Using  for statistical analyses

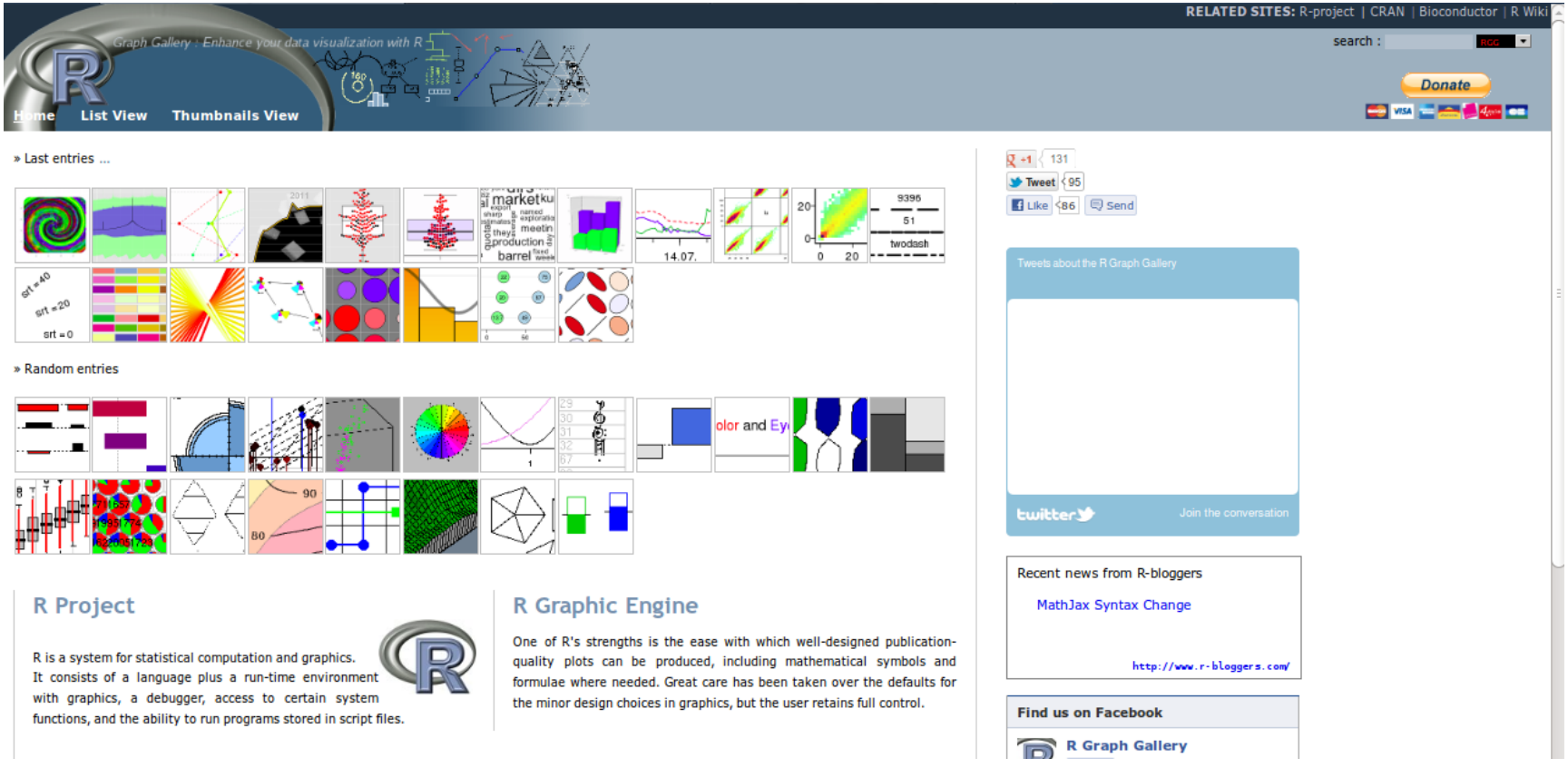
Robert Bauer

Warnemünde, 05/30/2012

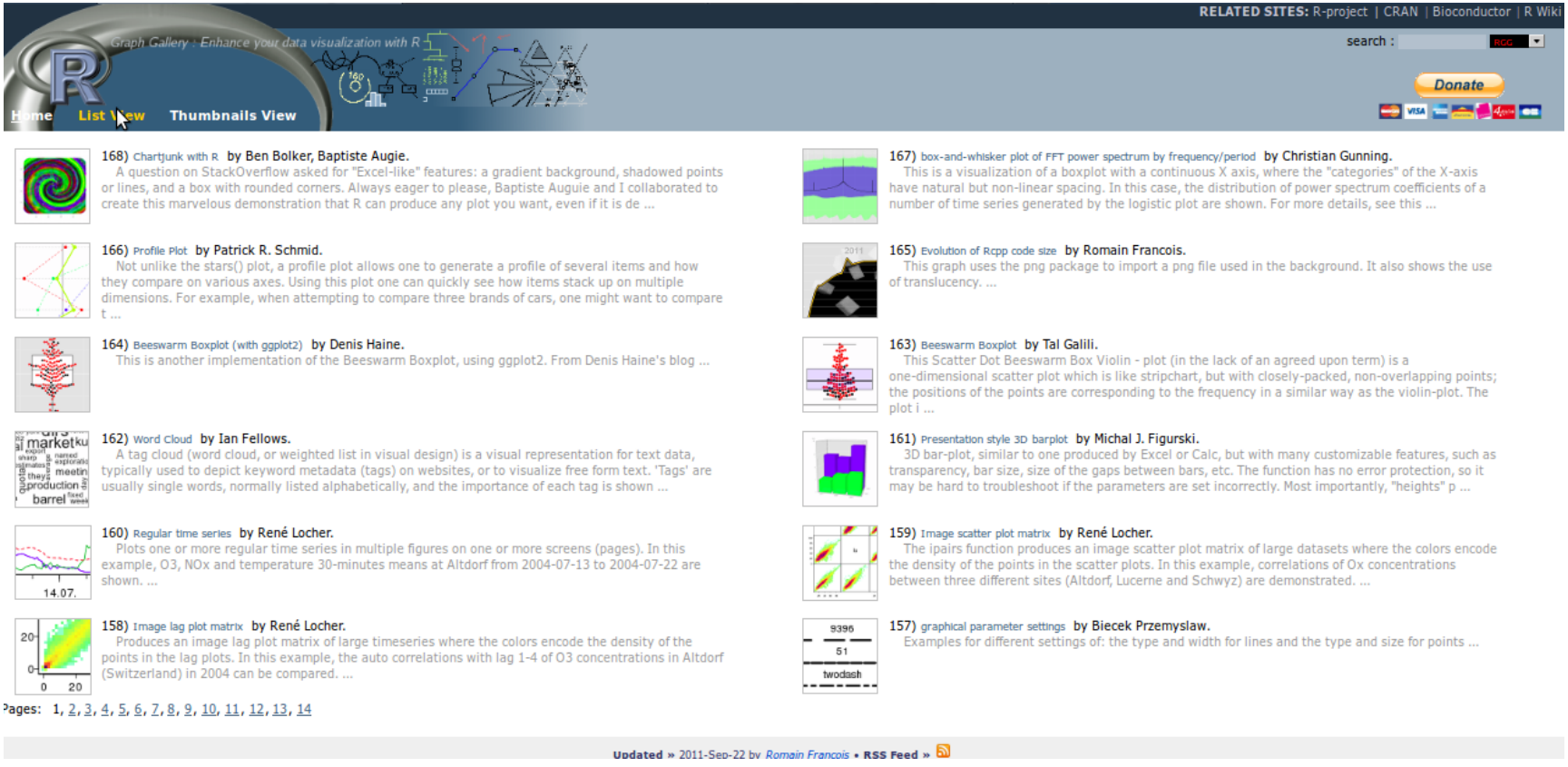
Day 7 - Agenda:

-  Graph Gallery
- wind rose plots
- matrices and arrays
- image plots

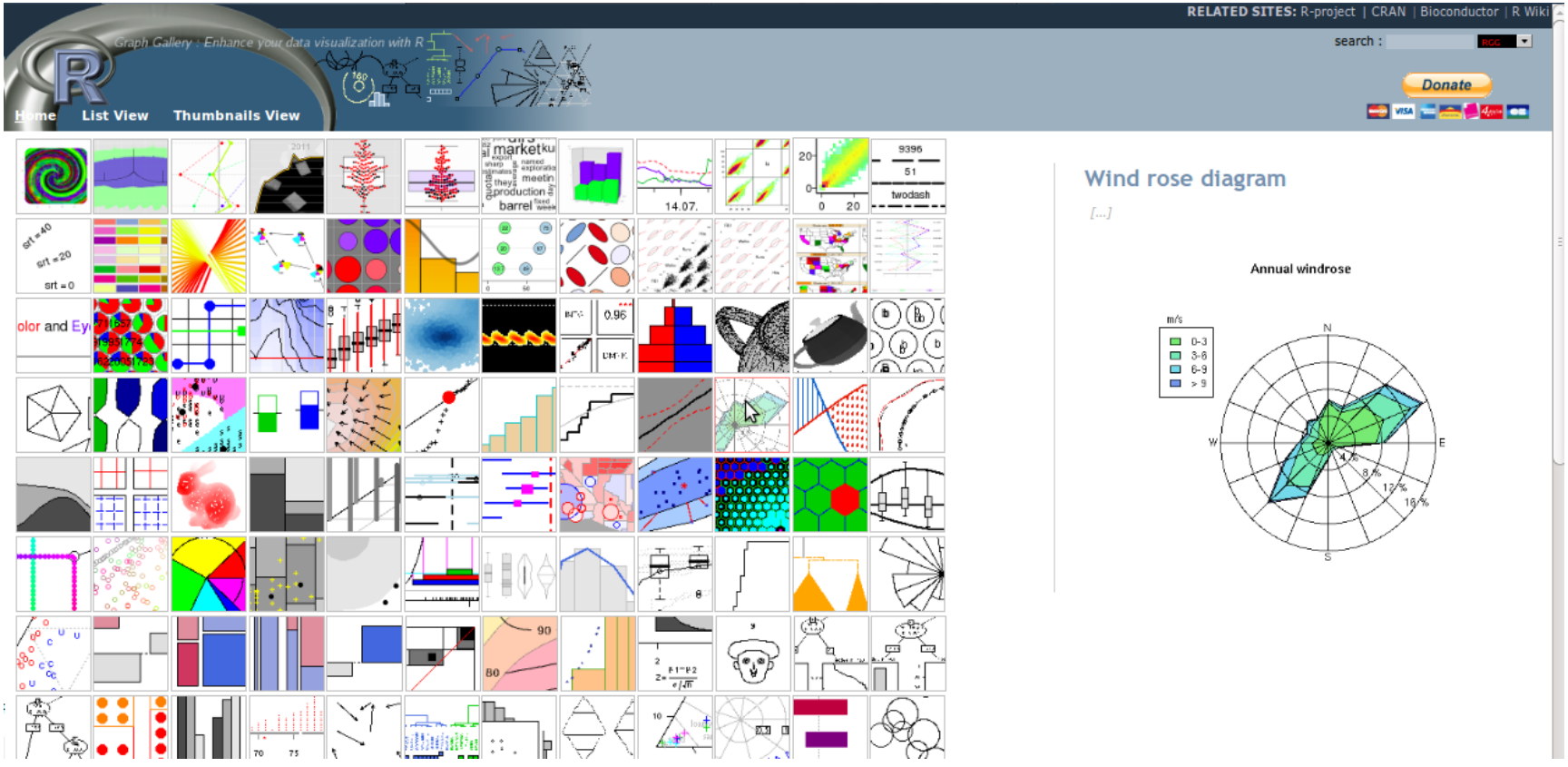
Graph Gallery



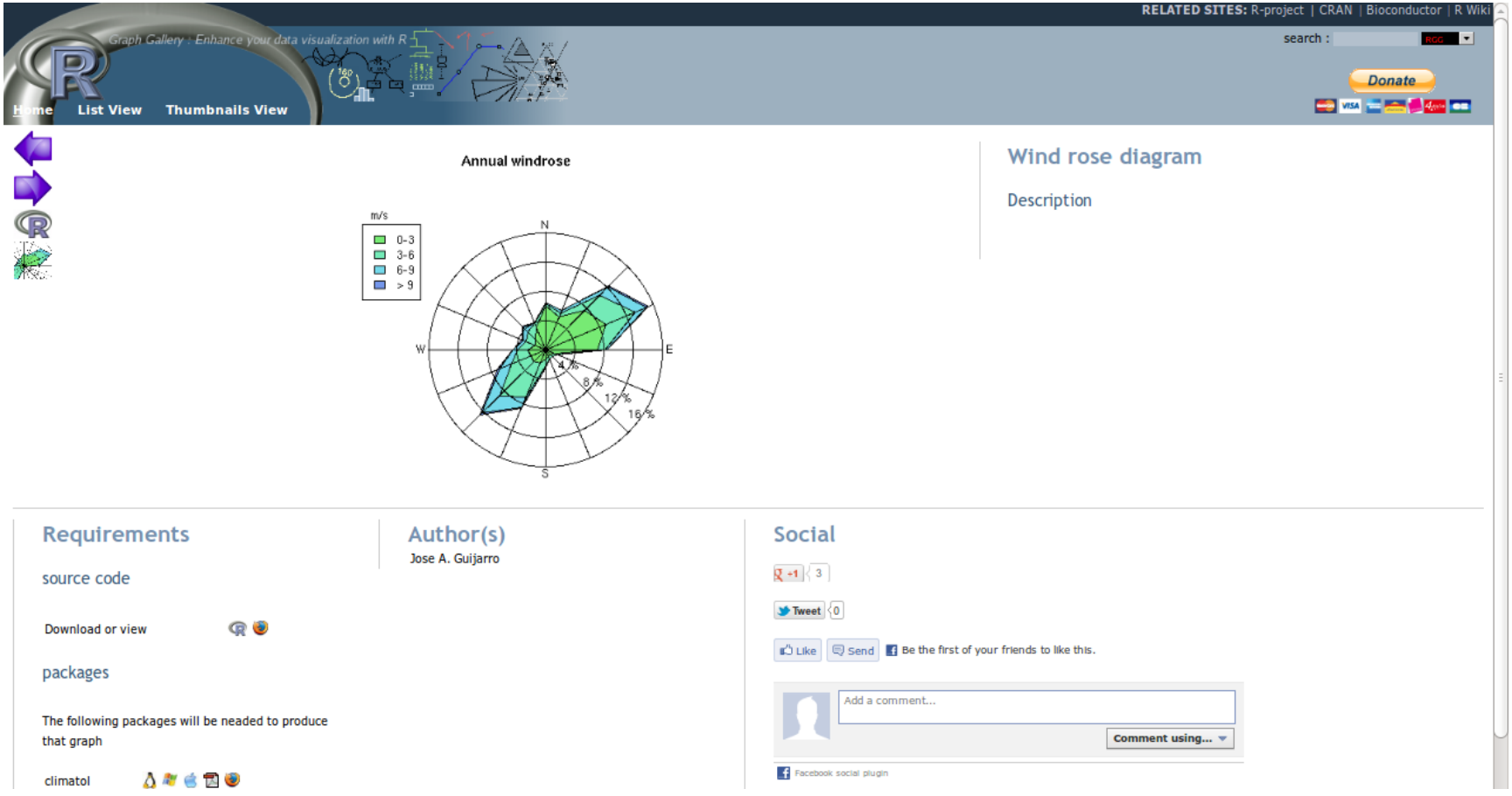
Graph Gallery



Graph Gallery



Graph Gallery



wind roses

```
require(climatol)
data(windfreq.dat)
rosavent(windfreq.dat,4,4,ang=-3*pi/16,main="Annual windrose")
```

```
install.packages("climatol")
require(climatol)
data("windfr")
rosavent(windfr,4,4,ang=-3*pi/16,main="Annual windrose")
```

wind roses

RStudio

File Edit View Project Workspace Plots Tools Help

day7_with_exercises.r × windfr ×

4 observations of 17 variables

	row.names	N	NNE	NE	ENE	E	ESE	SE	SSE	S	SSW	SW	WSW	W	WNW	NW	NNW
1	0-3	59	48	75	90	71	15	10	11	14	20	22	22	24	15	19	33
2	3-6	3	6	29	42	11	3	4	3	9	50	67	28	14	13	15	5
3	6-9	1	3	16	17	2	0	0	0	2	16	33	17	6	5	9	2
4	> 9	0	1	2	3	0	0	0	0	0	1	4	3	1	1	2	0

Workspace History

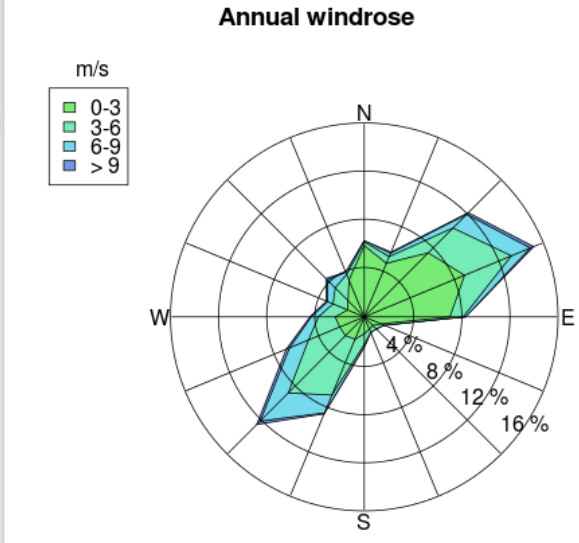
Load Save Import Dataset Clear All

Data

windfr 4 obs. of 16 variables

Files Plots Packages Help

Zoom Export Clear All



Console ~/Dropbox/R_course/day7/

```
> require(climatol)
Loading required package: climatol
> data("windfr")
> rosavent(windfr,4,4,ang=-3*pi/16,main="Annual windrose")
>
> View(windfr)
> View(windfr)
> head(windfr)
     N NNE NE ENE  E ESE SE SSE  S SSW SW WSW  W WNW NW NNW
0-3 59  48 75  90 71  15 10  11 14  20 22  22 24  15 19  33
3-6  3   6 29  42 11   3  4   3  9  50 67  28 14  13 15   5
6-9  1   3 16  17  2   0  0   0  2  16 33  17  6   5  9   2
> 9  0   1  2   3  0   0  0   0  0   1  4   3  1   1  2   0
> names(windfr)
 [1] "N"   "NNE" "NE"  "ENE" "E"   "ESE" "SE"  "SSE" "S"   "SSW" "SW"  "WSW" "W"   "WNW" "NW"
[16] "NNW"
>
```

wind roses

```
u <- windfr
names(u) <- 1:16 # seq(0,360-22.5, length.out=16)
rosavent(windfr,4,4,ang=-3*pi/16,main="Annual windrose")
```

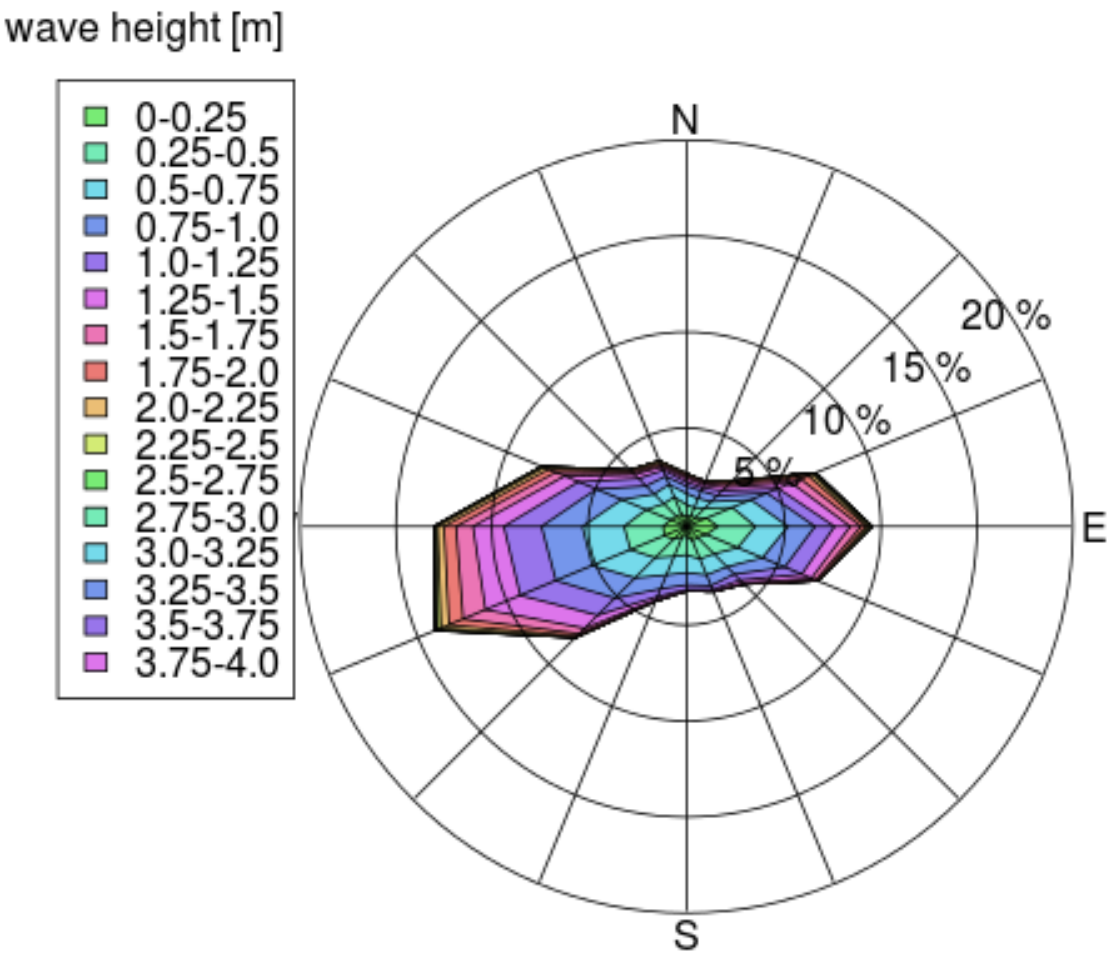
Exercise

1. Import data from “wave_height_frequencies.xls”
2. create wind rose from data
3. rename legend (hint: use the help command)

Exercise

```
data <- read.table('wave_height_frequencies.csv')
rosavent(data, uni="wave height [m]")
?rosavent
```

Exercise


wave height [m]
0-0.25
0.25-0.5
0.5-0.75
0.75-1.0
1.0-1.25
1.25-1.5
1.5-1.75
1.75-2.0
2.0-2.25
2.25-2.5
2.5-2.75
2.75-3.0
3.0-3.25
3.25-3.5
3.5-3.75
3.75-4.0
N
E
S
20 %
15 %
10 %

Exercise

```
data <- read.table('wave_height_frequencies.csv')
rosavent(data, uni="wave height [m]")
?rosavent
```

defining matrices

```
matrix(1:10,nrow=5,ncol=2) # matrix(data,rows,columns)
matrix(1:10,5,2)           # option 2
matrix(1:10,5)             # option 3
dim(m)                     # checking matrix dimensions
```

defining matrices

```
matrix(1:10,nrow=5,ncol=2) # matrix(data,rows,columns)
matrix(1:10,5,2)           # option 2
matrix(1:10,5)             # option 3
dim(m)                     # checking matrix dimensions
```

```
matrix(0,5,2)              # 5x2 matrix of zeros
matrix(1,5,2)              # 5x2 matrix of ones
matrix(NA,5,2)             # 5x2 matrix of NAs
```

defining matrices

```
matrix(1:10,nrow=5,ncol=2) # matrix(data,rows,columns)
matrix(1:10,5,2)           # option 2
matrix(1:10,5)             # option 3
dim(m)                     # checking matrix dimensions

# matrices with uniform values
matrix(0,5,2)              # 5x2 matrix of zeros
matrix(1,5,2)              # 5x2 matrix of ones
matrix(NA,5,2)             # 5x2 matrix of NAs
```

```
matrix(0,5)                # single column matrix

matrix(1:10,ncol=5,nrow=2) # changing arguments order
matrix(1:10,5,2,byrow=T)   # fill matrix by rows; default:
    byrow=F
```

defining matrices

```
matrix(1:10,nrow=5,ncol=2) # matrix(data,rows,columns)
```

```
## other options:
## a) aligning vectors
cbind(rep(0,5),rep(0,5))    # 5x2 matrix of zeros

## b) converting vectors
m <- 1:10
dim(m) <- c(5,2)

## c) arrays
array(1:10,dim=c(5,2))      # arrays
array(0, dim=c(5,2))        # 5x2 matrix of zeros
array(1:10, dim=c(5,2,3))   # multiple dimensions
```

matrix operations

```
m <- matrix(1:10,5,2)       # matrix(data,rows,columns)
m

## 1) changing values
## a) accessing specific elements
i <- c(2,4,8,10)
m[i]
m[i] <- 0
m

m[3,2] <- NA
m
m[8]
m[8] <- 100
m
```

matrix operations

```
## b) matrix wide operations
m <- matrix(rnorm(10),5,2) # matrix(data,rows,columns)
m
m <- round(m)
m

m*2
m*m  # element by element product
```

matrix operations

```
## applying functions
# apply(matrix, margin, fun, ...)
# margin = 1; --> apply functions per row
# margin = 2; --> apply functions per column
# margin = c(1,2); --> apply functions per row & column
apply(m, 2, mean)
apply(m, 2, max)
apply(m, 2, sort) # sorting values

## 2) transpose matrix (changing rows and columns)
t(m)

## 3) extend matrix
cbind(m,1:5)
n <- 1:5
cbind(m,n)
```

matrix operations

```
## 4) naming columns and vectors
colnames(m) <- paste("col",1:2, sep="")
rownames(m) <- paste("row",1:5, sep="")
m

# caution when accessing data!
m$col1 # works only on data frames
m[,1]
m <- data.frame(m)
m
str(m) # check structure
m$col1 # works only on data frames

# converting dataframes
m <- as.matrix(m)
str(m) # check structure
```

matrix operations

```
## 4) plotting matrices
m
dim(m) # 5x2 matrix
image(m)
```

matrix operations

```
## 4) plotting matrices
m
dim(m) # 5x2 matrix
image(m)

image(1:5,1:2,m) # change axes tick marks
# attention:
# plot from the lower left margin
# rows and columns are switched!
```

matrix operations

```
## 4) plotting matrices
m
dim(m) # 5x2 matrix
image(m)

image(1:5,1:2,m) # change axes tick marks
# attention:
# plot from the lower left margin
# rows and columns are switched!

# show indices
text(c(row(m)), c(col(m))-.25,
     paste("[",c(row(m)), ",",c(col(m)),"]", sep=""))
```

matrix operations

```
## 4) plotting matrices
m
dim(m) # 5x2 matrix
image(m)

image(1:5,1:2,m) # change axes tick marks
# attention:
# plot from the lower left margin
# rows and columns are switched!

# show indices
text(c(row(m)), c(col(m))-.25,
     paste("[",c(row(m)), ",",c(col(m)),"]", sep=""))

# show values
text(c(row(m)), c(col(m)), m)
```

how to plot a matrix

```
# set figure margins
par(mar=c(10,5,5,6)) # mar=c(bottom, left, top, right);
    default:c(5, 4, 4, 2) + 0.1.

# start plotting procedure
image(1:5,1:2,m, axes=FALSE) # plot know axes!

# add axes & box
axis(1,at=1:5,lab=2:6)
axis(2,at=1:2,lab=6:5)
box()
```

```
# show indices
text(c(row(m)), c(col(m))-.25,
     paste("[",c(row(m)), ",",c(col(m)),"]", sep=""))

# show values
text(c(row(m)), c(col(m)), m)
```

```
# add colorbar
# install.packages("fields")
library(fields)
image.plot(m, legend.only=TRUE, col = heat.colors(12))
```

how to plot a matrix

```
datasheet <- read.table('SST_data.csv', header=F, sep=',',
    dec=".")
head(datasheet)
z <- t(as.matrix(datasheet))
x <- 1:dim(z)[1]
y <- 1:dim(z)[2]
colorbar.colors <- tim.colors(64)

# set figure margins
par(mar=c(10,5,5,6)) # mar=c(bottom, left, top, right);
    default:c(5, 4, 4, 2) + 0.1.

# start plotting procedure
image(x, y, z[,180:1], col=colorbar.colors,
      xlab='time', ylab='latitude', main="average monthly
          SST from 2000-2010 at 335Â°E", axes=F)

dates <- paste(7, '/', 2000:2010, sep="")
axis(1,seq(7,length(x),12),dates)
axis(2,seq(1,180,44.5),c(-90,45,0,45,90))
box()
image.plot(zlim=range(z), legend.only=TRUE, col=colorbar.
    colors)
```

Exercise

1. Calculate the min, mean, median, max SST of each latitude
2. create an image plot from the wind rose data of Exercise 1
   2.1 add a colorbar and contour lines

Exercises

```
# 1. Calculate the min, mean, median, max SST of each
    latitude
apply(z,2,min)
apply(z,2,mean)
apply(z,2,median)
apply(z,2,max)
apply(z,2,range)

colMeans(z)
apply(z,2,quantile, probs = c(0.25, 0.5, 0.75))
```