

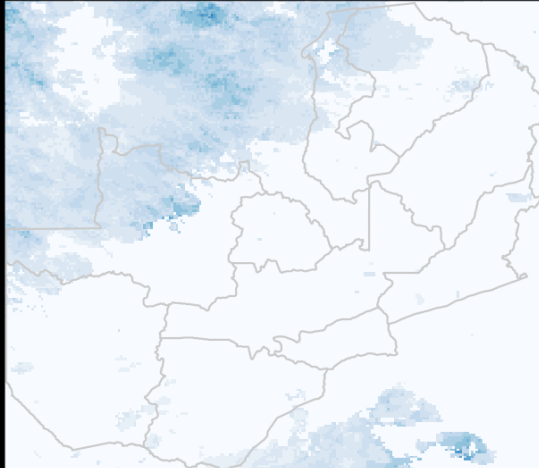
Geospatial Analysis with R

Class 4

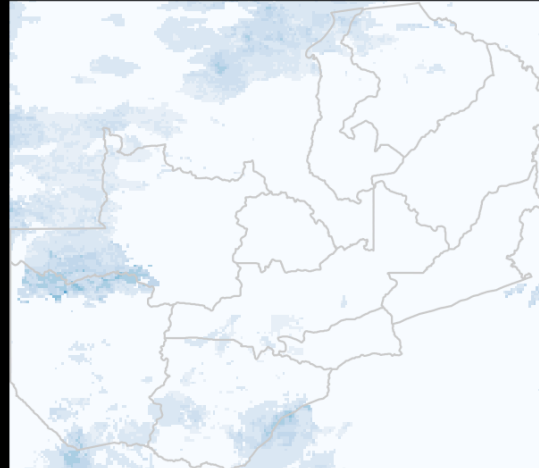


Weekly Rainfall (mm)

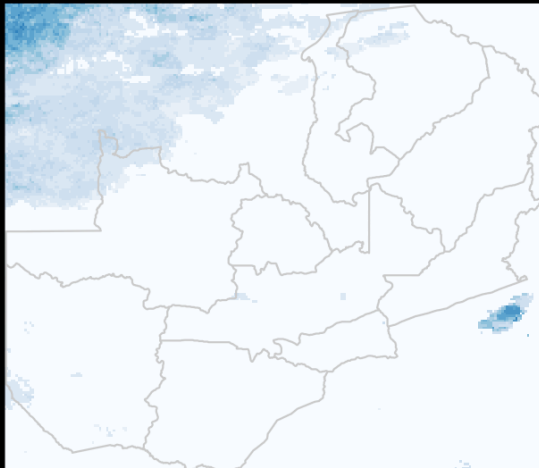
Week of 2016-10-25



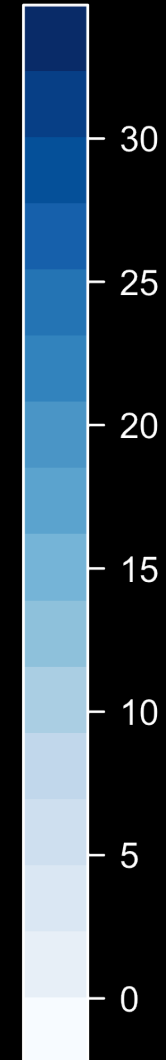
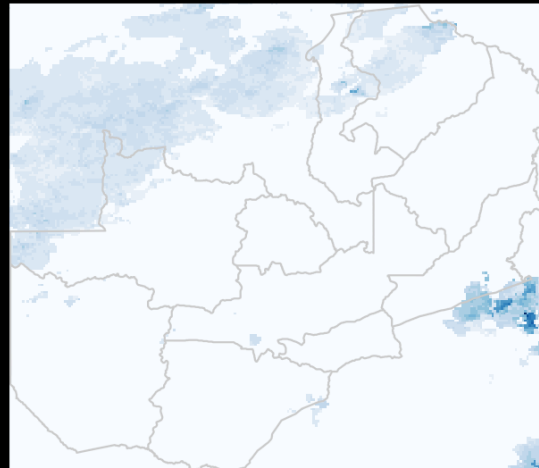
Week of 2016-11-01



Week of 2016-11-08



Week of 2016-11-15



```

data("chirps")
library(rasterVis)
library(lubridate)
zam <- getData("GADM", country = "ZMB", level = 1)

dates <- seq(as_date("2016-10-25"), as_date("2016-11-21"), by = "day")
dates <- data.frame(dates, w = unlist(lapply(1:4, function(x) rep(x, 7))))
rfweek <- stack(lapply(unique(dates$w), function(x) {
  w <- which(dates$w == x)
  calc(chirps[[x]], sum)
})))
wk <- as_date(sapply(unique(dates$w), function(x) dates[dates$w == x, 1][1]))
names(rfweek) <- wk

mapTheme <- rasterTheme(region = brewer.pal(9, "Blues"))
pstrip <- list(cex = 1, col = "white")
png("inst/slides/figures/weekly-rf.png", height = 5, width = 5, res = 300,
    units = "in", bg = "transparent")
p <- levelplot(rfweek, scales=list(draw = FALSE), par.settings = mapTheme,
  names.attr = paste("Week of", wk), axes = FALSE, xlab = "",
  ylab = "", par.strip.text = pstrip,
  main = list("Weekly Rainfall (mm)", col = "White"),
  colorkey = list(axis.line = list(col = "white"),
    axis.text = list(col = "white")))
p2 <- p + layer(sp.polygons(zam, col = "grey80", lwd = 0.7))
p2
dev.off()

```

Today

- A review of where we are so far
- Looking a bit more at Rmarkdown
- `git` merging exercise

What we should know by now

- Key concepts/tools of reproducibility and why we use them
- We should know:
 - How to set up R package project with `git` VCS
 - How to keep project synced between local and remote repos
 - How to document functions
 - Where your library lives
 - How package source differs from installed package
 - What the key ingredients in a package are

What we should know by now

- Key concepts/tools of reproducibility and why we use them
- We should know:
 - How to set up R package project with `git` VCS
 - How to keep project synced between local and remote repos
 - How to document functions
 - Where your library lives
 - How package source differs from installed package
 - What the key ingredients in a package are
- What we might not know (about packages):
 - data folder and lazy loads
 - inst folder and how to get at it

Data in packages

```
1 Package: geospaar
2 Type: Package
3 Title: Course materials for Clark University's GEOG 246/346
4 Version: 1.0.1|
5 Author: Lyndon Estes
6 Maintainer: Lyndon Estes <LEstes@clarku.edu>
7 Description: This package contains the course materials for Geospatial
8               Analysis with R.
9 License: GPL (>= 3)
10 Encoding: UTF-8
11 LazyData: true
12 Imports: raster, dplyr
13 Suggests: knitr, rmarkdown
14 VignetteBuilder: knitr
15 RoxygenNote: 6.0.1
```

- packages often include example data
- Lazy loading data only loads when used
- Lazy loaded data in data/ folder
 - formats: .R, .rda, .RData, .tab, .txt, .csv
- Non lazy loads (raw data) in inst/extdata

```
ls()
```

```
## character(0)
```

```
data("chirps", package = "geospaar")  
ls()
```

```
## [1] "chirps"
```

```
chirps
```

```
## Loading required package: raster
```

```
## Loading required package: sp
```

```
## class      : RasterBrick  
## dimensions : 197, 234, 46098, 28 (nrow, ncol, ncell, nlayers)  
## resolution : 0.05, 0.05 (x, y)  
## extent      : 21.95, 33.65, -18.05, -8.200001 (xmin, xmax, ymin, ymax)  
## coord. ref. : +proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=0,0,0  
## data source : in memory  
## names       : Y16299, Y16300, Y16301, Y16302, Y16303, Y16304, Y16305, Y16306,  
## min values  : 0, 0, 0, 0, 0, 0, 0, 0,  
## max values  : 21.33322, 17.76521, 22.12555, 32.39063, 19.46936, 28.53870, 19.03967, 26.27214,
```

```
rm(list = ls())  
ls()
```

```
## character(0)
```



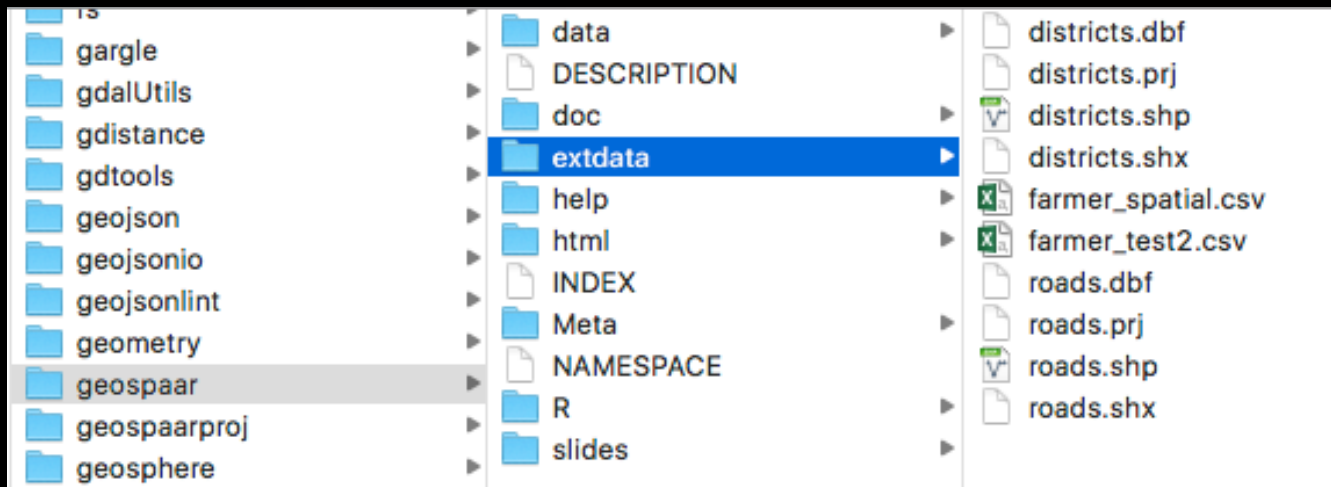
```
library(geospaar)
ls()
```

```
## character(0)
```

```
chirps
```

```
## class      : RasterBrick
## dimensions  : 197, 234, 46098, 28  (nrow, ncol, ncell, nlayers)
## resolution  : 0.05, 0.05  (x, y)
## extent     : 21.95, 33.65, -18.05, -8.200001  (xmin, xmax, ymin, ymax)
## coord. ref. : +proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=0,0,0
## data source : in memory
## names       : Y16299, Y16300, Y16301, Y16302, Y16303, Y16304, Y16305, Y16306,
## min values  : 0, 0, 0, 0, 0, 0, 0, 0,
## max values  : 21.33322, 17.76521, 22.12555, 32.39063, 19.46936, 28.53870, 19.03967, 26.27214,
```

Raw data in inst/extdata



```
system.file("extdata", package = "geospaar")
```

```
## [1] "/Library/Frameworks/R.framework/Versions/3.5/Resources/library/geospaar/extdata"
```

```
dir(system.file("extdata", package = "geospaar"))
```

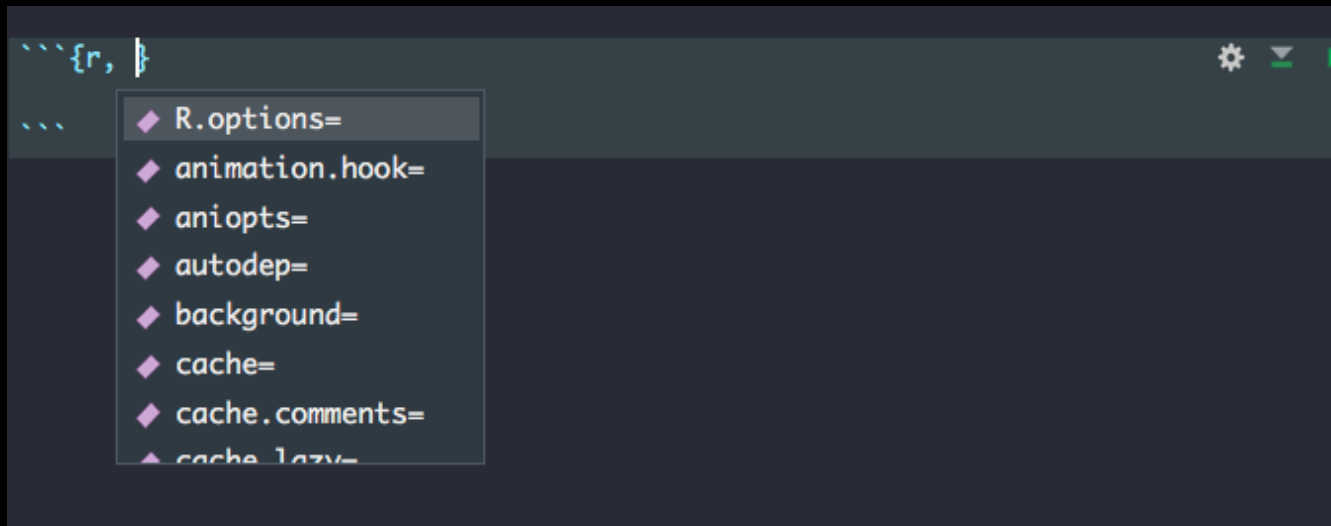
```
## [1] "districts.dbf"      "districts.prj"      "districts.shp"
## [4] "districts.shx"      "farmer_spatial.csv" "farmer_test2.csv"
## [7] "roads.dbf"          "roads.prj"          "roads.shp"
## [10] "roads.shx"
```

```
f <- system.file("extdata", "farmer_spatial.csv", package = "geospaar")
head(read.csv(f))
```

```
##      uuid      lat      lon      date pl ra
## 1 009a8424 -16.9257 27.2559 2016-10-31 NA 0
## 2 00df166f -16.5036 26.9425 2016-10-31 NA 0
## 3 019d99f9 -13.1991 24.7820 2016-10-31 NA 0
## 4 02671a00 -16.9137 27.2537 2016-10-31 NA 0
## 5 02be9843 -15.4070 27.0849 2016-10-31 NA 0
## 6 0360d642 -16.6044 26.8255 2016-10-31 NA 0
```

A look at RMarkdown

Chunk options



Merging

