Calculate landscape parameters from Ridgefield geoTIFF DEM

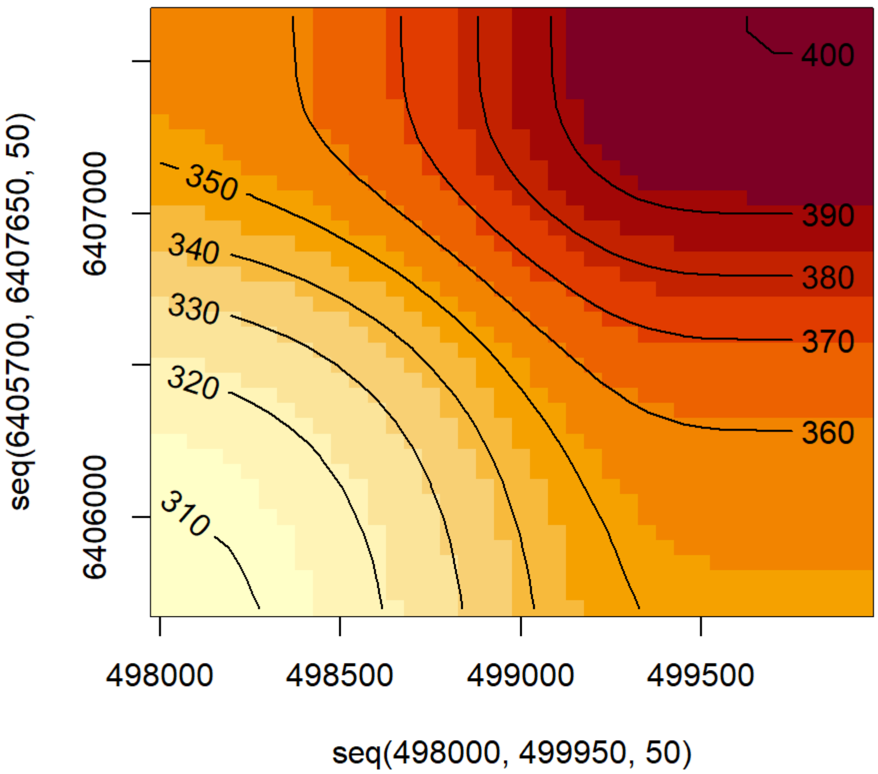
Note: not all the code is shown.

First load packages and some other setup stuff

```
library(sf)
library(raster) # slope parameters from raster::terrain() function
library(rgdal)
library(ggplot2)
library(ggpubr)
library(viridis)
library(fgeo.analyze)
GDA94 <- CRS("+proj=geocent +ellps=GRS80 +units=m +no_defs +type=crs")
```

Create matrix of test elevations in Excel using summed exponential functions & import into R data frame `testElev`, check with plot:

```
# testElev <- read.table("clipboard", sep="\t", header=TRUE) # no accidental run
image(seq(498000,499950,50), seq(6405700,6407650,50),
      as.matrix(testElev[,2:41]))
contour(seq(498000,499950,50), seq(6405700,6407650,50),
        as.matrix(testElev[,2:41]), add=TRUE, labcex=1.)
```



Make into a 3-column dataframe (x,y,elev)

```
elevDF <- data.frame(x=rep(seq(498000,499950,50),40),
                     y=rep(seq(6405700,6407650,50),each=40),
                     elev=c(as.matrix(testElev[,2:41])))
summary(elevDF)
##        x                y                elev
##  Min.   :498000   Min.   :6405700   Min.   :306.0
##  1st Qu.:498488   1st Qu.:6406188   1st Qu.:339.9
##  Median :498975   Median :6406675   Median :356.2
##  Mean   :498975   Mean   :6406675   Mean   :357.2
##  3rd Qu.:499463   3rd Qu.:6407162   3rd Qu.:375.6
##  Max.   :499950   Max.   :6407650   Max.   :400.0
```

Make RasterLayer from this 3-column dataframe, and add a CRS

```
testRast <- raster::rasterFromXYZ(elevDF)
raster::crs(testRast) <- "+proj=utm +zone=50 +south"
```

Then calculate slope using `raster::terrain()`:

```
test_slope <- raster::terrain(testRast, opt = "slope", unit = "degrees")
```

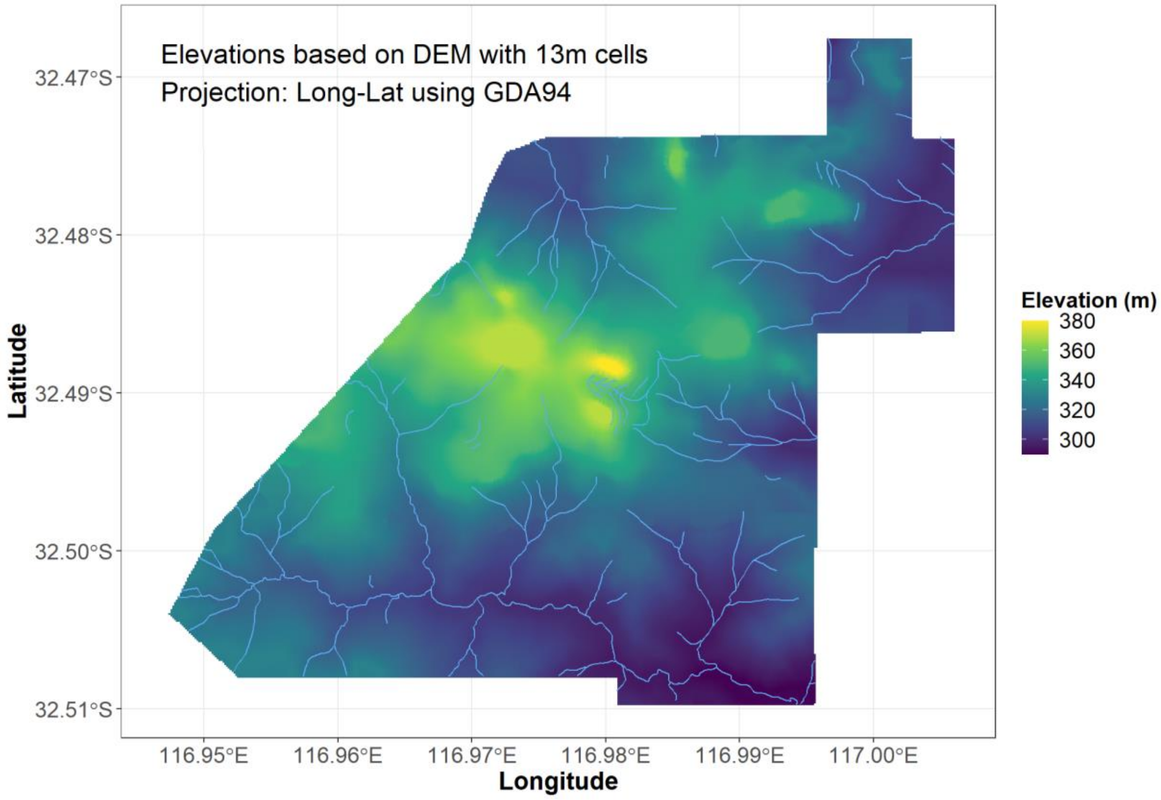
This section based on code at *https://datacarpentry.org/r-raster-vector-geospatial/01-raster-structure/*

Read Ridgefield geoTIFF into R rasterLayer

```
## GDALinfo("Ridgefield_DEM_gda94_utm50_epsg28350.tif") # just file info
RF_elev <- raster("Ridgefield_DEM_gda94_utm50_epsg28350.tif")
  # also make a raster subset with no NA
  NWrect <- extent(498000,500500,6405700,6407000)
  NW_elev <- raster::crop(RF_elev,NWrect)
  RF_elevDF <- as.data.frame(RF_elev, xy=TRUE)
  colnames(RF_elevDF)[3] <- "elev"
```

plot the data frame made from elevation raster

```
elevMapGG <- ggplot() +
  geom_raster(data = RF_elevDF , aes(x = x, y = y, fill = elev)) +
  geom_sf(data = rf_hydrology, col="steelblue2") + # add the streams
  scale_fill_viridis_c(na.value = "#FFFFFF00", name="Elevation (m)") +
  labs(x = "Longitude", y="Latitude") +
  coord_sf(crs = st_crs(28350)) +
  annotate(geom="text", x=495000, y=6407500,
           label=paste0("Elevations based on DEM with 13m cells",
                        "\nProjection: Long-Lat using GDA94"),
           size=6, hjust = 0) +
  theme_bw() +
  theme(axis.title = element_text(size=16, face="bold"),
        axis.text = element_text(size=14),
        legend.title = element_text(size=14, face="bold"),
        legend.text = element_text(size=14))
elevMapGG
```



For the `terrain()` function see *https://arc2r.github.io/book/Surface.html*

Working with slope

Extract slope from elevation data

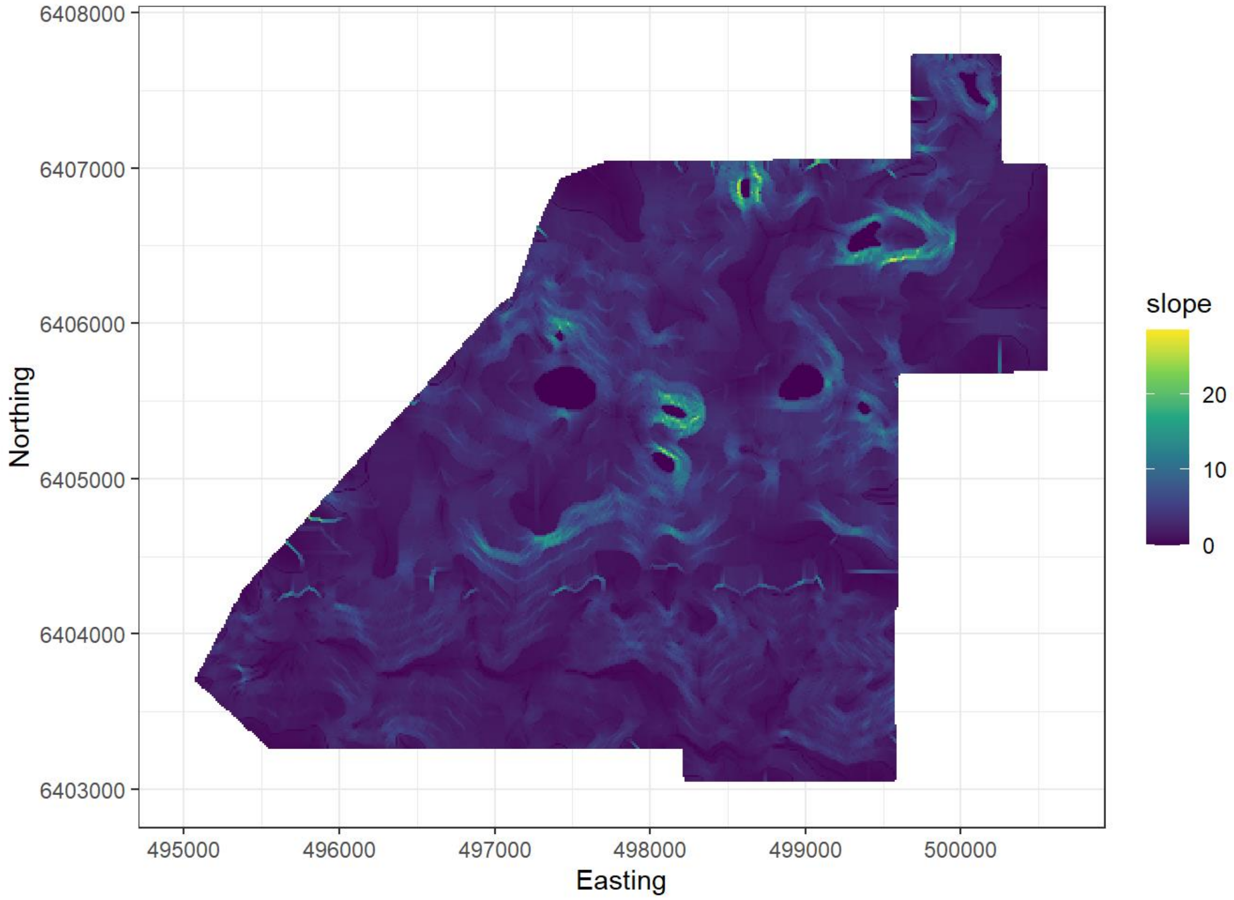
```
RF_slope <- terrain(RF_elev, opt = "slope", unit = "degrees", neighbors=4)
RF_slope@crs@projargs ; bbox(RF_slope) ; summary(RF_slope)
## [1] "+proj=utm +zone=50 +south +ellps=GRS80 +units=m +no_defs"
##         min     max
## s1  495053  500578
## s2 6403059 6407739
##                slope
## Min.        0.000000
## 1st Qu.     1.631004
## Median      2.455780
## 3rd Qu.     3.472014
## Max.       28.543020
## NA's    66074.000000
```

Make a data frame version of slope

```
RF_slopeDF <- as.data.frame(RF_slope_MGA, xy=TRUE)
summary(RF_slopeDF)
##        x                y                slope
##  Min.   :494995   Min.   :6403000   Min.   : 0.00
##  1st Qu.:496399   1st Qu.:6404196   1st Qu.: 1.63
##  Median :497816   Median :6405399   Median : 2.45
##  Mean   :497816   Mean   :6405399   Mean   : 2.92
##  3rd Qu.:499233   3rd Qu.:6406602   3rd Qu.: 3.47
##  Max.   :500637   Max.   :6407798   Max.   :28.54
##                                     NA's   :73370
```

Plot the raw slope data

```
ggplot() +
  geom_raster(data = RF_slopeDF , aes(x = x, y = y, fill = slope)) +
  scale_fill_viridis_c(na.value = "#00000000") +
  labs(x = "Easting", y="Northing") +
  coord_fixed() +
  theme_bw()
```

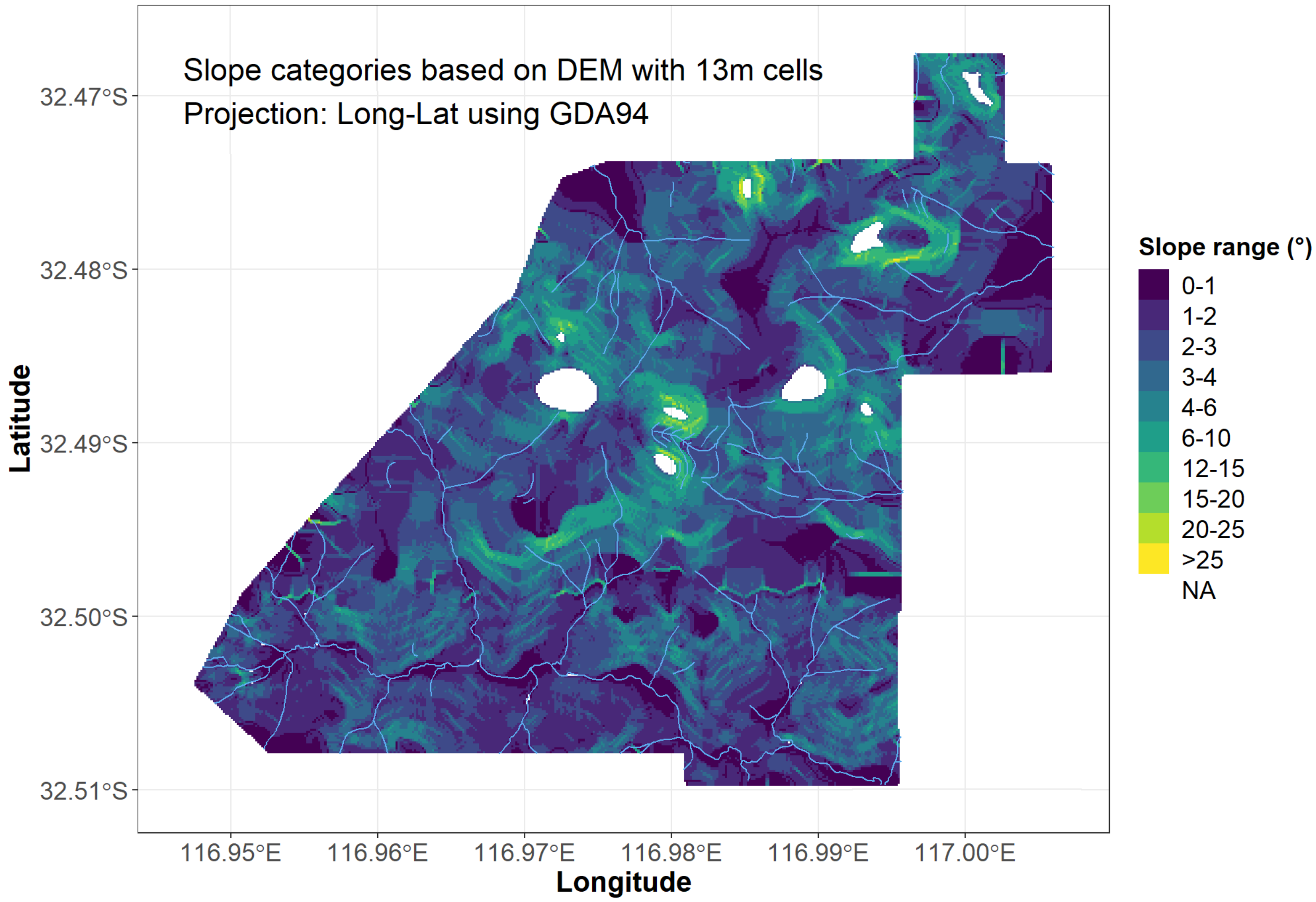


Categorise the slopes into range classes and summarise
(nice breaks between classes found by trial and error!)

```
RF_slopeDF$SlopeCat <- cut(RF_slopeDF$slope,
                       breaks = c(0,1,2,3,4,6,10,15,20,25,99),
                       labels=c("0-1","1-2","2-3","3-4", "4-6","6-10","12-15",
                                "15-20","20-25",">25"))
# summarise the slope categories
cat("Percentage of pixels in each slope category:")
## Percentage of pixels in each slope category:
(pctslopes <-
  round(100*table(RF_slopeDF$SlopeCat)/sum(table(RF_slopeDF$SlopeCat)),2))
##
##   0-1   1-2   2-3   3-4   4-6  6-10 12-15 15-20 20-25   >25
## 10.05 25.18 28.93 17.90 10.70  5.33  1.56  0.25  0.10  0.01
```

plot the categorised slope map

```
slopeCatMapGG <- ggplot() +
  geom_raster(data = RF_slopeDF , aes(x = x, y = y, fill = SlopeCat)) +
  geom_sf(data = rf_hydrology, col="steelblue2") + # add streams
  scale_fill_viridis_d(na.value = "#FFFFFF00", name="Slope range (\u00B0)") +
  labs(x = "Longitude", y="Latitude") +
  coord_sf(crs=st_crs(28350)) +
  annotate(geom="text", x=495000, y=6407500,
          label=paste0("Slope categories based on DEM with 13m cells",
                       "\nProjection: Long-Lat using GDA94"),
           size=6, hjust = 0) +
  theme_bw() +
  theme(axis.title = element_text(size=16, face="bold"),
        axis.text = element_text(size=14),
        legend.title = element_text(size=14, face="bold"),
        legend.text = element_text(size=14))
slopeCatMapGG
```



Working with aspect

Extract aspect from elevation data

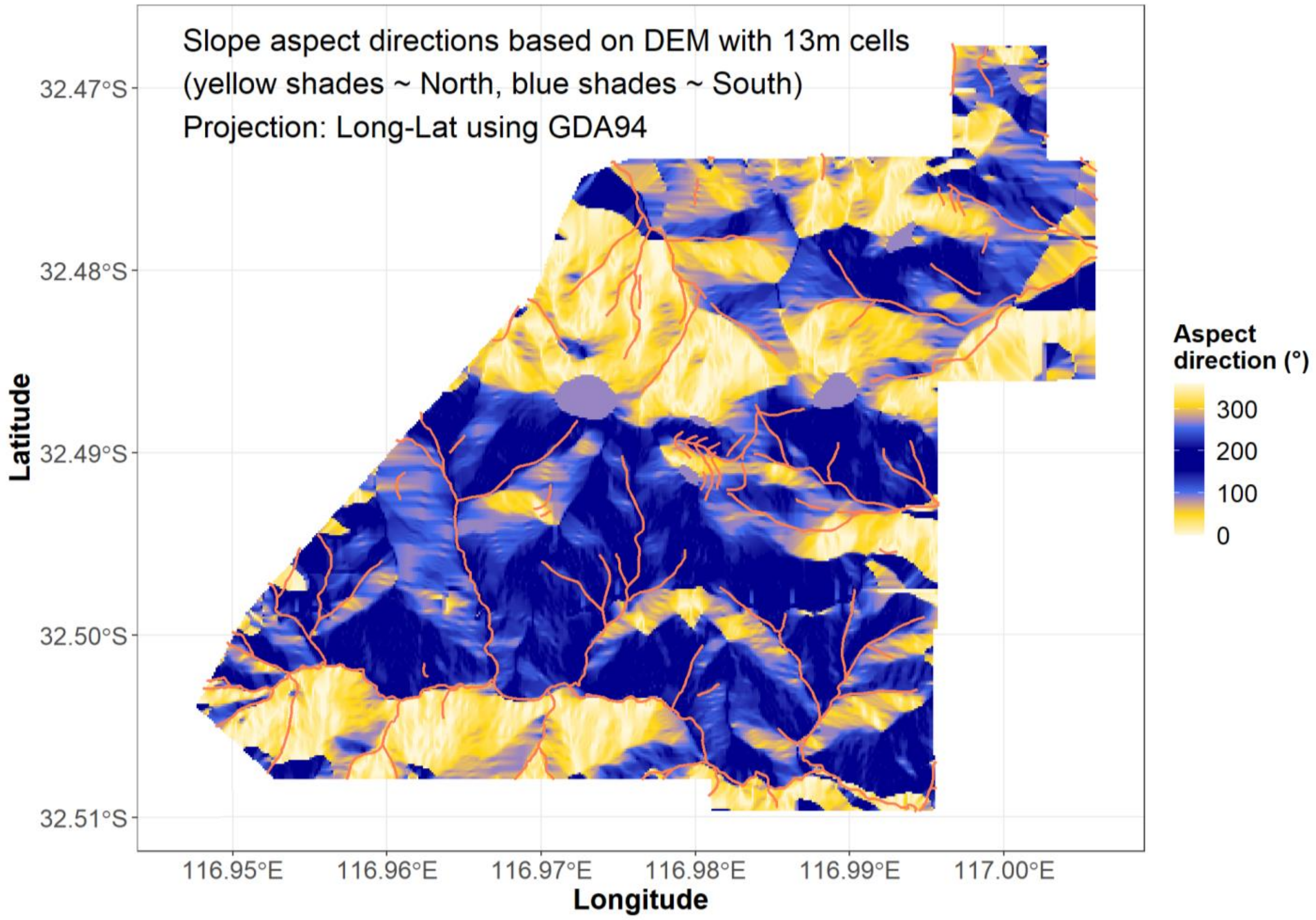
```
RF_aspect <- terrain(RF_elev, opt = "aspect", unit = "degrees")
```

Make data frame version of aspect data

```
RF_aspectDF <- as.data.frame(RF_aspect, xy=TRUE)
```

Map aspect with ggplot

```
RF_aspectGG <- ggplot() +
  geom_raster(data = RF_aspectDF, aes(x = x, y = y, fill = aspect)) +
  # make interpretable colour gradient
  scale_fill_gradientn(na.value = "#00000000",
                       colors = c("cornsilk","gold","royalblue2","blue4",
                                  "blue4","royalblue2","gold","cornsilk"),
                       name="Aspect\ndirection (\u00B0)") +
  geom_sf(data = rf_hydrology, col="coral", size=0.8) +
  labs(x = "Longitude", y="Latitude") +
  annotate(geom="text", x=495000, y=6407500,
           label=paste0("Slope aspect directions based on DEM with 13m cells",
                        "\n(yellow shades ~ North, blue shades ~ South)",
                        "\nProjection: Long-Lat using GDA94"),
           size=6, hjust = 0) +
  coord_sf(crs=st_crs(28350)) +
  theme_bw() +
  theme(axis.title = element_text(size=16, face="bold"),
        axis.text = element_text(size=14),
        legend.title = element_text(size=14, face="bold"),
        legend.text = element_text(size=14))
RF_aspectGG
```



Terrain water flow direction

Extract flow direction from elevation data

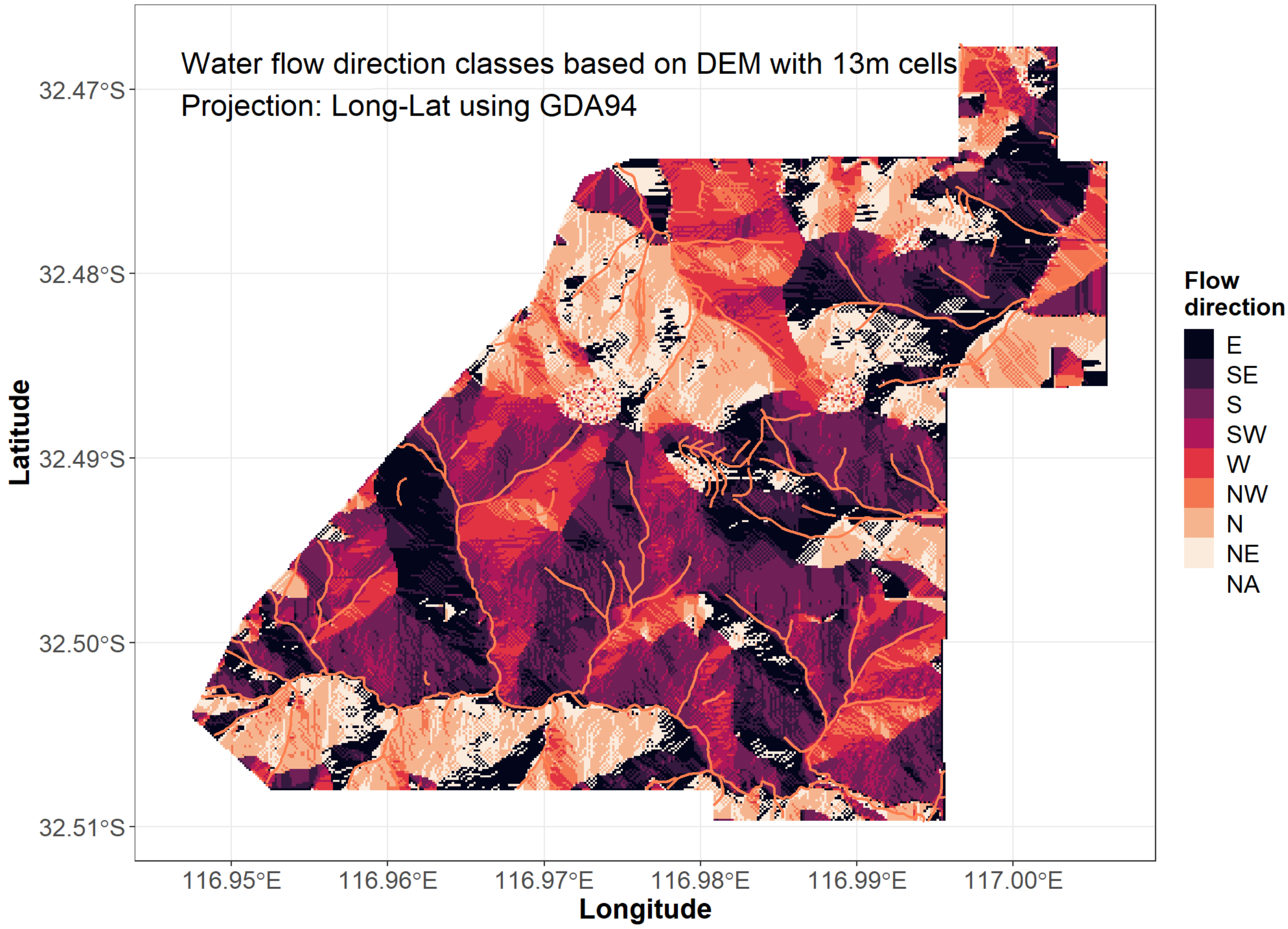
```
RF_flowdir <- terrain(RF_elev, opt = "flowdir")
```

Make data frame version of flow directions

```
RF_flowdirDF <- as.data.frame(RF_flowdir, xy=TRUE)
```

Flow direction map with ggplot

```
RF_flowdirDF$flowQ <- cut(RF_flowdirDF$flowdir, breaks=c(0,2^seq(0,7)+0.5),
                          labels=c("E","SE","S","SW","W","NW","N","NE"))
RF_flowGG <- ggplot() +
  geom_raster(data = RF_flowdirDF, aes(x = x, y = y, fill = flowQ)) +
  scale_fill_viridis_d(na.value = "#00000000",
                       option = "rocket",
                       name="Flow\ndirection") +
  geom_sf(data = rf_hydrology, col="coral", size=0.8) +
  labs(x = "Longitude", y="Latitude") +
  annotate(geom="text", x=495000, y=6407500,
           label=paste0("Water flow direction classes based on DEM with 13m cells",
                        "\nProjection: Long-Lat using GDA94"),
           size=6, hjust = 0) +
  coord_sf(crs=st_crs(28350)) +
  theme_bw() +
  theme(axis.title = element_text(size=16, face="bold"),
        axis.text = element_text(size=14),
        legend.title = element_text(size=14, face="bold"),
        legend.text = element_text(size=14))
RF_flowGG
```



Writing data back to geoTIFF

Write the `.tif` files

```
writeRaster(RF_slope, filename="Ridgefield_Slope_EPSG28350.tif", format="GTiff")
writeRaster(RF_aspect, filename="Ridgefield_Aspect_EPSG28350.tif", format="GTiff")
writeRaster(RF_flowdir, filename="Ridgefield_flowdir_EPSG28350.tif", format="GTiff")
```

Check files with `rgdal::GDALinfo()`

```
GDALinfo("Ridgefield_Slope_EPSG28350.tif") # just file info
## Warning in getProjectionRef(x, OVERRIDE_PROJ_DATUM_WITH_TOWGS84
## = OVERRIDE_PROJ_DATUM_WITH_TOWGS84, : Discarded datum
## Geocentric_Datum_of_Australia_1994 in Proj4 definition: +proj=utm +zone=50
## +south +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 +units=m +no_defs
## rows        360
## columns     425
## bands       1
## lower left origin.x        495053
## lower left origin.y        6403059
## res.x       13
## res.y       13
## ysign       -1
## oblique.x   0
## oblique.y   0
## driver      GTiff
## projection  +proj=utm +zone=50 +south +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 +units=m
## +no_defs
## file        Ridgefield_Slope_EPSG28350.tif
## apparent band summary:
##    GDType hasNoDataValue NoDataValue blockSize1 blockSize2
## 1 Float32           TRUE    -3.4e+38          4        425
## apparent band statistics:
##   Bmin     Bmax Bmean   Bsd
## 1    0 28.54302 -9999 -9999
## Metadata:
## AREA_OR_POINT=Area
GDALinfo("Ridgefield_Aspect_EPSG28350.tif") # just file info
## Warning in getProjectionRef(x, OVERRIDE_PROJ_DATUM_WITH_TOWGS84
## = OVERRIDE_PROJ_DATUM_WITH_TOWGS84, : Discarded datum
## Geocentric_Datum_of_Australia_1994 in Proj4 definition: +proj=utm +zone=50
## +south +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 +units=m +no_defs
## rows        360
## columns     425
## bands       1
## lower left origin.x        495053
## lower left origin.y        6403059
## res.x       13
## res.y       13
## ysign       -1
## oblique.x   0
## oblique.y   0
## driver      GTiff
## projection  +proj=utm +zone=50 +south +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 +units=m
## +no_defs
## file        Ridgefield_Aspect_EPSG28350.tif
## apparent band summary:
##    GDType hasNoDataValue NoDataValue blockSize1 blockSize2
## 1 Float32           TRUE    -3.4e+38          4        425
## apparent band statistics:
##   Bmin Bmax Bmean   Bsd
## 1    0  360 -9999 -9999
## Metadata:
## AREA_OR_POINT=Area
GDALinfo("Ridgefield_flowdir_EPSG28350.tif") # just file info
## Warning in getProjectionRef(x, OVERRIDE_PROJ_DATUM_WITH_TOWGS84
## = OVERRIDE_PROJ_DATUM_WITH_TOWGS84, : Discarded datum
## Geocentric_Datum_of_Australia_1994 in Proj4 definition: +proj=utm +zone=50
## +south +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 +units=m +no_defs
## rows        360
## columns     425
## bands       1
## lower left origin.x        495053
## lower left origin.y        6403059
```

```
## res.x        13
## res.y        13
## ysign        -1
## oblique.x    0
## oblique.y    0
## driver       GTiff
## projection   +proj=utm +zone=50 +south +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 +units=m
## +no_defs
## file         Ridgefield_flowdir_EPSG28350.tif
## apparent band summary:
##    GDType hasNoDataValue NoDataValue blockSize1 blockSize2
## 1 Float32           TRUE    -3.4e+38          4        425
## apparent band statistics:
##   Bmin Bmax Bmean   Bsd
## 1    1  128 -9999 -9999
## Metadata:
## AREA_OR_POINT=Area
```

---

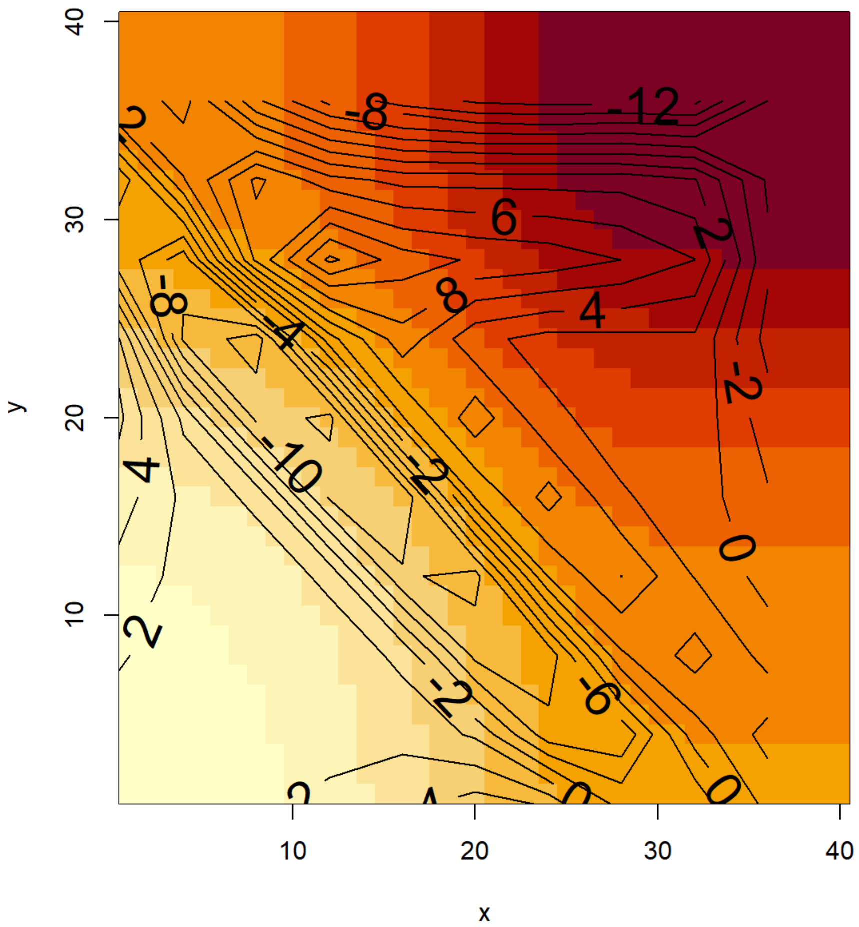
not sure if this works! \ trying on the basis that slope curvature ~ second derivative of elevation???

```
RF_curvature <- terrain(RF_slope, opt = "slope", unit="degrees")
plot(RF_curvature, col=cividis(128))
```

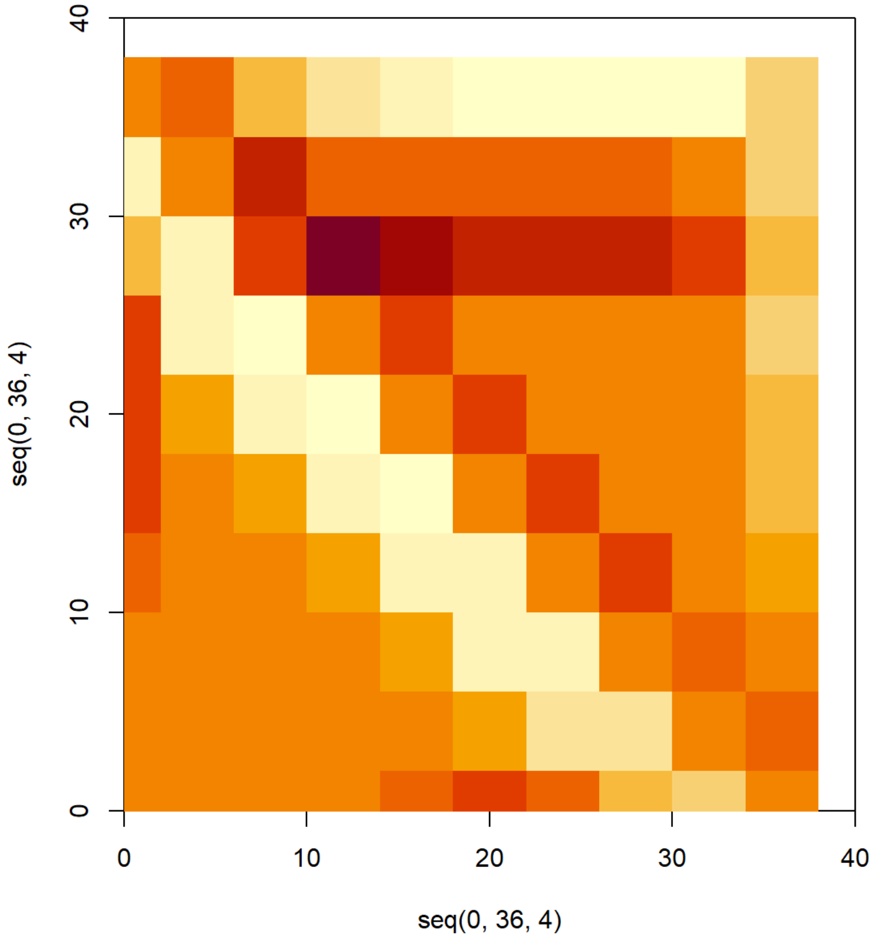
Trying the `fgeo.analyze::` R package for curvature

Trying the test data first (made near top of this file)

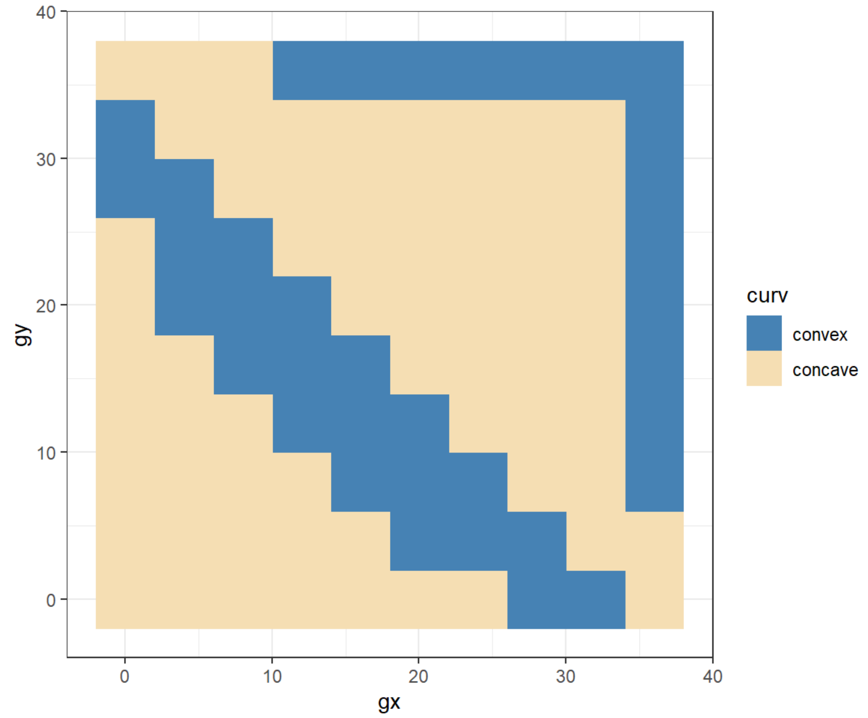
```
## elevTopoTib <- fgeo_topography(elevDF,gridsize=4,xdim=40,ydim=40, edgecorrect = F)
elevTopo <- as.data.frame(elevTopoTib)
image(seq(1,40), seq(1,40), as.matrix(testElev[,2:41]),xlab="x",ylab="y")
contour(seq(0,36,4), seq(0,36,4), matrix(elevTopo$convex,ncol=10),add=T,
        labcex=2)
```



```r
# -=-=-=- OR -=-=-=-
image(seq(0,36,4), seq(0,36,4), matrix(elevTopo$convex,ncol=10),
      xlim=c(0,40), ylim=c(0,40))
```




```r
elevTopo$curv <- cut(elevTopo$convex, breaks=c(-999999,0,999999),
                     labels=c("convex","concave"))
ggplot() +
  geom_raster(data = elevTopo, aes(x = gx, y = gy, fill = curv)) +
  scale_fill_manual(values = c("steelblue","wheat"),na.value = "#FFFFFF00") +
  coord_equal() +
  theme_bw()
```



```
test_slopeDF <- as.data.frame(test_slope, xy=T)
```

```
fgeo_topography() function still not working on Ridgefield data :( ####
even when using a raster subset with no NA values
NW_elevDF <- as.data.frame(NW_elev, xy=TRUE)
names(NW_elevDF)[3] <- "elev"
summary(NW_elevDF)
```

```
NW_topo_pars <- fgeo_topography(NW_elevDF, gridsize=20, xdim=425, ydim = 360)
```

```
Error: cannot allocate vector of size 59.7 Gb
```

```
NW_topo_pars <- fgeo_topography(NW_elevDF, gridsize=40, xdim=425, ydim = 360)
```

```
Error in if (theta1 <= theta2) { : missing value where TRUE/FALSE needed
```

```
NW_topo_pars <- fgeo_topography(NW_elevDF, gridsize=40, xdim=425, ydim = 360,
                                edgecorrect = F)
```

```
Error in if (theta1 <= theta2) { : missing value where TRUE/FALSE needed
```