---

title: "Warden and Gore waterbird analyses 2020-2021"

author: "Adrian Pinder"

date and time: '`r Sys.time()`'

output:

html\_document:

toc: true

toc\_depth: 4

software: 'RStudio: Version 1.3.1093 – © 2009-2020 RStudio, Inc. R version: `r getRversion()`'

editor\_options:

chunk\_output\_type: console

---

Git repository https://github.com/AdrianMP62/Warden-2019

RStudio: Version 1.1.463 – © 2009-2018 RStudio, Inc. R version: `r getRversion()`

Date and time: `r Sys.time()`

Uses the following datafiles:

\* Warden\_Gore\_by\_suite\_06\_20.csv (System, SurveyType, SiteName, SiteCode, Subsite Date, CommonName, SpeciesCode, Count) (e.g. Warden, aerial, Neridup Suite, WRP001, combined, 11/10/2006, Great Egret, greg, 1). Note some double counts for Oct 2006 and Feb 2015 removed.

\* Average depths.csv (average depths, rainfall variables and pipeline operation and timing of surveys of Warden wetlands for each survey period).

\* Taxonomy (CommonName, Order and informal group name in consecutive columns. CommonNames match those in Warden\_Gore\_by\_suite\_06\_19.csv)

Date and time: `r Sys.time()`

```{r setup, echo=FALSE}

#knitr::opts\_chunk$set(echo = FALSE)

#knitr::opts\_chunk$set(message = FALSE)

#knitr::opts\_chunk$set(include = FALSE)

knitr::opts\_knit$set(root.dir = '../')

```

```{r}

options(scipen=999) # prevent-scientific notation for numbers

```

```{r}

set.seed(1234)

```

```{r}

Packages <- c("gridExtra","vegan", "tidyr", "ggrepel","ggplot2", "dplyr", "plyr", "scales", "ggpubr", "stringr","reshape2", "knitr", "zoo", "car","reshape2","readxl","lubridate")

lapply(Packages, library, character.only = TRUE)

source("./functions/bio\_env\_ext.R") #The Bio\_env and bv\_step\_ext routines below comes from http://menugget.blogspot.com.au/2011/06/clarke-and-ainsworths-bioenv-and-bvstep.html

source("./functions/bv\_step\_ext.R")

source("./functions/bv.step.R")

source("./functions/extract\_legend.R")

```

#Analyses of waterbird data from the Warden and Gore Ramsar wetlands 2006 to 2019

The following is the output from an Rmarkdown R script for analyses of waterbird communities of the Warden and Gore wetland systems surveyed between Oct 2006 and Feb 2021.

These analyses are based primarily on ground counts. Aerial counts are only for a few Warden system wetlands in 2006 to 2009 where ground counts were not undertaken. These include Neridup and Bandy Creek (Oct 2006-Nov 2008), Ewans (Oct 2006) and Ewans and Mullet combined (Nov 2008 - have to ignore Nov 2008 Mullet ground counts), North Wheatfield (Oct 2006, Oct 2007, Nov 2008). Note that the Nov 2008 aerial data was collected two weeks prior to the ground counts. Gore data is included only for November 2009 onwards when ground counts are available. Some replicate count data for Oct 2006 and Feb 2015 (North Wheatfield) are excluded from the dataset above prior to import. Pink Lake is excluded from analyses because it wasnt consistently surveyed. Birds counted only while in air and not allocated to any particular wetland are given the site code 'In flight'. These conditions apply to all following analyses.

##IMPORT DATA AND FORMAT COLUMNS

```{r}

#create data table with unidentified birds included - for column plots of

input.data <- read.csv("./raw\_data/Warden\_Gore\_by\_suite\_06\_21.csv")

input.data <- input.data[-grep("Grassbird", input.data$CommonName), ] #remove records of little grassbird

input.data$Date <- as.Date(input.data$Date, format="%d/%m/%y")

input.data$Survey <- input.data$Date #create Survey field from Date

input.data$Survey <- format(as.Date(input.data$Survey), "%b-%Y") #change format to mmm-YYYY

input.data$Survey <- gsub("Oct-2018", "Nov-2018", input.data$Survey) #lump survey dates that straddle Oct/Nov 2018 and call Nov-2018

yq <- as.yearqtr(as.yearmon(input.data$Date, "%Y/%m/%d"))

input.data$Season <- factor(format(yq, "%q"), levels = 1:4, labels = c("summer", "autumn", "winter", "spring")) #create Season (spring vs summer) variable

input.data$Sys\_Sur <- paste(input.data$System, input.data$Survey, sep = " ") # combine System and date to create one 'sample' variable for mds - i.e. combine system with season

input.data <- input.data[-grep("No birds", input.data$CommonName), ] #remove rows where wetland surveyed but no birds present or where wetlands dry

input.data <- input.data[-grep("No data collected", input.data$CommonName), ] #remove rows where wetland not surveyed

#import taxonomic nomenclature table

input.data.2 <- input.data[-grep("Unidentified", input.data$CommonName), ] #remove rows where birds not identified and save as a separate file

#import taxonomic nomenclature table

Taxonomy <- read.csv("./raw\_data/Taxonomy.csv")

```

###Totals of species x year for Warden+gore

```{r}

input.data.2$Year <- year(input.data.2$Date)

WG\_SxY <- as.matrix(xtabs(input.data.2$Count ~ input.data.2$CommonName + input.data.2$Year))

write.csv(WG\_SxY, "./outputs/WG\_SxY.csv")

```

##MULTI-VARIATE ANALYSIS OF WARDEN AND GORE DATA COMBINED, BY SEASON

```{r}

WG.bySeason <- input.data.2

```

###Aggregate data by Season and create matrix

```{r}

WG.bySeason.agg <- aggregate(data=WG.bySeason, Count ~ CommonName + Survey + Season, FUN="sum") #sum data by common name and survey period

```

#### Create matrix for ordination

```{r}

WG.bySeason.m <- spread(WG.bySeason.agg, CommonName, Count)

rownames(WG.bySeason.m) <- WG.bySeason.m[,1]

WG.bySeason.m[,1] <- NULL

WG.bySeason.m[is.na(WG.bySeason.m)] <- 0

WG.bySeason.m [,1] <- NULL #remove season column

WG.bySeason.m[, 1-75] <- sqrt(WG.bySeason.m[, 1-75])

```

###Extract survey and season variables for ordination plots

```{r}

surv.seas <- unique(WG.bySeason[c("Survey", "Season")]) #extract a unique list of survey codes (mmm-yy) and season so can later allocate a season to each survey in the MDS output

surv.seas <- surv.seas[order(surv.seas$Survey), ] #reorder alphabetically by Survey (Dec-2011 to Oct-2012)

```

###Run 2D ordination

```{r}

WG.MDS.bySeas <- metaMDS(WG.bySeason.m, distance = "bray")

```

####Extract ordination axis coordinates

```{r}

WG.MDS.bySeas.pts <- as.data.frame(WG.MDS.bySeas$points) #extract 2D ordination coordinates (points)

WG.MDS.bySeas.pts$Survey <- row.names(WG.MDS.bySeas.pts) #use survey period as row.names

WG.MDS.bySeas.pts$Season <- surv.seas[match(WG.MDS.bySeas.pts$Survey, surv.seas$Survey), 2] #adds 'season' (spring versus summer) (from column '2' of Surv.Seas) to ordination points by matching survey between points and surv.seas

```

####2D ordination plot of Warden and Gore data combined by season.

The ordination below shows that, by and large, with data from systems (Warden and Gore) combined, there is a difference between waterbird communities present in spring and those present in late summer.

```{r}

Xmin <- min(WG.MDS.bySeas.pts$MDS1)\*1.2

Xmax <- max(WG.MDS.bySeas.pts$MDS1)\*1.2

Ymin <- min(WG.MDS.bySeas.pts$MDS2)\*1.2

Ymax <- max(WG.MDS.bySeas.pts$MDS2)\*1.2

ggplot(WG.MDS.bySeas.pts, aes(x=MDS1, y=MDS2)) + xlim(Xmin,Xmax) + ylim(Ymin,Ymax) + geom\_point(aes(colour=Season), size=5) + coord\_fixed(ratio = 1) + geom\_text(aes(label=Survey), size=3, vjust=1, hjust=1.3)

```

However, the stress value of the above ordination is high at `r WG.MDS.bySeas$stress`, so a 3D ordination was also produced.

#### 3D ordination

```{r}

WG.3dMDS.bySeas <- metaMDS(WG.bySeason.m, distance = "bray", k=3)

```

####Extract ordination axis coordinates

```{r}

WG.3dMDS.bySeas.pts <- as.data.frame(WG.3dMDS.bySeas$points) #extract 3D ordination coordinates (points)

WG.3dMDS.bySeas.pts$Survey <- row.names(WG.3dMDS.bySeas.pts) #use survey period as row.names

WG.3dMDS.bySeas.pts$Season <- surv.seas[match(WG.3dMDS.bySeas.pts$Survey, surv.seas$Survey), 2] #adds 'season' (spring versus summer) (from column '2' of Surv.Seas) to ordination points by matching survey between points and surv.seas

```

####3D ordination plot of Warden and Gore data combined by season, stress = `r WG.3dMDS.bySeas$stress`.

```{r}

Xmin <- min(WG.3dMDS.bySeas.pts$MDS1)\*1.3

Xmax <- max(WG.3dMDS.bySeas.pts$MDS1)\*1.3

Ymin <- min(WG.3dMDS.bySeas.pts$MDS2)\*1.3

Ymax <- max(WG.3dMDS.bySeas.pts$MDS2)\*1.3

ggplot(WG.3dMDS.bySeas.pts, aes(x=MDS1, y=MDS2)) + xlim(Xmin, Xmax) + ylim(Ymin, Ymax) + geom\_point(aes(colour=Season), size=5) + coord\_fixed(ratio = 1) + geom\_text(aes(label=Survey), size=3, vjust=1, hjust=1.3) + theme (plot.margin=unit(c(0,0,0,5),"mm"))

```

####A 'Simper' analysis to extract a short-list of species most responsible for the differences between seasons

```{r}

WG.bySeas.Simper <- simper(WG.bySeason.m, surv.seas$Season) #Simper analysis to determine which species best correlated with differences between seasons of whole dataset

```

####Extract top 8 species names from the Simper analysis

```{r}

WG.Simper.ext <- WG.bySeas.Simper$spring\_summer #extract results from Simper analysis

WG.Sim.Average <- as.data.frame(WG.Simper.ext$average) #extract the average difference data

WG.Sim.Spec <- as.data.frame(WG.Simper.ext$species) #extract species names

WG.bySeas.Simper2 <- cbind(WG.Sim.Spec, WG.Sim.Average) #combine species with averages

WG.bySeas.Simper2 <- WG.bySeas.Simper2[ order(-WG.bySeas.Simper2[, 2]), ] #order species and averages by average

colnames(WG.bySeas.Simper2) = c("Species", "cont") #add new column names

WG.bySeas.Simper2 <- as.vector(WG.bySeas.Simper2[1:8, 1]) #extracts top 8 species names

```

####Aggregate species x site listing by season, survey and common name to later extract data for just the 8 species

```{r}

agg.spec.data <- aggregate(data=WG.bySeason, Count ~ CommonName + Season + Survey, FUN="sum") #aggregate counts by common name, season and survey

#agg.spec.data <- as.data.frame(sapply(agg.spec.data, gsub, pattern = "-", replacement = " ")) #replace dashes with spaces

```

####Extract top 8 species from aggregated data

```{r}

agg.spec.data.8 <- agg.spec.data[agg.spec.data$CommonName %in% WG.bySeas.Simper2, ]

rownames(agg.spec.data.8) <- NULL #reset row names

agg.spec.data.8 <- as.data.frame(agg.spec.data.8[, -3]) #remove Survey field

agg.spec.data.8$CommonName <- as.character(agg.spec.data.8$CommonName)

write.csv(agg.spec.data.8, "./outputs/agg.spec.data.8.csv")

agg.spec.data.8 <- read.csv("./outputs/agg.spec.data.8.csv")

```

####box plots

These box plots show the abundance of the top 8 species contributiong to waterbird communities being different in spring and summer (Gore and Warden systems combined).

```{r}

qplot(x=Season, y=Count, data=agg.spec.data.8, geom="boxplot") + facet\_wrap(~ CommonName, scales="free\_y", ncol=2)

```

##ORDINATION OF GORE SYSTEM VERSUS WARDEN SYSTEM

```{r}

WG.bySystem <- input.data.2

```

#### Aggregate data by species and system/survey

```{r}

WG.bySystem.agg <- aggregate(data=WG.bySystem, Count ~ CommonName + Sys\_Sur, FUN="sum")

```

#### Create matrix for ordination (uses spread function from package tidyr)

```{r}

WG.bySystem.m <- spread(WG.bySystem.agg, CommonName, Count)

rownames(WG.bySystem.m) <- WG.bySystem.m[,1]

WG.bySystem.m[,1] <- NULL

WG.bySystem.m[is.na(WG.bySystem.m)] <- 0

WG.bySystem.m[, 1-75] <- sqrt(WG.bySystem.m[, 1-75])

```

#### MDS unconstrained ordination using metaMDS in package vegan to compare Warden and Gore wetland systems

```{r}

WG.MDS.bySyst <- metaMDS(WG.bySystem.m, try=100, distance = "bray")

```

#### Extract coordinates from ordination and add season to data frame for Warden vs Gore analysis

```{r}

system <- unique(input.data.2$System) #extract a unique list of systems (Gore, Warden)

WG.MDS.bySyst.pts <- as.data.frame(WG.MDS.bySyst$points) #extract coordinates from metaMDS

WG.MDS.bySyst.pts$Sys\_Sur <- row.names(WG.MDS.bySyst.pts) #use row.names to create new System+Survey period variable

WG.MDS.bySyst.pts$Survey <- str\_sub(WG.MDS.bySyst.pts$Sys\_Sur, -8, -1) #extract Survey period from Sys-Sur variable by extracting last 8 characters

WG.MDS.bySyst.pts$System <- substr(WG.MDS.bySyst.pts$Sys\_Sur, 1, 4) #extract System from Sur-sys variable by extracting first 4 variables

WG.MDS.bySyst.pts$System <- gsub("Ward", "Warden", WG.MDS.bySyst.pts$System) #convert 'Ward' to 'Warden'

#WG.MDS.bySyst.pts <- WG.MDS.bySyst.pts[with(WG.MDS.bySyst.pts, order(Survey)), ] #order by survey

WG.MDS.bySyst.pts$Season <- substr(WG.MDS.bySyst.pts$Survey, 1, 3) #extract months

WG.MDS.bySyst.pts$Season <- recode(WG.MDS.bySyst.pts$Season, "'Feb'='summer';c('Dec','Oct','Nov')='spring'") #convert months to season

WG.MDS.bySyst.pts$SystSeas <- paste(WG.MDS.bySyst.pts$System, WG.MDS.bySyst.pts$Season, sep=" in ")

str(WG.MDS.bySyst.pts) #join System and season

```

#### Ordination plot with symbols coloured by system (Warden versus Gore), `r WG.MDS.bySyst$stress`

```{r}

ggplot(WG.MDS.bySyst.pts, aes(x=MDS1, y=MDS2, shape=SystSeas, colour=SystSeas)) + xlim(min(WG.MDS.bySyst.pts$MDS1)\*1.3, max(WG.MDS.bySyst.pts$MDS1)\*1.3) + ylim(min(WG.MDS.bySyst.pts$MDS2)\*1.3, max(WG.MDS.bySyst.pts$MDS2)\*1.3) + coord\_fixed(ratio = 1) + geom\_point(size=5) + scale\_color\_manual(values=c("blue","blue","red","red")) + scale\_shape\_manual(values=c(0,15,2,17)) + geom\_text(aes(label=Survey), colour="black", size=3, vjust=1, hjust=1.3)

```

However, the stress level of the above two-dimensional ordintion is unacceptably high at `r WG.MDS.bySyst$stress`. A three-dimensional ordination was therefore undertaken.

#### 3D MDS unconstrained ordination using metaMDS in package vegan to compare Warden and Gore wetland systems

```{r}

WG.3dMDS.bySyst <- metaMDS(WG.bySystem.m, distance = "bray", k=3, transform="false")

```

#### Extract coordinates from 3D ordination and add season to data frame for Warden vs Gore analysis

```{r}

system <- unique(input.data.2$System) #extract a unique list of systems (Gore, Warden)

WG.3dMDS.bySyst.pts <- as.data.frame(WG.3dMDS.bySyst$points) #extract coordinates from metaMDS

WG.3dMDS.bySyst.pts$Sys\_Sur <- row.names(WG.3dMDS.bySyst.pts) #use row.names to create new System+Survey period variable

WG.3dMDS.bySyst.pts$Survey <- str\_sub(WG.3dMDS.bySyst.pts$Sys\_Sur, -8, -1) #extract Survey period from Sys-Sur variable by extracting last 8 characters

WG.3dMDS.bySyst.pts$System <- substr(WG.3dMDS.bySyst.pts$Sys\_Sur, 1, 4) #extract System from Sur-sys variable by extracting first 4 variables

WG.3dMDS.bySyst.pts$System <- gsub("Ward", "Warden", WG.3dMDS.bySyst.pts$System) #convert 'Ward' to 'Warden'

#WG.3dMDS.bySyst.pts <- WG.3dMDS.bySyst.pts[with(WG.3dMDS.bySyst.pts, order(Survey)), ] #order by survey

WG.3dMDS.bySyst.pts$Season <- substr(WG.3dMDS.bySyst.pts$Survey, 1, 3) #extract months

WG.3dMDS.bySyst.pts$Season <- recode(WG.3dMDS.bySyst.pts$Season, "'Feb'='summer';c('Dec','Oct','Nov')='spring'") #convert months to season

WG.3dMDS.bySyst.pts$SystSeas <- paste(WG.3dMDS.bySyst.pts$System, WG.3dMDS.bySyst.pts$Season, sep=" in ")

str(WG.3dMDS.bySyst.pts) #join System and season

```

### plot of 3D Warden vs Gore by season ordination

The ordination plot below shows that the Warden and Gore wetland systems consistently support different waterbird communities, i.e. for waterbirds they have different conservation values. It can also be seen that there is greater seasonal differences in the Warden system than in the Gore system (i.e. greater overlap between positions of the summer and spring surveys amongst the Gore surveys).

the Nov 2018 and Feb 2019 surveys are within the range of community compositions surveyed since 2006 for both systems, despite the very high counts in Feb 2019 and low Warden count in Nov 2018.

```{r}

ggplot(WG.3dMDS.bySyst.pts, aes(x=MDS1, y=MDS2, shape=SystSeas, colour=SystSeas)) + xlim(min(WG.3dMDS.bySyst.pts$MDS1)\*1.3,max(WG.3dMDS.bySyst.pts$MDS1)\*1.3) + ylim(min(WG.3dMDS.bySyst.pts$MDS2)\*1.3,max(WG.3dMDS.bySyst.pts$MDS2)\*1.3) + coord\_fixed(ratio = 1) + geom\_point(size=5) + scale\_color\_manual(values=c("blue","blue","red","red")) + scale\_shape\_manual(values=c(0,15,2,17)) + theme(legend.title = element\_blank()) + theme(legend.position = "top") + theme(legend.text=element\_text(size=12)) + theme(plot.margin=unit(c(0,0,0,0),"mm")) + geom\_text\_repel(aes(label=Survey), colour="grey", size=5, vjust=1, hjust=1.5)

```

```{r new-plot-for-Birdlife-talk}

# ggplot(WG.3dMDS.bySyst.pts, aes(x=MDS1, y=MDS2, shape=SystSeas, colour=SystSeas)) + xlim(min(WG.3dMDS.bySyst.pts$MDS1)\*1.3,max(WG.3dMDS.bySyst.pts$MDS1)\*1.3) + ylim(min(WG.3dMDS.bySyst.pts$MDS2)\*1.3,max(WG.3dMDS.bySyst.pts$MDS2)\*1.3) + coord\_fixed(ratio = 1) + geom\_point(size=5) + scale\_color\_manual(values=c("black","black","black","black")) + scale\_shape\_manual(values=c(15,15,15,15)) + theme(legend.title = element\_blank()) + theme(legend.position = "top") + theme(legend.text=element\_text(size=12)) + theme(plot.margin=unit(c(0,0,0,0),"mm")) + theme(axis.title.x=element\_blank(), axis.text.x=element\_blank(), axis.ticks.x=element\_blank()) + theme(axis.title.y=element\_blank(), axis.text.y=element\_blank(), axis.ticks.y=element\_blank()) + theme(legend.title = element\_blank()) + theme(legend.position = "none")

```

#### Simper analysis to extract species separating Warden versus Gore

```{r}

WG.bySyst.Sim <- simper(WG.bySystem.m, WG.3dMDS.bySyst.pts$System) #Simper analysis to determine which species best correlated with differences between systems

summary(WG.bySyst.Sim, ordered=TRUE)

```

#### Extract the top 10 species from the Simper analysis

```{r}

WG.bySyst.Sim.ext <- WG.bySyst.Sim$Gore\_Warden #extract name and contribution data from simper analysis

WG.bySyst.Sim.average <- as.data.frame(WG.bySyst.Sim.ext$average)#extract average contribution values

WG.bySyst.Sim.spec <- as.data.frame(WG.bySyst.Sim.ext$species)#extract species names

WG.bySyst.Sim.ext <- cbind(WG.bySyst.Sim.spec, WG.bySyst.Sim.average) #combine average with species name

WG.bySyst.Sim.ext <- WG.bySyst.Sim.ext[ order(-WG.bySyst.Sim.ext[,2]), ] #order by average value

colnames(WG.bySyst.Sim.ext) = c("Species", "cont") #add new column names

WG.bySyst.Sim.ext <- as.vector(WG.bySyst.Sim.ext[1:10, 1]) #extract top 10 species: first 10 rows

#WG.bySyst.Sim.ext <- gsub("\\.", " ", WG.bySyst.Sim.ext)

WG.bySyst.Sim.ext

```

#### Extract count data for top 10 species

```{r}

agg.spec.data.2 <- aggregate(data=WG.bySystem, Count ~ CommonName + Survey + System, FUN="sum") #aggregare count data by system and survey

#agg.spec.data.2 <- as.data.frame(sapply(agg.spec.data.2, gsub, pattern = "-", replacement = " ")) #replace dashes in species names with spaces to match simper output

agg.spec.data.2 <- agg.spec.data.2[agg.spec.data.2$CommonName %in% WG.bySyst.Sim.ext, ] #restrict aggregated counts to just those top 8 species from simper

rownames(agg.spec.data.2) <- NULL

agg.spec.data.2$CommonName <- as.character(agg.spec.data.2$CommonName)

colnames(agg.spec.data.2)[4] <- "Abundance"

write.csv(agg.spec.data.2, "./outputs/agg.spec.data.2.csv")

agg.spec.data.2 <- read.csv("./outputs/agg.spec.data.2.csv")

```

#### Box plots for Warden v Gore for the top 10 simper species

```{r}

qplot(x=System, y=Abundance, data=agg.spec.data.2, geom="boxplot") + facet\_wrap(~ CommonName, scales="free\_y", ncol=2) + theme(axis.title.y = element\_text(hjust=0.5)) #produce facet wrapped box plots showing differences in top 10 simper species by wetland system

```

#### Undertake Mann-Whitney for each of the 10 top Simper species and combine results into file called ttest.results<p/>

```{r}

WG.bySyst.Sim.ext <- WG.bySyst.Sim.ext[-8]

WG.bySyst.Sim.ext

ttest.v <- vector()

for (i in 1:9) {

tt.data <- subset(agg.spec.data.2, CommonName == WG.bySyst.Sim.ext[i])

print(ttest <- wilcox.test(log10(Abundance) ~ System, data = tt.data))

ttest.v <- c(ttest.v, ttest$p.value)

}

ttest.v

ttest.results <- data.frame(CommonName=WG.bySyst.Sim.ext, p.value=ttest.v)

ttest.results

```

#WARDEN SYSTEM ANALYSIS ONLY

##REGRESSIONS OF ESPERANCE RAINFALL STATION DATA TO FILL MISSING VALUES

```{r}

raincheck <- read.csv("./raw\_data/rain for regr.csv")

ggdensity(raincheck$MYRU)

fit <- lm(raincheck$ESPE ~ sqrt(raincheck$MYRU) + sqrt(raincheck$ESPA))

plot(fit)

summary(fit)

plot(raincheck$ESPE, -38.914+(11.19\*sqrt(raincheck$MYRU)))

```

##Graphing lake depths

###Load depth data and convert to AHD

```{r}

Wdepths <- read\_xlsx("./raw\_data/Warden depths.xlsx") #note: all values of "dry", "pooling" and "below DG" counted as zero depth. Missing values retained.

Wdepths[, c(2:7)] <- round(Wdepths[, c(2:7)], digits = 2)

# create year and month variables

Wdepths$month <- month(as.POSIXlt(Wdepths$Date, format="%d/%m/%Y"))

Wdepths$year <- year(as.POSIXlt(Wdepths$Date, format="%d/%m/%Y"))

Wdepths$year <- as.factor(Wdepths$year)

Wdepths$doy <- as.numeric(strftime(Wdepths$Date, format = "%j")) #create day of year variable

```

###Extract and graph Ewans dataset

```{r}

Wdepths\_Ewans <- Wdepths[, c("Ewans","year","doy")] #extract Ewans data

Wdepths\_Ewans <- na.omit(Wdepths\_Ewans) #remove NAs

#graph data)

ggplot(Wdepths\_Ewans, aes(x=doy, y=Ewans, color=year)) +

geom\_line() +

geom\_point() +

scale\_color\_manual(values=c('grey','grey','grey','grey','grey','grey','grey','grey','grey','grey','grey','green','grey','grey',"grey",'grey','blue','red')) +

ggtitle("Ewans annual depths") +

theme(plot.title = element\_text(vjust = -10)) +

theme(plot.title = element\_text(hjust = 0.5)) +

ylab("Depth (m)") +

xlab("Days in Year")

```

###Extract and graph Wheatfield dataset

```{r}

Wdepths\_Wheat <- Wdepths[, c("Wheatfield","year","doy")] #extract Wheatfield data

Wdepths\_Wheat <- na.omit(Wdepths\_Wheat) #remove NAs

#graph data)

ggplot(Wdepths\_Wheat, aes(x=doy, y=Wheatfield, color=year)) +

geom\_line() +

geom\_point() +

scale\_color\_manual(values=c('grey','grey',"grey",'grey','grey','grey','grey','grey','grey','blue','grey','grey','grey','grey','grey',"grey",'grey','grey','red')) +

ggtitle("Wheatfield annual depths") +

theme(plot.title = element\_text(vjust = -10)) +

theme(plot.title = element\_text(hjust = 0.5)) +

ylab("Depth (m)") +

xlab("Days in Year")

```

###Extract and graph Warden dataset

```{r}

Wdepths\_Ward <- Wdepths[, c("Warden","year","doy")] #extract Warden data

Wdepths\_Ward <- na.omit(Wdepths\_Ward) #remove NAs

#graph data)

ggplot(Wdepths\_Ward, aes(x=doy, y=Warden, color=year)) +

geom\_line() +

geom\_point() +

scale\_color\_manual(values=c('grey','grey',"grey",'grey','grey','grey','grey','grey','grey','grey','grey','grey','grey','blue','grey',"grey",'grey','grey','red')) +

ggtitle("Warden annual depths") +

theme(plot.title = element\_text(vjust = -10)) +

theme(plot.title = element\_text(hjust = 0.5)) +

ylab("Depth (m)") +

xlab("Days in Year")

```

##continuous scale plots

##select just Sep to Nov depths and create seperate column for Sep depths only.

```{r}

Wdepths\_AHD\_sp <- Wdepths\_AHD

Wdepths\_AHD\_sp$month <- month(as.POSIXlt(Wdepths\_AHD\_sp$Date, format="%d/%m/%Y"))

Wdepths\_AHD\_sp$year <- year(as.POSIXlt(Wdepths\_AHD\_sp$Date, format="%d/%m/%Y"))

Wdepths\_AHD\_sp <- Wdepths\_AHD\_sp[Wdepths\_AHD\_sp$month %in% c(9,10,11), ]

Wdepths\_AHD\_sp$Wheat\_sep <- NA

Wdepths\_AHD\_sp$Woody\_sep <- NA

Wdepths\_AHD\_sp$year <- as.factor(Wdepths\_AHD\_sp$year)

Wdepths\_AHD\_sp$month <- as.factor(Wdepths\_AHD\_sp$month)

Wdepths\_AHD\_sp$Wheat\_sep <- ifelse(Wdepths\_AHD\_sp$month == 9, Wdepths\_AHD\_sp$Wheatfield, 'NA')

Wdepths\_AHD\_sp$Wheat\_sep <- as.numeric(Wdepths\_AHD\_sp$Wheat\_sep)

Wdepths\_AHD\_sp$Woody\_sep <- ifelse(Wdepths\_AHD\_sp$month == 9, Wdepths\_AHD\_sp$Woody, 'NA')

Wdepths\_AHD\_sp$Woody\_sep <- as.numeric(Wdepths\_AHD\_sp$Woody\_sep)

```

##plot eastern suite depths

```{r}

ggp1 <-

ggplot(data = Wdepths\_AHD\_sp, aes(x = Date)) + # Create ggplot2 plot

geom\_point(aes(y = Wheatfield), color = 'red') +

geom\_point(aes(y = Wheat\_sep), color = 'black') +

ggtitle("Wheatfield") +

theme(plot.title = element\_text(vjust = -10)) +

theme(plot.title = element\_text(hjust = 0.5)) +

ylab("Depth AHD (m)\n")

ggp2 <- ggplot(data = Wdepths\_AHD\_sp, aes(x = Date)) + # Create ggplot2 plot

geom\_point(aes(y = Woody), color = 'red') +

geom\_point(aes(y = Woody\_sep), color = 'black') +

ggtitle("Woody") +

theme(plot.title = element\_text(vjust = -10)) +

theme(plot.title = element\_text(hjust = 0.5)) +

ylab("Depth AHD (m)\n")

ggp3 <- ggplot(data = Wdepths\_AHD\_sp, aes(x = Date)) + # Create ggplot2 plot

geom\_point(aes(y = Windabout, color = year)) +

ggtitle("Windabout") +

theme(plot.title = element\_text(vjust = -10)) +

theme(plot.title = element\_text(hjust = 0.5)) +

ylab("Depth AHD (m)\n")

ggp4 <- ggplot(data = Wdepths\_AHD\_sp, aes(x = Date)) + # Create ggplot2 plot

geom\_point(aes(y = Warden, color = year)) +

ggtitle("Warden") +

theme(plot.title = element\_text(vjust = -10)) +

theme(plot.title = element\_text(hjust = 0.5)) +

ylab("Depth AHD (m)\n")

```

```{r}

ggarrange(ggp1, ggp2, ggp3, ggp4, nrow=4, common.legend=TRUE, legend = "right")

```

###Totals of species abundances x year (for spring)

```{r}

Warden <- input.data[input.data$System == "Warden", ]

Warden <- Warden[Warden$Season == "spring", ]

Warden$Year <- year(Warden$Date)

Warden\_SxY <- as.matrix(xtabs(Warden$Count ~ Warden$CommonName + Warden$Year))

Warden\_SxY

write.csv(Warden\_SxY, "./outputs/Warden\_SxY.csv")

```

###Ordinations and associated analyses

####Input waterbird data and split into spring and summer datasets, including creating a survey variable

```{r}

Warden <- input.data.2[input.data.2$System == "Warden", ]

Warden.sum <- Warden[Warden$Season == "summer", ] #summer only data

Warden.spr <- Warden[Warden$Season == "spring", ] #spring only data

```

####Create a matrix for checking counts - this not used in analysis

```{r}

Warden\_matrix <- acast(Warden, CommonName ~ Survey, value.var="Count", fun.aggregate = sum) #sums counts for each species and survey

write.csv(Warden\_matrix, "./outputs/Warden\_matrix.csv")

```

####Aggregate count data by just common name and survey for creating season matrices

```{r}

Warden.sum.agg <- aggregate(data=Warden.sum, Count ~ CommonName + Survey, FUN="sum") #aggregate count data by species and survey

#Warden.sum.agg$Count <- sqrt(Warden.sum.agg$Count) removed sqrt because metaMDS routine in vegan transforms species abundance data is required.

Warden.spr.agg<- aggregate(data=Warden.spr, Count ~ CommonName + Survey, FUN="sum") #aggregate count data by species and survey

#Warden.spr.agg$Count <- sqrt(Warden.spr.agg$Count) removed sqrt because metaMDS routine in vegan transforms species abundance data is required.

```

####Create waterbird matrices by season

```{r}

#summer

Warden.sum.m <- melt(Warden.sum.agg)

Warden.sum.m <- dcast(Warden.sum.m, Survey ~ CommonName, fill="0")

rownames(Warden.sum.m) <- Warden.sum.m[,1]

Warden.sum.m[,1] <- NULL

write.csv(Warden.sum.m, "./outputs/Warden\_sum\_m.csv")

Warden.sum.m <- read.csv("./outputs/Warden\_sum\_m.csv", row.names=1)

#spring

Warden.spr.m <- melt(Warden.spr.agg)

Warden.spr.m <- dcast(Warden.spr.m, Survey ~ CommonName, fill="0")

rownames(Warden.spr.m) <- Warden.spr.m[,1]

Warden.spr.m[,1] <- NULL

write.csv(Warden.spr.m, "./outputs/Warden\_spr\_m.csv")

Warden.spr.m <- read.csv("./outputs/Warden\_spr\_m.csv", row.names=1)

```

####MDS unconstrained ordination using metaMDS in package vegan to analyse effects of season within the Warden system

```{r}

Warden.sum.MDS <- metaMDS(Warden.sum.m, distance = "bray", autotransform = TRUE)

Warden.spr.MDS <- metaMDS(Warden.spr.m, distance = "bray", autotransform = TRUE)

```

####Extract ordination point data for ggplots

```{r}

Warden.sum.MDS.pts <- as.data.frame(Warden.sum.MDS$points) #extract coordinates from ordination

Warden.sum.MDS.pts$Survey <- row.names(Warden.sum.MDS.pts)

Warden.spr.MDS.pts <- as.data.frame(Warden.spr.MDS$points) #extract coordinates from ordination

Warden.spr.MDS.pts$Survey <- row.names(Warden.spr.MDS.pts)

#W.points$season <- c("spring","summer","summer","summer","summer","summer","summer","summer","spring","spring","spring","spring","spring","spring","spring","spring") # add season labels as new column in dataframe

```

####Load environmental data, extract pipeline operation days and create seasonal env matrices (for pipeline days only and for depth and other data)

```{r}

av.depths <- read.csv("./raw\_data/average depths.csv") #load data file with average depths (per survey) for gauged Warden wetlands plus other env data

av.depths <- av.depths[order(av.depths$Survey), ] # order data by survey

row.names(av.depths) <- av.depths$Survey

av.depths <- av.depths[order(av.depths$Season, av.depths$Survey), ] #order alphabetically by season then survey

Surv.spr <- av.depths[av.depths$Season == "Spring", "Days", drop=FALSE]

Surv.sum <- av.depths[av.depths$Season == "Summer", "Days", drop=FALSE]

Surv.Pipe.spr <- av.depths[av.depths$Season == "Spring", c("Pipeline\_days","Days"), drop=FALSE]

Surv.Pipe.sum <- av.depths[av.depths$Season == "Summer", c("Pipeline\_days","Days"), drop=FALSE]

Pipeline.spr <- av.depths[av.depths$Season == "Spring", "Pipeline\_days", drop=FALSE]

Pipeline.sum <- av.depths[av.depths$Season == "Summer", "Pipeline\_days", drop=FALSE]

av.depths$Pipeline\_days <- NULL #remove pipeline data from av.depths

av.depths$Days<-NULL

av.depths$Month <- NULL

BEdat.spr <- av.depths[av.depths$Season == "Spring", 3:ncol(av.depths)] #restrict av.depths to spring

row.names(BEdat.spr) <- row.names(av.depths[av.depths$Season == "Spring", ])

BEdat.spr2 <- BEdat.spr #copy env data for spring

BEdat.spr2$Pipeline\_days <- Pipeline.spr$Pipeline\_days #add spring pipeline data

BEdat.sum <- av.depths[av.depths$Season == "Summer", 3:ncol(av.depths)] #restrict av.depths to summer

row.names(BEdat.sum) <- row.names(av.depths[av.depths$Season == "Summer", ])

BEdat.sum2 <- BEdat.sum #copy env data for spring

BEdat.sum2$Pipeline\_days <- Pipeline.sum$Pipeline\_days #add spring pipeline data

```

####Produce environmental data correlation matrices

```{r}

cor(BEdat.spr, method="pearson")

cor(BEdat.sum, method="pearson")

```

####Range standardise env data without pipeline data

```{r}

BEdat.spr.st <-BEdat.spr

for (i in 1:ncol(BEdat.spr)){

BEdat.spr.st[, i] <- rescale(BEdat.spr[, i], to = c(0,1))

}

Pipeline.spr$Pipeline\_days\_st <- rescale(Pipeline.spr$Pipeline\_days, to = c(0,1))

BEdat.sum.st <-BEdat.sum

for (i in 1:ncol(BEdat.sum)){

BEdat.sum.st[, i] <- rescale(BEdat.sum[, i], to = c(0,1))

}

Pipeline.sum$Pipeline\_days\_st <- rescale(Pipeline.sum$Pipeline\_days, to = c(0,1))

```

####Range standardise env data with pipeline

```{r}

BEdat.spr.st2 <-BEdat.spr2

for (i in 1:ncol(BEdat.spr2)){

BEdat.spr.st2[, i] <- rescale(BEdat.spr2[, i], to = c(0,1))

}

Surv.Pipe.spr$Days.st <- rescale(Surv.Pipe.spr$Days, to = c(0,1))

Surv.Pipe.sum$Days.st <- rescale(Surv.Pipe.sum$Days, to = c(0,1))

```

####Ordination graphs of Warden system waterbird communities by survey with surveys undertaken in spring and summer scaled and coloured by average depth of gauged wetlands

```{r}

#summer

ggplot(Warden.sum.MDS.pts, aes(x=MDS1, y=MDS2)) + xlim(min(Warden.sum.MDS.pts$MDS1)\*1.3, max(Warden.sum.MDS.pts$MDS1)\*1.3) + ylim(min(Warden.sum.MDS.pts$MDS2)\*1.3, max(Warden.sum.MDS.pts$MDS2)\*1.3) + coord\_fixed(ratio = 1) + geom\_point(aes(size = BEdat.sum$Depth, colour=BEdat.sum$Depth)) + scale\_size(range = c(9,20)) + geom\_text(aes(label = Survey), cex=3, vjust=2, hjust=1.25) + theme(legend.position = "none")

#spring

ggplot(Warden.spr.MDS.pts, aes(x=MDS1, y=MDS2)) + xlim(min(Warden.spr.MDS.pts$MDS1)\*1.3, max(Warden.spr.MDS.pts$MDS1)\*1.3) + ylim(min(Warden.spr.MDS.pts$MDS2)\*1.3, max(Warden.spr.MDS.pts$MDS2)\*1.3) + coord\_fixed(ratio = 1) + geom\_point(aes(size = BEdat.spr.st$Depth, colour=BEdat.spr$Depth)) + scale\_size(range = c(5,20)) + geom\_text(aes(label = Survey), cex=4, vjust=2, hjust=1.25) + theme(legend.position = "none") + scale\_x\_reverse(limits=c(0.3, -0.3)) + theme(legend.title = element\_blank()) + theme(legend.position = "top") + theme(legend.text=element\_text(size=12)) + theme(plot.margin=unit(c(0,0,0,0),"mm")) + theme(axis.title.x=element\_blank(), axis.text.x=element\_blank(), axis.ticks.x=element\_blank()) + theme(axis.title.y=element\_blank(), axis.text.y=element\_blank(), axis.ticks.y=element\_blank()) + theme(legend.title = element\_blank()) + theme(legend.position = "none")

```

####Create euclidean distance matrices for datasets to be used to partial bioenv analysis

```{r}

Pipeline.spr.m <-vegdist(Pipeline.spr$Pipeline\_days, method="euclidean")

Pipeline.sum.m <- vegdist(Pipeline.sum$Pipeline\_days, method ="euclidean")

Surv.spr.m <- vegdist(Surv.Pipe.spr$Days, method="euclidean")

Surv.sum.m <- vegdist(Surv.Pipe.sum$Days, method="euclidean")

Surv.Pipe.spr.m <- vegdist(Surv.Pipe.spr, method="euclidean")

Surv.Pipe.sum.m <- vegdist(Surv.Pipe.sum, method="euclidean")

```

####Test for effect of Pipeline operation

```{r}

Pipeline.sp <- bioenv(Warden.spr.m ~ Pipeline\_days, Pipeline.spr, method = "pearson", index = "bray", ) #bioenv to testfor effect of pipeline operation days

Pipeline.sp

Pipeline.sp.be <- bioenvdist(Pipeline.sp, which = "best") #extract euclidean matrix based on Pipeline\_days

mantel(vegdist(Warden.spr.m, method = "bray"), Pipeline.spr.m, method="pearson", permutations=999)

Pipeline.su <- bioenv(Warden.sum.m ~ Pipeline\_days, Pipeline.sum, method = "pearson", index = "bray", ) #bioenv to testfor effect of pipeline operation days

Pipeline.su

Pipeline.su.be <- bioenvdist(Pipeline.su, which = "best") #extract euclidean matrix based on Pipeline\_days

mantel(vegdist(Warden.sum.m, method = "bray"), Pipeline.sum.m, method="pearson", permutations=999)

```

####Bio-env analysis to determine influence of pipeline operation, depth and rainfall on waterbrd communities on the Warden wetlands. Uses pearson correlation because env vars without major outliers and not too far off normal in qqnorm plots.

```{r}

#SPRING

#step-wise analysis with pipeline variable partialled out.

Warden.spr.be1 <- bioenv(Warden.spr.m, BEdat.spr, method = "pearson", index = "bray", metric="euclidean", partial = Pipeline.spr$Pipeline\_days)

Warden.spr.be1

#mantel tests

Warden.spr.md <- vegdist(Warden.spr.m, method = "bray") #calculate Warden spring waterbird bray-curtis dissimilarity index

BEdat.spr.md <-bioenvdist(Warden.spr.be1, which="best") #extract best subset of env vars

mantel(Warden.spr.md, BEdat.spr.md, method="pearson", permutations=999) #mantel test to determine permutational significance

#step-wise analysis with pipeline variable allowed in (use dataset BEdat.spr2)

Warden.spr.be2 <- bioenv(Warden.spr.m, BEdat.spr2, method = "pearson", index = "bray", metric="euclidean")

Warden.spr.be2

#mantel tests

BEdat.spr.md2 <-bioenvdist(Warden.spr.be2, which="best")

mantel(Warden.spr.md, BEdat.spr.md2, method="pearson", permutations=999) #mantel test to determine permutational significance

#SUMMER

Warden.sum.be <- bioenv(Warden.sum.m, BEdat.sum2, method = "pearson", index = "bray", metric="euclidean")

Warden.sum.be

Warden.sum.md <- vegdist(Warden.sum.m, method = "bray") #calculate Warden spring waterbird bray-curtis dissimilarity index

BEdat.sum.md <-bioenvdist(Warden.sum.be, which="best")

mantel(Warden.sum.md, BEdat.sum.md, method="pearson", permutations=999) #mantel test to determine permutational significance

```

####Bio-env analysis to determine which species best match patterns in overall community ordination

```{r}

#spring

Warden.spr.msq <- sqrt(Warden.spr.m) #sqrt dataset since bv.step (as opposed to bvstep) does not do this, but needs to be done for consistency.

Warden.spr.bio <- bv.step(Warden.spr.msq, Warden.spr.msq, scale.fix=FALSE, scale.var=FALSE, fix.dist.method = "bray", var.dist.method = "bray", max.rho=0.95, min.delta.rho=0.01, output.best=10, num.restarts = 100)

Warden.spr.bio

#note: warnings are not important, see http://menugget.blogspot.com/2011/06/clarke-and-ainsworths-bioenv-and-bvstep.html

spring.species <- c("Australian.Shelduck","Banded.Stilt","Grey.Teal","Pacific.Black.Duck","Red.necked.Stint","Silver.Gull","Red.necked.Avocet")

Warden.spr7 <- Warden.spr.m[, spring.species] # create new survey x specie matrix with the 8 species from the bv.step analysis

write.csv(Warden.spr7, "./outputs/Warden 7 species.csv")

Warden.spr7.m <- vegdist(sqrt(Warden.spr7), method= "bray") #create dissimilarity matrix for just 8 species, for use in mantel tests

Warden.spr.md <- vegdist(sqrt(Warden.spr.m), method = "bray") #calculate Warden spring

mantel(Warden.spr.md, Warden.spr7.m, method="pearson", permutations=999) #mantel test to determine permutational significance

Warden.sprsp.m <- Warden.spr.msq[, spring.species] #extract just the 7 species for spring with square root abundances (because the ordination being used for the envfit (below) was performed on square root distances

#summer

Warden.sum.msq <- sqrt(Warden.sum.m) #sqrt dataset since bv.step (as opposed to bvstep) does not do this, but needs to be done for consistency.

Warden.sum.bio <- bv.step(Warden.sum.msq, Warden.sum.msq, scale.fix=FALSE, scale.var=FALSE, fix.dist.method = "bray", var.dist.method = "bray", max.rho=0.95, min.delta.rho=0.01, output.best=10, num.restarts = 200)

Warden.sum.bio

#note: warnings are not important, see http://menugget.blogspot.com/2011/06/clarke-and-ainsworths-bioenv-and-bvstep.html

summer.species <- c("Australian.Shelduck","Grey.Teal","Little.Black.Cormorant")

Warden.sum3 <- Warden.sum.m[, summer.species] # create new survey x specie matrix with the 8 species from the bv.step analysis

write.csv(Warden.sum3, "./outputs/Warden 3 species.csv")

Warden.sum3.m <- vegdist(sqrt(Warden.sum3), method= "bray") #create dissimilarity matrix for just 8 species, for use in mantel tests

Warden.sum.md <- vegdist(sqrt(Warden.sum.m), method = "bray") #calculate Warden suming

mantel(Warden.sum.md, Warden.sum3.m, method="pearson", permutations=999) #mantel test to determine permutational significance

Warden.sumsp.m <- Warden.sum.msq[, summer.species] #extract just the 8 species for spring with square root abundances (because the ordination being used for the envfit (below) was performed on square root distances

```

####db-RDA analyses including extended plotting

```{r}

#spring

#test individual variables

mod1 <- capscale(Warden.spr.m ~ Depth, BEdat.spr.st2, dist="bray") #test individual variables (use BEdat.spr2 to include Pipeline\_days

mod1

anova(mod1) #signif of individual variable

#stepwise model procedure with 'pipeline' data partialled out

mod2 <- capscale(Warden.spr.m ~ Depth + CDM3MO\_ESPER + CDM6MO\_ESPER + CDM12MO\_ESPER + CDM3MO\_EAERO + CDM6MO\_EAERO + CDM12MO\_EAERO + CDM3MO\_MYRUP + CDM6MO\_MYRUP + CDM12MO\_MYRUP + Pipeline\_days, BEdat.spr.st2, dist="bray") #full model

mod0 <- capscale(Warden.spr.m ~ 1, BEdat.spr.st2, dist="bray") #Null model

ind <- ordistep(mod0, scope = formula(mod2), perm.max = 999) #stepwise model build from all

anova(ind) #signif of stepwise model

mod3 <- capscale(Warden.spr.m ~ CDM12MO\_EAERO, BEdat.spr.st2, dist ="bray") #model selected by ordistep

mod3 #view results of stepwise model

anova(mod3) #signif or stepwise model

#build db-RDA plot from components

#plot(mod3, type = "n", choices = c(1, 2))

#points(mod3, display = "sites", pch=21, bg = "grey", cex=Warden.sprsp.m$Straw.necked.Ibis^2/50) #scale #sites by abundances of the 8 species

# text(mod3, display = "sites", cex = 0.7)

# points(mod3, display = "bp", cex=2)

# text(mod3, display = "bp", cex=0.7, col="blue")

#summer

mod1 <- capscale(Warden.sum.m ~ CDM12MO\_MYRUP, BEdat.sum.st, dist="bray") #test individual variables

mod1

anova(mod1) #signif of individual variable

#stepwise model build

mod2 <- capscale(Warden.sum.m ~ ., BEdat.sum.st, dist="bray") #full model

mod0 <- capscale(Warden.sum.m ~ 1, BEdat.sum.st, dist="bray") #Null model

ind <- ordistep(mod0, scope = formula(mod2), perm.max = 999) #stepwise model build from all

anova(ind) #signif of stepwise model

mod3 <- capscale(Warden.sum.m ~ CDM6MO\_MYRUP, BEdat.sum.st, dist ="bray")

mod3

```

#### test whether pipeline is still a factor when Lake Warden data removed

Create a Warden matrix without Lake Warden data

```{r}

Warden.spr.NoWa <- Warden.spr[-grep("WRP013", Warden.spr$SiteCode), ] #remove records for Lake Warden

Warden.spr.NoWa.agg<- aggregate(data=Warden.spr.NoWa, Count ~ CommonName + Survey, FUN="sum")

Warden.spr.NoWa.m <- melt(Warden.spr.NoWa.agg)

Warden.spr.NoWa.m <- dcast(Warden.spr.NoWa.m, Survey ~ CommonName, fill="0")

rownames(Warden.spr.NoWa.m) <- Warden.spr.NoWa.m[,1]

Warden.spr.NoWa.m[,1] <- NULL

write.csv(Warden.spr.NoWa.m, "./outputs/Warden\_spr\_NoWa\_m.csv")

Warden.spr.NoWa.m <- read.csv("./outputs/Warden\_spr\_NoWa\_m.csv", row.names=1)

```

####Create a warden matrix of just Lake Warden

```{r}

Warden.spr.Wa <- Warden.spr[Warden.spr$SiteCode == "WRP013", ] #include only records for Lake Warden

Warden.spr.Wa.agg<- aggregate(data=Warden.spr.Wa, Count ~ CommonName + Survey, FUN="sum")

Warden.spr.Wa.m <- melt(Warden.spr.Wa.agg)

Warden.spr.Wa.m <- dcast(Warden.spr.Wa.m, Survey ~ CommonName, fill="0")

rownames(Warden.spr.Wa.m) <- Warden.spr.Wa.m[,1]

Warden.spr.Wa.m[,1] <- NULL

write.csv(Warden.spr.Wa.m, "./outputs/Warden\_spr\_NoWa\_m.csv")

Warden.spr.Wa.m <- read.csv("./outputs/Warden\_spr\_NoWa\_m.csv", row.names=1)

Warden.spr.Wa.md <- vegdist(Warden.spr.Wa.m, method = "bray")

write.csv(as.matrix(Warden.spr.Wa.md), file = "./outputs/Warden\_spr\_Wa\_md.csv")

```

#####Bioenv without Lake Warden

```{r}

Warden.spr.NoWa.be <- bioenv(Warden.spr.NoWa.m, Pipeline.spr, method = "pearson", index = "bray", metric="euclidean")

Warden.spr.NoWa.be

Warden.spr.NoWa.md <- vegdist(Warden.spr.NoWa.m, method = "bray") #calculate Warden spring waterbird bray-curtis dissimilarity index

Pipeline.NoWa.be.m <-bioenvdist(Warden.spr.NoWa.be, which="best")

mantel(Warden.spr.NoWa.md, Pipeline.NoWa.be.m, method="pearson", permutations=999) #mantel test to determine permutational significance

```

#####db-RDA without lake Warden

```{r}

mod2 <- capscale(Warden.spr.NoWa.m ~ ., BEdat.spr2, dist="bray") #full model

mod0 <- capscale(Warden.spr.NoWa.m ~ 1, BEdat.spr2, dist="bray") #Null model

ind <- ordistep(mod0, scope = formula(mod2), perm.max = 999) #stepwise model build from all

anova(ind) #signif of stepwise model

mod3 <- capscale(Warden.spr.NoWa.m ~ Pipeline\_days, BEdat.spr2, dist ="bray") #model selected by ordistep

anova(mod3) #signif or stepwise model

mod3 #stepwise model

```

####create env dataset with just those vars from the rda and bio-env analyses

```{r}

env.restricted.spr <- c("CDM12MO\_ESPER", "CDM12MO\_EAERO", "Depth", "Pipeline\_days")

#non-standardised

BEdat.spr.rest <- BEdat.spr2[, env.restricted.spr]

#standardised

BEdat.spr.st2.rest <- BEdat.spr.st2[, env.restricted.spr]

```

####env fit to add waterbird and environmental vectors to unconstrained ordination

```{r}

#uses Warden.sprsp (from the bio-bio bv.step analysis above) and BEdat.spr.rest (from code above)

fit.spr.birds <- envfit(Warden.spr.MDS, Warden.sprsp.m, permutations = 999, choices=c(1,2)) #fit these species as vectors in unconstrained ordination

Warden.arrows.spr.birds <-as.data.frame(fit.spr.birds$vectors$arrows\*sqrt(fit.spr.birds$vectors$r)) #extract vector coordinates

Warden.arrows.spr.birds$Species <- row.names(Warden.arrows.spr.birds) #add species names to vectors

#restricted env vars envfit

plot(Warden.spr.MDS, type = "n", xlim=c(-0.4, 0.5), ylim=c(-0.4, 0.3))

points(Warden.spr.MDS, display = "sites", bg = "grey", pch=21, cex = Warden.spr7$Australian.Shelduck/500)

text(Warden.spr.MDS, display = "sites", cex = 0.7)

plot(fit.spr.birds)

fit.spr.env <- envfit(Warden.spr.MDS, BEdat.spr.st2.rest, permutations = 999) #fit env vars as vectors in unconstrained ordination

plot(Warden.spr.MDS, type = "n", xlim=c(-0.4, 0.5), ylim=c(-0.4, 0.3))

points(Warden.spr.MDS, display = "sites", bg = "grey", pch=21, cex=BEdat.spr.rest$CDM12MO\_ESPER/15)

text(Warden.spr.MDS, display = "sites", cex = 0.7)

plot(fit.spr.env)

```

```{r}

ggplot(Warden.spr.MDS.pts, aes(x=MDS1, y=MDS2)) + xlim(-0.8, 0.6) + ylim(-0.6, 0.6) + coord\_fixed(ratio = 1) + geom\_point(aes(size = 2)) + geom\_text(aes(label = Survey), colour="blue", cex=3, vjust=2, hjust=1.25) + theme(legend.position = "none") + scale\_x\_reverse(limits=c(0.6, -0.8)) + geom\_segment(data=Warden.arrows.spr.birds,aes(x=0,xend=NMDS1,y=0,yend=NMDS2), arrow = arrow(length = unit(0.5, "cm")),colour="grey",inherit\_aes=FALSE) + geom\_text(data=Warden.arrows.spr.birds,aes(x=NMDS1,y=NMDS2,label=Species),size=3)

#see https://stackoverflow.com/questions/14711470/plotting-envfit-vectors-vegan-package-in-ggplot2 for adding vectors to ggplot ordination

```

###Stacked abundance column plots for Warden system waterbirds

```{r}

Warden.stack <- input.data[input.data$System == "Warden", ]

Warden.stack$Group <- Taxonomy[match(Warden.stack$CommonName, Taxonomy$CommonName), 3]

```

#### Aggregate data by taxonomic group and survey, with survey and season variables for plotting

plus add dummy dates for years not surveyed

```{r}

Warden.stack.agg <- aggregate(data=Warden.stack, Count ~ Group + Survey + Season, FUN="sum") #aggregate count data by species and survey

#create dummy data for missing survey years for spring and summer

md.Group <- rep("Grebes", times = 7)

md.Survey <- c("Nov-2015","Nov-2016","Nov-2017","Nov-2019","Feb-2016","Feb-2017","Feb-2018")

md.Season <- c(rep("spring", times = 4), rep("summer", times = 3))

md.Count <- rep(0, times = 7)

Missing.dates <- data.frame(md.Group, md.Survey, md.Season, md.Count)

colnames(Missing.dates) <- c("Group","Survey","Season","Count")

Warden.stack.agg <- rbind(Warden.stack.agg, Missing.dates)

Warden.stack.agg$SurvOrd <- as.character(Warden.stack.agg$Survey)

Warden.stack.agg$SurvOrd <- as.numeric(str\_replace\_all(Warden.stack.agg$SurvOrd, c("Feb-2008" = "1", "Feb-2010" = "2", "Feb-2011" = "3", "Feb-2012" = "4", "Feb-2013" = "5", "Feb-2014" = "6", "Feb-2015" = "7", "Feb-2016" = "8", "Feb-2017" = "9", "Feb-2018" = "10", "Feb-2019" = "11", "Oct-2006" = "1", "Oct-2007" = "2", "Nov-2008" = "3", "Nov-2009" = "4", "Nov-2010" = "5", "Dec-2011" = "6", "Oct-2012" = "7", "Nov-2013"= "8", "Nov-2014" = "9", "Nov-2015" = "10", "Nov-2016" = "11", "Nov-2017" = "12", "Nov-2018" = "13", "Nov-2019" = "14", "Nov-2020" = "15")))

```

#### stacked column plots for Warden by season

```{r}

Warden.stack.spr <- Warden.stack.agg[Warden.stack.agg$Season=="spring", ] #cut down to just spring data

Warden.stack.spr <- Warden.stack.spr[ order(Warden.stack.spr$SurvOrd), ]

ggplot(Warden.stack.spr, aes(x=reorder(Survey, SurvOrd), y=Count, fill=Group)) + geom\_bar(stat="identity", colour = "black") + scale\_y\_continuous(breaks = seq(0,30000, by=5000), limits=c(0,30000)) + labs(x="Survey", y="Abundance") + theme (axis.text.x=element\_text(angle=90)) + theme(axis.title.y = element\_text(size = 20, margin = margin(t = 0, r = 20, b = 0, l = 0))) + theme(axis.title.x = element\_text(size = 20, margin = margin(t = 20, r = 0, b = 0, l = 0))) + theme(axis.text.x = element\_text(size=12)) + theme(axis.text.y = element\_text(size=15)) + scale\_fill\_manual(values = c("#CD9B9B","#912CEE","#104E8B","#698B22","#00EEEE","#A2B5CD","#FFA500","#FFE1FF","#B4CDCD","#C1FFc1","#A52A2A"))

#summer

Warden.stack.sum <- Warden.stack.agg[Warden.stack.agg$Season=="summer", ] #cut down to just summer data

Warden.stack.sum <- Warden.stack.sum[ order(Warden.stack.sum$SurvOrd), ]

ggplot(Warden.stack.sum, aes(x=reorder(Survey, SurvOrd), y=Count, fill=Group)) + geom\_bar(stat="identity", colour = "black") + scale\_y\_continuous(breaks = seq(0,30000, by=5000), limits=c(0,30000)) + labs(x="Survey", y="Abundance") + theme (axis.text.x=element\_text(angle=90)) + theme(axis.title.y = element\_text(size = 20, margin = margin(t = 0, r = 20, b = 0, l = 0))) + theme(axis.title.x = element\_text(size = 20, margin = margin(t = 20, r = 0, b = 0, l = 0))) + theme(axis.text.x = element\_text(size=12)) + theme(axis.text.y = element\_text(size=15)) + scale\_fill\_manual(values = c("#CD9B9B","#912CEE","#104E8B","#698B22","#00EEEE","#A2B5CD","#FFA500","#FFE1FF","#B4CDCD","#C1FFc1","#A52A2A")) #uses Date to reorder survey on x axis

```

###Stacked column plots for species richness at Warden

```{r}

Warden.stack.rich <- input.data.2[input.data.2$System == "Warden", ]

Warden.stack.rich$Group <- Taxonomy[match(Warden.stack.rich$CommonName, Taxonomy$CommonName), 3]

```

####create matrix for examining data

```{r}

Warden\_matrix <- acast(Warden.stack.rich, Group+CommonName ~ Survey, value.var="Count", fun.aggregate = sum) #sums counts for each species and survey

write.csv(Warden\_matrix, "./outputs/Warden\_matrix.csv")

```

####Aggregate counts by common name, then count number of species within groups

```{r}

Warden.stack.rich <- aggregate(data=Warden.stack.rich, Count ~ CommonName + Group + Survey + Season, FUN="sum") #aggregate count data by species and survey

Warden.stack.rich <- Warden.stack.rich[, c("Survey","Group","Season")] #restrict data to 3 columns

Warden.stack.rich <- count(Warden.stack.rich, c("Group","Survey","Season")) #count no. species

#create dummy years

md.Group <- rep("Grebes", times = 7)

md.Survey <- c("Nov-2015","Nov-2016","Nov-2017","Nov-2019","Feb-2016","Feb-2017","Feb-2018")

md.Season <- c(rep("spring", times = 4), rep("summer", times = 3))

md.freq <- rep(0, times = 7)

Missing.dates <- data.frame(md.Group, md.Survey, md.Season, md.freq)

colnames(Missing.dates) <- c("Group","Survey","Season","freq")

Warden.stack.rich <- rbind(Warden.stack.rich, Missing.dates)

Warden.stack.rich$SurvOrd <- as.character(Warden.stack.rich$Survey)

Warden.stack.rich$SurvOrd <- as.numeric(str\_replace\_all(Warden.stack.rich$SurvOrd, c("Feb-2008" = "1", "Feb-2010" = "2", "Feb-2011" = "3", "Feb-2012" = "4", "Feb-2013" = "5", "Feb-2014" = "6", "Feb-2015" = "7", "Feb-2016" = "8", "Feb-2017" = "9", "Feb-2018" = "10", "Feb-2019" = "11", "Oct-2006" = "1", "Oct-2007" = "2", "Nov-2008" = "3", "Nov-2009" = "4", "Nov-2010" = "5", "Dec-2011" = "6", "Oct-2012" = "7", "Nov-2013"= "8", "Nov-2014" = "9", "Nov-2015" = "10", "Nov-2016" = "11", "Nov-2017" = "12", "Nov-2018" = "13", "Nov-2019" = "14", "Nov-2020" = "15")))

```

#### stacked richness column plots for Warden by season

```{r}

Warden.stack.rich.spr <- Warden.stack.rich[Warden.stack.rich$Season=="spring", ]

#cut down to just spring data

Warden.stack.rich.spr <- Warden.stack.rich.spr[order(Warden.stack.rich.spr$SurvOrd), ]

ggplot(Warden.stack.rich.spr, aes(x=reorder(Survey, SurvOrd), y=freq, fill=Group)) + geom\_bar(stat="identity", colour = "black") + scale\_y\_continuous(breaks = seq(0,50, by=10), limits=c(0,50)) + labs(x="Survey", y="Number of species") + theme (axis.text.x=element\_text(angle=90)) + theme(axis.title.y = element\_text(size=20, margin = margin(t = 0, r = 20, b = 0, l = 0))) + theme(axis.title.x = element\_text(size=20, margin = margin(t = 20, r = 0, b = 0, l = 0))) + theme(axis.text.x = element\_text(size=12)) + theme(axis.text.y = element\_text(size=15)) + scale\_fill\_manual(values = c("#CD9B9B","#912CEE","#104E8B","#698B22","#00EEEE","#A2B5CD","#FFA500","#FFE1FF","#B4CDCD","#C1FFc1","#A52A2A"))

#summer

Warden.stack.rich.sum <- Warden.stack.rich[Warden.stack.rich$Season=="summer", ] #cut down to just summer data

Warden.stack.rich.sum <- Warden.stack.rich.sum[ order(Warden.stack.rich.sum$SurvOrd), ]

ggplot(Warden.stack.rich.sum, aes(x=reorder(Survey, SurvOrd), y=freq, fill=Group)) + geom\_bar(stat="identity", colour = "black") + scale\_y\_continuous(breaks = seq(0,50, by=10), limits=c(0,50)) + labs(x="Survey", y="Number of species") + theme (axis.text.x=element\_text(angle=90)) + theme(axis.title.y = element\_text(size=20, margin = margin(t = 0, r = 20, b = 0, l = 0))) + theme(axis.title.x = element\_text(size = 20, margin = margin(t = 20, r = 0, b = 0, l = 0))) + theme(axis.text.x = element\_text(size=12)) + theme(axis.text.y = element\_text(size=15))+ scale\_fill\_manual(values = c("#CD9B9B","#912CEE","#104E8B","#698B22","#00EEEE","#A2B5CD","#FFA500","#FFE1FF","#B4CDCD","#C1FFc1","#A52A2A"))

```

### Esperance rainfall for Warden wetlands

```{r}

rainfall <- read.csv("./raw\_data/multi station CDM.csv")

rainfall$Month <- as.yearmon(rainfall$Month, "%b-%y")

rainfall$Month.2 <- rainfall$Month

rainfall$Month.2 <- as.Date(rainfall$Month)

databreaks <- seq(as.Date("2004-01-01"), as.Date("2019-03-01"), by="3 month")

labels <- as.yearmon(databreaks, "%b-%y")

labels

```

#### Plot of rainfall

```{r}

ggplot(rainfall, aes(x = Month.2, y = CDMM, colour=Station)) + geom\_line() + theme (axis.text.x=element\_text(angle=90)) + scale\_x\_date(breaks=databreaks, labels = labels)

```

###Calculate total abundances per survey by season

```{r}

#spring

Warden.spr$year <- lubridate::year(Warden.spr$Date)

Warden.spr.yearsums <- aggregate(data=Warden.spr, Count ~ year + year, FUN="sum")

write.csv(Warden.spr.yearsums, "./outputs/warden\_spr\_yearsums.csv")

#summer

Warden.sum$year <- lubridate::year(Warden.sum$Date)

Warden.sum.yearsums <- aggregate(data=Warden.sum, Count ~ year + year, FUN="sum")

write.csv(Warden.sum.yearsums, "./outputs/warden\_sum\_yearsums.csv")

```

### Import data for species x species analysis

```{r}

Species <- input.data.2

Species.sum <- Species[Species$Season == "summer", ] #summer only data

Species.spr <- Species[Species$Season == "spring", ] #spring only data

```

#### extract data for particular species

```{r}

sp.id <- "Chestnut Teal"

sp.selected <- Species[Species$CommonName == sp.id, ]

sp.selected <- sp.selected[order(sp.selected$Date), ]

write.csv(sp.selected, file = sp.id)

```

##GORE SYSTEM ANALYSIS ONLY

###Totals of species abundances x year (for spring)

```{r}

Gore\_SxY <- input.data[input.data$System == "Gore", ]

Gore\_SxY <- Gore\_SxY[Gore\_SxY$Season == "spring", ]

Gore\_SxY$Year <- year(Gore\_SxY$Date)

Gore\_SxY <- as.matrix(xtabs(Gore\_SxY$Count ~ Gore\_SxY$CommonName + Gore\_SxY$Year))

Gore\_SxY

write.csv(Gore\_SxY, "./outputs/Gore\_SxY.csv")

```

###Ordinations and associated analyses

####Input waterbird data and split into spring and summer datasets, including creating a survey variable

```{r}

Gore <- input.data.2[input.data.2$System == "Gore", ]

Gore.sum <- Gore[Gore$Season == "summer", ] #summer only data

Gore.sum$Group <- Taxonomy[match(Gore.sum$CommonName, Taxonomy$CommonName), 3]

Gore.spr <- Gore[Gore$Season == "spring", ] #spring only data

Gore.spr$Group <- Taxonomy[match(Gore.spr$CommonName, Taxonomy$CommonName), 3]

```

####Create a matrix for checking counts - this not used in analysis

```{r}

Gore\_matrix <- as.data.frame(acast(Gore, CommonName ~ Survey, value.var="Count", fun.aggregate = sum)) #sums counts for each species and survey

Gore\_matrix$Group <- Taxonomy[match(row.names(Gore\_matrix), Taxonomy$CommonName), 3]

write.csv(Gore\_matrix, "./outputs/Gore\_matrix.csv")

```

####Aggregate count data by just common name and survey for creating season matrices

```{r}

Gore.sum.agg <- aggregate(data=Gore.sum, Count ~ CommonName + Survey, FUN="sum") #aggregate count data by species and survey

Gore.spr.agg<- aggregate(data=Gore.spr, Count ~ CommonName + Survey, FUN="sum") #aggregate count data by species and survey

```

####Create waterbird matrices by season

```{r}

#summer

Gore.sum.m <- melt(Gore.sum.agg)

Gore.sum.m <- dcast(Gore.sum.m, Survey ~ CommonName, fill="0")

rownames(Gore.sum.m) <- Gore.sum.m[,1]

Gore.sum.m[,1] <- NULL

write.csv(Gore.sum.m, "./outputs/Gore\_sum\_m.csv")

Gore.sum.m <- read.csv("./outputs/Gore\_sum\_m.csv", row.names=1)

#spring

Gore.spr.m <- melt(Gore.spr.agg)

Gore.spr.m <- dcast(Gore.spr.m, Survey ~ CommonName, fill="0")

rownames(Gore.spr.m) <- Gore.spr.m[,1]

Gore.spr.m[,1] <- NULL

write.csv(Gore.spr.m, "./outputs/Gore\_spr\_m.csv")

Gore.spr.m <- read.csv("./outputs/Gore\_spr\_m.csv", row.names=1)

```

####2D MDS unconstrained ordinations with vegan plots of Gore surveys by each season based on sqrt counts or raw counts

```{r}

#summer - sqrt abundance

Gore.sum.m.sqrt <- sqrt(Gore.sum.m)

Gore.sum.MDS.sqrt <- metaMDS(Gore.sum.m.sqrt, distance = "bray", k=2, autotransform = FALSE)

plot(Gore.sum.MDS.sqrt, display = "sites", cex=5, xlim=c(-0.7, 0.5))

points(Gore.sum.MDS.sqrt, display = "species", cex=2, col="red")

text(Gore.sum.MDS.sqrt, display = "sites", cex=1, col="blue")

text(Gore.sum.MDS.sqrt, display = "species", cex=1)

stressplot(Gore.sum.MDS.sqrt)

#summer - raw abundance - which does not work - produces degenerate ordination.

Gore.sum.MDS <- metaMDS(Gore.sum.m, distance = "bray", k=2, autotransform = FALSE)

plot(Gore.sum.MDS, display = "sites", cex=5, xlim=c(-0.7, 0.5))

points(Gore.sum.MDS, display = "species", cex=2, col="red")

text(Gore.sum.MDS, display = "sites", cex=1, col="blue")

text(Gore.sum.MDS, display = "species", cex=1)

stressplot(Gore.sum.MDS)

#spring - sqrt abundance

Gore.spr.m.sqrt <- sqrt(Gore.spr.m)

Gore.spr.MDS.sqrt <- metaMDS(Gore.spr.m.sqrt, distance = "bray", k=2, autotransform = FALSE) #use Gore.spr.m (raw abundances) or Gore.spr.m.sqrt (sqrt transformed abundances)

Gore.spr.spscores.sqrt <- scores(Gore.spr.MDS.sqrt, display=c("species"))

plot(Gore.spr.MDS.sqrt)

text(Gore.spr.MDS.sqrt, display="sites", cex=0.8)

text(Gore.spr.MDS.sqrt, display="species", cex=0.8)

stressplot(Gore.spr.MDS.sqrt)

#spring - raw abundance

Gore.spr.MDS <- metaMDS(Gore.spr.m, distance = "bray", k=2, autotransform = FALSE) #use Gore.spr.m (raw abundances) or Gore.spr.m.sqrt (sqrt transformed abundances)

Gore.spr.spscores <- scores(Gore.spr.MDS, display=c("species"))

plot(Gore.spr.MDS)

text(Gore.spr.MDS, display="sites", cex=0.8)

text(Gore.spr.MDS, display="species", cex=0.8)

stressplot(Gore.spr.MDS)

```

####Extract ordination point data for ggplots for metaMDS

Note - need to ensure the right ordination has been run (above), i.e. sqrt or raw count data

```{r}

#summer #sqrt only

Gore.sum.MDS.sqrt.pts <- as.data.frame(Gore.sum.MDS.sqrt$points) #extract coordinates from ordination

Gore.sum.MDS.sqrt.pts$Label2 <- row.names(Gore.sum.MDS.sqrt.pts) #add survey name from row names and call variable Label2 so that species names can be added later

Gore.sum.MDS.sqrt.pts$Label2 <- as.yearmon(Gore.sum.MDS.sqrt.pts$Label, "%b-%Y") #convert label to date format using package zoo

Gore.sum.MDS.sqrt.pts <- Gore.sum.MDS.sqrt.pts[order(as.Date(Gore.sum.MDS.sqrt.pts$Label2, format="%b-%Y")),] #order by Survey date

Gore.sum.MDS.sqrt.pts$Label <- as.character(Gore.sum.MDS.sqrt.pts$Label2) #convert Survey to character name and create new variable 'Label'

Gore.sum.MDS.sqrt.pts$Type <- "Survey" #add data type (survey here and species later)

Gore.sum.MDS.sqrt.pts$Label2 <- NULL

Gore.sum.MDS.sqrt.spscores <- as.data.frame(scores(Gore.sum.MDS, display = "species")) #extract species scores for later restricting to bio-bio (bv.step) outcome

#spring - raw counts

Gore.spr.MDS.pts <- as.data.frame(Gore.spr.MDS$points) #extract coordinates from ordination

Gore.spr.MDS.pts$Label2 <- row.names(Gore.spr.MDS.pts) #add survey name from row names and call variable Label2 so that species names can be added later

Gore.spr.MDS.pts$Label2 <- as.yearmon(Gore.spr.MDS.pts$Label, "%b-%Y") #convert label to date format using package zoo

Gore.spr.MDS.pts <- Gore.spr.MDS.pts[order(as.Date(Gore.spr.MDS.pts$Label2, format="%b-%Y")),] #order by Survey date

Gore.spr.MDS.pts$Label <- as.character(Gore.spr.MDS.pts$Label2) #convert Survey to character name and create new variable 'Label'

Gore.spr.MDS.pts$Type <- "Survey" #add data type (survey here and species later)

Gore.spr.MDS.pts$Label2 <- NULL

Gore.spr.MDS.spscores <- as.data.frame(scores(Gore.spr.MDS, display = "species")) #extract species scores for later restricting to bio-bio (bv.step) outcome

#spring - sqrt counts

Gore.spr.MDS.sqrt.pts <- as.data.frame(Gore.spr.MDS.sqrt$points) #extract coordinates from ordination

Gore.spr.MDS.sqrt.pts$Label2 <- row.names(Gore.spr.MDS.sqrt.pts) #add survey name from row names and call variable Label2 so that species names can be added later

Gore.spr.MDS.sqrt.pts$Label2 <- as.yearmon(Gore.spr.MDS.sqrt.pts$Label, "%b-%Y") #convert label to date format using package zoo

Gore.spr.MDS.sqrt.pts <- Gore.spr.MDS.sqrt.pts[order(as.Date(Gore.spr.MDS.sqrt.pts$Label2, format="%b-%Y")),] #order by Survey date

Gore.spr.MDS.sqrt.pts$Label <- as.character(Gore.spr.MDS.sqrt.pts$Label2) #convert Survey to character name and create new variable 'Label'

Gore.spr.MDS.sqrt.pts$Type <- "Survey" #add data type (survey here and species later)

Gore.spr.MDS.sqrt.pts$Label2 <- NULL

Gore.spr.MDS.sqrt.spscores <- as.data.frame(scores(Gore.spr.MDS, display = "species")) #extract species scores for later restricting to bio-bio (bv.step) outcome

```

####Ordination graphs of Gore system waterbird communities by survey with surveys undertaken in spring and summer

```{r}

#summer sqrt data

xmin <- min(Gore.sum.MDS.sqrt.pts$MDS1)\*1.3

xmax <- max(Gore.sum.MDS.sqrt.pts$MDS1)\*1.3

ymin <- min(Gore.sum.MDS.sqrt.pts$MDS2)\*1.3

ymax <- max(Gore.sum.MDS.sqrt.pts$MDS2)\*1.3

ggplot(Gore.sum.MDS.sqrt.pts, aes(x=MDS1, y=MDS2)) + xlim(xmin, xmax) + ylim(ymin, ymax) + coord\_fixed(ratio = 1) + geom\_point(cex=8) + geom\_text(aes(label = Label), cex=3, vjust=2, hjust=1.6) + theme(legend.position = "none") + geom\_path()

#spring raw count data

Gore.spr.MDS.pts.2 <- Gore.spr.MDS.pts

Gore.spr.MDS.pts.2$Label <- gsub(' ', '-', Gore.spr.MDS.pts.2$Label) #in Survey replace space with dash

Gore.spr.MDS.pts.2$Label <- paste("01-", Gore.spr.MDS.pts.2$Label, sep = "") #add dummy day to Survey

Gore.spr.MDS.pts.2$Label <- as.Date(Gore.spr.MDS.pts.2$Label, format = "%d-%b-%Y") #change survey to date

Gore.spr.MDS.pts.2<- Gore.spr.MDS.pts.2[order(Gore.spr.MDS.pts.2$Label), ] #order by date

xmin <- min(Gore.spr.MDS.pts.2$MDS1)\*1.3

xmax <- max(Gore.spr.MDS.pts.2$MDS1)\*1.3

ymin <- min(Gore.spr.MDS.pts.2$MDS2)\*1.3

ymax <- max(Gore.spr.MDS.pts.2$MDS2)\*1.3

ggplot(Gore.spr.MDS.pts, aes(x=MDS1, y=MDS2)) + xlim(xmin, xmax) + ylim(ymin, ymax) + coord\_fixed(ratio = 1) + geom\_point(cex=8) + geom\_text(aes(label = Label), cex=3, vjust=2, hjust=1.3) + theme(legend.position = "none")

#spring sqrt count data

Gore.spr.MDS.sqrt.pts.2 <- Gore.spr.MDS.sqrt.pts

Gore.spr.MDS.sqrt.pts.2$Label <- gsub(' ', '-', Gore.spr.MDS.sqrt.pts.2$Label) #in Survey replace space with dash

Gore.spr.MDS.sqrt.pts.2$Label <- paste("01-", Gore.spr.MDS.sqrt.pts.2$Label, sep = "") #add dummy day to Survey

Gore.spr.MDS.sqrt.pts.2$Label <- as.Date(Gore.spr.MDS.sqrt.pts.2$Label, format = "%d-%b-%Y") #change survey to date

Gore.spr.MDS.sqrt.pts.2<- Gore.spr.MDS.sqrt.pts.2[order(Gore.spr.MDS.sqrt.pts.2$Label), ] #order by date

xmin <- min(Gore.spr.MDS.sqrt.pts.2$MDS1)\*1.3

xmax <- max(Gore.spr.MDS.sqrt.pts.2$MDS1)\*1.3

ymin <- min(Gore.spr.MDS.sqrt.pts.2$MDS2)\*1.3

ymax <- max(Gore.spr.MDS.sqrt.pts.2$MDS2)\*1.3

ggplot(Gore.spr.MDS.sqrt.pts, aes(x=MDS1, y=MDS2)) + xlim(xmin, xmax) + ylim(ymin, ymax) + coord\_fixed(ratio = 1) + geom\_point(cex=8) + geom\_text(aes(label = Label), cex=3, vjust=2, hjust=1.6) + theme(legend.position = "none") + geom\_path()

```

#### Bio-bio analysis of spring waterbird species (raw count data)

```{r}

# use bv.stepo to undertake bio-bio analysis and extract ordination coordinates for the 4 species

Gore.spr.bio <- bv.step(Gore.spr.m, Gore.spr.m, scale.fix=FALSE, scale.var=FALSE, fix.dist.method = "bray", var.dist.method = "bray", max.rho=0.95, min.delta.rho=0.01, output.best=10, num.restarts = 100)

#note: warnings are not important, see http://menugget.blogspot.com/2011/06/clarke-and-ainsworths-bioenv-and-bvstep.html

Gore.spr.bio

Gore.spring.species <- c("Australian.Shelduck","Banded.Stilt","Grey.Teal","Black.Swan") #4 top species from bv.step

#Gore.spr.bio.4 <- Gore.spr.MDS.spscores[Gore.spring.species, ] # extract species scores for these 4 species from ordination

#Gore.spr.bio.4$Species <- row.names(Gore.spr.bio.4) #create new species variable

#Gore.spr.bio.4$Type <- "Species" #add data type

#colnames(Gore.spr.bio.4) <- c("MDS1", "MDS2","Label","Type")

#extract abundances for 4 species from spring bio-env analysis

Gore.spr.bio.4.abund <- Gore.spr.m[, Gore.spring.species]

Gore.spr.bio.4.abund$Label2 <- row.names(Gore.spr.bio.4.abund) #add survey name from row names and call variable Label2 so that species names can be added later

Gore.spr.bio.4.abund$Label2 <- as.yearmon(Gore.spr.bio.4.abund$Label, "%b-%Y") #convert label to date format using package zoo

Gore.spr.bio.4.abund$Label2 <- NULL

#Gore.spring.bio.4.abund <- normalize(Gore.spr.bio.4.abund, method = "range", range = c(0, 1), margin=2)

Gore.spr.bio.4.dist <- vegdist(Gore.spr.bio.4.abund, method = "bray") #B-C dissimilarity matrix of 4 species

Gore.spr.m.dist <- vegdist(Gore.spr.m, method = "bray") # B-C dissimilarity matrix of all species - raw count data

mantel(Gore.spr.m.dist, Gore.spr.bio.4.dist, method="pearson", permutations=999) #mantel test to determine permutational significance

```

#ordination plots of Gore spring surveys (based on raw coiunt data) scaled by species abundances

```{r}

xmin <- min(Gore.spr.MDS.pts.2$MDS1)\*1.3

xmax <- max(Gore.spr.MDS.pts.2$MDS1)\*1.3

ymin <- min(Gore.spr.MDS.pts.2$MDS2)\*1.3

ymax <- max(Gore.spr.MDS.pts.2$MDS2)\*1.3

Gore.spr.MDS.pts.2 <- Gore.spr.MDS.pts.2[order(row.names(Gore.spr.MDS.pts.2)), ]

ggplot(Gore.spr.MDS.pts.2, aes(x=MDS1, y=MDS2)) + xlim(xmin, xmax) + ylim(ymin, ymax) + coord\_fixed(ratio = 1) + geom\_point(size = Gore.spr.bio.4.abund$Australian.Shelduck/350) + geom\_text(aes(label = Label), cex=3, vjust=2, hjust=1.6) + theme(legend.position = "none")

```

###Stacked abundance column plots for Gore system waterbirds

```{r}

Gore.stack <- input.data[input.data$System == "Gore", ]

Gore.stack$Group <- Taxonomy[match(Gore.stack$CommonName, Taxonomy$CommonName), 3]

```

#### Aggregate data by taxonomic group and survey, with survey and season variables for plotting

plus add dummy dates for years not surveyed

```{r}

Gore.stack.agg <- aggregate(data=Gore.stack, Count ~ Group + Survey + Season, FUN="sum") #aggregate count data by species and survey

#create dummy data for missing survey years for spring and summer

md.Group <- rep("Grebes", times = 7)

md.Survey <- c("Nov-2015","Nov-2016","Nov-2017","Nov-2019","Feb-2016","Feb-2017","Feb-2018")

md.Season <- c(rep("spring", times = 4), rep("summer", times = 3))

md.Count <- rep(0, times = 7)

Missing.dates <- data.frame(md.Group, md.Survey, md.Season, md.Count)

colnames(Missing.dates) <- c("Group","Survey","Season","Count")

Gore.stack.agg <- rbind(Gore.stack.agg, Missing.dates)

Gore.stack.agg$SurvOrd <- as.character(Gore.stack.agg$Survey)

Gore.stack.agg$SurvOrd <- as.numeric(str\_replace\_all(Gore.stack.agg$SurvOrd, c("Feb-2010" = "1", "Feb-2011" = "2", "Feb-2012" = "3", "Feb-2013" = "4", "Feb-2014" = "5", "Feb-2015" = "6", "Feb-2016" = "7", "Feb-2017" = "8", "Feb-2018" = "9", "Feb-2019" = "10", "Nov-2009" = "1", "Nov-2010" = "2", "Dec-2011" = "3", "Oct-2012" = "4", "Nov-2013"= "5", "Nov-2014" = "6", "Nov-2015" = "7", "Nov-2016" = "8", "Nov-2017" = "9", "Nov-2018" = "10", "Nov-2019" = "11", "Nov-2020" = "12")))

```

#### stacked column plots for Gore by season

```{r}

Gore.stack.spr <- Gore.stack.agg[Gore.stack.agg$Season=="spring", ] #cut down to just spring data

Gore.stack.spr <- Gore.stack.spr[ order(Gore.stack.spr$SurvOrd), ]

ggplot(Gore.stack.spr, aes(x=reorder(Survey, SurvOrd), y=Count, fill=Group)) + geom\_bar(stat="identity", colour = "black") + scale\_y\_continuous(breaks = seq(0,25000, by=2500), limits=c(0,25000)) + labs(x="Survey", y="Abundance") + theme (axis.text.x=element\_text(angle=90)) + theme(axis.title.y = element\_text(margin = margin(t = 0, r = 20, b = 0, l = 0))) + theme(axis.title.x = element\_text(margin = margin(t = 20, r = 0, b = 0, l = 0))) + scale\_fill\_manual(values = c("#CD9B9B","#912CEE","#104E8B","#698B22","#00EEEE","#FFA500","#FFE1FF","#B4CDCD","#C1FFc1","#A52A2A"))

#summer

Gore.stack.sum <- Gore.stack.agg[Gore.stack.agg$Season=="summer", ] #cut down to just summer data

Gore.stack.sum <- Gore.stack.sum[ order(Gore.stack.sum$SurvOrd), ]

ggplot(Gore.stack.sum, aes(x=reorder(Survey, SurvOrd), y=Count, fill=Group)) + geom\_bar(stat="identity", colour = "black") + scale\_y\_continuous(breaks = seq(0,25000, by=2500), limits=c(0,25000)) + labs(x="Survey", y="Abundance") + theme (axis.text.x=element\_text(angle=90)) + theme(axis.title.y = element\_text(margin = margin(t = 0, r = 20, b = 0, l = 0))) + theme(axis.title.x = element\_text(margin = margin(t = 20, r = 0, b = 0, l = 0))) + scale\_fill\_manual(values = c("#CD9B9B","#912CEE","#104E8B","#698B22","#00EEEE","#A2B5CD","#FFA500","#FFE1FF","#B4CDCD","#C1FFc1","#A52A2A")) #uses Date to reorder survey on x axis

```

###Stacked column plots for species richness at Gore

```{r}

Gore.stack.rich <- input.data.2[input.data.2$System == "Gore", ]

Gore.stack.rich$Group <- Taxonomy[match(Gore.stack.rich$CommonName, Taxonomy$CommonName), 3]

```

####create matrix for examining data

```{r}

Gore\_matrix <- acast(Gore.stack.rich, Group + CommonName ~ Survey, value.var="Count", fun.aggregate = sum) #sums counts for each species and survey

```

####Aggregate counts by common name, then count number of species within groups

```{r}

Gore.stack.rich <- aggregate(data=Gore.stack.rich, Count ~ CommonName + Group + Survey + Season, FUN="sum") #aggregate count data by species and survey

Gore.stack.rich <- Gore.stack.rich[, c("Survey","Group","Season")] #restrict data to 3 columns

Gore.stack.rich <- count(Gore.stack.rich, c("Group","Survey","Season")) #count no. species

#craete dummy years

md.Group <- rep("Grebes", times = 7)

md.Survey <- c("Nov-2015","Nov-2016","Nov-2017","Nov-2019","Feb-2016","Feb-2017","Feb-2018")

md.Season <- c(rep("spring", times = 4), rep("summer", times = 3))

md.freq <- rep(0, times = 7)

Missing.dates <- data.frame(md.Group, md.Survey, md.Season, md.freq)

colnames(Missing.dates) <- c("Group","Survey","Season","freq")

Gore.stack.rich <- rbind(Gore.stack.rich, Missing.dates)

Gore.stack.rich$SurvOrd <- as.character(Gore.stack.rich$Survey)

Gore.stack.rich$SurvOrd <- as.numeric(str\_replace\_all(Gore.stack.rich$SurvOrd, c("Feb-2010" = "1", "Feb-2011" = "2", "Feb-2012" = "3", "Feb-2013" = "4", "Feb-2014" = "5", "Feb-2015" = "6", "Feb-2016" = "7", "Feb-2017" = "8", "Feb-2018" = "9", "Feb-2019" = "10", "Nov-2009" = "1", "Nov-2010" = "2", "Dec-2011" = "3", "Oct-2012" = "4", "Nov-2013"= "5", "Nov-2014" = "6", "Nov-2015" = "7", "Nov-2016" = "8", "Nov-2017" = "9", "Nov-2018" = "10", "Nov-2019" = "11", "Nov-2020" = "12")))

```

#### stacked richness column plots for Gore by season

```{r}

Gore.stack.rich.spr <- Gore.stack.rich[Gore.stack.rich$Season=="spring", ]

#cut down to just spring data

Gore.stack.rich.spr <- Gore.stack.rich.spr[order(Gore.stack.rich.spr$SurvOrd), ]

ggplot(Gore.stack.rich.spr, aes(x=reorder(Survey, SurvOrd), y=freq, fill=Group)) + geom\_bar(stat="identity", colour = "black") + scale\_y\_continuous(breaks = seq(0,40, by=10), limits=c(0,40)) + labs(x="Survey", y="Number of species") + theme (axis.text.x=element\_text(angle=90)) + theme(axis.title.y = element\_text(margin = margin(t = 0, r = 20, b = 0, l = 0))) + theme(axis.title.x = element\_text(margin = margin(t = 20, r = 0, b = 0, l = 0))) + scale\_fill\_manual(values = c("#CD9B9B","#912CEE","#104E8B","#698B22","#00EEEE","#FFA500","#FFE1FF","#B4CDCD","#C1FFc1","#A52A2A"))

#summer

Gore.stack.rich.sum <- Gore.stack.rich[Gore.stack.rich$Season=="summer", ] #cut down to just summer data

Gore.stack.rich.sum <- Gore.stack.rich.sum[ order(Gore.stack.rich.sum$SurvOrd), ]

ggplot(Gore.stack.rich.sum, aes(x=reorder(Survey, SurvOrd), y=freq, fill=Group)) + geom\_bar(stat="identity", colour = "black") + scale\_y\_continuous(breaks = seq(0,40, by=10), limits=c(0,40)) + labs(x="Survey", y="Number of species") + theme (axis.text.x=element\_text(angle=90)) + theme(axis.title.y = element\_text(margin = margin(t = 0, r = 20, b = 0, l = 0))) + theme(axis.title.x = element\_text(margin = margin(t = 20, r = 0, b = 0, l = 0))) + scale\_fill\_manual(values = c("#CD9B9B","#912CEE","#104E8B","#698B22","#00EEEE","#A2B5CD","#FFA500","#FFE1FF","#B4CDCD","#C1FFc1","#A52A2A"))

```

#Create ground survey matrix

```{r}

WGdata <- read.csv("./raw\_data/Warden\_Gore\_by\_suite\_06\_19.csv")

WGdata$SSD <- paste(WGdata$SiteCode, WGdata$Subsite, WGdata$Date, sep=" ")

ground.data <-WGdata[WGdata$SurveyType == "ground", ]

ground.data.W <- ground.data[ground.data$System == "Warden", ]

ground.data.W <- as.data.frame(acast(ground.data.W, CommonName ~ SSD, value.var="Count")) #sums counts for each species and survey

ground.data.W <- as.data.frame(t(ground.data.W))

ground.data.W$SSD <- row.names(ground.data.W)

ground.data.W$Site <- substr(ground.data.W$SSD, start=1, stop=6)

```

```{r}

Taxonomy <- read.csv("./raw\_data/Taxonomy.csv")

```