

Access WFS as csv using R

Introduction

With the monitoring project "Shortlist Wind op Zee", birds have been counted. The data have been managed after the project by Deltares using OpenEarth and are now available as a.o. WFS from the "[Marine Projects](#)" data catalogue

The bird data set

Exploring the bird data set starts with reading the data from the Geoserver as csv. Not all columns are perhaps necessary.

```
require(readr) #(faster and simpler read functions than the standard functions)

## Loading required package: readr
##
## Attaching package: 'readr'
##
## The following objects are masked from 'package:scales':
##
##     col_factor, col_numeric

header <- read_lines("http://marineprojects.openearth.nl/geoserver/ihm_viewer/ows?service=WFS&version=1.0.0
&request=GetFeature&typeName=ihm_viewer:shwoz_vogels&outputFormat=csv", n_max = 1)
strsplit(header, ",")
```

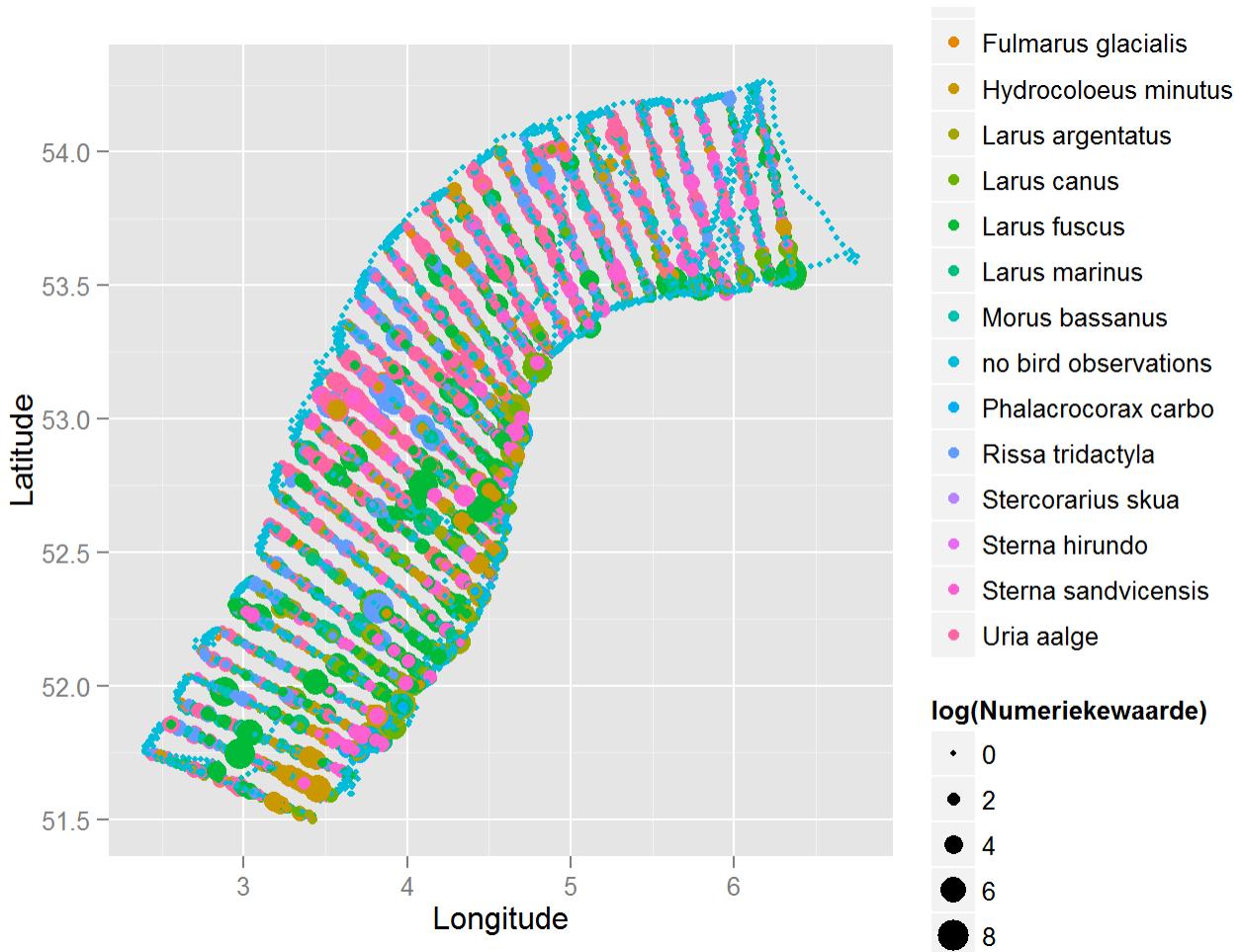
[[1]]	
## [1] "FID"	"Latitude"
## [3] "Longitude"	"Identificatie"
## [5] "Meetpunt.identificatie"	"compartiment.code"
## [7] "orgaan.code"	"BiotaXon.naam"
## [9] "begindiepte_m"	"referentievlek.code"
## [11] "monsterbewerkingsmethode.code"	"bemonsteringsmethode.code"
## [13] "Veldapparaat.omschrijving"	"monsternemingsdatum"
## [15] "UTCoffset"	"Parameter.omschrijving"
## [17] "Eenheid.code"	"Hoedanigheid.code"
## [19] "Bemonsteringsmethode.code"	"Begindatum"
## [21] "Tijd_UTCoffset"	"Limietsymbool"
## [23] "Numeriekewaarde"	"Kwaliteitsoordeel.code"
## [25] "Geometrie"	

Read only columns that are you need, and define column types.

```
birds <- read_csv("http://marineprojects.openearth.nl/geoserver/ihm_viewer/ows?service=WFS&version=1.0.0
&request=GetFeature&typeName=ihm_viewer:shwoz_vogels&outputFormat=csv", col_types = "_ddi____c_____ccc__cc_di_")
## base function read.csv() # would also work equally well.
```

Plot the data

```
require(ggplot2)
plot <- ggplot(birds, aes(Longitude, Latitude))
plot + geom_point(aes(color = BiotaXon.naam, size = log(Numeriekewaarde)))
```



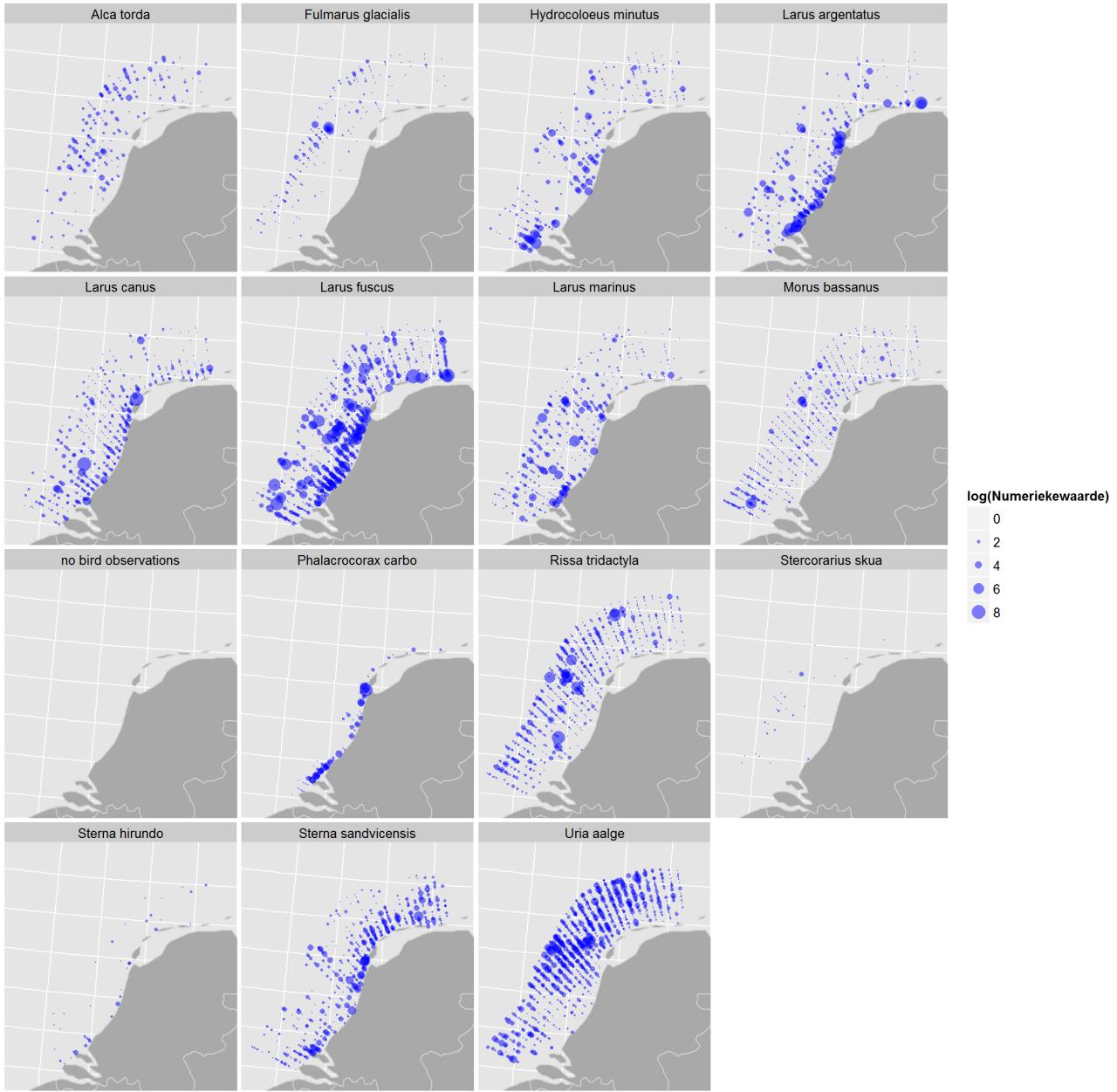
Or make distribution maps

```
data("countriesLow")
world <- fortify(countriesLow)

## Regions defined for each Polygons

map <- ggplot() + geom_polygon(data = world,
                                 aes(x=long, y=lat, group=group),
                                 color = "lightgrey", fill = "darkgrey")
xxlim <- expand_range(c(min(birds$Longitude),max(birds$Longitude)), add = 0.1)
yylim <- expand_range(c(min(birds$Latitude),max(birds$Latitude)), add = 0.1)

map +
  coord_map("ortho", xlim = xxlim, ylim = yylim, orientation=c(55, 10, 0)) +
  geom_point(data = birds,
             aes(Longitude, Latitude, size = log(Numeriekewaarde)), color = "blue", alpha = 0.5) +
  scale_size(range = c(0, 5)) +
  facet_wrap(~ Biotaxon.naam) +
  ggtitle(paste("")) +
  theme(axis.text = element_blank(),
        axis.title = element_blank(),
        axis.line = element_blank(),
        axis.ticks = element_blank())
```



Interactive map

```
#color scale
factpal <- colorFactor(topo.colors(40), birds$Biotaxon.naam)
#invoke leaflet map
leaflet() %>%
  addTiles(group = "OSM (default)") %>%
  addCircleMarkers(data=birds,
    popup = sprintf(as.character(paste(birds$Biotaxon.naam,":", round(birds$Numeriekewaarde, 0),
birds$Eenheid.code))), 
    # clusterOptions = markerClusterOptions(),
    radius = ~log(Numeriekewaarde)*2,
    color = ~factpal(Biotaxon.naam),
    stroke = F,
    fillOpacity = 0.7) %>%
  addLayersControl(options = layersControlOptions(collapsed = FALSE))
## Assuming 'Longitude' and 'Latitude' are longitude and latitude, respectively
```

(The interactive map does not appear in the wiki page, but will run when you run the R code.)