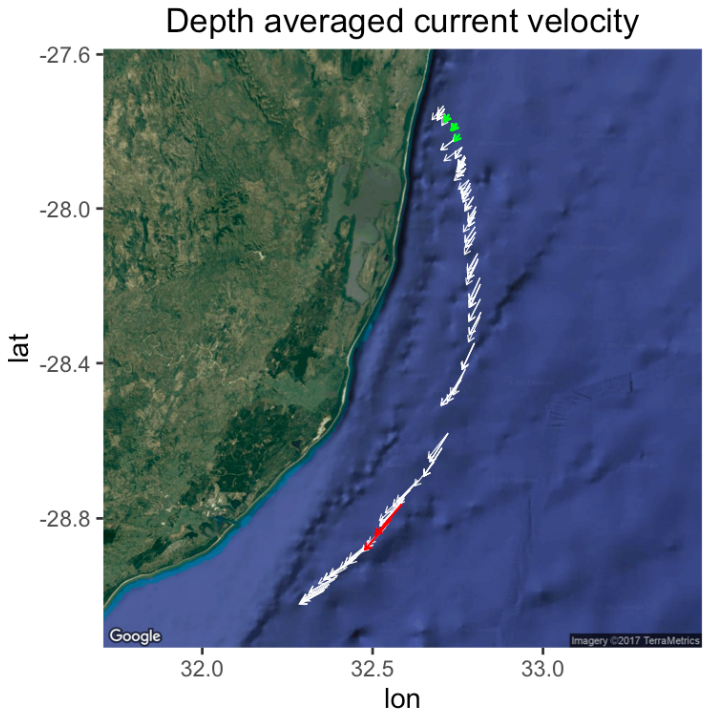
plot velocities on map

```
g1 <- ggmap(map) + geom_segment(data = mycoord, mapping = aes(x = lon, y = lat,
    xend = lon + u, yend = lat + v), arrow = arrow(length = unit(0.1, "cm")),
    size = 0.2, color = "white") + ggtitle(...)
```



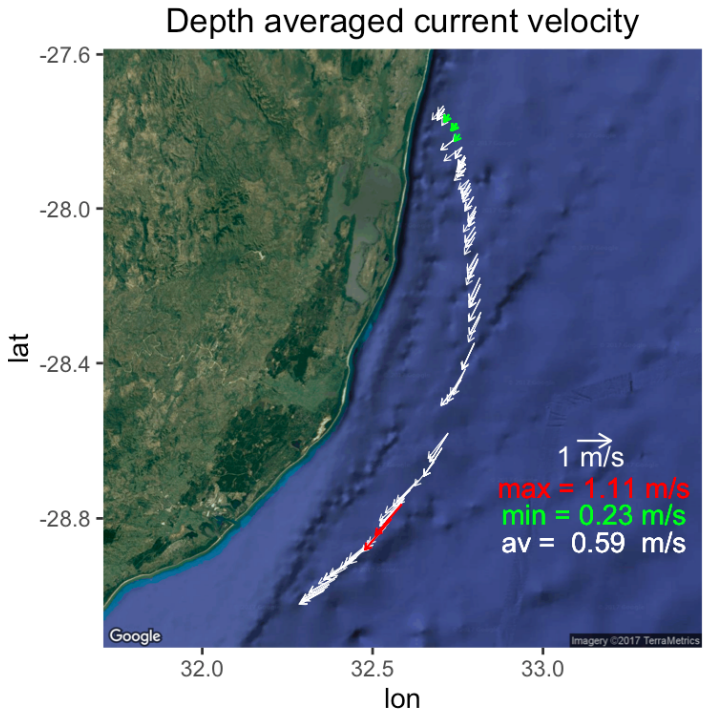
add min and max vel

```
g2 <- g1 + geom_segment(data = max, mapping = ..., arrow = arrow(..., color = "red")) +
    geom_segment(data = min, mapping = ..., arrow = arrow(..., color = "green"))
```



add title and labels

```
g3 <- g2 + geom_segment(mapping = aes(x = 33.1, y = -28.6, xend = 33.1 + 1 *
    scale, yend = -28.6), arrow = ...) + geom_text(mapping = aes(x = 33.1 +
    0.04, y = -28.6 - 0.04, label = "1 m/s"), data = label, color = "white") +
    ...
```



summary

- 'oce' for processing of various ocean data
- other R packages can help to solve specific questions
- oceanography still very Matlab focused, need sharing