Comparing velocity packages

Brunno Oliveira

02 November, 2023

Contents

setup

```r
# devtools::install_github("cbrown5/vocc")
# devtools::install_github("oliveirab/VoCC", build_vignettes = FALSE)

list.of.packages <- c("climetrics","vocc","VoCC1","terra","ggplot2","GGally","data.table","geodata","ti
# new.packages <- list.of.packages[!(list.of.packages %in% installed.packages()[,"Package"])]
# if(length(new.packages)) install.packages(new.packages)
sapply(list.of.packages, require, character.only = TRUE)
```

```
## climetrics        vocc       VoCC1       terra     ggplot2      GGally data.table
##        TRUE        TRUE        TRUE        TRUE        TRUE        TRUE        TRUE
##     geodata   tidyterra   gridExtra
##        TRUE        TRUE        TRUE
```

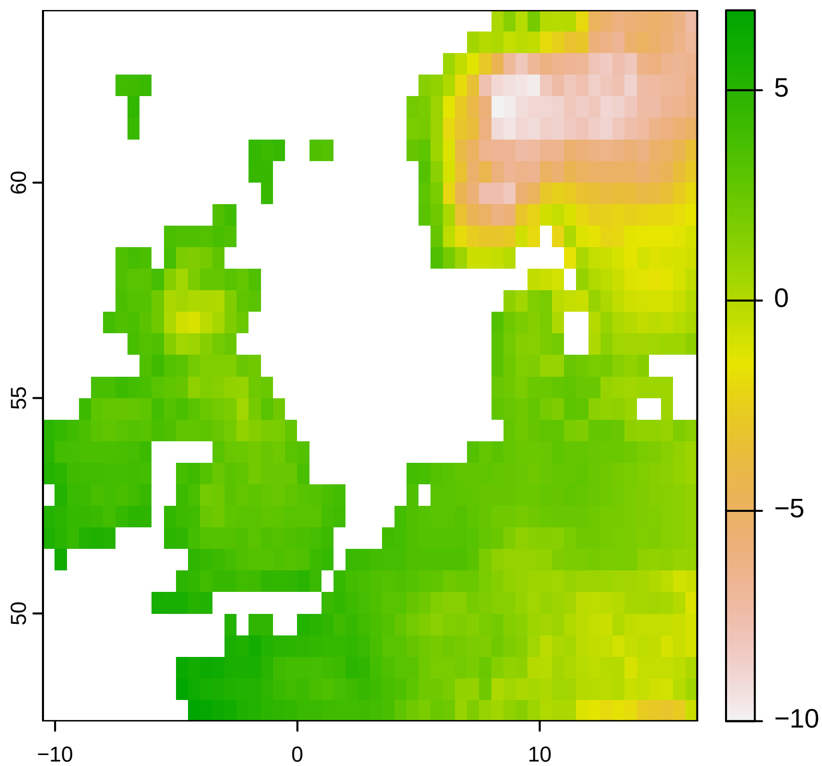
```r
# Source velocity functions adapted from the package climetrics after applying some corrections describ
# source("/storage/simple/projects/t_cesab/brunno/Exposure-SDM/R/velocity_functions.R")
source(here::here("R/velocity_functions.R"))
# Source function generated by biov1 to create the spatial gradient at the latitudinal direction.
source(here::here("R/gael_velocity.R"))
# Source settings (here used only to the Eckert equal area projection)
source(here::here("R/settings.R"))

bearing_to_angle <- function(x) ifelse(x <=90, (90-x), (360 - (x - 90)))
```

Compare velocity R-packages

In order to generate comparable results across multiple R packages, we sourced climate data from the package `climetrics`. These represent monthly mean annual temperature from Jan 1991 to Dec 2020. The geographical area covers North of France, Great Britain and West of Norway.

Get temperature data

```r
# path to the dataset folder
filePath <- system.file("external/", package="climetrics")
tmean1 <- rast(paste0(filePath,'/tmean.tif'))
plot(tmean1[[1]])
```




```r
# corresponding dates
n <- readRDS(paste0(filePath,'/dates.rds'))

# get averages per year
terra::time(tmean1) <- as.Date(n)
tmean1 <- tapp(tmean1, "years", mean)

# project to equal area
tmean <- terra::project(tmean1,Eckt)

# Global continental lines
globe_lines <- vect(rnaturalearth::ne_countries(scale = "small"))
```

```
## Warning: The ‘returnclass‘ argument of ‘ne_download()‘ sp as of rnaturalearth 1.0.0.
## i Please use ‘sf‘ objects with {rnaturalearth}, support for Spatial objects
##   (sp) will be removed in a future release of the package.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```
globe_lines <- terra::aggregate(globe_lines)
```

Calculate velocities

Climetrics

This package is based on the package vocc (not VoCC!). I found that both packages have an issue when calculating the spatial gradient. The issue is that when extracting environmental data from the RasterStack they use only the first year to calculate the spatial gradient. Instead, the package VoCC uses the mean of all years (calc(r,mean)), which is the correct thing to do! To prove this issue, I calculate velocity with this package using the both the default (wrong) method and the corrected procedure (using mean raster).

```
start <- Sys.time()

### Wrong way
# Velocity
v_climetrics1 <- climetrics:::gVelocity(tmean)

climetrics_time <- Sys.time() - start

# Spatial gradient (wrong way)
spgrad_climetrics1 <- climetrics:::.spatialgradTerra(tmean)

spgrad_climetrics1$NS[is.na(spgrad_climetrics1$NS)] <- 0
spgrad_climetrics1$WE[is.na(spgrad_climetrics1$WE)] <- 0
spgrad_climetrics1$NAsort <- ifelse((abs(spgrad_climetrics1$NS)+abs(spgrad_climetrics1$WE)) == 0, NA, 1)
spgrad_climetrics1$Grad <- spgrad_climetrics1$NAsort * sqrt((spgrad_climetrics1$WE^2) + (spgrad_climetri

# Angles
angle_climetrics1 <- spgrad_climetrics1$angle


### Right way
# Trend
trend_climetrics <- climetrics:::.tempgradTerra(tmean)
# Spatial gradient
spgrad_climetrics2 <- climetrics:::.spatialgradTerra(mean(tmean))

spgrad_climetrics2$NS[is.na(spgrad_climetrics2$NS)] <- 0
spgrad_climetrics2$WE[is.na(spgrad_climetrics2$WE)] <- 0
spgrad_climetrics2$NAsort <- ifelse((abs(spgrad_climetrics2$NS)+abs(spgrad_climetrics2$WE)) == 0, NA, 1)
spgrad_climetrics2$Grad <- spgrad_climetrics2$NAsort * sqrt((spgrad_climetrics2$WE^2) + (spgrad_climetri

# Angles
angle_climetrics2 <- spgrad_climetrics2$angle

# Velocity
v_climetrics2 <- climetrics:::.calcvelocity(grad = spgrad_climetrics2,
                                            slope = trend_climetrics,
                                            .terra = TRUE)
```

vocc

Note that the same issue described above happens here. Thus, I calculate velocity with this package using the both the default (wrong) and the correct procedure (using mean raster).

```
# Trend
start <- Sys.time()

trend_vocc = vocc::calcslope(rx = stack(tmean),
                             divisor = 1) # to get C/year

### Wrong way
# Spatial gradient
spgrad_out_vocc1 = vocc::spatialgrad(rx = stack(tmean),
                                     y_dist = res(tmean),
                                     y_diff = NA)
# Angle
angle_vocc1 = spgrad_out_vocc1$angle

# Calculating temperature velocity
v_vocc1 = vocc::calcvelocity(grad = spgrad_out_vocc1,
                             slope = trend_vocc)
v_vocc_rast1 <- rast(tmean[[1]])
v_vocc_rast1[] <- v_vocc1$velocity

vocc_time <- Sys.time() - start

### Right way
# spatial gradient
spgrad_out_vocc2 = vocc::spatialgrad(rx = stack(mean(tmean)),
                                     y_dist = res(tmean),
                                     y_diff = NA)
# Angle
angle_vocc2 = spgrad_out_vocc2$angle

# Calculating temperature velocity
v_vocc2 = vocc::calcvelocity(grad = spgrad_out_vocc2,
                             slope = trend_vocc)
v_vocc_rast2 <- rast(tmean[[2]])
v_vocc_rast2[] <- v_vocc2$velocity
```

VoCC

```
start <- Sys.time()

# calculate the trend
trend_VoCC = VoCC1::tempTrend(r = stack(tmean),
                              th = 0.25*nlyr(tmean) ## set minimum # obs. to 1/4 time series length
)

# calculate the spatial gradient
spgrad_VoCC = VoCC1::spatGrad(r = stack(tmean),
```

```
                               projected = TRUE)

# Calculating temperature velocity
v_VoCC = VoCC1::gVoCC(tempTrend = trend_VoCC,
                      spatGrad = spgrad_VoCC)

# Angle
angle_VoCC = spgrad_VoCC$Ang[]

VoCC_time <- Sys.time() - start
```

bioshifts v3

Functions adapted from climetrics (because it is based on the package `terra`, which allows faster computation relative to other packages that are based on `raster`), appling modifications to generate results in km/year.

```
start <- Sys.time()

# Trend
trend_biov3 <- temp_grad(tmean, th = 0.25*nlyr(tmean))

# Spatial gradient
spgrad_biov3 <- spatial_grad(tmean)

# Velocity
v_biov3 <- gVelocity(spgrad_biov3,trend_biov3,truncate = FALSE)
v_biov3 <- v_biov3$GradVel

# Angle
angle_biov3 = spgrad_biov3$angle

biov3_time <- Sys.time() - start
```

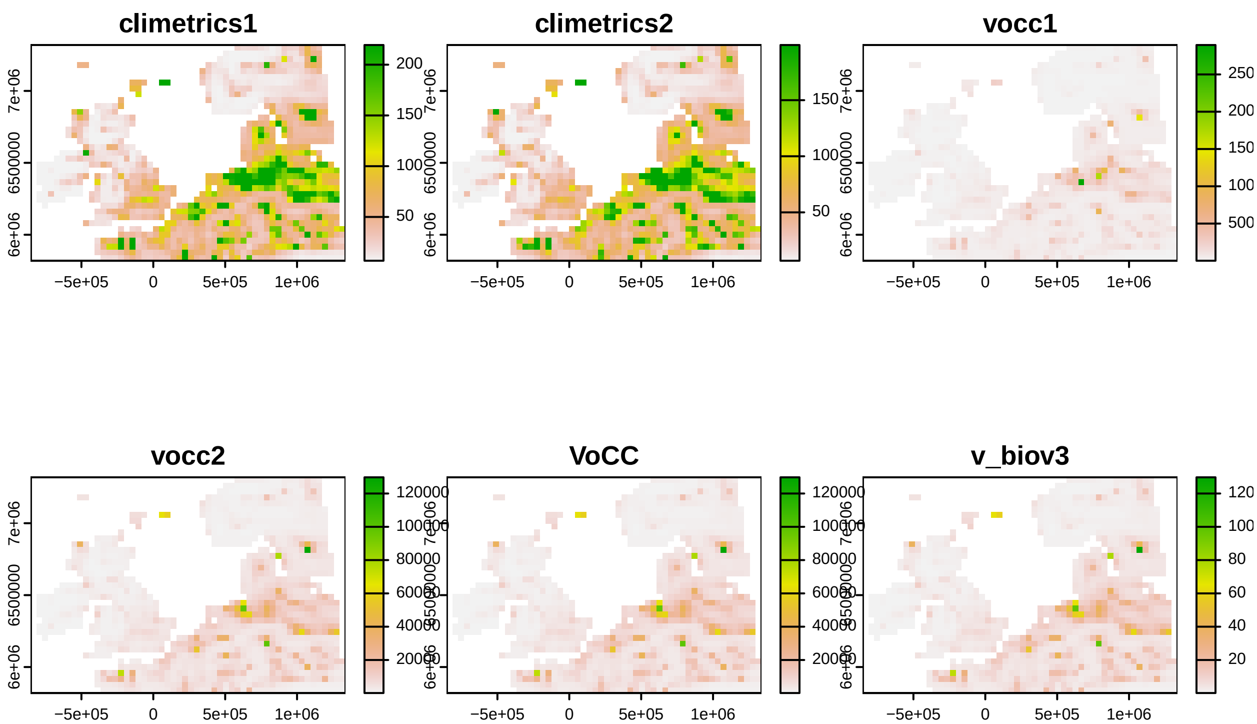
bioshifts v1

```
# spatial gradient
spgrad_biov1 <- Gael_grad(stack(tmean))
# trend (use VoCC trend function)
trend_biov1 <- VoCC1::tempTrend(r = stack(tmean), th = 10)
trend_biov1 = resample(trend_biov1, spgrad_biov1)

v_biov1 = trend_biov1[[1]] / spgrad_biov1
v_biov1 = resample(v_biov1, stack(tmean))
```

Compare results

Velocities maps

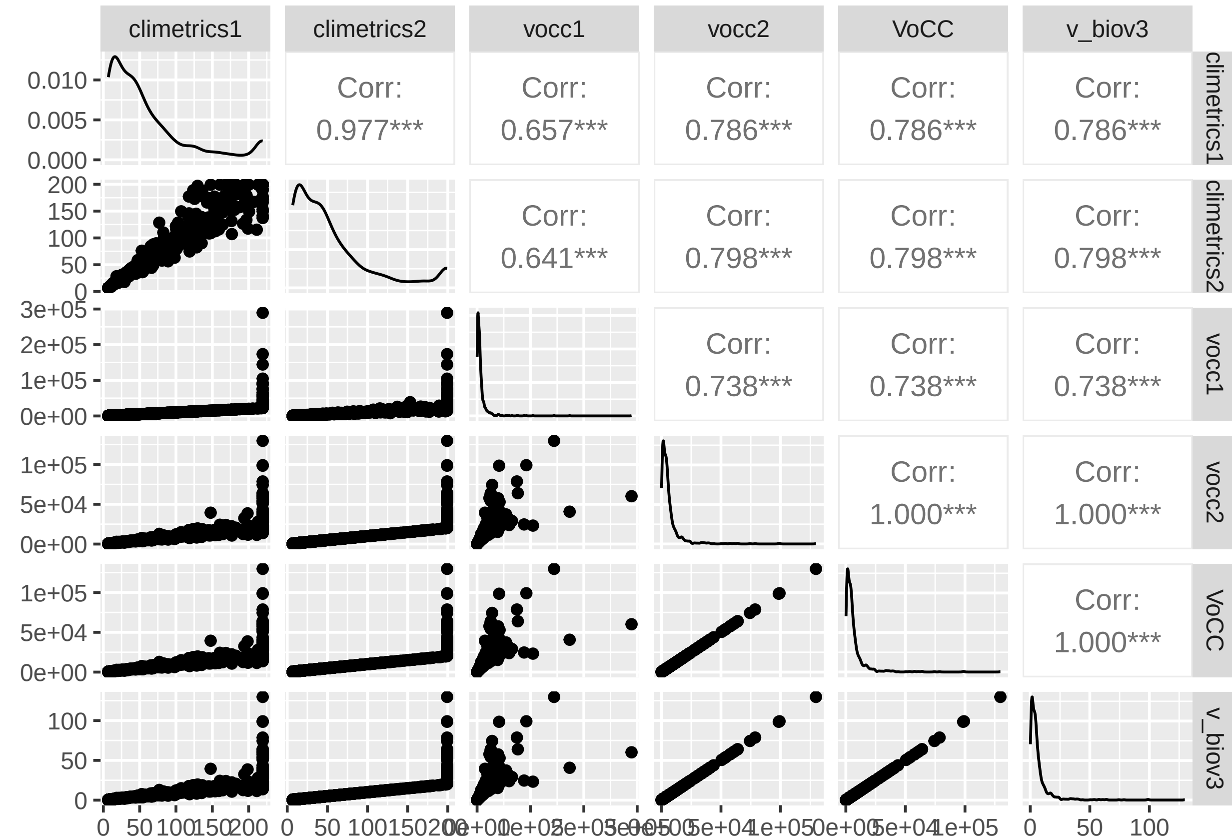
```r
tmp <- rast(c(climetrics1=v_climetrics1,
              climetrics2=v_climetrics2,
              vocc1=v_vocc_rast1,
              vocc2=v_vocc_rast2,
              VoCC=rast(v_VoCC[[1]]),
              v_biov3=v_biov3))

plot(tmp)
```



Quite different values.

Velocities correlations

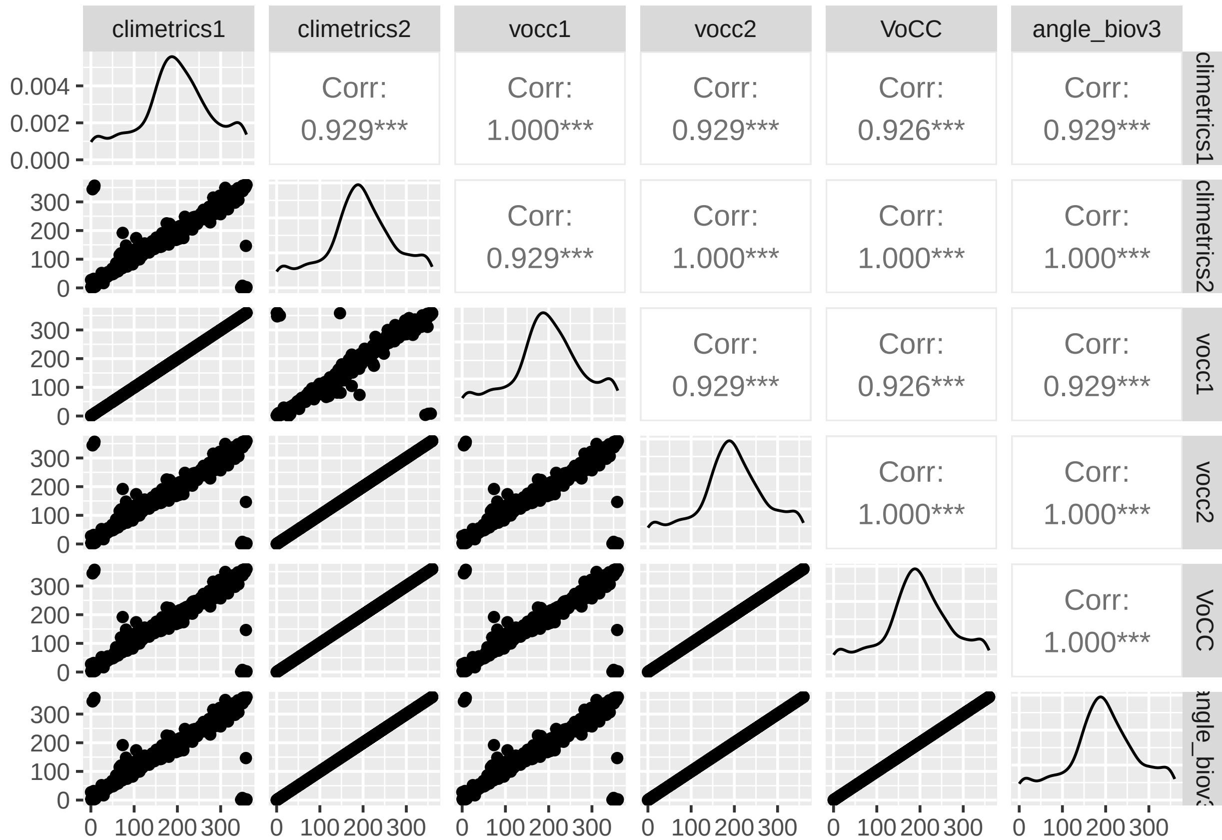
```r
velocities <- data.frame(tmp)

ggpairs(velocities)
```



Velocity values for `climetrics1` and `vocc1` are identical (but not perfectly because `climetrics1` remove outliers by default). `climetrics1` and `vocc1` are different from VoCC because they handle the spatial gradient differently (`climetrics1` and `vocc1` use the first year of the time series to generate the spatial gradient). When we correct the functions from `climetrics` and `vocc` (`climetrics2` and `vocc2`, respectively) to use mean values across the time series, velocity values become identical to the ones in `VoCC`. If using projected rasters to a equal-area, values from `bioshifts v3` are in km/year while in `VoCC` they are presented in the scale of the raster (m/year).

Angle correlations

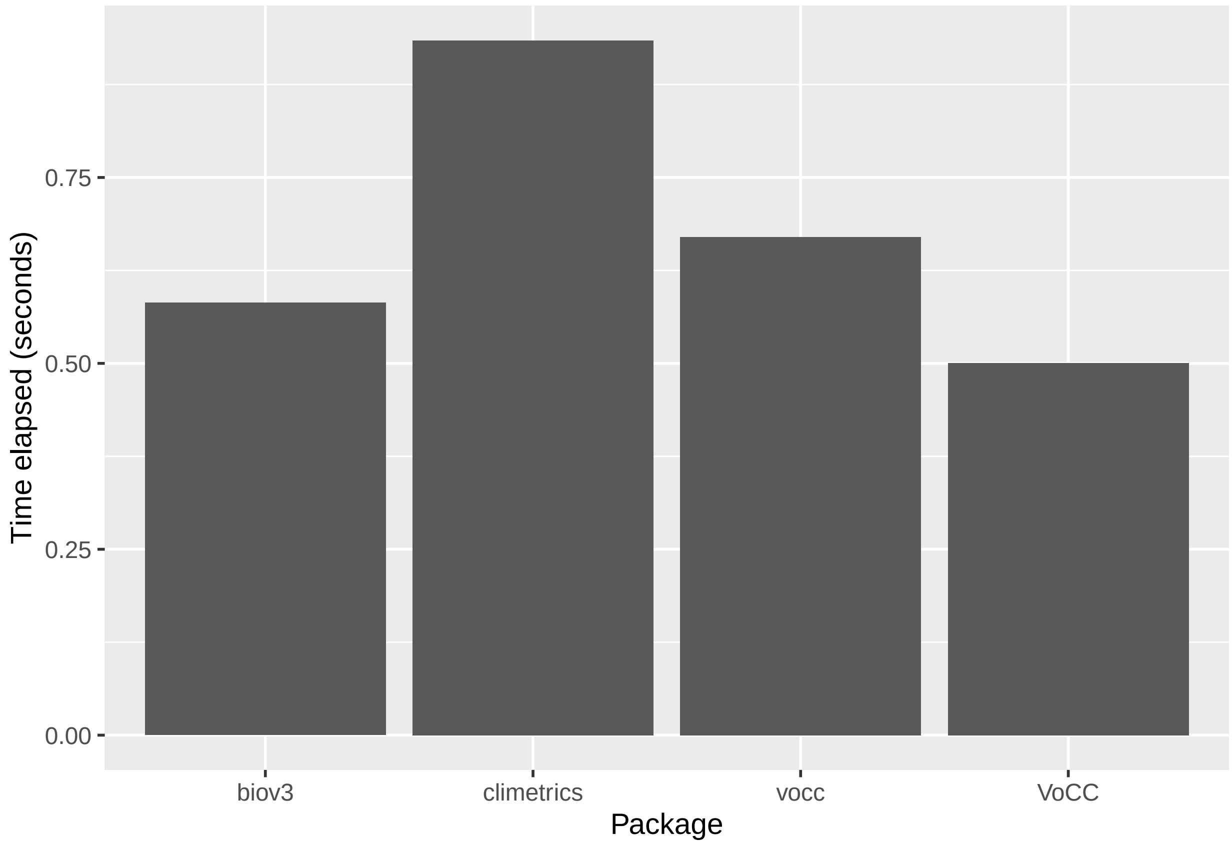
```
angles <- data.frame(climetrics1=angle_climetrics1,
                     climetrics2=angle_climetrics2,
                     vocc1=angle_vocc1,
                     vocc2=angle_vocc2,
                     VoCC=angle_VoCC,
                     angle_biov3=angle_biov3)

ggpairs(angles)
```



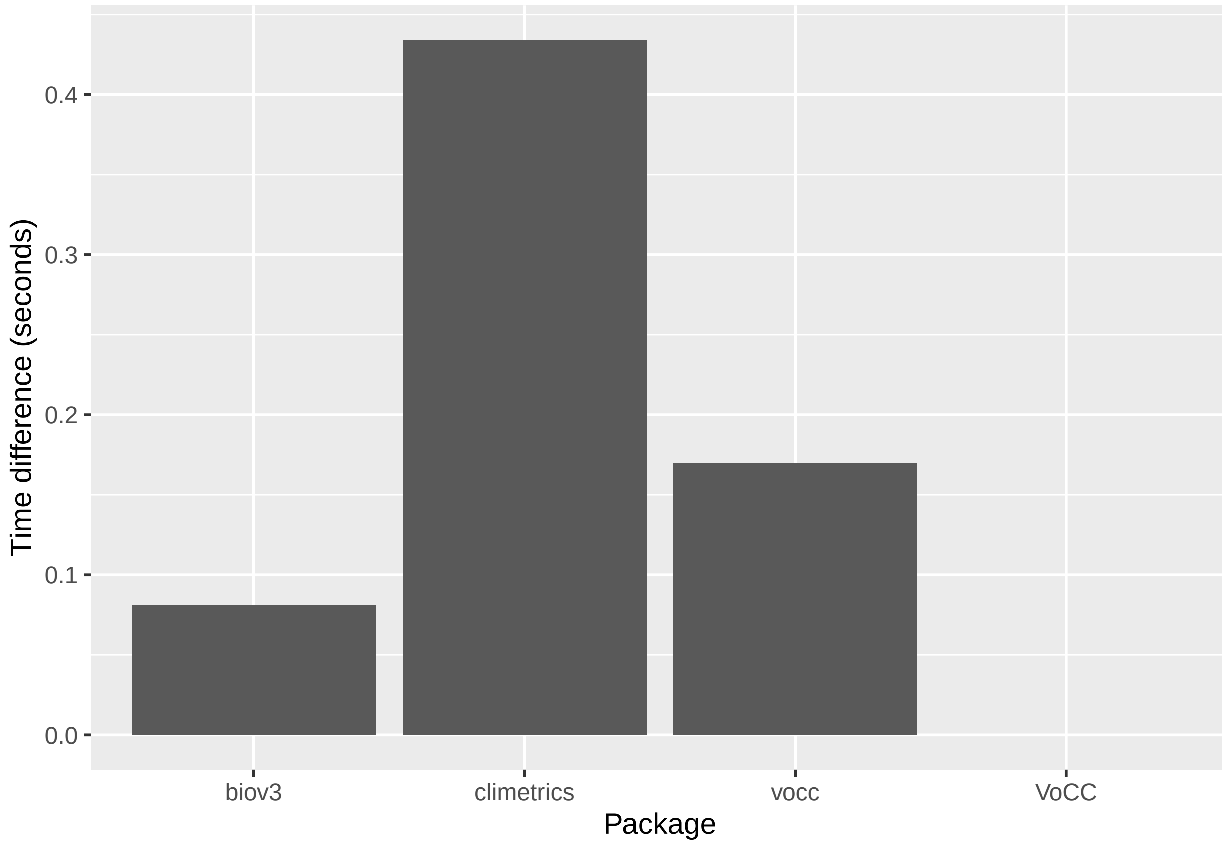
Angles are very similar across packages and they all peak at around 200 degrees, meaning mainly South direction velocities. P.S.: these are angles of the environmental gradient, which is the opposite of velocity trajectory.

Time for calculating velocities

```
timesFuncs <- data.frame(t(data.frame(climetrics_time[[1]],vocc_time[[1]],VoCC_time[[1]],biov3_time[[1]]
names(timesFuncs) <- "elapsed"
timesFuncs$Package <- gsub("_time..1..","",rownames(timesFuncs))

ggplot(timesFuncs, aes(x = Package, y = elapsed))+
    geom_col() + ylab("Time elapsed (seconds)")
```



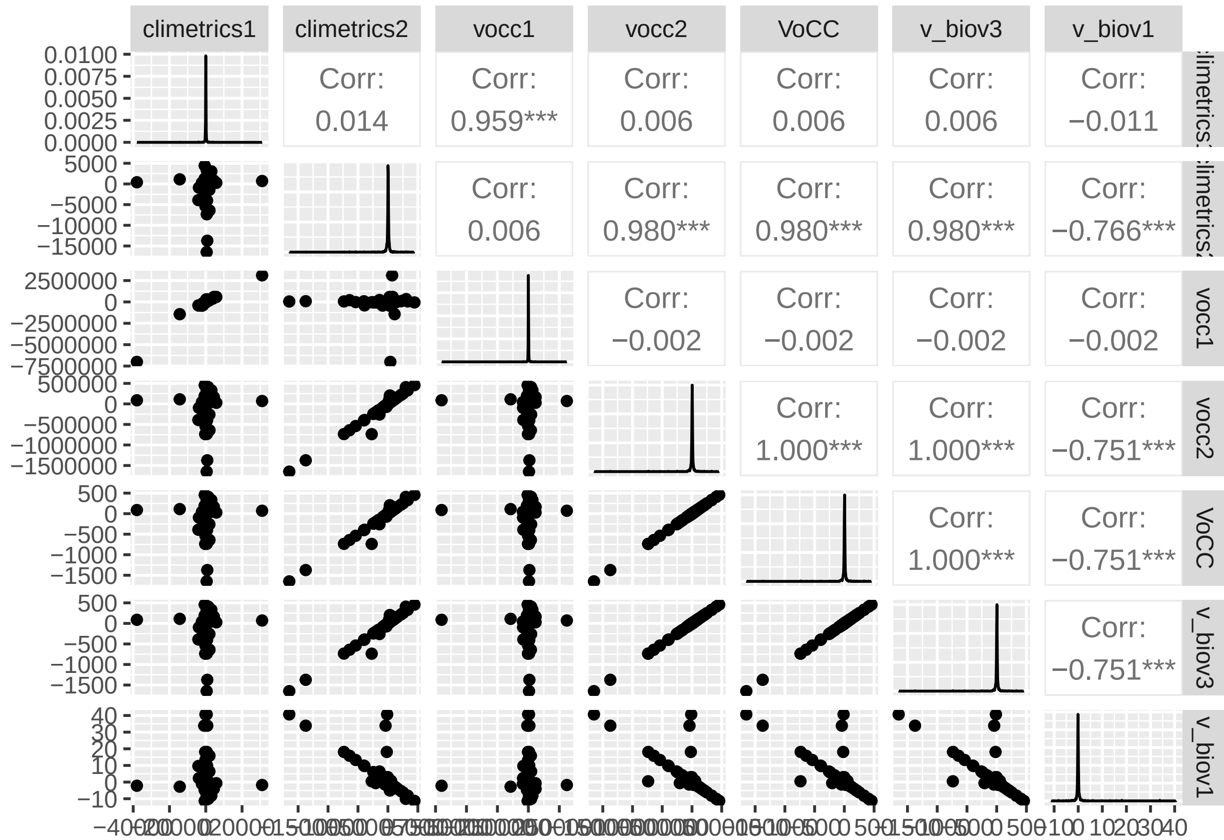

```
ggplot(timesFuncs, aes(x = Package, y = elapsed-min(elapsed)))+
    geom_col() + ylab("Time difference (seconds)")
```



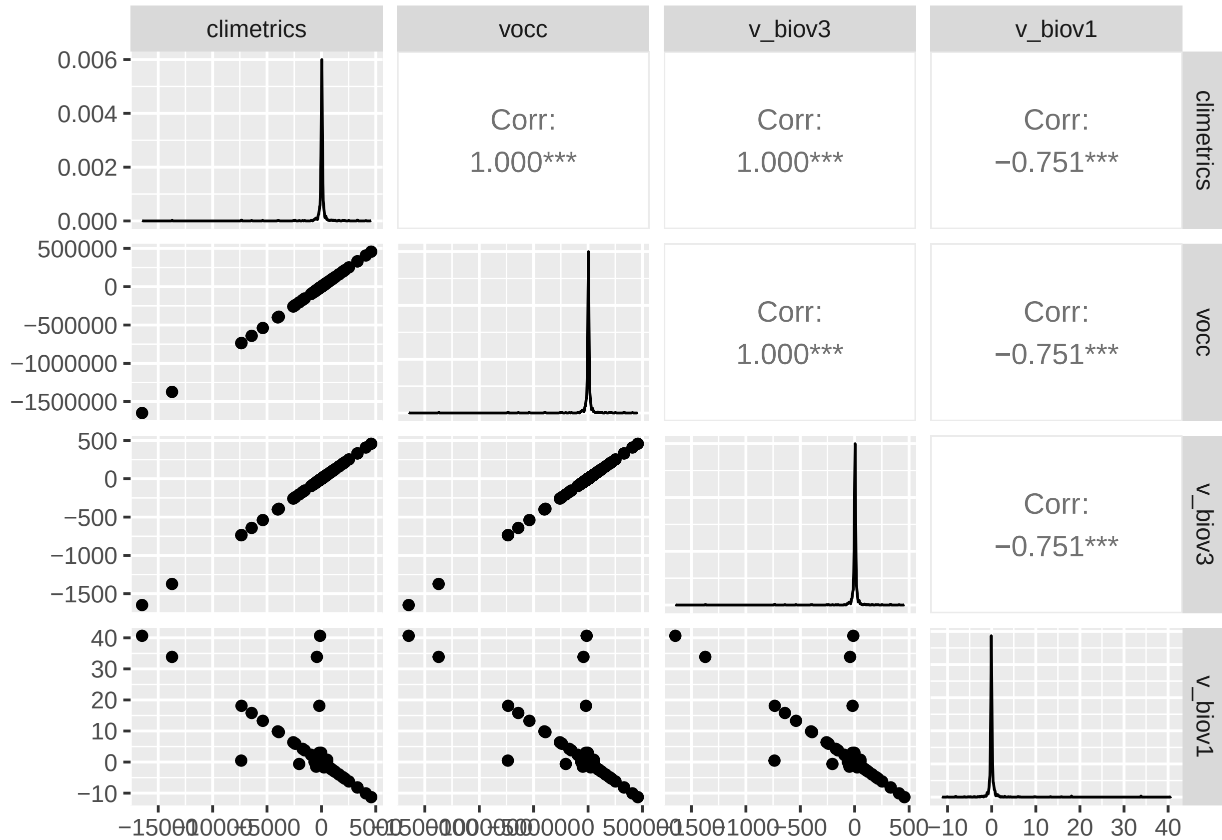
North direction velocities

Here, we can add biov1's results for North velocity

```
# Convert all velocities to the North direction
velocities_North <- data.frame(
    climetrics1=v_climetrics1[][,1] / cos(deg_to_rad((angle_climetrics1 + 180) %% 360)),
    climetrics2=v_climetrics2[][,1] / cos(deg_to_rad((angle_climetrics2 + 180) %% 360)),
    vocc1=v_vocc_rast1[][,1] / cos(deg_to_rad((angle_vocc1 + 180) %% 360)),
    vocc2=v_vocc_rast2[][,1] / cos(deg_to_rad((angle_vocc2 + 180) %% 360)),
    VoCC=v_VoCC$voccMag[]/1000 / cos(deg_to_rad((angle_VoCC + 180) %% 360)),
    v_biov3=v_biov3[][,1] / cos(deg_to_rad((angle_biov3 + 180) %% 360)),
    v_biov1=v_biov1[])

ggpairs(na.omit(velocities_North))
```

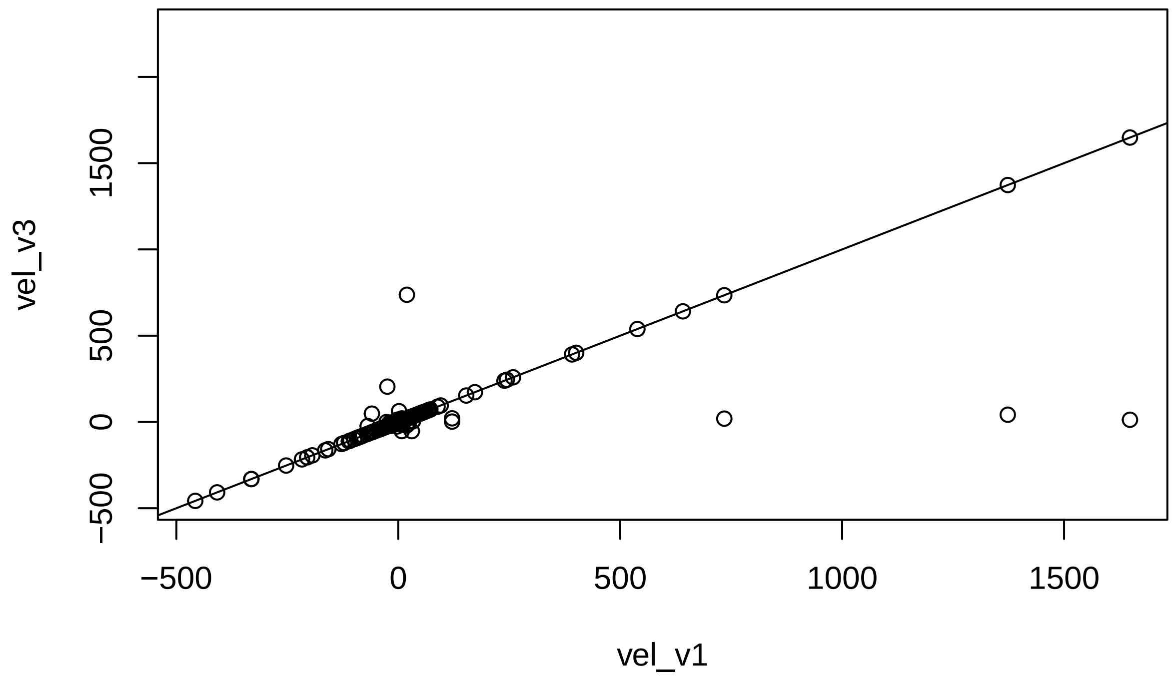



```r
# same as doing
velocities_North <- data.frame(
    climetrics=trend_climetrics[][,1] / spgrad_climetrics2$NS,
    vocc=trend_vocc[][,1] / spgrad_out_vocc2$NS,
    v_biov3=trend_biov3[][,1] / spgrad_biov3$NS,
    v_biov1=v_biov1[])

ind <- cells(v_climetrics1 > 0)

velocities_North$climetrics[ind] <- velocities_North$climetrics[ind] * -1
velocities_North$vocc[ind] <- velocities_North$vocc[ind] * -1
velocities_North$v_biov3[ind] <- velocities_North$v_biov3[ind] * -1

ggpairs(na.omit(velocities_North))
```



VoCC and bioshifts still identical. Velocities from `biov1` and `biov3` are very similar, but the magnitudes are different. This is because v1 provides the gradient in C/pixel. After fixing the scale of biov1 gradient by adding raster resolution we get similar scales relative to biov3.

```
test <- data.frame(vel_v3 = trend_biov3[][,1] / spgrad_biov3$NS,
                   vel_v1 = (v_biov1[] * res(tmean)[1])/1000)

plot(vel_v3~vel_v1, test)
abline(a=0, b = 1)
```

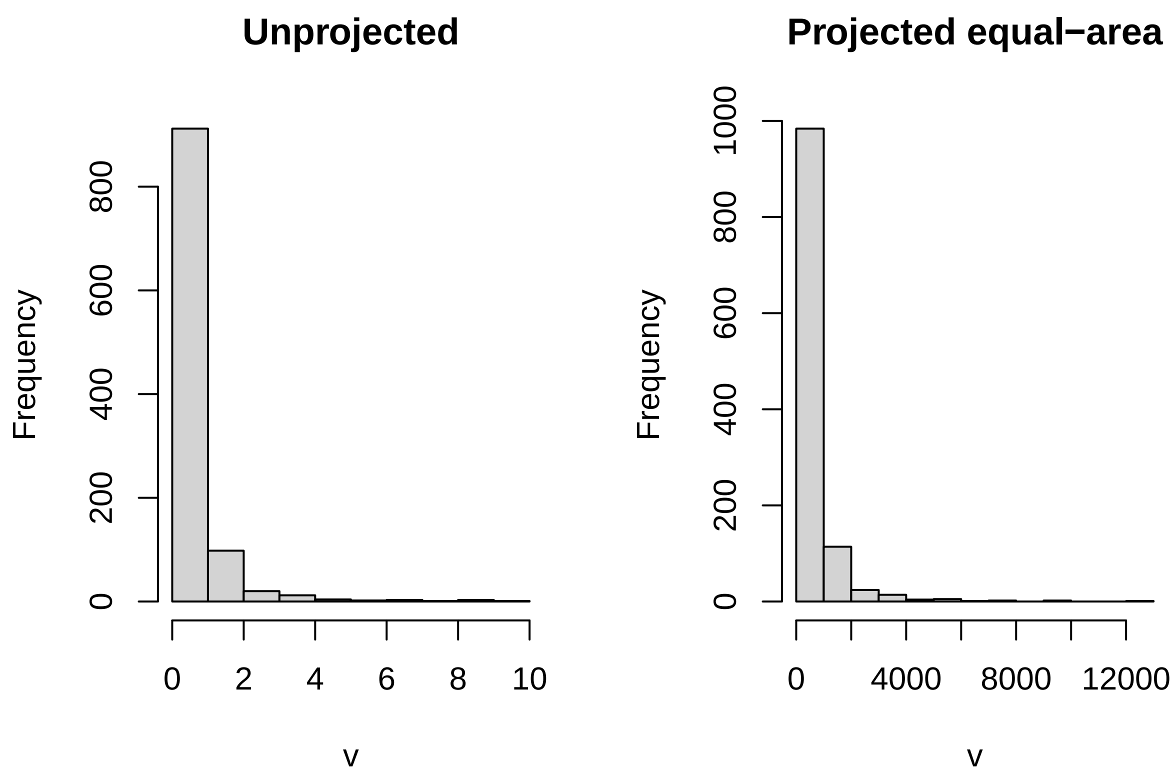


Extra

Compare velocities with projected and unprojected raster files

```
# Use the same set of climate variables
tmean <- rast(paste0(filePath,'/tmean.tif'))
# project to equal-area
tmean_proj <- terra::project(tmean,Eckt)

# Use rts function in the rts package to make a raster time series:
tmean.t <- rts(tmean,n)
tmean.t_proj <- rts(tmean_proj,n)


####
# VoCC unprojected
# Trend
trend_VoCC = VoCC1::tempTrend(r = stack(tmean),
                              th = 0.25*nlyr(tmean))

# Spatial gradient
spgrad_VoCC = VoCC1::spatGrad(r = stack(tmean),
                              projected = FALSE)
```

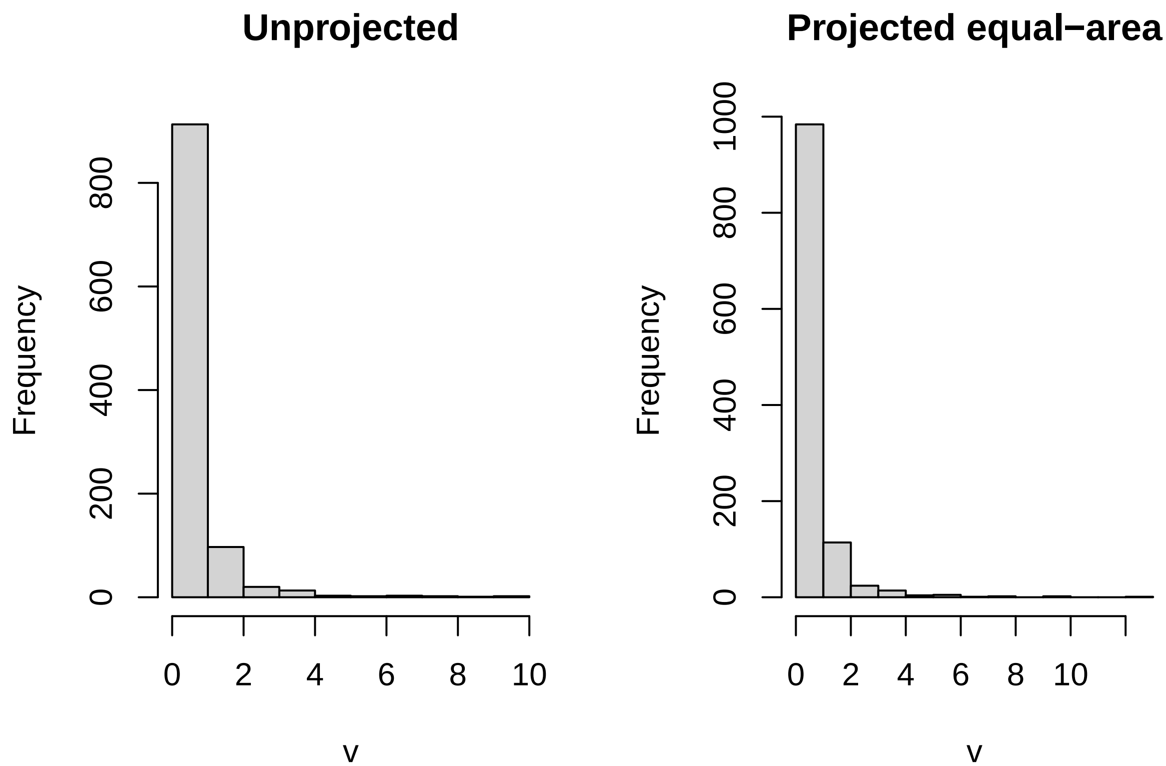
```
# Velocity
v_VoCC = VoCC1::gVoCC(tempTrend = trend_VoCC,
                      spatGrad = spgrad_VoCC)

####
# VoCC projected
# Trend
trend_VoCC = VoCC1::tempTrend(r = stack(tmean_proj),
                              th = 0.25*nlyr(tmean_proj) ## set minimum # obs. to 1/4 time series length
)
# Spatial gradient
spgrad_VoCC = VoCC1::spatGrad(r = stack(tmean_proj),
                              projected = TRUE)
# Velocity
v_VoCC2 = VoCC1::gVoCC(tempTrend = trend_VoCC,
                       spatGrad = spgrad_VoCC)

## Compare velocities
{
    par(mfrow=c(1,2))
    hist(v_VoCC[[1]],main="Unprojected")
    hist(v_VoCC2[[1]],main="Projected equal-area")
    par(mfrow=c(1,1))
}
```



```r
####
# bioshifts unprojected
# Trend
trend_biov3_ <- temp_grad(tmean, th = 0.25*nlyr(tmean))
# Spatial gradient
spgrad_biov3_ <- spatial_grad(tmean)
```

```
## Loading required namespace: fields
```

```r
# Velocity
v_biov3_ <- gVelocity(spgrad_biov3_,trend_biov3_)

####
# bioshifts projected
# Trend
trend_biov3_ <- temp_grad(tmean_proj, th = 0.25*nlyr(tmean_proj))
# Spatial gradient
spgrad_biov3_ <- spatial_grad(tmean_proj)
# Velocity
v_biov32 <- gVelocity(spgrad_biov3_,trend_biov3_)

## Compare velocities
{
    par(mfrow=c(1,2))
    hist(v_biov3_[[1]],main="Unprojected")
    hist(v_biov32[[1]],main="Projected equal-area")
    par(mfrow=c(1,1))
}
```



Why VoCC results are different when using projected and unprojected data?

Inspection of VoCC package functions indicates that when using projected data, the results are given in the unit of the environmental data provided. As the unit of projected data is usually in meters, the results of the unprojected data are 1000 smaller then the projected (1km = 1000 meters). To make results identical between projected and unprojected data (both in C/km), transform units:

```r
summary(v_VoCC[[1]])
```

```
##              voccMag
## Min.      0.02282359
## 1st Qu.   0.16517678
## Median    0.34051253
## 3rd Qu.   0.63029609
## Max.      9.90459499
## NA's    726.00000000
```

```r
summary(v_VoCC2[[1]]/1000)
```

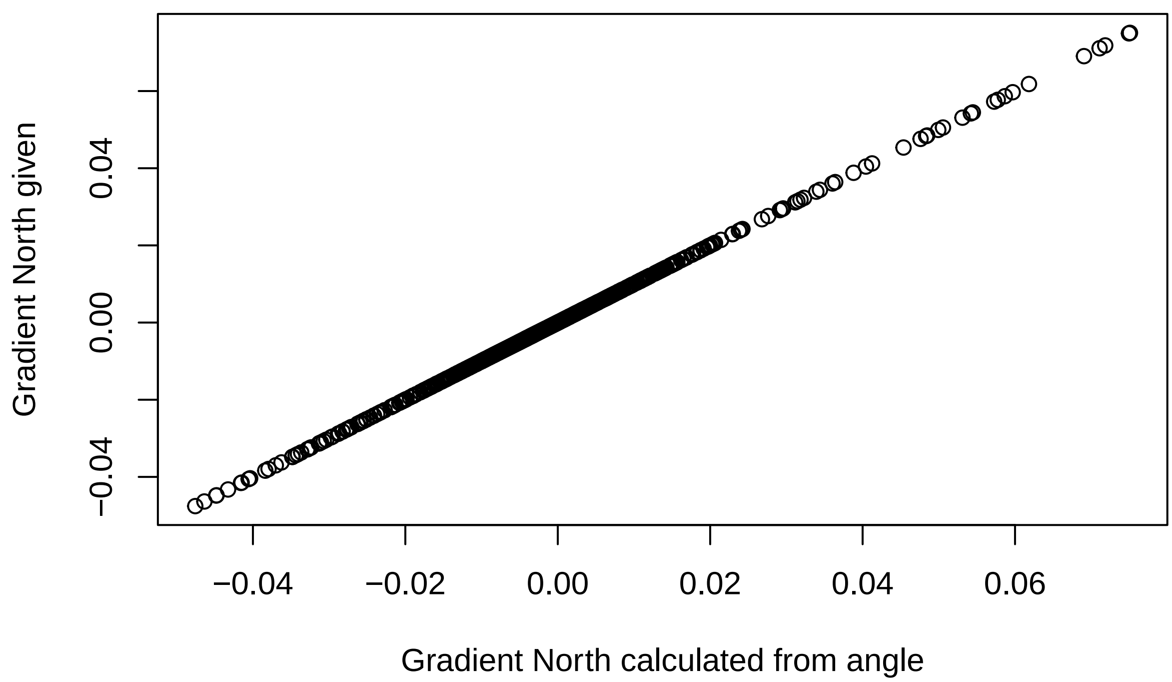
```
##              voccMag
## Min.      0.02477245
## 1st Qu.   0.18575202
## Median    0.39378280
## 3rd Qu.   0.69185021
```

```
## Max.             Inf
## NA's    846.00000000
```

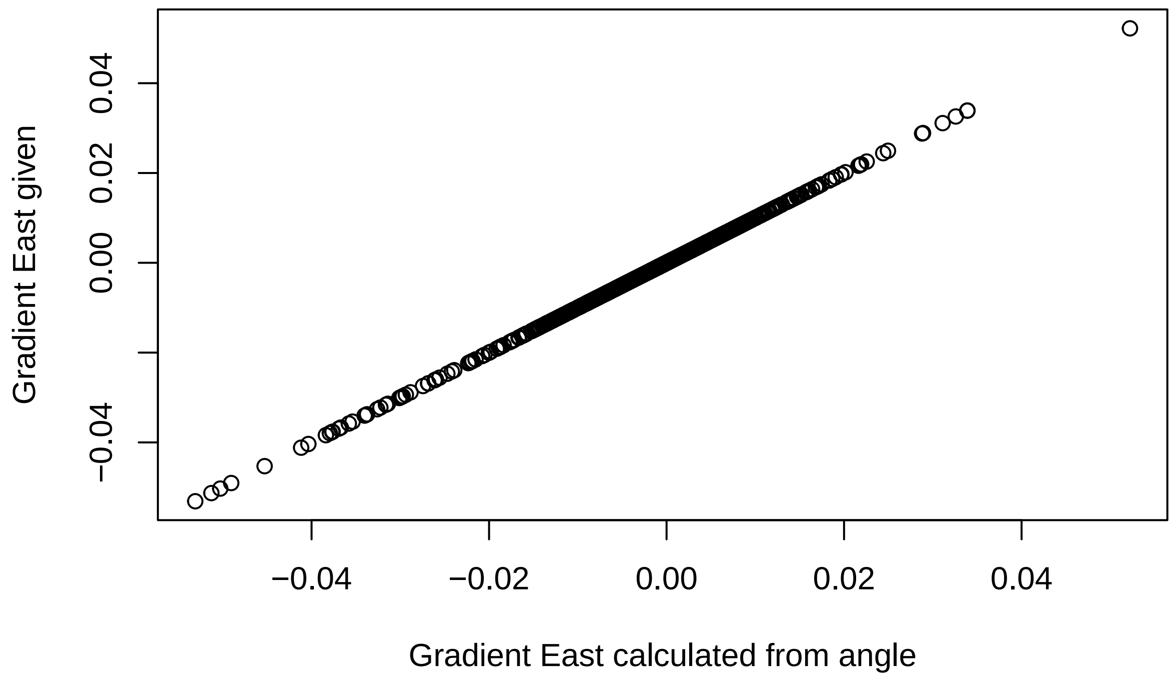
Get gradient North from undirectional gradient

Tests whether we can use velocity North provided directly from the function output, or if it's necessary to calculate the resultant velocity North from the angle of the gradient.

```
# get data
v_data <- na.omit(spgrad_biov3)

# Convert gradients to the North direction using the angle
vel_gradient <- v_data$Grad  # °C/km
vel_angle <- v_data$angle

# Calculate temperature change in North and East directions
gradient_north_calculated <- vel_gradient * cos(deg_to_rad(vel_angle))
gradient_east_calculated <- vel_gradient * sin(deg_to_rad(vel_angle))

plot(gradient_north_calculated,v_data$NS,
     xlab="Gradient North calculated from angle",ylab = "Gradient North given")
```




```
plot(gradient_east_calculated,v_data$WE,
     xlab="Gradient East calculated from angle",ylab = "Gradient East given")
```




```r
all.equal(v_data$NS,gradient_north_calculated)
```

```
## [1] TRUE
```

```r
all.equal(v_data$WE,gradient_east_calculated)
```

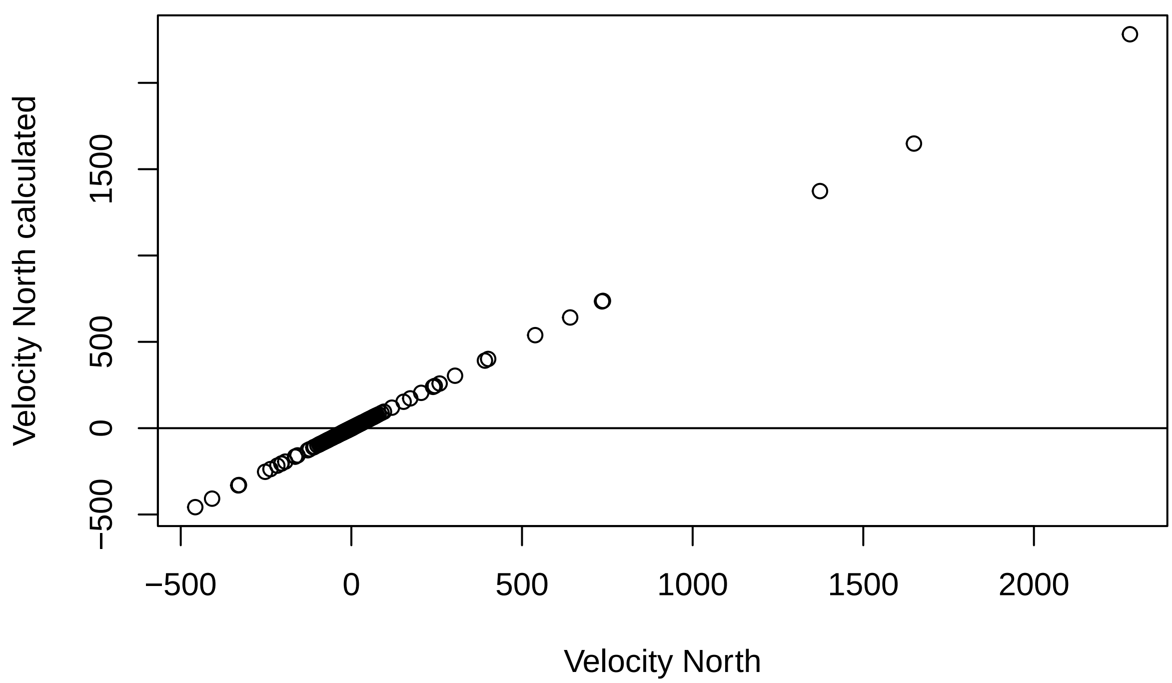
```
## [1] TRUE
```

According to the calculations below, we can use the gradient North directly. It is not necessary to convert velocity rates to the North direction using angles.

Get vel North from unidirectional vel and angle

```r
test <- spgrad_biov3

# add coordinates
test <- cbind(test,xyFromCell(v_biov3, test$icell))

# add vel
test$vel <- v_biov3[spgrad_biov3$icell][,1]

# add trend
```

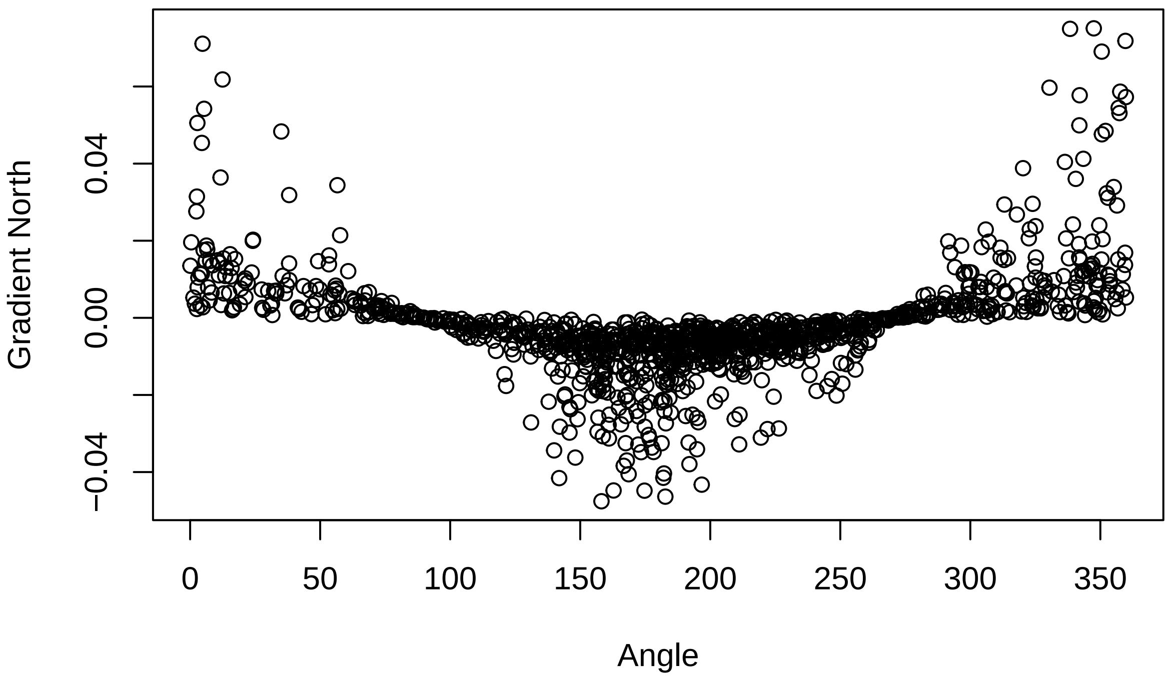
```r
test$trend <- trend_biov3[spgrad_biov3$icell][,1]

# get vel north
test$v_north <- test$trend / test$NS

# convert vel undirection to north based on angle
test$v_north_calc <- test$vel / cos(deg_to_rad(test$angle))

{
    plot(test$v_north,
         test$v_north_calc,
         xlab="Velocity North",
         ylab="Velocity North calculated")
    abline(h=0)
}
```



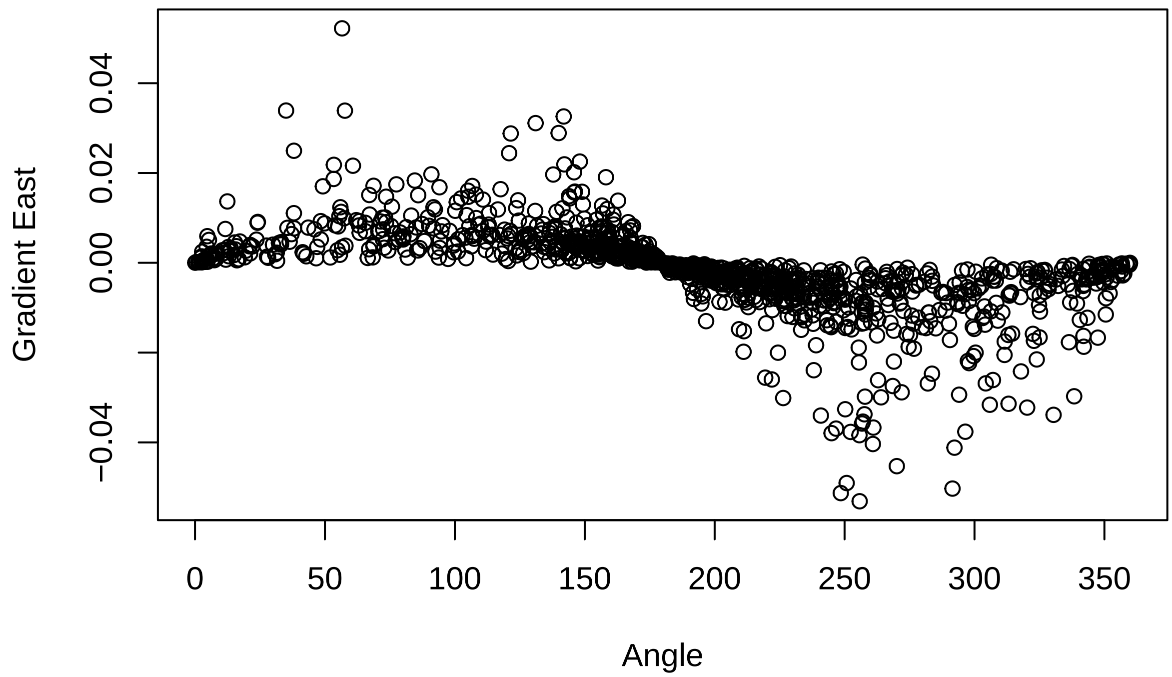
Understanding angles

We expect velocity North to be greater when angle is North, and same direct comparison for South direction.

```r
plot(v_data$angle,v_data$NS,
     xlab="Angle",ylab = "Gradient North")
```




```r
plot(v_data$angle,v_data$WE,
     xlab="Angle",ylab = "Gradient East")
```

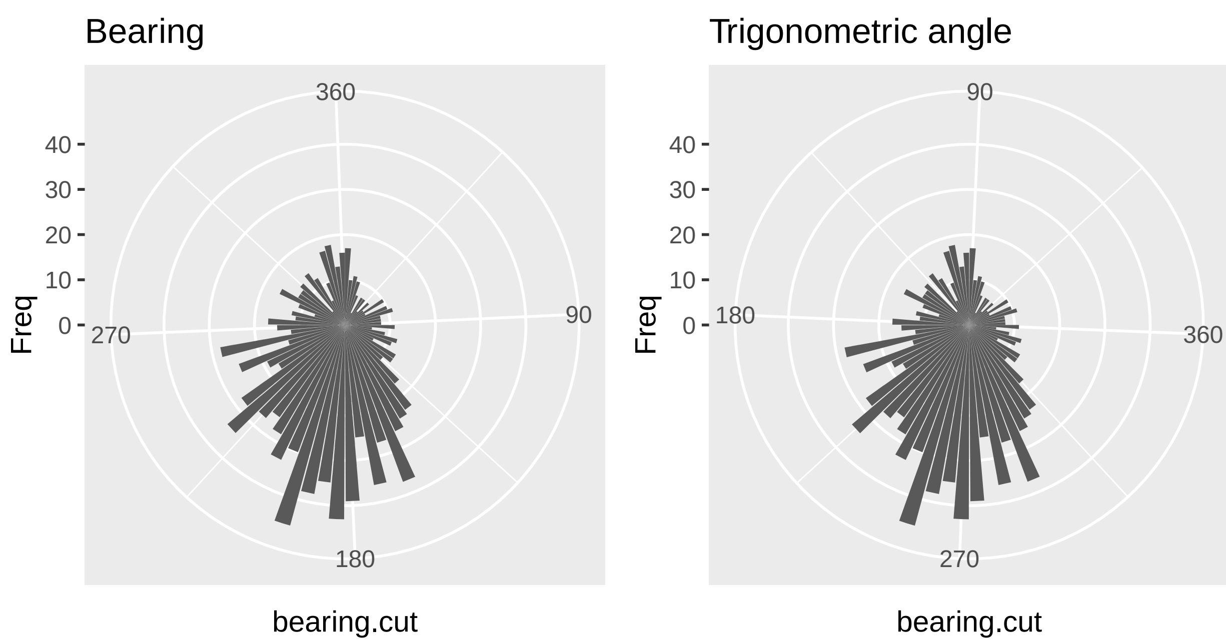


Indeed! Angles 0 and 180 represent North and South, respectively. Angles 90 and 270 represent West and East, respectively.

Visualize angles and bearings

```
# 0º at North and clockwise
bearing <- spgrad_biov3$angle

#create histogram
breaks = seq(0, 360, by=5)     # half-integer sequence
bearing.cut = cut(bearing, breaks, right=FALSE)
bearing.freq = as.data.frame(table(bearing.cut))
bearing.freq$bearing.cut <- seq(5,360, by = 5)

#plot
p1 <- ggplot(bearing.freq, aes(x = bearing.cut, y = Freq)) +
    coord_polar(theta = "x", start =0, direction = 1) +
    geom_bar(stat = "identity") +
    scale_x_continuous(breaks = seq(0, 360, 90))+
    ggtitle("Bearing")

# transform to 0º at E and counterclockwise
bearing2 <- bearing_to_angle(bearing)
```

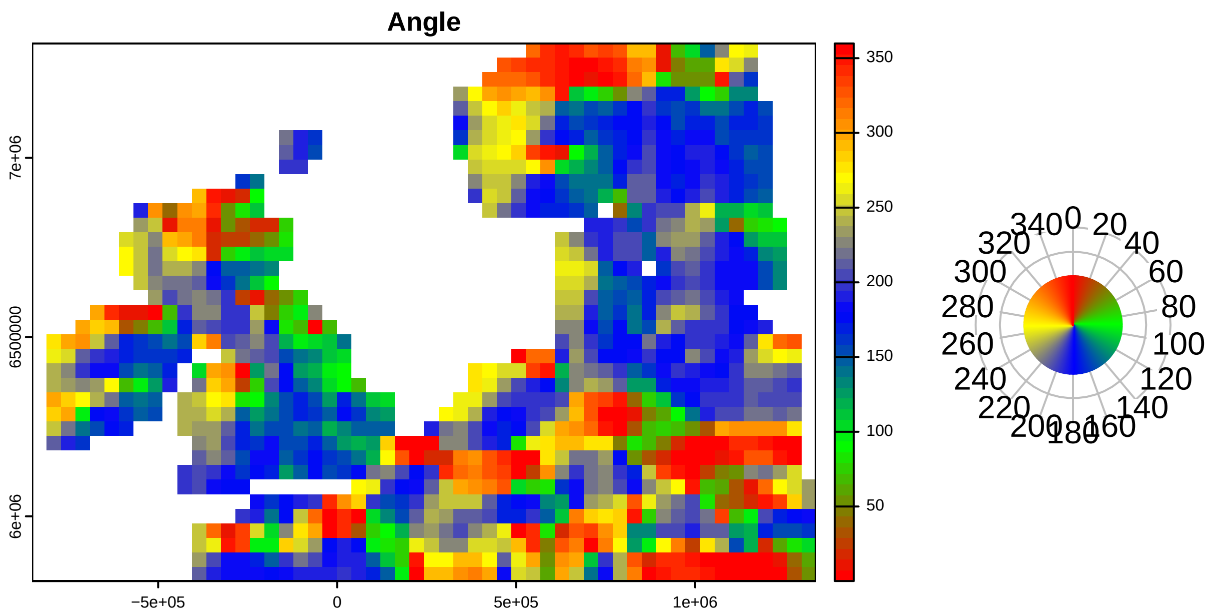
```r
#create histogram
bearing.cut2 = cut(bearing2, breaks, right=FALSE)
bearing.freq2 = as.data.frame(table(bearing.cut2))
bearing.freq2$bearing.cut <- seq(5,360, by = 5)

# plot
p2 <- ggplot(bearing.freq2, aes(x = bearing.cut, y = Freq)) +
    coord_polar(theta = "x", start = -pi/2, direction = -1) +
    geom_bar(stat = "identity") +
    scale_x_continuous(breaks = seq(0, 360, 90))+
    ggtitle("Trigonometric angle")

grid.arrange(p1, p2, ncol=2)
```



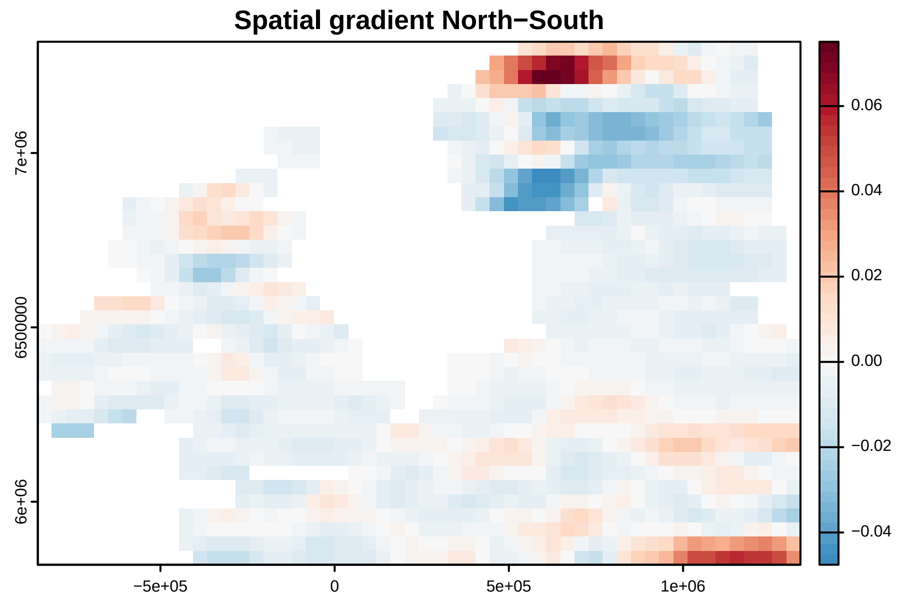

```r
############################################
# Map angle
angle_rast <- v_biov3
angle_rast[spgrad_biov3$icell] <- spgrad_biov3$angle


angle_map(angle_rast, main = "Angle")
```

```
## Warning in cos(radial.pos[i, ]) * lengths[i, ]: longer object length is not a
## multiple of shorter object length
```

```
## Warning in sin(radial.pos[i, ]) * lengths[i, ]: longer object length is not a
## multiple of shorter object length
```




```r
# Map gradient NS
grad_rast <- v_biov3
grad_rast[spgrad_biov3$icell] <- spgrad_biov3$NS

velocity_map(grad_rast, main = "Spatial gradient North-South")
```



Most of the velocities in this area are in the South direction