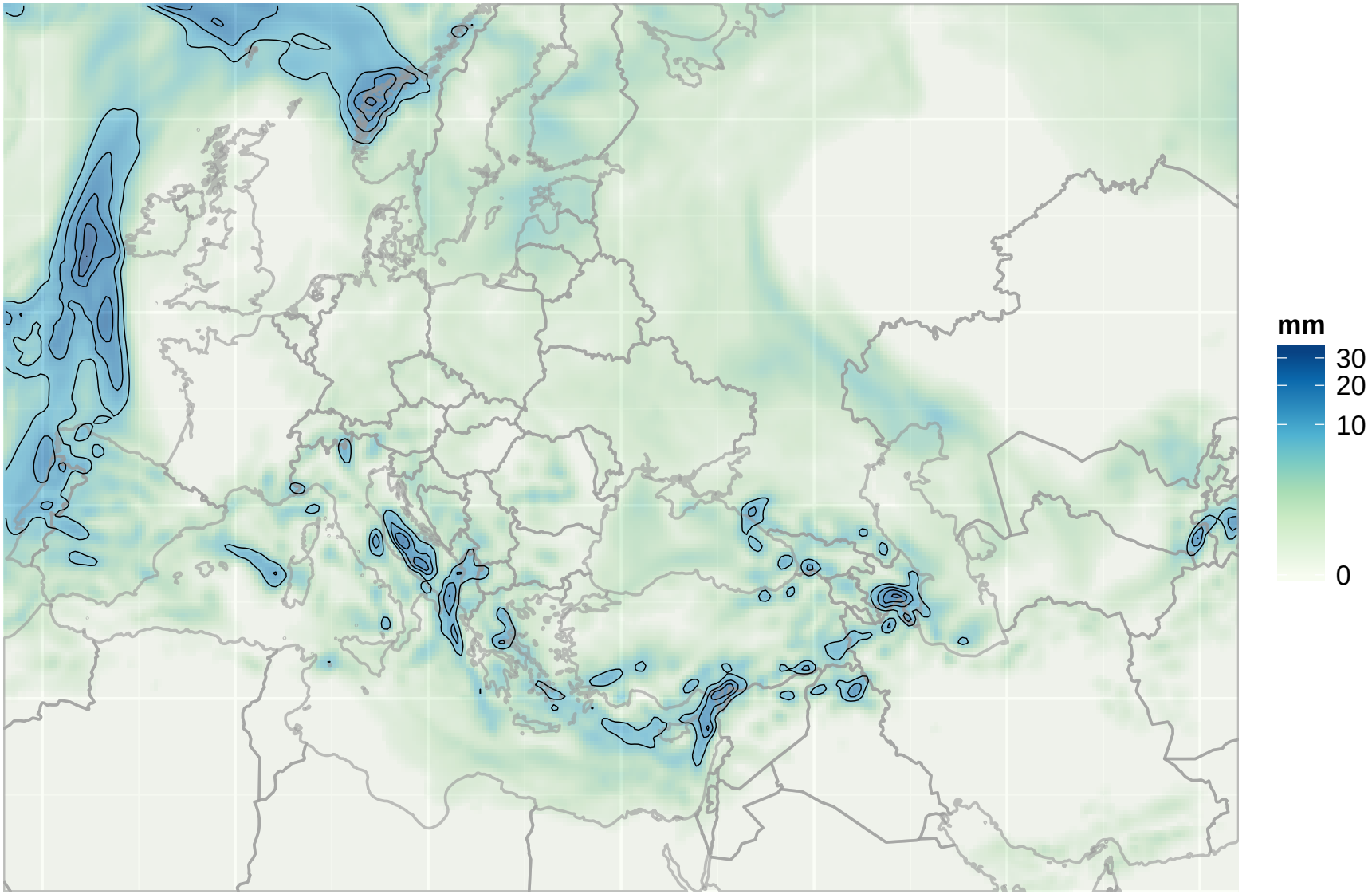
Spatial data with R

some applications to weather and agriculture

Joe Wheatley

Biospherica Risk

Dublin R
21 Feb 2013


mm
30
20
10
0

Packages

- **sp** vector data
- **raster** grid data
- **rgdal** input/output, coordinate systems
- **rgeos** geometric calculations on vector data

Vector example: admin boundary

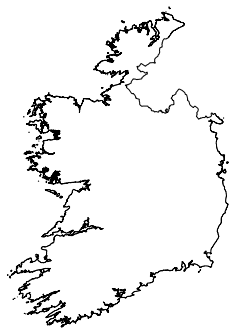
```
>map <- getData('GADM',country='IRL', level=0)
>projection(map)
[1] "+proj=longlat +datum=WGS84"
```

Vector example: admin boundary

```
>map <- getData('GADM',country='IRL', level=0)
>projection(map)
[1] "+proj=longlat +datum=WGS84"

>str(map,2)
Formal class 'SpatialPolygonsDataFrame' [package "sp"] with 5 slots
  ..@ data       :'data.frame': 1 obs. of  70 variables:
  ..@ polygons   :List of 1
  ..@ plotOrder  : int 1
  ..@ bbox       : num [1:2, 1:2] -10.66 51.42 -5.99 55.45
  .. ..- attr(*, "dimnames")=List of 2
  ..@ proj4string:Formal class 'CRS' [package "sp"] with 1 slots
```

Vector example: admin boundary

```
>map <- getData('GADM',country='IRL', level=0)
>projection(map)
[1] "+proj=longlat +datum=WGS84"

>str(map,2)
Formal class 'SpatialPolygonsDataFrame' [package "sp"] with 5 slots
  ..@ data       :'data.frame': 1 obs. of  70 variables:
  ..@ polygons   :List of 1
  ..@ plotOrder  : int 1
  ..@ bbox       : num [1:2, 1:2] -10.66 51.42 -5.99 55.45
  .. ..- attr(*, "dimnames")=List of 2
  ..@ proj4string:Formal class 'CRS' [package "sp"] with 1 slots

>plot(map)
```



Transforming vector data

change projection and units ...

```
>newproj <- "+proj=aea +lat_1=20 ... +ellps=WGS84 +units=km"
>map <- spTransform(map,CRS(newproj))

>gArea(map)
[1] 70250.45   # wikipedia 70,273 km2
```

Transforming vector data

change projection and units ...

```
>newproj <- "+proj=aea +lat_1=20 ... +ellps=WGS84 +units=km"
>map <- spTransform(map,CRS(newproj))

>gArea(map)
[1] 70250.45   # wikipedia 70,273 km2
```

separate polygons ..

```
>polylist <- lapply(1:421,
  function(i) Polygons(list(map@polygons[[1]]@Polygons[[i]]),i) )
>map <- SpatialPolygons(polylist,proj4string=CRS(newproj))
```

Transforming vector data

change projection and units ...

```
>newproj <- "+proj=aea +lat_1=20 ... +ellps=WGS84 +units=km"
>map <- spTransform(map,CRS(newproj))

>gArea(map)
[1] 70250.45   # wikipedia 70,273 km2
```

separate polygons ..

```
>polylist <- lapply(1:421,
  function(i) Polygons(list(map@polygons[[1]]@Polygons[[i]]),i) )
>map <- SpatialPolygons(polylist,proj4string=CRS(newproj))
```

attach data ..

```
>map.areas <-  sapply(1:421, function(i) gArea(map[i,]))
>df <- data.frame(island=NA, area=map.areas)
>map <- SpatialPolygonsDataFrame(map, df)
```

Transforming vector data: cntd

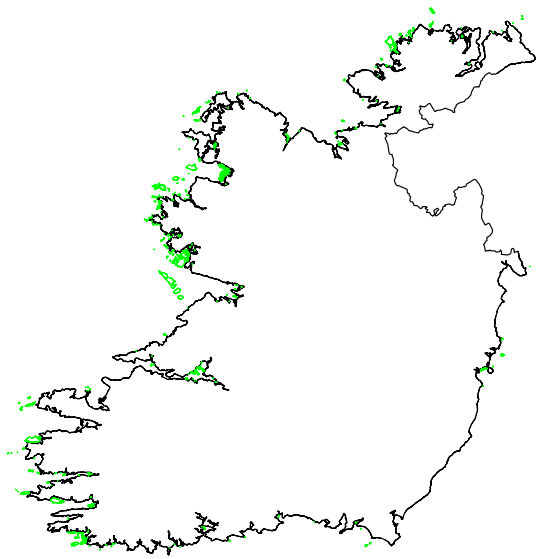
```
>str(map,2)
Formal class 'SpatialPolygonsDataFrame' [package "sp"] with 5 slots
  ..@ data :'data.frame': 421 obs. of  2 variables:
  ..@ polygons :List of 421
  ..@ plotOrder : int [1:421] 1 2 3 4 5 6 7 8 9 10 ...
  ..@ bbox : num [1:2, 1:2] -2585 1287 -2214 1671....
  ..@ proj4string:Formal class 'CRS' [package "sp"] with 1 slots
```

Transforming vector data: cntd

```
>str(map,2)
Formal class 'SpatialPolygonsDataFrame' [package "sp"] with 5 slots
  ..@ data :'data.frame': 421 obs. of  2 variables:
  ..@ polygons :List of 421
  ..@ plotOrder : int [1:421] 1 2 3 4 5 6 7 8 9 10 ...
  ..@ bbox : num [1:2, 1:2] -2585 1287 -2214 1671....
  ..@ proj4string:Formal class 'CRS' [package "sp"] with 1 slots
```

sorting and subsetting ..

```
>map <- map[order(-map$area),]
>coastline <- map[1,]
>islands <- map[2:421,]
```



Coastline Paradox (Richardson 1961)

```
>gLength(gSimplify(coastline,1))
[1] 4329.766
>gLength(gSimplify(coastline,10))
[1] 2488.165
```

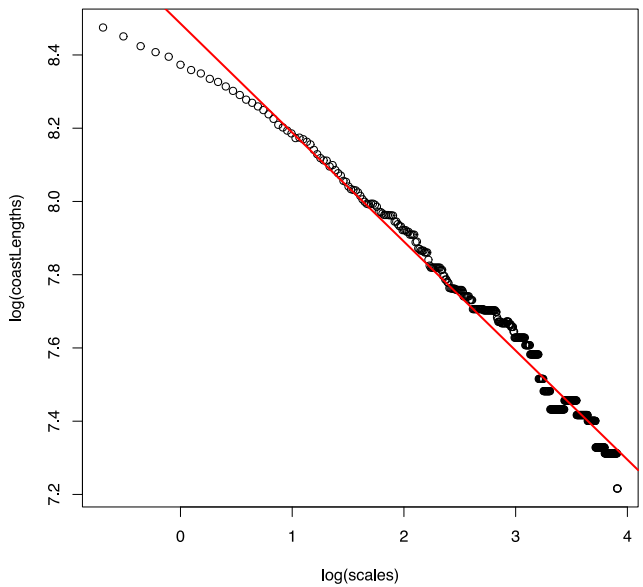
Coastline Paradox (Richardson 1961)

```
>gLength(gSimplify(coastline,1))
[1] 4329.766
>gLength(gSimplify(coastline,10))
[1] 2488.165
```

$$L \propto \text{scale}^{1-D}$$

```
>scales <- seq(0.5,50,by=0.1)
>coastLengths <- sapply(scales, function(s)
                                    gLength(gSimplify(coastline,s)))
>lm(log(coastLengths)~log(scales))
```

$\implies D = 1.3$



Vector example: soil map of world

ESRI shapefile from fao.org

```
>soils <- readOGR("~/DSMW/", "DSMW") #read
>projection(soils) <- "+proj=longlat +datum=WGS84"
>dim(soils@data)
[1] 34112    12
>soils@data[5000:5002,]
```

	SNUM	FAOSOIL	DOMSOI	PHASE1	PHASE2	MISCLU1	MISCLU2	PERMAFROST	CNTCODE	CNTNAME	SQKM	COUNTRY
4999	3030	Ch1-3a	Ch	<NA>	<NA>	0	0	0	228	UR	302	RUSSIAN FEDERATION
5000	6997	WAT	WR	<NA>	<NA>	0	0	0	33	CA	236	CANADA
5001	3048	De18-1a	De	<NA>	<NA>	0	0	0	306	WH	34	BALARUS

Vector example: soil map of world

ESRI shapefile from fao.org

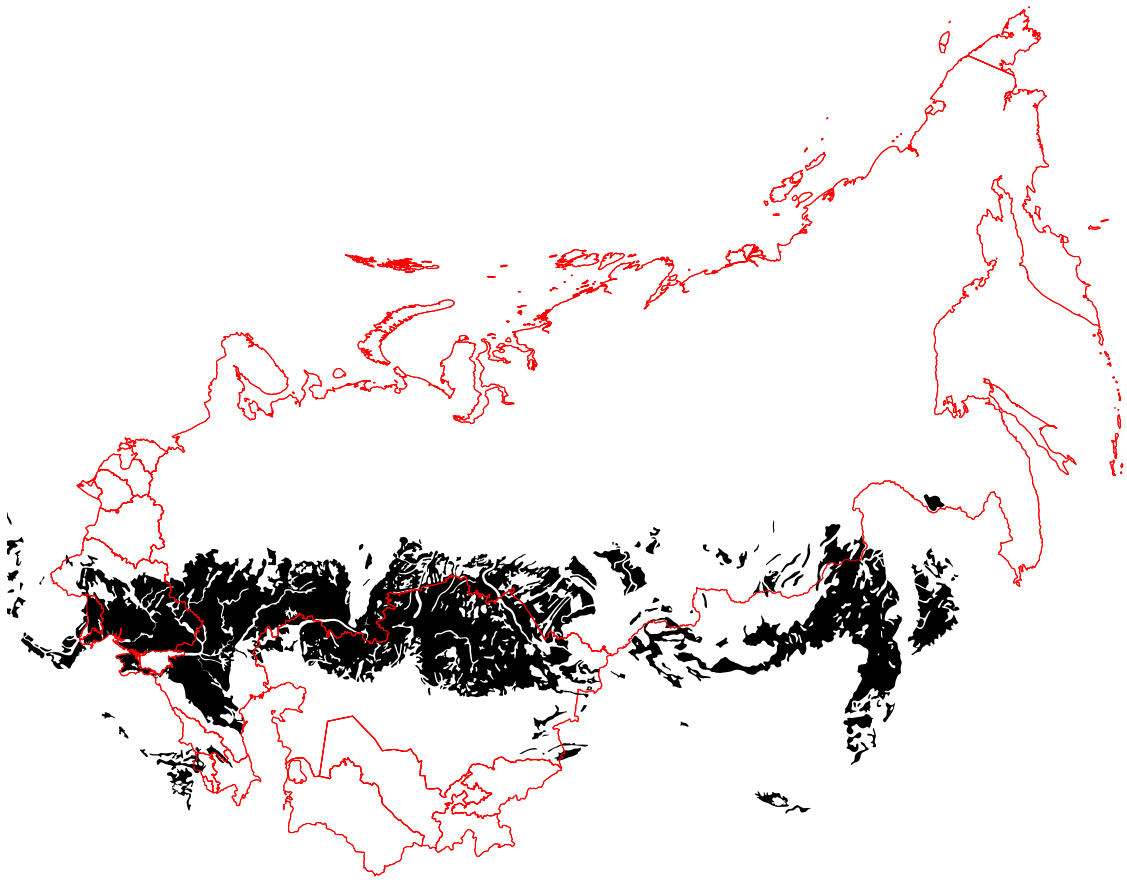
```
>soils <- readOGR("~/DSMW/", "DSMW") #read
>projection(soils) <- "+proj=longlat +datum=WGS84"
>dim(soils@data)
[1] 34112    12
>soils@data[5000:5002,]
```

	SNUM	FAOSOIL	DOMSOI	PHASE1	PHASE2	MISCLU1	MISCLU2	PERMAFROST	CNTCODE	CNTNAME	SQKM	COUNTRY
4999	3030	Ch1-3a	Ch	<NA>	<NA>	0	0	0	228	UR	302	RUSSIAN FEDERATION
5000	6997	WAT	WR	<NA>	<NA>	0	0	0	33	CA	236	CANADA
5001	3048	De18-1a	De	<NA>	<NA>	0	0	0	306	WH	34	BALARUS

Black Earth Soils

```
>blackearth <- subset(soils, substr(DOMSOI,1,1) %in% c("C","K"))
>blackearth <- gUnaryUnion(blackearth)
```

Black Earth



Raster example: global temperature maps

netCDF grid format x,y,z,t,...variable

monthly air temperature anomaly since 1850 metoffice.gov.uk/hadobs/hadcrut4/

```
>air <- brick("~/air.mon.anom.nc",varname="air")
>nlayers(air)
[1]1932
>extent(air)
class       : Extent
xmin        : -180
xmax        : 180
ymin        : -90
ymax        : 90
>projection(air)
[1] "+proj=longlat +datum=WGS84 +ellps=WGS84"


>cellStats(air[[1]],mean,na.rm=T)
[1] -3.547575
```

Raster $\rightarrow$ time-series using *extract*

area of interest = black earth soils of Volga district

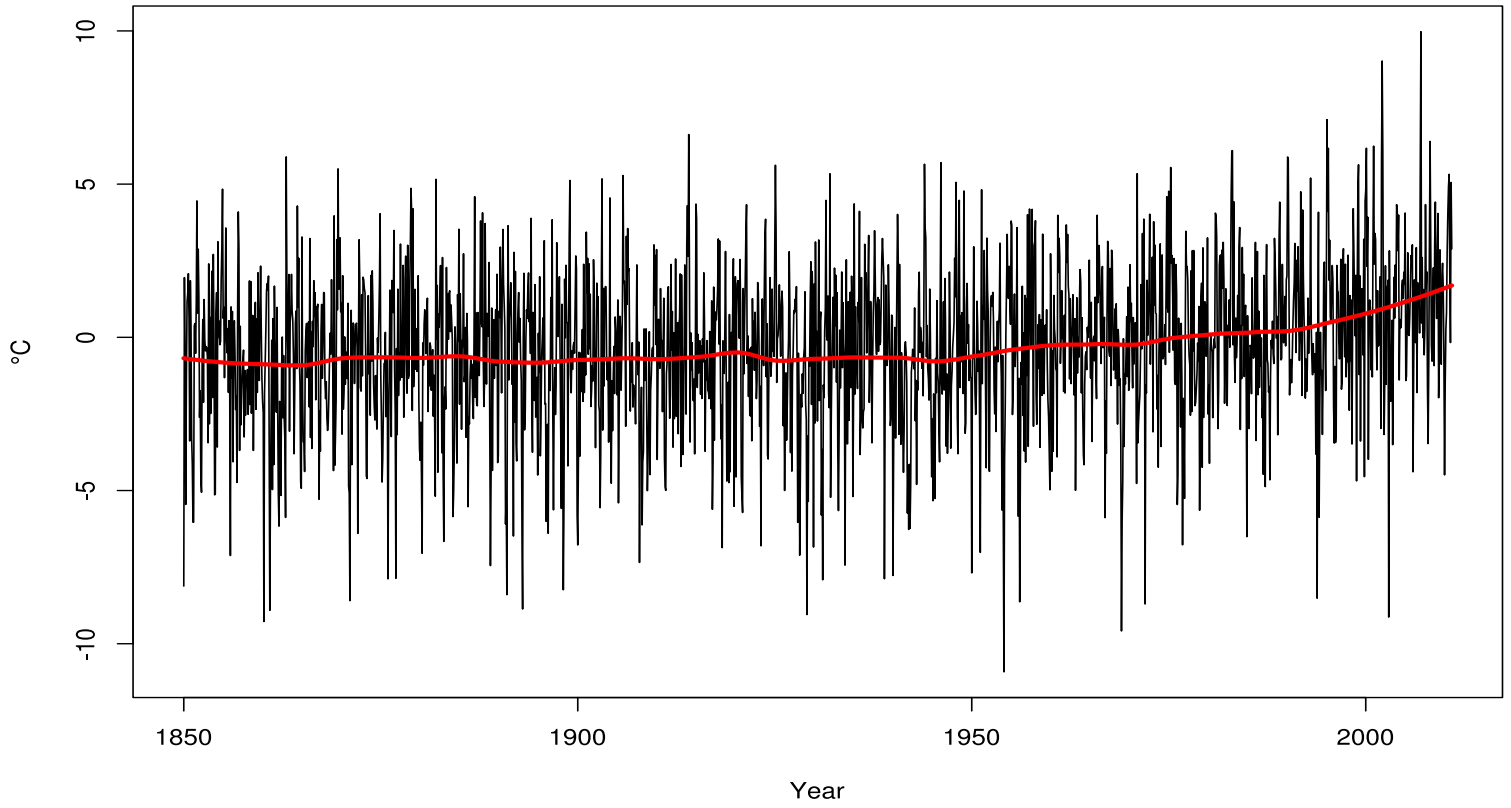
```
>blackearth.volga <- gIntersection(blackearth,volga)
```

extract temperature anomaly since 1850

```
>air <- projectRaster(air, crs=newproj)
```

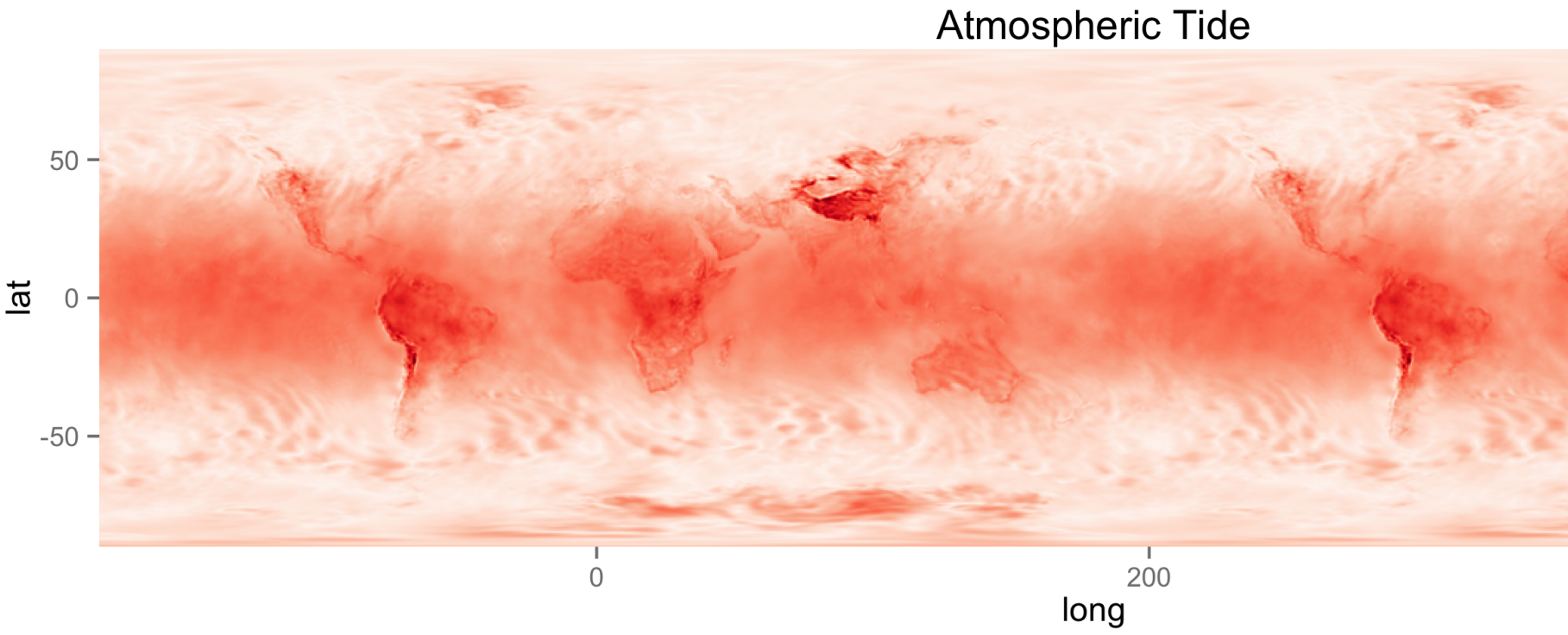
```
>extract(air, blackearth.volga,fun=mean,weights=T,na.rm=T)
[1] -8.1208610  1.9521706 -0.9950178 -5.4555962 -1.313737 ...
```

Peak Pasta?



Hadcrut4

Atmospheric tide



rasterStack $\rightarrow$ rasterLayer using *calc*

surface air pressure 24h/12h periodic variation $\leftarrow$ atmospheric tides

10 days hourly data (rasterBrick) nomads.ncep.noaa.gov

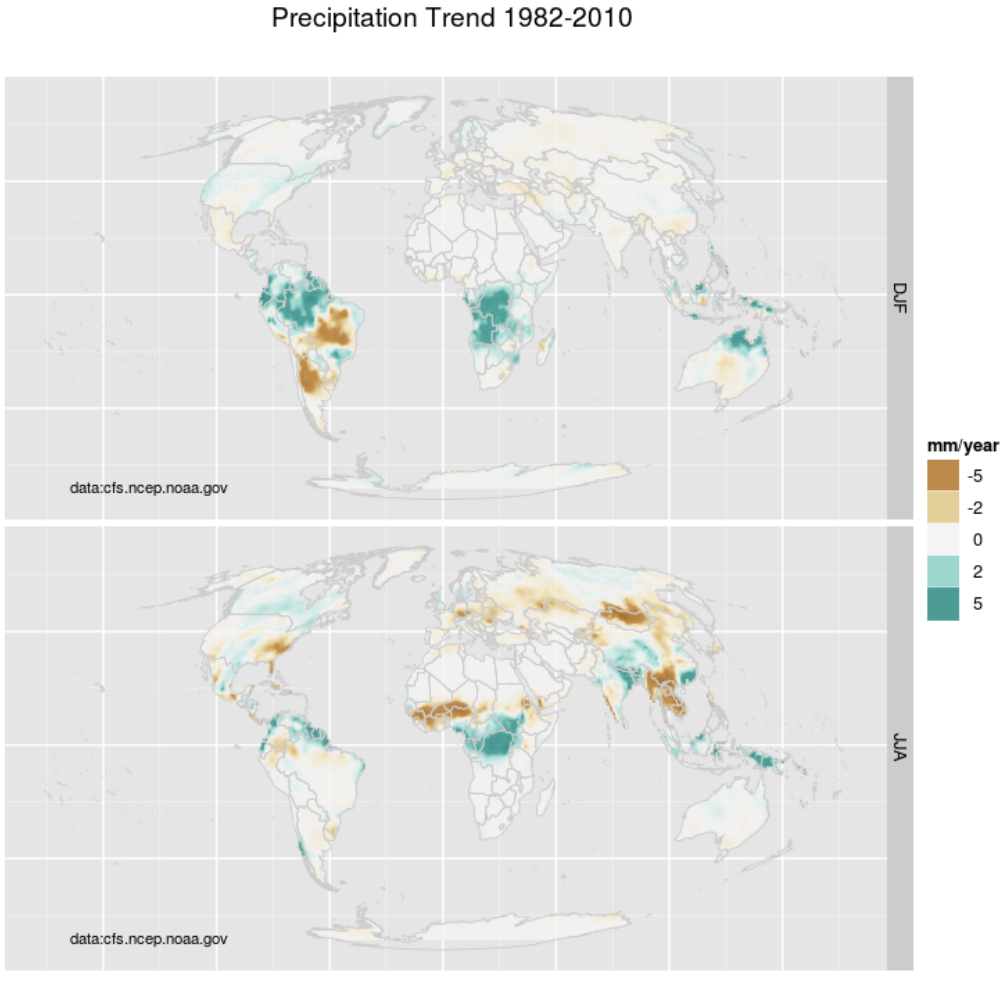
```
>pressure <- brick(c(map1,map2, ......., map240))
```

```
>fun <- function(p){
    #p = vector of hourly air pressure
    #time unit = day
    t <- seq(0,length.out=length(p),by=1/24)
    fit <- lm(p ~ sin(4*pi*t)+cos(4*pi*t))
    coefs <- coefficients(fit)[2:3]
    return(sqrt(coefs %*% coefs))
    }
```

atmosphere tide amplitude (rasterLayer)

```
>amplitude <- calc(pressure,fun)
```

Spatio-temporal: trend extraction



More Info

CRAN Task Views
cran.r-project.org/web/views/Spatial.html
cran.r-project.org/web/views/SpatioTemporal.html

cheatsheet & links
maths.lancs.ac.uk/∼rowlings/Teaching/UseR2012/cheatsheet.html