Missing data

- Missing data should be coded as blank cells
- Reported as "NA"
- Function `is.na()` can be used to look for missing data, returns TRUE/FALSE

```
> is.na(temp)
[1,] FALSE FALSE FALSE FALSE FALSE
[2,] FALSE FALSE FALSE FALSE FALSE
[3,] FALSE FALSE TRUE FALSE TRUE
[4,] FALSE FALSE TRUE FALSE TRUE
[5,] FALSE FALSE FALSE FALSE FALSE
```

Missing Data

- Some functions do not expect missing data
  - `mean(temp[,])`
    - [1] 0.5867246
  - `mean(temp[,3])`
    - [1] NA
  - `mean(temp[,3], na.rm = T)`
    - [1] 0.5134151

- How much missing data?
  - `sum(is.na(temp))`
    - 5

- Total missing data by row
  - `apply(is.na(temp), 1, sum)`
    - [1] 0 0 2 1 1 0 0 1 0

As we did before, you could exclude rows or columns based on the amount of missing data.

Functions handle missing data differently...be aware

- The `daisy` and `dist` functions (cluster and base package) deal with missing data by excluding those observations and re-weighing other variables.
- `hclust` function does not handle missing data.

This will generate an error:

```
> hclust(scaled_morph, method = "average")
```

This will work:

```
> euc_dist <- daisy(scaled_morph, metric = "euclidean")
> hclust(euc_dist, method = "average")
```

Geographic Distance

- With longitude and latitude variables, your geographic distance (Euclidean) can be calculated using the `dist` function.
- That assume the Earth is flat, and returns values in decimal degrees units.
- Function `earth.dist` corrects for this and returns values in km.
- For this dataset, the two are correlated (0.989)
Geographic Distance

- Another function is `distGeo` (geosphere package) that is supposed to be more accurate, allows you to specify ellipsoid parameters.
- Correlation = 0.9999
- Maximum discrepancy for the dataset was 5.7 km

Converting to/from square/triangular matrix

- Sometimes you need to import a distance matrix from other software.
- A square similarity matrix (mirrored above and below the diagonal) can be converted to a distance matrix using `as.dist`.

```
distances <- as.dist(distances)
distances
```

- A distance matrix can be converted into a square using `as.matrix`.
- Will fill in 0 along the diagonal, mirror the top and bottom halves.

```
as.dist(square)
```

More about Mapping

```
library(maps)
library(mapdata)
# North American fish example
nafish<-read.table("NA_fish_diversity.csv",header=T,sep="",row.names="HUC_8")
map("state", fill=F)
points(nafish$lat,nafish$long,cex=0.5,col="red",pch=19)
```

```
as.matrix(triangular)
```

![Map of North America](image)
More about Mapping

```r
div_bubble <- round(as.data.frame(div_bubble))
div_bubble$div_bubble <- as.numeric(div_bubble$div_bubble)
pdate <- format(div_bubble$div_bubble, digits = 2)
```

Adding shapefiles to maps

- Package `maptools` contains functions to import and work with GIS data.
- Function `readShapeSpatial` (and other `readShape` functions) will import standard file formats.
- Downloaded HUC shapefiles for MS.

```r
nafish <- read.csv("NA_fish_diversity.csv", header = T, sep = ",", row.names = "HUC_8")
HUCs <- readShapePoly("wbdhu8_a_ms.shp")
MS_fish_huc <- nafish[match(HUCs$HUC8, na_huc_names),]
plot(HUCs, col = HUC_colors)
text(MS_fish_huc$lat, MS_fish_huc$long, label = MS_fish_huc$diversity, col = "gray", cex = 0.75)
map("state", fill = F, col = "white")
```

Hist function

```r
hist(rnorm(10000))
hist(rnorm(10000), breaks = 50, axes = FALSE)
hist(rnorm(10000), plot = FALSE)
```

Hist function

```r
var1 <- rnorm(1000, mean = 1)
var2 <- rnorm(1000, mean = 2)
hist(var1, col = "blue")
hist(var2, col = "red", add = T)
hist(var1, col = rgb(0, 0, 1, 0.5), main = "var1 and var2")
hist(var2, col = rgb(1, 0, 0, 0.5), add = T)
```
Sequences of Colors

color_func=colorRampPalette(c("blue","red"))
color_func()

color_func=colorRampPalette(c(rgb(.5,0,1),rgb(.0,0,1)),alpha=T)
color_func()

color_func=colorRampPalette(c("blue","red"),alpha=T)
color_func(10)