Assignment: Temporal Diversity

Venus Kuo; Z620: Quantitative Biodiversity, Indiana University

13 February, 2017

OVERVIEW

In this Assignment, we extend our understanding of diversity from the spatial dimension to the temporal dimension.

After completing this exercise you will know how to:

1. wrangle a large dataset to visualize and analyze time series data
2. test hypotheses from experiments with temporal data
3. quantify temporal β-diversity and stability

Directions:

1. Change "Student Name" on line 3 (above) with your name.
2. Complete as much of the exercise as possible during class; what you do not complete in class will need to be done on your own outside of class.
3. Use the Handout as a guide; it contains a more complete description of data sets along with the proper scripting needed to carry out the exercise.
4. Be sure to **answer the questions** in this exercise document; they also correspond to the Handout. Space for your answer is provided in this document and indicated by the ">" character. If you need a second paragraph be sure to start the first line with ">".
5. Before you leave the classroom, **push** this file to your GitHub repo.
6. When you are done with the Assignment, **Knit** the text and code into a html file.
7. After Knitting, please submit the completed Assignment by creating a **pull request** via GitHub. Your pull request should include this file *temporal_assignment.Rmd* and the html output of `Knitr` (*temporal_assignment.html*).

1) R SETUP

Typically, the first thing you will do in either an R script or an RMarkdown file is setup your environment. This includes things such as setting the working directory and loading any packages that you will need.

In the R code chunk below, provide the code to:

1. clear your R environment,
2. print your current working directory,
3. set your working directory to your "*/Week5-Temporal*" folder, and
4. load any packages you need to complete the assignment.

```r
# Set working directory #
rm(list = ls())
getwd()
```

```
## [1] "C:/Users/Venus/Github/QB2017_Kuo/Week5-Temporal"
```

```r
setwd("C:/Users/Venus/Github/QB2017_Kuo/Week5-Temporal")
#setwd("/Users/vkuo/GitHub/QB2017_Kuo/Week5-Temporal/")
```

```
# Require or install packages #
package.list <- c('vegan', 'tidyr', 'dplyr', 'codyn', 'ggplot2', 'cowplot', 'MullerPlot', 'RColorBrewer
for (package in package.list) {
  if (!require(package, character.only=T, quietly=T)) {
    install.packages(package)
    library(package, character.only=T)
} }
```

```
## This is vegan 2.4-1
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
##
## Attaching package: 'cowplot'
```

```
## The following object is masked from 'package:ggplot2':
##
##     ggsave
```

```
##
## Attaching package: 'reshape2'
```

```
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
##
## Attaching package: 'TH.data'
```

```
## The following object is masked from 'package:MASS':
##
##     geyser
```

```
##
## Attaching package: 'nlme'
```

```
## The following object is masked from 'package:dplyr':
##
```

```
##     collapse
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
##
## Attaching package: 'timeDate'
```

```
## The following object is masked from 'package:xtable':
##
##     align
```

```
## This is forecast 7.3
```

```
##
## Attaching package: 'forecast'
```

```
## The following object is masked from 'package:nlme':
##
##     getResponse
```

```
##
## Attaching package: 'ade4'
```

```
## The following object is masked from 'package:vegan':
##
##     cca
```

```
##
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:stats':
##
##     lowess
```

```
## BiodiversityR 2.7-2: use function 'BiodiversityRGUI()' to launch the BiodiversityR Graphical User In
```

2) LOADING DATA

Load dataset

In the R code chunk below, do the following:

1. load the portal dataset from in the "*/Week5/data*" folder, and
2. explore the structure of the dataset.

```
# Load portal dataset #
portal <- read.table("data/combined.csv", sep = ",", header = TRUE)

# Venus the Explorer #
str(portal)
```

```
## 'data.frame':    34786 obs. of  13 variables:
##  $ record_id      : int  1 72 224 266 349 363 435 506 588 661 ...
##  $ month          : int  7 8 9 10 11 11 12 1 2 3 ...
##  $ day            : int  16 19 13 16 12 12 10 8 18 11 ...
##  $ year           : int  1977 1977 1977 1977 1977 1977 1977 1978 1978 1978 ...
##  $ plot_id        : int  2 2 2 2 2 2 2 2 2 2 ...
##  $ species_id     : Factor w/ 48 levels "AB","AH","AS",..: 16 16 16 16 16 16 16 16 16 16 ...
##  $ sex            : Factor w/ 3 levels "","F","M": 3 3 1 1 1 1 1 1 3 1 ...
##  $ hindfoot_length: int  32 31 NA NA NA NA NA NA NA NA ...
##  $ weight         : int  NA NA NA NA NA NA NA NA 218 NA ...
##  $ genus          : Factor w/ 26 levels "Ammodramus","Ammospermophilus",..: 13 13 13 13 13 13 13 13
##  $ species        : Factor w/ 40 levels "albigula","audubonii",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ taxa           : Factor w/ 4 levels "Bird","Rabbit",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ plot_type      : Factor w/ 5 levels "Control","Long-term Krat Exclosure",..: 1 1 1 1 1 1 1 1 1 1
```

```r
length(unique(portal$plot_id))
```

```
## [1] 24
```

```r
length(unique(portal$species))
```

```
## [1] 40
```

Question 1: Describe some of the attributes of the `portal` dataset.

a. How many plots are in `portal`?
b. How many rodent species are there in the `portal` dataset?

> ***Answer 1a***: There are 24 plots. ***Answer 1b***: There are 40 rodent species.

3) WRANGLING THE PORTAL DATASET

In the R code chunk below, do the following:

1. Create a site-by-species matrix for any year of your choosing.
2. Create a vector of plot_type for sites in the site-by-species matrix.
3. Analyze alpha diversity (e.g., Shannon/Simpson) across the sites for that year.
4. Create a PCoA ordination of your site-by-species matrix.
5. Using the hypothesis testing tools you learned in the beta-diversity module, test the hypothesis that species abundances across sites vary as a factor of treatment type (i.e., plot_type).

```r
# Create new date vector #
portal <- unite(portal, col = date , c(year, month, day), sep = "-", remove = FALSE)
# Create new taxon vector #
portal <- unite(portal, col = taxon, c(genus, species), sep = "_" , remove = FALSE)
# Making time by species matrix #
time.by.species <- group_by(portal, year, plot_id, plot_type) %>% count(taxon) %>% spread(key = taxon,
# Chose year 1984 too look at #
dplyr::filter(time.by.species, year == 1984)
```

```
## Source: local data frame [24 x 51]
## Groups: year, plot_id, plot_type [24]
##
##     year plot_id                 plot_type Ammodramus_savannarum
##    <int>   <int>                    <fctr>                 <dbl>
## 1   1984       1         Spectab exclosure                     0
## 2   1984       2                   Control                     0
## 3   1984       3  Long-term Krat Exclosure                     0
## 4   1984       4                   Control                     0
## 5   1984       5          Rodent Exclosure                     0
## 6   1984       6 Short-term Krat Exclosure                     0
## 7   1984       7          Rodent Exclosure                      0
## 8   1984       8                   Control                      0
## 9   1984       9         Spectab exclosure                      0
## 10  1984      10          Rodent Exclosure                      0
## # ... with 14 more rows, and 47 more variables:
## #   Ammospermophilus_harrisi <dbl>, Amphispiza_bilineata <dbl>,
## #   Baiomys_taylori <dbl>, Calamospiza_melanocorys <dbl>,
## #   Callipepla_squamata <dbl>, Campylorhynchus_brunneicapillus <dbl>,
## #   Chaetodipus_baileyi <dbl>, Chaetodipus_intermedius <dbl>,
## #   Chaetodipus_penicillatus <dbl>, Chaetodipus_sp. <dbl>,
## #   Cnemidophorus_tigris <dbl>, Cnemidophorus_uniparens <dbl>,
## #   Crotalus_scutalatus <dbl>, Crotalus_viridis <dbl>,
## #   Dipodomys_merriami <dbl>, Dipodomys_ordii <dbl>, Dipodomys_sp. <dbl>,
## #   Dipodomys_spectabilis <dbl>, Lizard_sp. <dbl>, Neotoma_albigula <dbl>,
## #   Onychomys_leucogaster <dbl>, Onychomys_sp. <dbl>,
## #   Onychomys_torridus <dbl>, Perognathus_flavus <dbl>,
## #   Perognathus_hispidus <dbl>, Peromyscus_eremicus <dbl>,
## #   Peromyscus_leucopus <dbl>, Peromyscus_maniculatus <dbl>,
## #   Pipilo_chlorurus <dbl>, Pipilo_fuscus <dbl>, Pipilo_sp. <dbl>,
## #   Pooecetes_gramineus <dbl>, Reithrodontomys_fulvescens <dbl>,
## #   Reithrodontomys_megalotis <dbl>, Reithrodontomys_montanus <dbl>,
## #   Reithrodontomys_sp. <dbl>, Rodent_sp. <dbl>, Sceloporus_clarki <dbl>,
## #   Sceloporus_undulatus <dbl>, Sigmodon_fulviventer <dbl>,
## #   Sigmodon_hispidus <dbl>, Sigmodon_ochrognathus <dbl>,
## #   Sparrow_sp. <dbl>, Spermophilus_spilosoma <dbl>,
## #   Spermophilus_tereticaudus <dbl>, Sylvilagus_audubonii <dbl>,
## #   Zonotrichia_leucophrys <dbl>
```

```r
# Chose plot 5 to look at #
dplyr::filter(time.by.species, plot_id == 5)
```

```
## Source: local data frame [26 x 51]
## Groups: year, plot_id, plot_type [26]
##
##     year plot_id        plot_type Ammodramus_savannarum
##    <int>   <int>           <fctr>                 <dbl>
## 1   1977       5 Rodent Exclosure                     0
## 2   1978       5 Rodent Exclosure                     0
## 3   1979       5 Rodent Exclosure                     0
## 4   1980       5 Rodent Exclosure                     0
## 5   1981       5 Rodent Exclosure                     0
## 6   1982       5 Rodent Exclosure                     0
## 7   1983       5 Rodent Exclosure                     0
## 8   1984       5 Rodent Exclosure                     0
## 9   1985       5 Rodent Exclosure                     0
## 10  1986       5 Rodent Exclosure                     0
## # ... with 16 more rows, and 47 more variables:
## #   Ammospermophilus_harrisi <dbl>, Amphispiza_bilineata <dbl>,
## #   Baiomys_taylori <dbl>, Calamospiza_melanocorys <dbl>,
## #   Callipepla_squamata <dbl>, Campylorhynchus_brunneicapillus <dbl>,
## #   Chaetodipus_baileyi <dbl>, Chaetodipus_intermedius <dbl>,
## #   Chaetodipus_penicillatus <dbl>, Chaetodipus_sp. <dbl>,
## #   Cnemidophorus_tigris <dbl>, Cnemidophorus_uniparens <dbl>,
## #   Crotalus_scutalatus <dbl>, Crotalus_viridis <dbl>,
## #   Dipodomys_merriami <dbl>, Dipodomys_ordii <dbl>, Dipodomys_sp. <dbl>,
## #   Dipodomys_spectabilis <dbl>, Lizard_sp. <dbl>, Neotoma_albigula <dbl>,
## #   Onychomys_leucogaster <dbl>, Onychomys_sp. <dbl>,
## #   Onychomys_torridus <dbl>, Perognathus_flavus <dbl>,
## #   Perognathus_hispidus <dbl>, Peromyscus_eremicus <dbl>,
## #   Peromyscus_leucopus <dbl>, Peromyscus_maniculatus <dbl>,
## #   Pipilo_chlorurus <dbl>, Pipilo_fuscus <dbl>, Pipilo_sp. <dbl>,
## #   Pooecetes_gramineus <dbl>, Reithrodontomys_fulvescens <dbl>,
## #   Reithrodontomys_megalotis <dbl>, Reithrodontomys_montanus <dbl>,
## #   Reithrodontomys_sp. <dbl>, Rodent_sp. <dbl>, Sceloporus_clarki <dbl>,
## #   Sceloporus_undulatus <dbl>, Sigmodon_fulviventer <dbl>,
## #   Sigmodon_hispidus <dbl>, Sigmodon_ochrognathus <dbl>,
## #   Sparrow_sp. <dbl>, Spermophilus_spilosoma <dbl>,
## #   Spermophilus_tereticaudus <dbl>, Sylvilagus_audubonii <dbl>,
## #   Zonotrichia_leucophrys <dbl>
```

```r
# Converting tidyr object into data frame #
time.by.species <- as.data.frame(time.by.species)

## Made site by species matrix for the year 1977 ##
portal1977 <- subset(time.by.species, year == 1977)

# Make vector for plot treatment type #
treatment <- portal1977$plot_type

# Take out first 3 columns #
portal1977 <- portal1977[, -c(1:3)]

## Analyze alpha diversity across sites ##
alpha <- rbind(diversity(portal1977[1:23,], index="shannon"))

## Ordination of site by species matrix ##
# Principal Coordinate Analysis #
portal.db <- vegdist(portal1977, method = "bray", upper = TRUE , diag = TRUE)
pcoa <- cmdscale(portal.db, eig = TRUE, k = 3)
# Interpreting PCoA output #
explainvar1 <- round(pcoa$eig[1] / sum(pcoa$eig), 3) * 100
explainvar2 <- round(pcoa$eig[2] / sum(pcoa$eig), 3) * 100
explainvar3 <- round(pcoa$eig[3] / sum(pcoa$eig), 3) * 100
sum.eig <- sum(explainvar1, explainvar2, explainvar3)

explainvar1
```

```
## [1] 39.2
```

```r
explainvar2
```

```
## [1] 26.9
```

```r
explainvar3
```

```
## [1] 16.2
```

```r
sum.eig
```

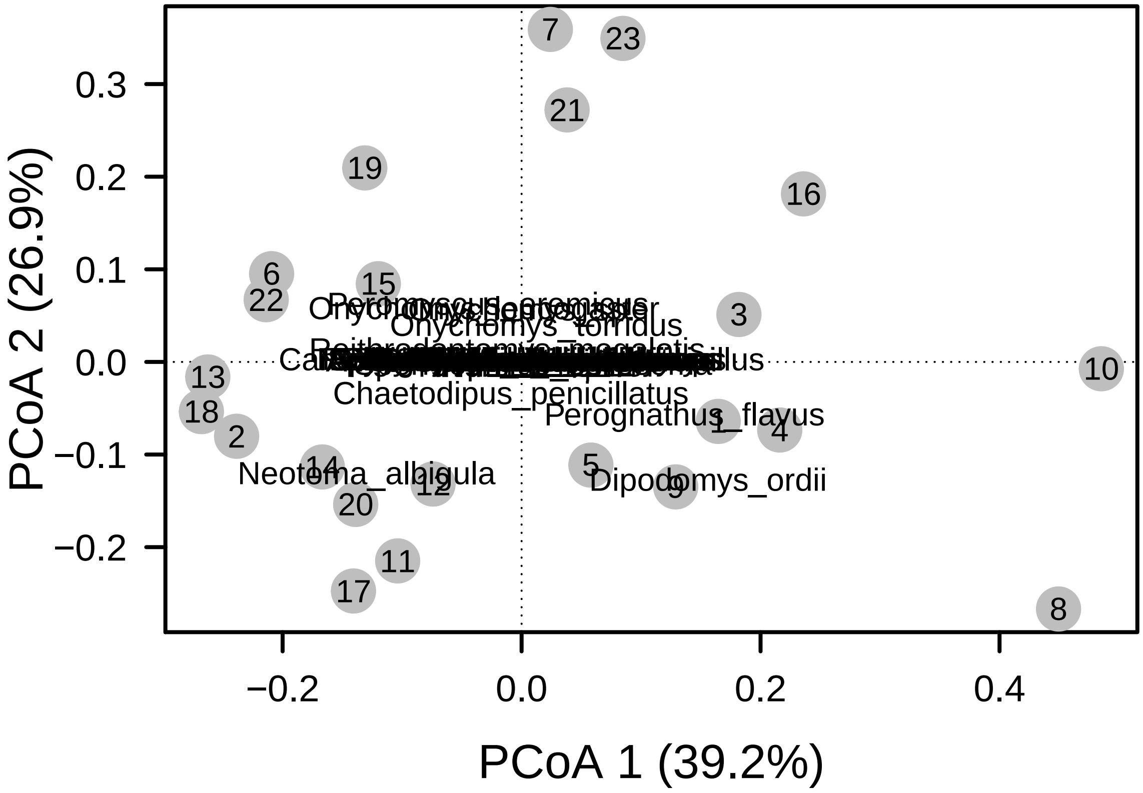
```
## [1] 82.3
```

```r
## Plot the PCoA ##
# Define Plot Parameters
```

```r
par(mar = c(5, 5, 1, 2) + 0.1)
# Initiate Plot
plot(pcoa$points[ ,1], pcoa$points[ ,2],
     xlab = paste("PCoA 1 (", explainvar1, "%)", sep = ""),
     ylab = paste("PCoA 2 (", explainvar2, "%)", sep = ""),
     pch = 16, cex = 2.0, type = "n", cex.lab = 1.5, cex.axis = 1.2, axes = FALSE)

# Add Axes
axis(side = 1, labels = T, lwd.ticks = 2, cex.axis = 1.2, las = 1)
axis(side = 2, labels = T, lwd.ticks = 2, cex.axis = 1.2, las = 1)
abline(h = 0, v = 0, lty = 3)
box(lwd = 2)

# Add Points & Labels
points(pcoa$points[ ,1], pcoa$points[ ,2],
       pch = 19, cex = 3, bg = "gray", col = "gray")
text(pcoa$points[ ,1], pcoa$points[ ,2],
     labels = row.names(pcoa$points))

REL <- portal1977
  for(i in 1:nrow(portal1977)){
    REL[i, ] = portal1977[i, ] / sum(portal1977[i, ])
  }

# Now, we use this information to calculate and add species scores
pcoa <- add.spec.scores(pcoa,REL,method = "pcoa.scores")
```

```
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero

## Warning in cor(comm[, i], ordiscores[, j]): the standard deviation is zero
```

```
text(pcoa$cproj[ ,1], pcoa$cproj[ ,2],
labels = row.names(pcoa$cproj), col = "black")
```



```
## Hypothesis testing: Beta-diversity ##

# Run PERMANOVA with adonis function #
```

```r
adonis(portal1977 ~ treatment, method = "bray", permutations = 999)
```

```
##
## Call:
## adonis(formula = portal1977 ~ treatment, permutations = 999,      method = "bray")
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs  MeanSqs F.Model      R2 Pr(>F)
## treatment  4    0.9626 0.240649   2.594 0.36566  0.003 **
## Residuals 18    1.6699 0.092772         0.63434
## Total     22    2.6325                  1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
indval <- multipatt(portal1977, cluster = treatment, func = "IndVal.g", control = how(nperm=999))
summary(indval)
```

```
##
##  Multilevel pattern analysis
##  ---------------------------
##
##  Association function: IndVal.g
##  Significance level (alpha): 0.05
##
##  Total number of species: 48
##  Selected number of species: 1
##  Number of species associated to 1 group: 1
##  Number of species associated to 2 groups: 0
##  Number of species associated to 3 groups: 0
##  Number of species associated to 4 groups: 0
##
##  List of species associated to each combination:
##
##  Group Spectab exclosure  #sps.  1
##                 stat p.value
## Dipodomys_ordii 0.784   0.041 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
portal.rel <- decostand(portal1977, method = "total")
phi <- multipatt(portal.rel, cluster = treatment, func = "r.g", control = how(nperm=999))
summary(phi)
```

```
##
##  Multilevel pattern analysis
##  ---------------------------
##
##  Association function: r.g
##  Significance level (alpha): 0.05
##
##  Total number of species: 48
##  Selected number of species: 2
##  Number of species associated to 1 group: 2
##  Number of species associated to 2 groups: 0
##  Number of species associated to 3 groups: 0
##  Number of species associated to 4 groups: 0
##
##  List of species associated to each combination:
##
##  Group Long-term Krat Exclosure  #sps.  1
##                 stat p.value
## Onychomys_sp. 0.734   0.015 *
##
##  Group Spectab exclosure  #sps.  1
##                       stat p.value
## Perognathus_flavus 0.647   0.043 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Question 2: Describe how different biodiversity estimates vary among sites.

a. Does diversity vary among sites? Does this correspond to treatment type?
b. Is treatment type a significant predictor of site dissimilarity?

Answer 2a: The alpha diversity of the 23 sites from the 1977 data sites vary between 0.37 and 1.84. And according to the PERMANOVA test I performed, it showed that site treatment significantly (p-value=0.005) affected species distribution across sites. So the alpha diversity is likely corresponding to treatment type ***Answer 2b***:Yes, treatment type is a significant predictor of site dissimilarity.

4) TIME SERIES ANALYSIS

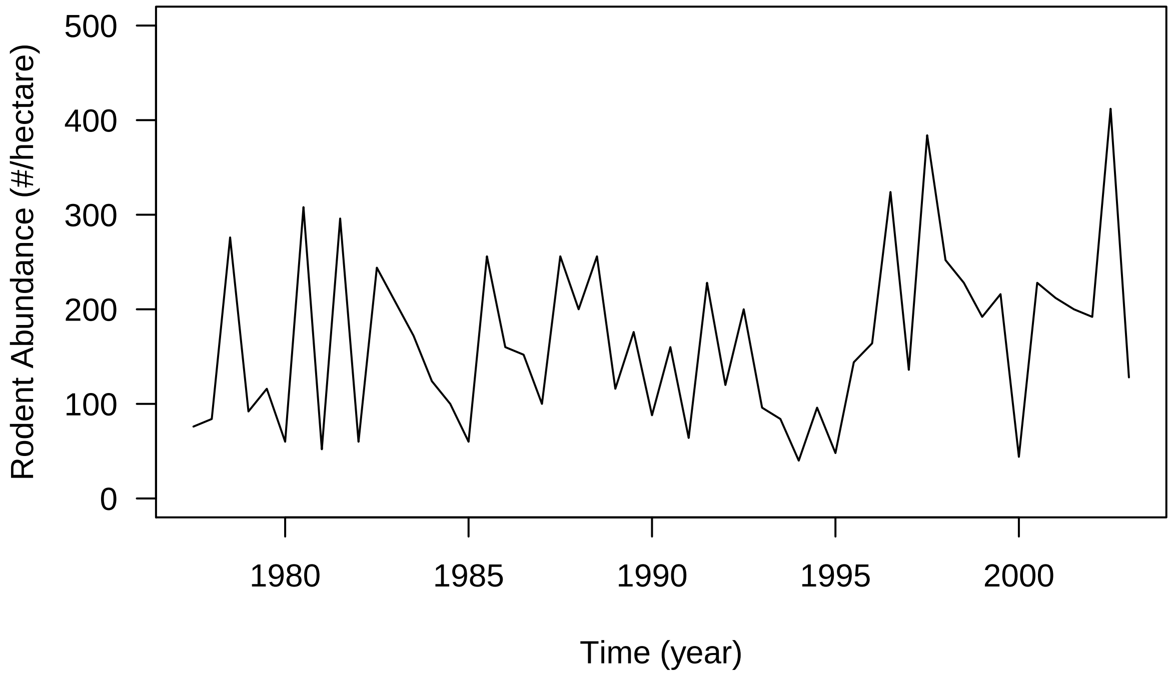
In the R code chunk below, do the following:

1. Create a time-by-species matrix that includes year, month, and plot_id for a site other than plot_id 2.
2. Examine per-hectare rodent abundance using simple moving average smoothing.
3. Test whether your data meets the assumption of stationarity.
4. If it does not meet this asumption, explore wasy to make your data stationary.
5. Examine and plot time lags using the partial autocorrelation function (PACF) and autocorrelation function (ACR).
6. Use the tools outlined in the Handout to create an ARMA model.

```r
# Time-by-species matrix with year, month, and plot_id #
time.by.spec.2 <- filter(portal, taxa == "Rodent") %>%
  group_by(year, month, plot_id) %>% count(taxon)
# Create seasonality variable using month number (6 = June; 10 = October) #
time.by.spec.2$season <- NA
time.by.spec.2$season <- time.by.spec.2$month %in% c(6:10)
# Rainy seasons are June - October #
time.by.spec.2$season <- ifelse(time.by.spec.2$season == TRUE, "rain", "norain")
# Group data by year and season #
group_by(time.by.spec.2, year, season)
```

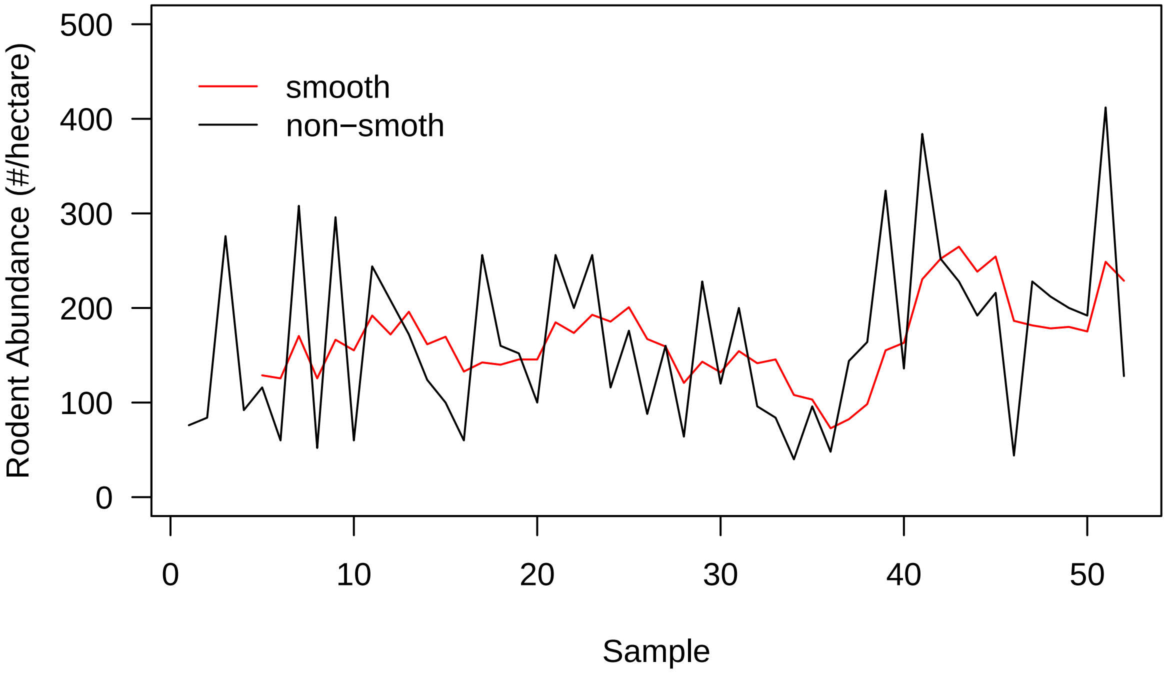
```
## Source: local data frame [16,391 x 6]
## Groups: year, season [52]
##
##     year month plot_id                     taxon     n season
##    <int> <int>   <int>                     <chr> <int>  <chr>
## 1   1977     7       1       Dipodomys_merriami     2   rain
## 2   1977     7       1       Perognathus_flavus     1   rain
## 3   1977     7       2 Chaetodipus_penicillatus     1   rain
## 4   1977     7       2       Dipodomys_merriami     1   rain
## 5   1977     7       2         Neotoma_albigula     1   rain
## 6   1977     7       2      Peromyscus_eremicus     1   rain
## 7   1977     7       3       Dipodomys_merriami     2   rain
## 8   1977     7       3    Dipodomys_spectabilis     1   rain
## 9   1977     7       3         Neotoma_albigula     1   rain
## 10  1977     7       4       Dipodomys_merriami     1   rain
## # ... with 16,381 more rows
```

```
# Filter to pick plot #
abund <- filter(time.by.spec.2, plot_id == 2) %>%
  group_by(year, season) %>%
  count(wt=n)
abund$nn <- abund$nn * 4
abund.ts <- ts(abund$nn, frequency=2, start=c(1977,2))
plot.ts(abund.ts, type="l", ylab="Rodent Abundance (#/hectare)",
        xlab = "Time (year)", las = 1, ylim = c(0, 500))
```

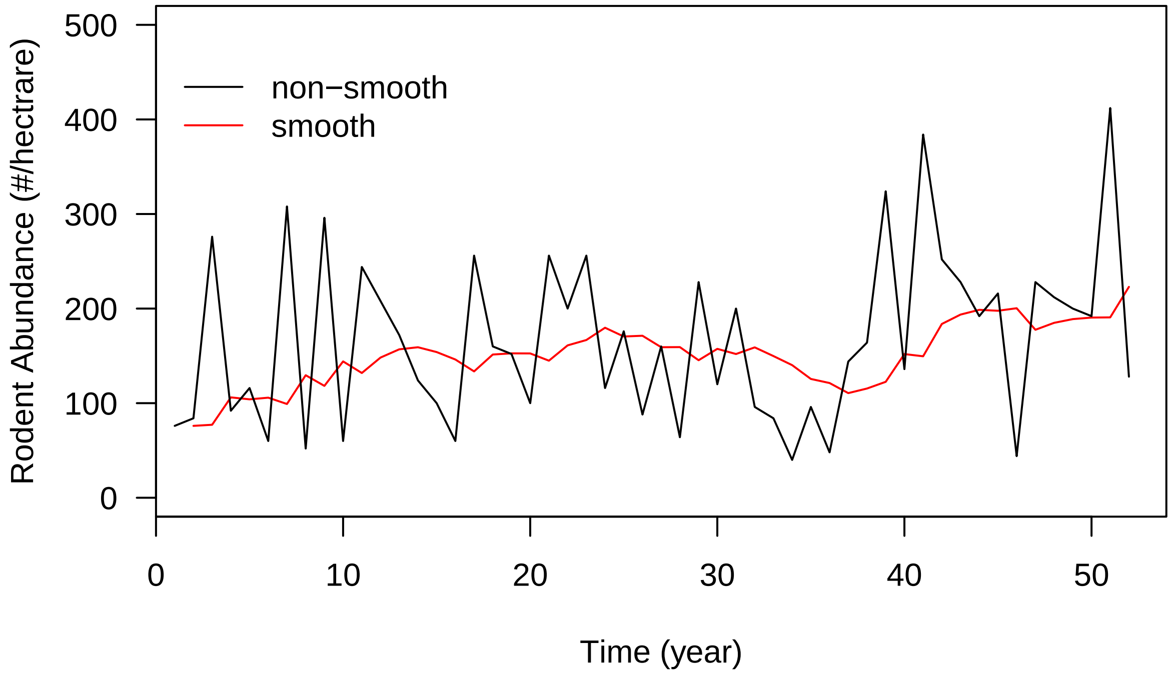


```
# Examine rodent abundance using moving average smoothing #
abund.sm <- SMA(abund$nn, n = 5)
plot(abund.sm, type="l", col="red", ylab="Rodent Abundance (#/hectare)",
     xlab="Sample", las=1, ylim=c(0,500))
lines(abund$nn, col="black")
```

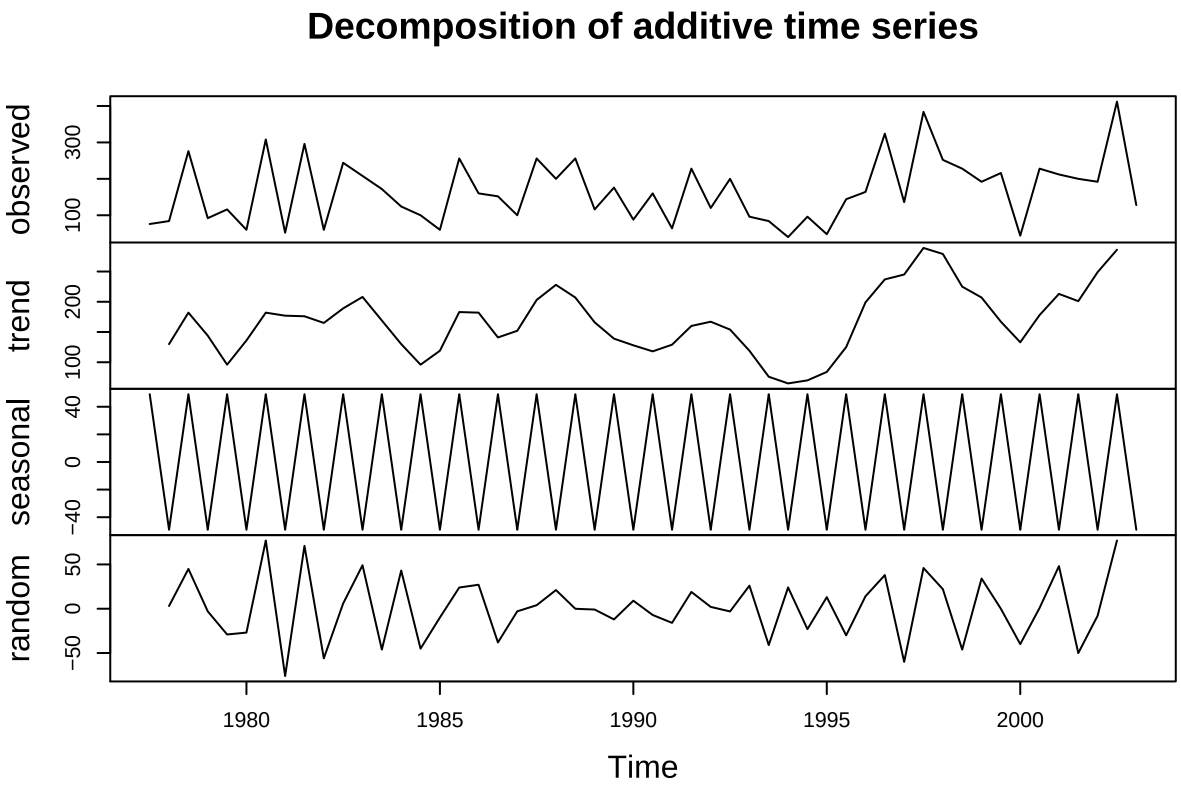
```
legend(0, 475, col=c("red", "black"), lty=c(1,1),
       c("smooth", "non-smoth"), bty="n", cex=1)
```



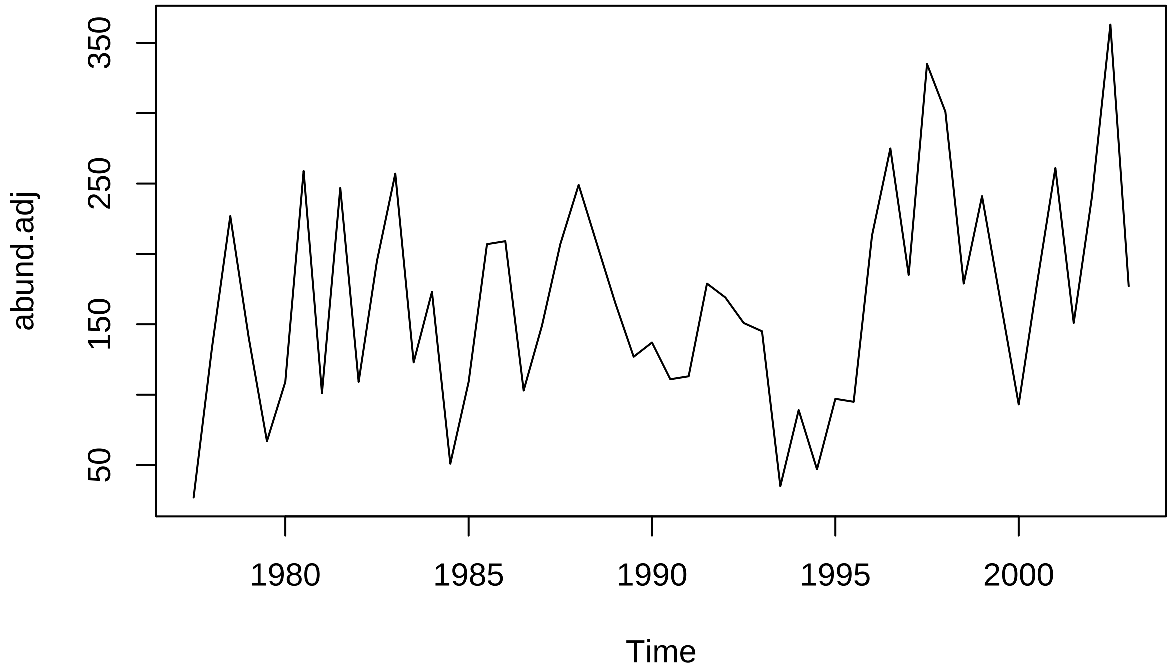
```
# Using Holt Winters function to specify two parameters #
abund.hw <- HoltWinters(abund$nn, beta = FALSE, gamma = FALSE)
# abund.hw$fitted
plot(abund.hw, xlab = "Time (year)", ylim = c(0,500),
     ylab = "Rodent Abundance (#/hectrare)", las = 1, main = NA)
legend(0, 475, col=c("black", "red"), lty = c(1,1),
        c("non-smooth", "smooth"), bty = "n", cex = 1)
```



```
# Decomposition of a time series #
# moving average decomposition #
abund.comp <- decompose(abund.ts)
# plot decomposition cateories #
plot(abund.comp)
```



```
# remove seasonality #
abund.adj <- abund.ts - abund.comp$seasonal
plot(abund.adj)
```



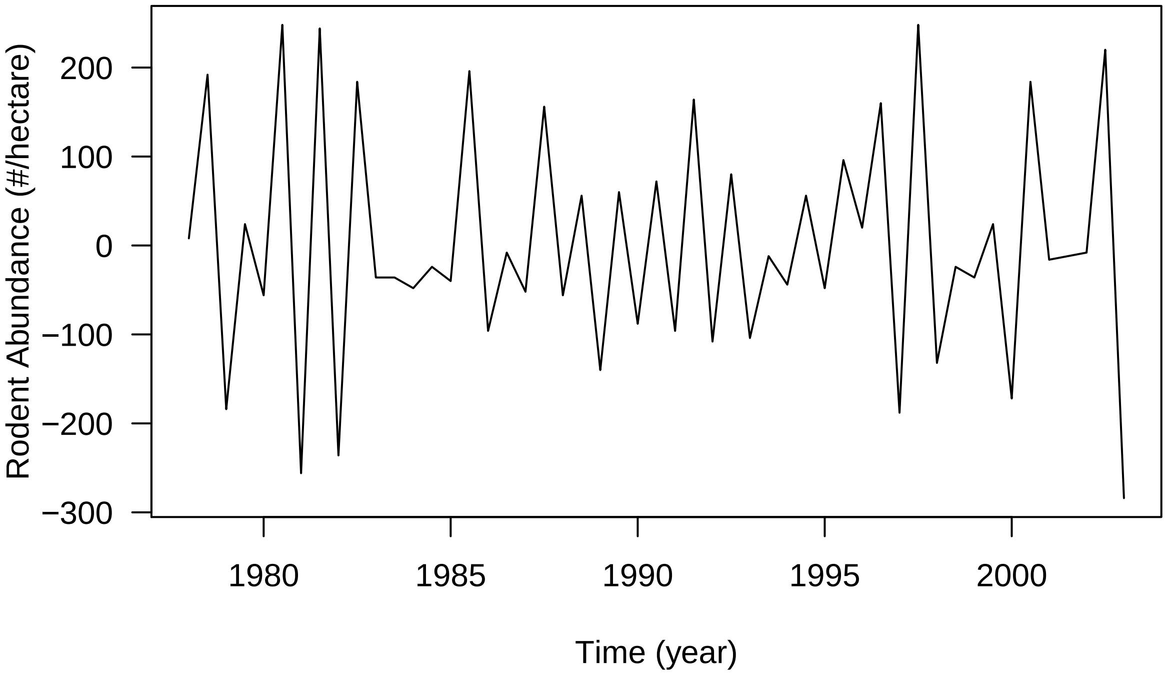
```
# Testing whether data meets assumption of stationarity #
adf.raw <- adf.test(abund.ts, alternative = "stationary")
adf.raw$p.value  # Fails Dickey-Fuller test #
```

```
## [1] 0.3229509
```

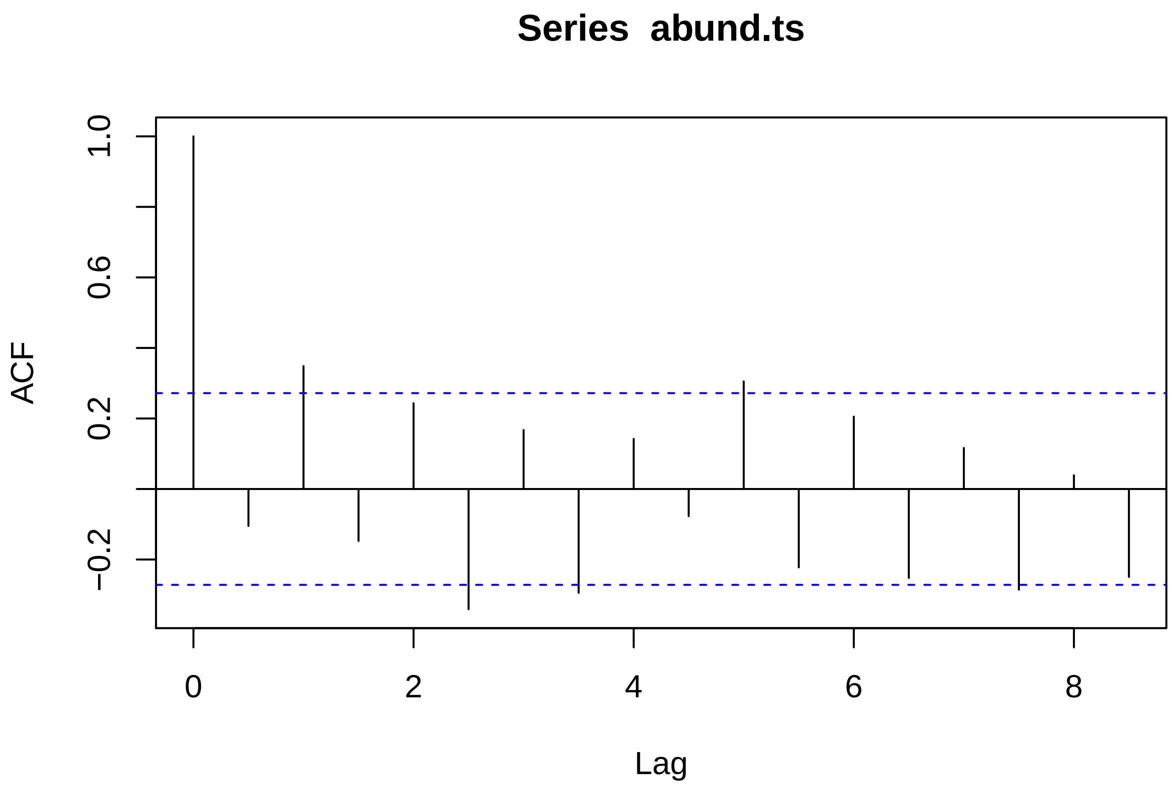
```
# Try differencing the time series and re-running #
abund.ts.diff <- diff(abund.ts)
adf.diff <- adf.test(abund.ts.diff, alternative = "stationary")
adf.diff$p.value # Meets assumption #
```

```
## [1] 0.02186602
```

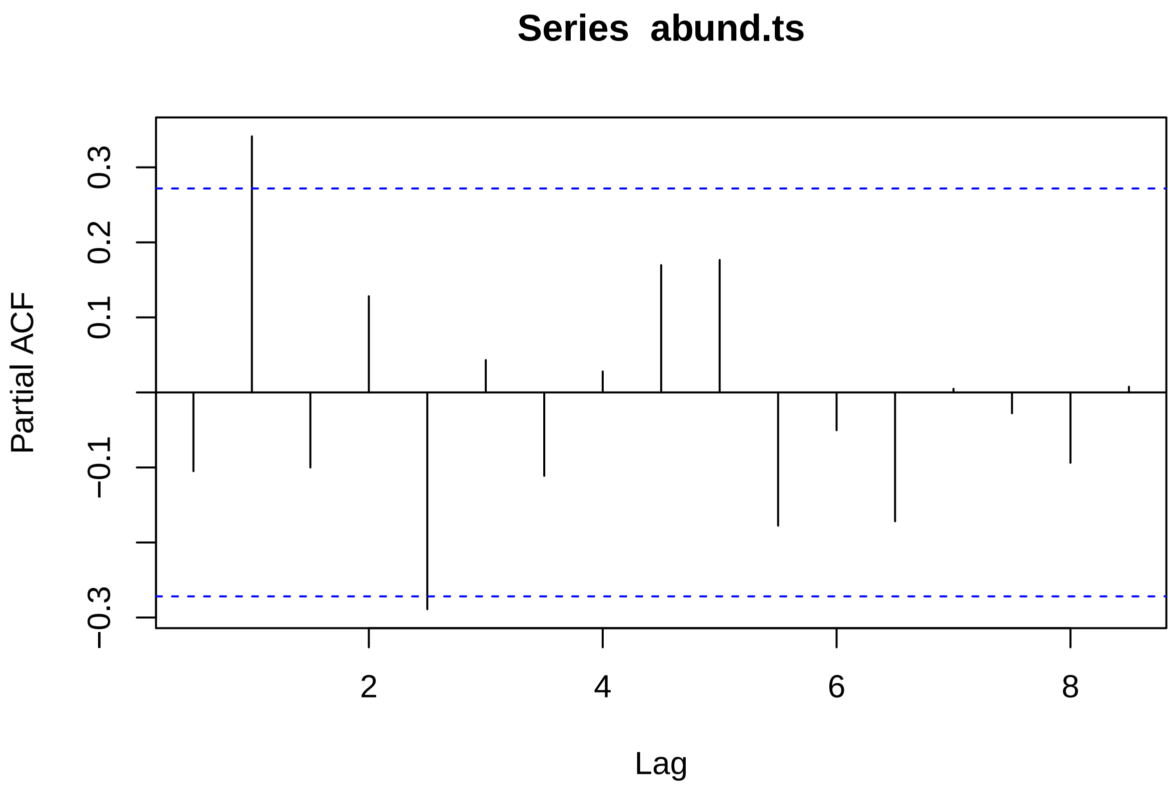
```
plot.ts(abund.ts.diff, ylab = "Rodent Abundance (#/hectare)", xlab = "Time (year)", las = 1)
```



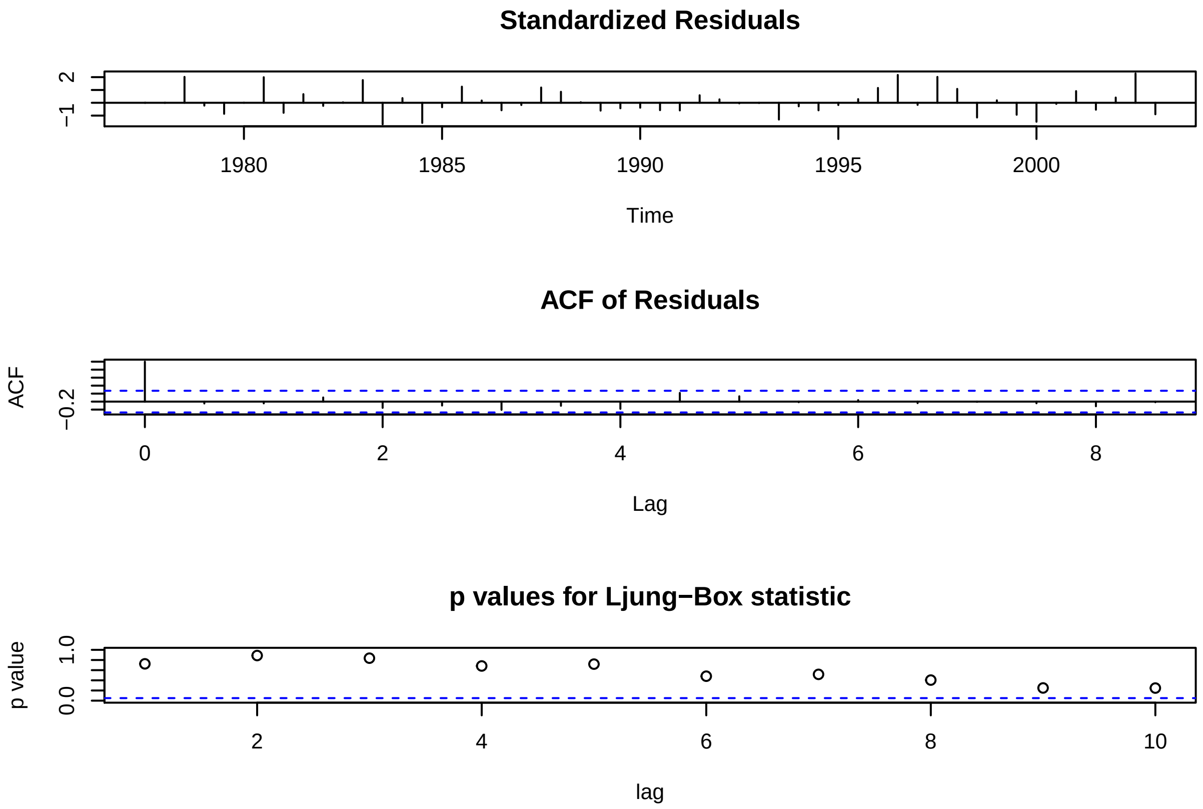
```
# Use autocorrelation function (ACF) to visualize and inform the parameterization of ARMA models #
acf(abund.ts)
```



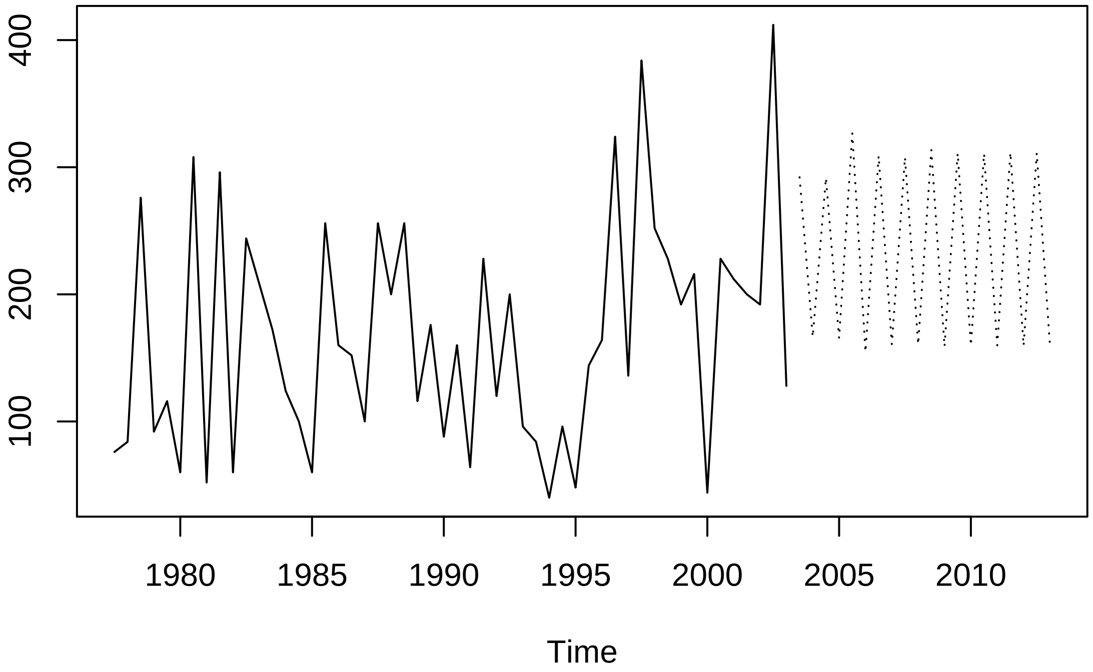
```
# Partial autocorrelation function (PACF) to calculate correlations between two series #
pacf(abund.ts)
```



```
# ARIMA model to descirbe the number of autoregressive lags, differencing, and order #
abund.arm <- auto.arima(abund.ts)
abund.arm <- arima((abund.ts), c(0, 0, 1), seasonal = list(order = c(2, 1, 0),
                     period=2), include.mean = TRUE)
tsdiag(abund.arm)
```



```
# Using predictions to forecast rodent abundance #
pred.arm <- predict(abund.arm, n.ahead = 20)
ts.plot(abund.ts, pred.arm$pred, lty = c(1,3)) #Interesting, looks funny #
```



Question 3: Describe the results from your time series analysis.

a. Does your data meet the assumption of stationarity? If not, what does this violation imply?
b. What does the ACF function do and how does it relate to the ARMA model? How does this differ from the autocorrelation function (ACF)?
c. What results can you conclude from your full ARMA model along with other methods outlined in the time series setcion of the Handout?

> ***Answer 3a***: Originally, the data did not meet the assumption of stationarity, meaning that the mean, variance, or covariance in a time series is affected by time. ***Answer 3b***: The autocorrelation function (ACF) shows us the lags of the forecast errors and provides information for the moving average (MA) component of the model. ACF looks at the correlation between lagged intervals in the time series. The Autoregressive moving average (ARMA) models are commonly used to identify trends in data and help make predictions of trends in the future. The autoregressive component of an ARMA model uses regression to obtain coefficients that predict current observations using previous or "lagged" observations from a time series. The ACF is used to obtain lags in the time series to identify trends for the future for ARMA models. ***Answer 3c***: Our time-series trend is increasing and this change is likely does not fail the assumption of stationarity.

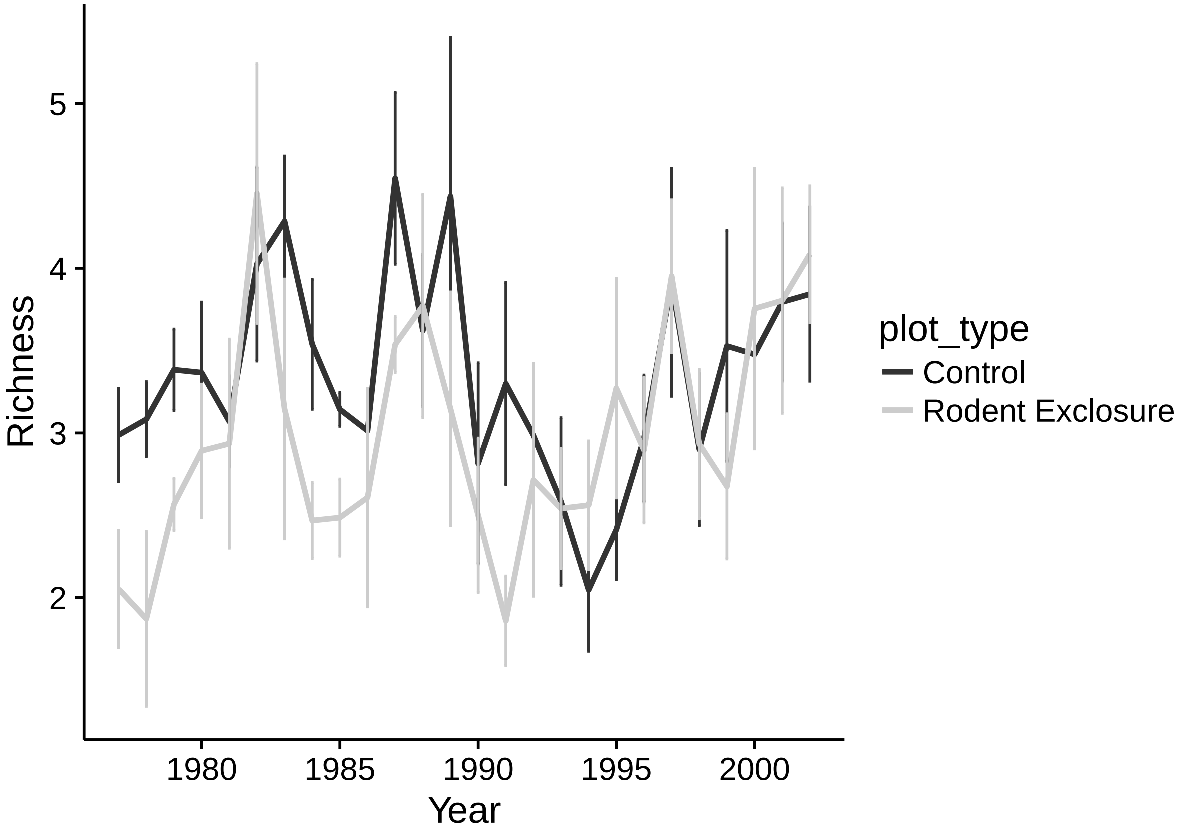
5) REPEATED MEASURES ANALYSIS OF VARIANCE (RM-ANOVA)

In the R code chunk below, do the following:

1. Create an appropriate data frame for RM-ANOVA (e.g., yearly species abundance values within plots).
2. Calculate the inverse of Simpson's diversity for each year, and plot it as a function of year for the Control and Rodent Exclosure plots.

3. Perform an RM-ANOVA and construct a F-test using the AR(1), compound symmetery, and unstructured covariance structures.

```
# Creating data frame for RM-ANOVA #
# Constructing time-vy-species matrix #
time.by.species <- group_by(portal, year, plot_id, plot_type) %>% count(taxon) %>% spread(key = taxon, 
# Calculate observed richness from time.by.species matrix #
# richness <- as.data.frame(rowSums(time.by.species[,-c(1:3)]))
# Calulate inverse Simpsons diversity from time.by.species matrix #
richness <- as.data.frame(diversity(time.by.species[,-c(1:3)], index="invsimpson"))
# Create data frame with experimental design and richness data #
rich.all <- data.frame(time.by.species[,1:3,], richness)
# Rename Column #
names(rich.all)[4] <- "richness"
# Pull out two of the five Portal treatments #
rich.treat <- rich.all[which(rich.all$plot_type == "Control" | rich.all$plot_type == "Rodent Exclosure"

# Plot data #
rich.treat.plot <- group_by(rich.treat, plot_type, year) %>% summarise (
  mean = mean(richness),
  sd = sd(richness),
  n = n(),
  sem = sd/sqrt(n))
rich.plot <- ggplot(rich.treat.plot, aes(x = year, y = mean, color = plot_type)) +
  geom_line(size = 1, show.legend = T) +
  geom_errorbar(aes(ymin = mean -sem, ymax = mean + sem), width = 0.1) +
  xlim(1977, 2002) +
  xlab("Year") +
  ylab("Richness") +
  scale_color_grey()
plot(rich.plot)
```



```
# Perform RM-ANOVA #
rich.rm <- lme(richness ~ plot_type*year, random = ~ 1 | plot_id,
               correlation = corAR1(form = ~1 | plot_id),
               data = rich.treat)
rich.cmp <- lme(richness ~ plot_type*year, random = ~ 1 | plot_id,
               correlation = corCompSymm(form = ~1 | plot_id),
               data = rich.treat)
# rich.unstr <- lme(richness ~ plot_type*year, random = ~ 1 | plot_id,
#                 correlation = corSymm(form = ~1 | plot_id),
#                 data = rich.treat)

# Look at detailed output #
summary(rich.rm)
```

```
## Linear mixed-effects model fit by REML
##  Data: rich.treat
##        AIC      BIC    logLik
##   1153.217 1180.322 -569.6087
##
## Random effects:
##  Formula: ~1 | plot_id
##         (Intercept) Residual
## StdDev:   0.6469478 1.250206
##
## Correlation Structure: AR(1)
##  Formula: ~1 | plot_id
##  Parameter estimate(s):
##       Phi
## 0.4250398
## Fixed effects: richness ~ plot_type * year
##                                    Value Std.Error  DF    t-value p-value
## (Intercept)                      1.36695  33.32814 343  0.0410150  0.9673
## plot_typeRodent Exclosure      -73.37575  52.01883  12 -1.4105613  0.1838
## year                             0.00100   0.01675 343  0.0595739  0.9525
## plot_typeRodent Exclosure:year   0.03670   0.02614 343  1.4039132  0.1612
##  Correlation: 
##                                (Intr) plt_RE year  
## plot_typeRodent Exclosure      -0.641              
## year                           -1.000  0.641       
## plot_typeRodent Exclosure:year  0.641 -1.000 -0.641
## 
## Standardized Within-Group Residuals:
##         Min          Q1         Med          Q3         Max 
## -1.93162654 -0.70434296 -0.08293334  0.45856267  5.06139168 
## 
## Number of Observations: 359
## Number of Groups: 14
```

```r
summary(rich.cmp)
```

```
## Linear mixed-effects model fit by REML
##  Data: rich.treat 
##        AIC      BIC    logLik
##   1215.368 1242.473 -600.6842
## 
## Random effects:
##  Formula: ~1 | plot_id
##         (Intercept) Residual
## StdDev:   0.7104222 1.214275
## 
## Correlation Structure: Compound symmetry
##  Formula: ~1 | plot_id 
##  Parameter estimate(s):
##          Rho 
## 6.672013e-18 
## Fixed effects: richness ~ plot_type * year 
##                                    Value Std.Error  DF    t-value p-value
## (Intercept)                     13.26069  22.33564 343  0.5937008  0.5531
## plot_typeRodent Exclosure      -68.60423  34.97126  12 -1.9617316  0.0734
## year                            -0.00498   0.01123 343 -0.4438008  0.6575
## plot_typeRodent Exclosure:year   0.03431   0.01758 343  1.9518712  0.0518
##  Correlation: 
##                                (Intr) plt_RE year  
## plot_typeRodent Exclosure      -0.639              
## year                           -1.000  0.639       
## plot_typeRodent Exclosure:year  0.639 -1.000 -0.639
## 
## Standardized Within-Group Residuals:
##        Min         Q1        Med         Q3        Max 
## -2.0346517 -0.6823806 -0.1001395  0.4790022  5.1406695 
## 
```

```
## Number of Observations: 359
## Number of Groups: 14
```

```r
# Obtain F-test #
anova(rich.rm)
```

```
##                 numDF denDF   F-value p-value
## (Intercept)         1   343 256.22009  <.0001
## plot_type           1    12   0.72274  0.4119
## year                1   343   1.56060  0.2124
## plot_type:year      1   343   1.97097  0.1612
```

```r
anova(rich.cmp)
```

```
##                 numDF denDF   F-value p-value
## (Intercept)         1   343 255.02615  <.0001
## plot_type           1    12   0.73483  0.4081
## year                1   343   1.08893  0.2974
## plot_type:year      1   343   3.80980  0.0518
```

```r
# Make cleaner ANOVA table #
set.caption("RMANOVA for Portal")
pander(anova(rich.rm))
```

Table 1: RMANOVA for Portal

	numDF	denDF	F-value	p-value
(Intercept)	1	343	256.2	0
plot_type	1	12	0.7227	0.4119
year	1	343	1.561	0.2124
plot_type:year	1	343	1.971	0.1612

```r
pander(anova(rich.cmp))
```

	numDF	denDF	F-value	p-value
(Intercept)	1	343	255	0
plot_type	1	12	0.7348	0.4081
year	1	343	1.089	0.2974
plot_type:year	1	343	3.81	0.05177

```r
# Use `lsmeans` package for time-corrected marginal means #
lsmeans(rich.rm, ~plot_type)
```

```
## NOTE: Results may be misleading due to involvement in interactions
```

```
##  plot_type           lsmean        SE df lower.CL upper.CL
##  Control           3.352478 0.2644674 13 2.781131 3.923825
##  Rodent Exclosure 3.002943 0.3066216 12 2.334872 3.671014
##
## Confidence level used: 0.95
```

```r
lsmeans(rich.cmp, ~plot_type)
```

```
## NOTE: Results may be misleading due to involvement in interactions
##  plot_type          lsmean        SE df lower.CL upper.CL
##  Control          3.348317 0.2649103 13 2.776013 3.920621
##  Rodent Exclosure 2.997377 0.3064250 12 2.329735 3.665020
##
## Confidence level used: 0.95
```

Question 4: Describe the results from your RM-ANOVA.

a. In your own words describe what a RM-ANOVA test is doing
b. Is there a noticeable trend in the inverse of Simpson's diversity over time?
c. What does the result of your F-test tell you?
d. Of the three RM-ANOVA models with different covariance structures, which one is best? How does this affect the interpretation of your data?

   Answer 4a: The RM-ANOVA test is used to test the equality of means for data sets when sample measurements are not independent from one another. ***Answer 4b***: There is alot of flucuation in the richness (invsimpson) for both control and exclosure treatments, but they behave similarily and seem to not exhibit any noticable trend. ***Answer 4c***: The F-test tells me that plot treatment/type, year, or the interaction of both year and plot type is not a significant predictor of species diversity. ***Answer 4d***: Of the two F tests I performed using the AR(1) and compound symmetry co variance structures, I found that the compound covariance structure worked the best. This might imply that all distinct samples within a treatment are correlated with one another in our dataset, while a autoregressive co variance structure implies that the weighted sum of past values will affect future values. ## 6) TEMPORAL BETA DIVERSITY

Turnover

In the R code chunk below, do the following:

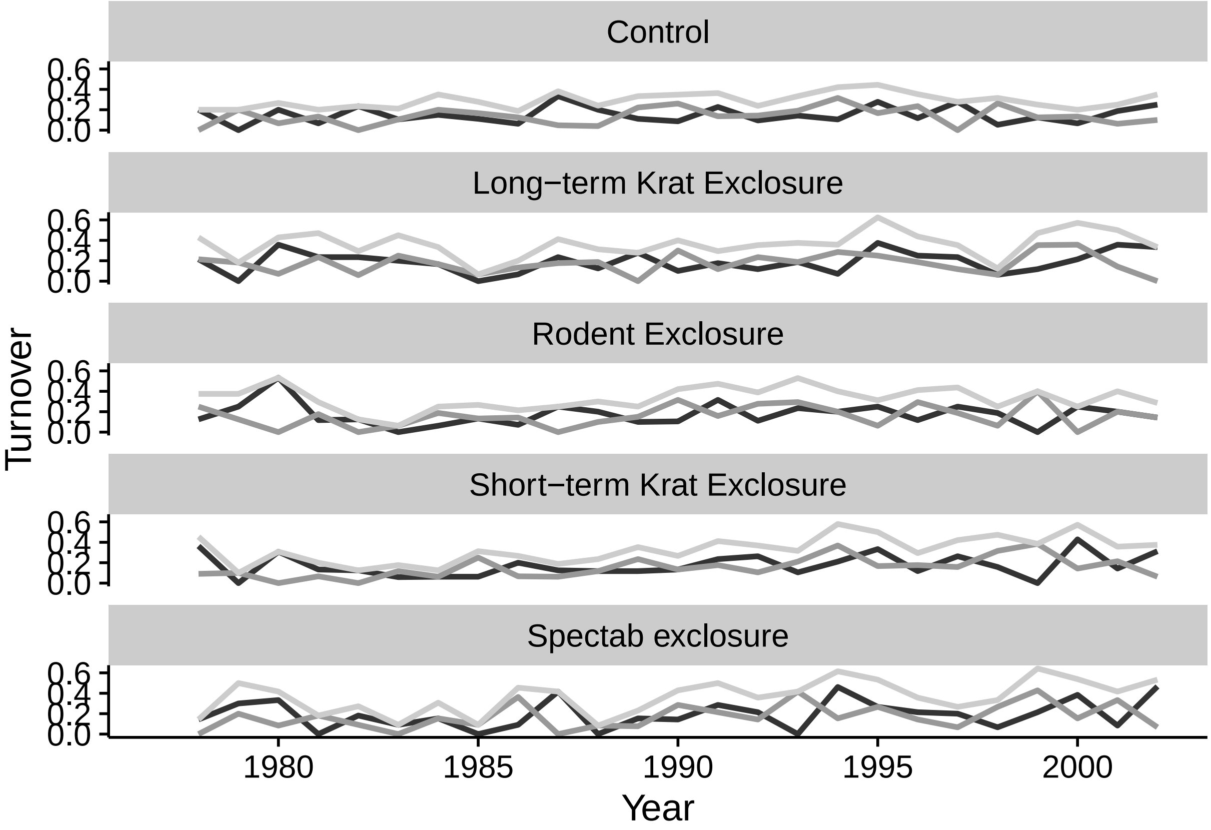
1. Calculate species abundances for each taxonomic group (the `taxa` column).
2. Calculate total turnover and turnover due to the gain/loss of species for each group.
3. Visualize turnover within each group

```r
# Calculate species abundances for each taxonomic group #
portal.species.abunds <- group_by(portal, year, plot_type) %>% count(taxon)
# Calculate total turnover #
portal.total <- turnover(df = portal.species.abunds,
                         time.var = "year",
                         species.var = "taxon",
                         abundance.var = "n",
                         replicate.var = "plot_type",
                         metric = "total")
# Calculate species gained #
portal.appearance <- turnover(df = portal.species.abunds,
                              time.var = "year",
                              species.var = "taxon",
                              abundance.var = "n",
                              replicate.var = "plot_type",
                              metric = "appearance")
# Calculate species lost #
portal.disappearance <- turnover(df = portal.species.abunds,
                                 time.var = "year",
                                 species.var = "taxon",
                                 abundance.var = "n",
                                 replicate.var = "plot_type",
                                 metric = "disappearance")
```

```
# Use join from dplyr to join columns by shared year and plot type colums #
portal.turnover <- full_join(portal.total, portal.disappearance) %>% full_join(portal.appearance)
```

```
## Joining, by = c("year", "plot_type")
## Joining, by = c("year", "plot_type")
```

```
portal.turnover <- gather(portal.turnover, key = metric,
                          value = turnover, total, appearance, disappearance)
turn.plot <- ggplot(
  portal.turnover, aes(x = year, y = turnover, color = metric)) +
  geom_line(size = 1, show.legend = T) +
  facet_wrap(~plot_type, ncol = 1) +
  xlim(1977, 2002) +
  xlab("Year") +
  ylab("Turnover") +
  theme(legend.position = "Bottom") +
  scale_color_grey()
plot(turn.plot)
```



Question 5:

a. How does temporal turnover relate to spatial turnover?
b. Which taxonomic group appears to be the most variable? Which group appears to be the least variable?

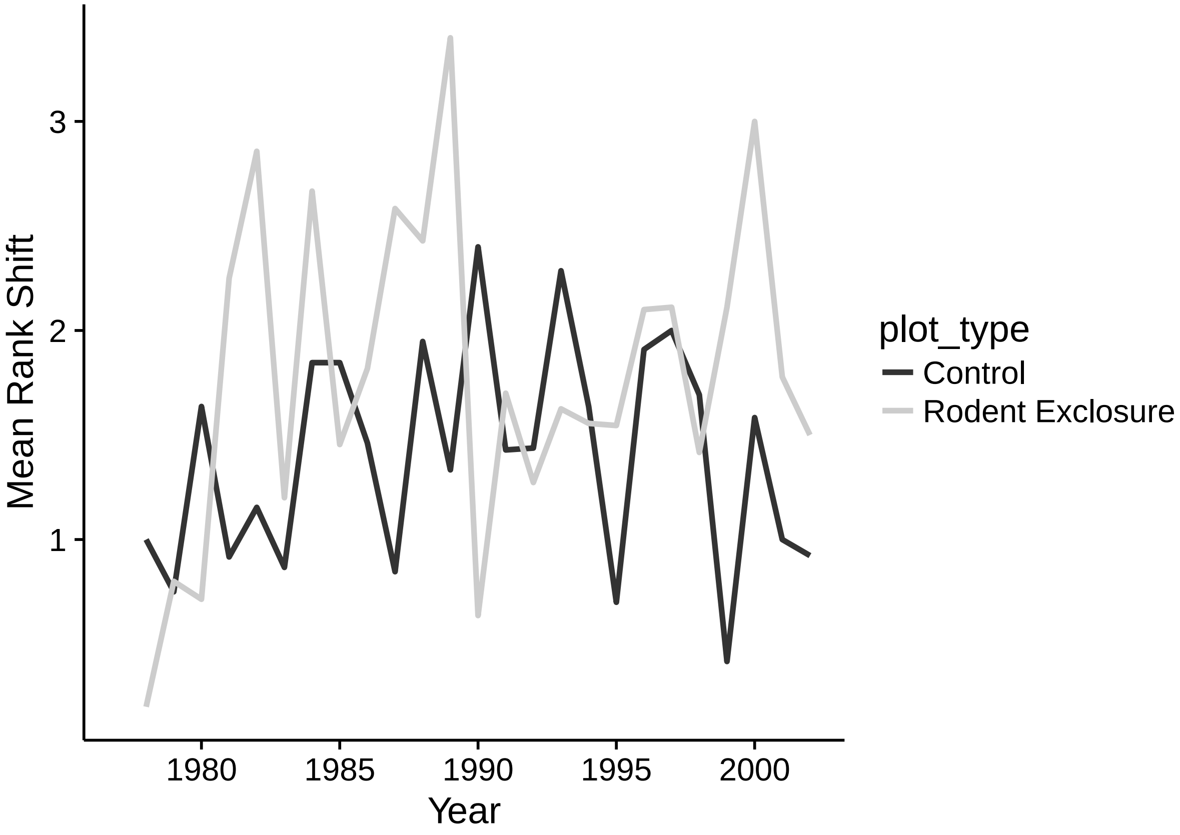
> ***Answer 5a***: Temporal turnover is the sum of species gained and lost over total species in both time points over a particular space - In other words, it is the change of species composition over time. ***Answer 5b***: The rodent exclosure seems to be the most variable, while the short-term Krat exclosure seems to be the least variable.

Mean Rank Shift

In the code chunk below, do the following:

1. Choose two plot_types or two plot_ids and compare the mean rank shift (MRS) between them.
2. Plot MRS for each through time.

```
# Pull out two treatments we analyzed eariler #
portal.abunds.cont.rodent <- filter(portal.species.abunds,
                                    plot_type == "Control" | plot_type == "Rodent Exclosure")
# Calculate MRS #
portal.rankshift <- rank_shift(
  df = as.data.frame(portal.abunds.cont.rodent),
  time.var = "year",
  species.var = "taxon",
  abundance.var = "n",
  replicate.var = "plot_type")
# Replace the year range with a single value to plot #
portal.rankshift$year <- as.numeric(substr(portal.rankshift$year_pair, 6, 9))
# Create ggplot #
rankshift.plot <- ggplot(portal.rankshift, aes(x = year, y = MRS, color = plot_type)) +
  geom_line(size = 1) +
  xlim(1977, 2002) +
  xlab("Year") +
  ylab("Mean Rank Shift") +
  scale_color_grey()
plot(rankshift.plot) #Interesting, higher mean rank shifting in rodent exclosure #
```



```
# Does one plot type show higher or lower MRS on average? Let's find out #
group_by(portal.rankshift, plot_type) %>%
  summarise(
    mean = mean(MRS),
    cv = sd(MRS)/mean)
```

```
## # A tibble: 2 × 3
##          plot_type     mean        cv
##              <chr>    <dbl>     <dbl>
## 1          Control 1.400675 0.3759483
## 2 Rodent Exclosure 1.788980 0.4361809
```

Question 6:

a. What does a change in the rank shift tell you about the community?
b. Interpret the analysis and figure you just made.

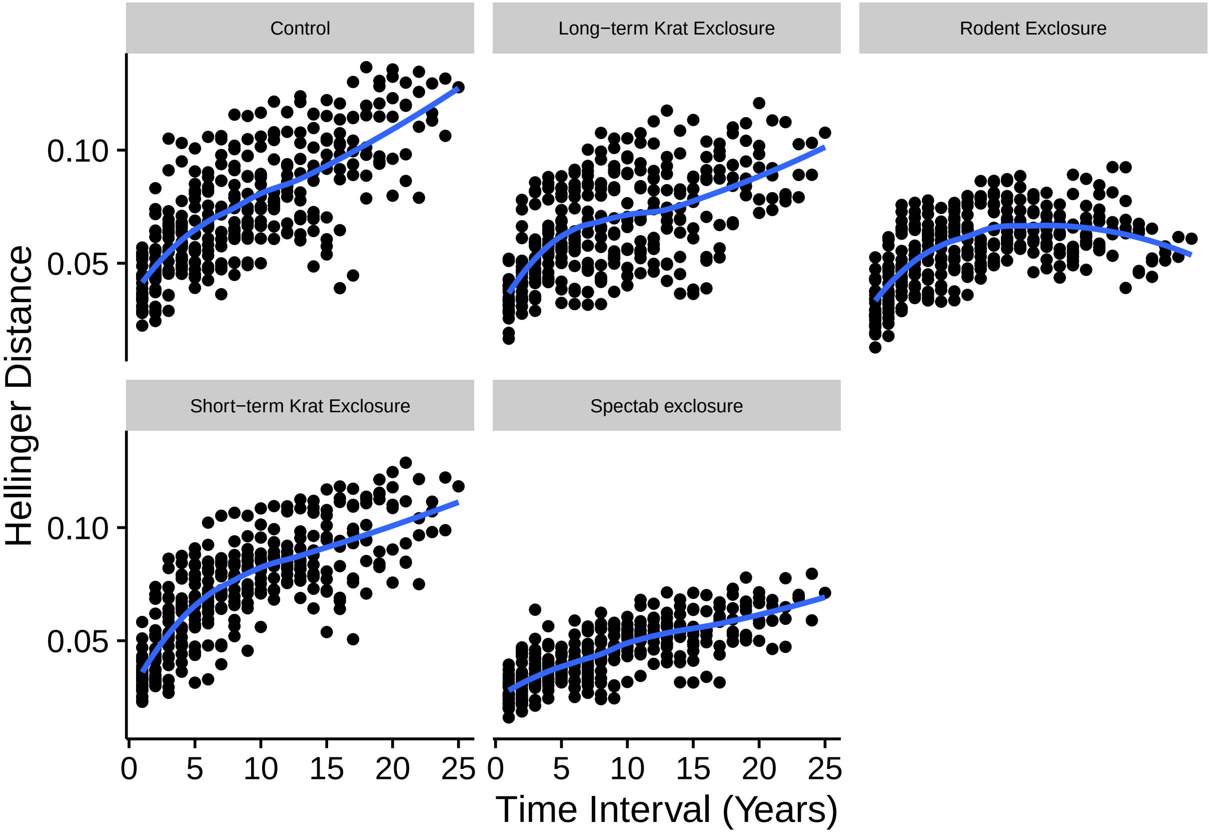
> ***Answer 6a***: A change in the rank shift tells me that commonness and rarity of taxa have changed - The higher the MRS index, the greater the change. ***Answer 6b***: From the figure and analysis, it is clear that Rodent exclosure treatment experienced greater changes in commonness and rarity of taxa across time.

Rate Change Interval

In the R code chunk below, do the following:

1. Calculate the rate change interval using the Hellinger distance.
2. Plot the results.

```
# First calculate the relative abundances #
# Calculate total abundance and add as column #
# Relativize species abundances across the whole dataset so the transformed distances are preserved #
portal.species.abunds$tot.abund <- rep(sum(portal.species.abunds$n),
                                       length(portal.species.abunds$n))
# Apply Hellinger transfrmation #
portal.hellinger.transf <- portal.species.abunds %>%
  mutate(hellinger.transf = sqrt(n / tot.abund))
# Use new column as our abundance vector #
portal.change.int <- rate_change_interval(portal.hellinger.transf,
                                          time.var = "year",
                                          species.var = "taxon",
                                          abundance.var = "hellinger.transf",
                                          replicate.var = "plot_type")
rate.plot <- ggplot(portal.change.int, aes(interval, distance)) +
  geom_point() +
  facet_wrap(~plot_type) +
  theme(strip.text.x = element_text(size = 7)) +
  stat_smooth(method = "loess", se = F, size = 1) +
  ylab("Hellinger Distance") +
  xlab("Time Interval (Years)")
rate.plot
```



Question 7:

a. What does it mean to calculate a distance metric across varying time intervals?
b. Interpret the overall results. Develop a hypothesis based on the different responses of each treatment.

> ***Answer 7a***: To calculate a distance metric across varying time intervals is to see how a community changes over time - To see how a community has diverged. ***Answer 7b***: Treatments besides the rodent exclosure increased in hellinger distance over time intervals, meaning that communities diverged across time, some faster and some slower. I hypothesize that excluding rodents from communities will prevent community divergence across time because rodent taxa may play the role of a keystone species?

7) STABILITY

In the R code chunk below, do the following:

1. Using total abundance as your focal variable, calculate stability (i.e., 1/CV) and synchrony for each plot type.
2. Test for a biodiversity-stability relationship by regressing community stability on mean richness.
3. Test for a biodiversity-stability relationship by regressing community stability on mean inverse Simpson's diversity.

```r
# Calculate stability within each plot type #
portal.stab <- community_stability(df = as.data.frame(portal.species.abunds),
                                   time.var = "year",
                                   abundance.var = "n",
                                   replicate.var = "plot_type")
```

```r
pander(portal.stab)
```

plot_type	stability
Control	3.044
Long-term Krat Exclosure	1.865
Rodent Exclosure	1.864
Short-term Krat Exclosure	2.462
Spectab exclosure	2.911

```r
# Calculate synchrony for each plot type #
portal.loreau <- synchrony(df = as.data.frame(portal.species.abunds),
                           time.var = "year",
                           species.var = "taxon",
                           abundance.var = "n",
                           replicate.var = "plot_type",
                           metric = "Loreau")
names(portal.loreau)[2] <- "loreau"
portal.gross <- synchrony(df = as.data.frame(portal.species.abunds),
                          time.var = "year",
                          species.var = "taxon",
                          abundance.var = "n",
                          replicate.var = "plot_type",
                          metric = "Gross")
names(portal.gross)[2] <- "gross"
pander(full_join(portal.loreau, portal.gross))
```

```
## Joining, by = "plot_type"
```

plot_type	loreau	gross
Control	0.1869	0.1263
Long-term Krat Exclosure	0.1578	0.07418
Rodent Exclosure	0.187	0.1423
Short-term Krat Exclosure	0.08773	0.001546
Spectab exclosure	0.1542	0.1393

```r
# Biodiversity-stability relationships #
# Summarise average annual richness in each plot type #
portal.mean.rich.plot <- rich.all %>%
  group_by(plot_id) %>%
  summarise(mean.rich = mean(richness))

# How does stability metrics relate to mean richness? #
portal.plot.abunds <- as.data.frame(
  group_by(portal, year, plot_id) %>% count(taxon))
portal.stab.plot <- community_stability(df = portal.plot.abunds,
                                        time.var = "year",
                                        abundance.var = "n",
                                        replicate.var = "plot_id")
# Join richness and stability #
portal.div.stab <- portal.mean.rich.plot %>%
  inner_join(portal.stab.plot)
```

```
## Joining, by = "plot_id"
```

```
# plot the relationship #
par(mar = c(5,5,1,1))
plot(portal.div.stab$stability ~ portal.div.stab$mean.rich,
     xlab = "", ylab= "", yaxt = "n", xaxt = "n",
     lim = c(2,10), ylim = c(1,4))
```

```
## Warning in plot.window(...): "lim" is not a graphical parameter
```

```
## Warning in plot.xy(xy, type, ...): "lim" is not a graphical parameter
```

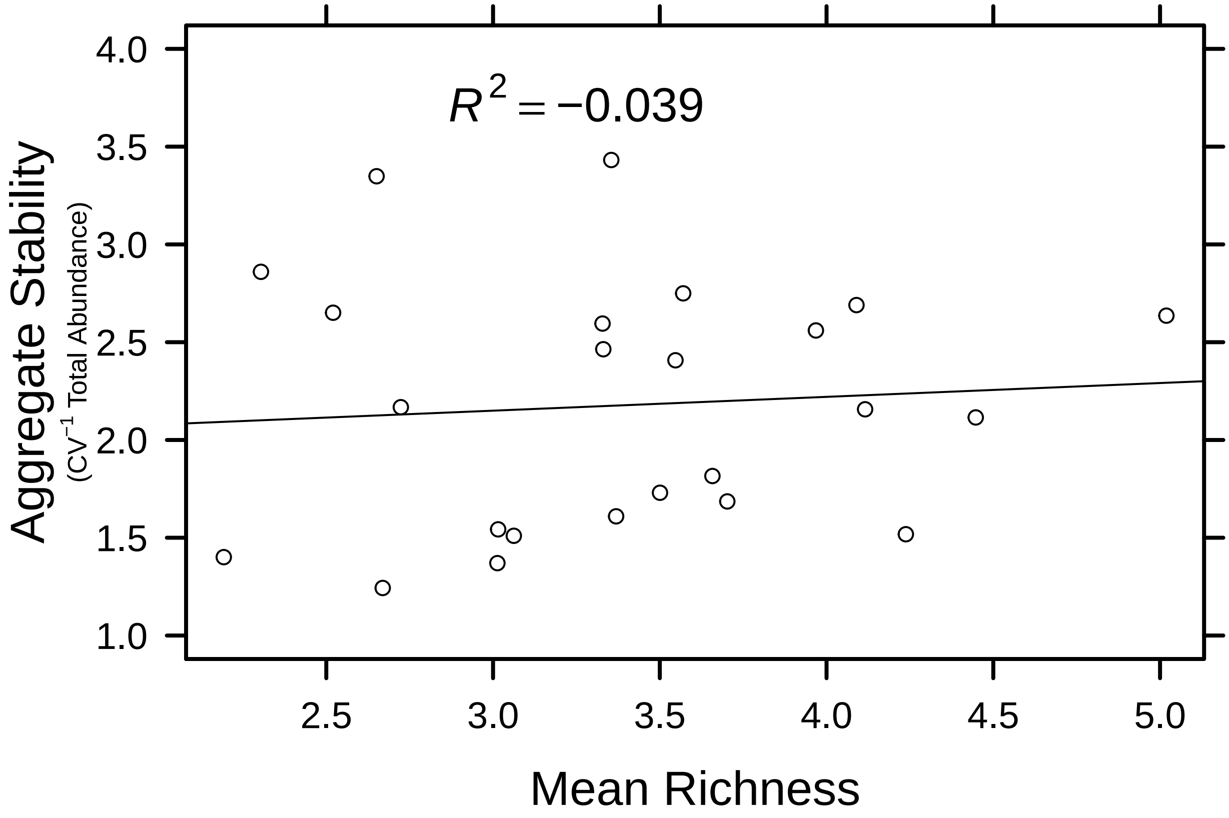
```
## Warning in axis(side = side, at = at, labels = labels, ...): "lim" is not a
## graphical parameter
```

```
## Warning in axis(side = side, at = at, labels = labels, ...): "lim" is not a
## graphical parameter
```

```
## Warning in box(...): "lim" is not a graphical parameter
```

```
## Warning in title(...): "lim" is not a graphical parameter
```

```
axis(side = 1, cex.axis = 1.2, lwd.ticks = 2, las =1)
axis(side = 2, cex.axis = 1.2, lwd.ticks = 2, las =1)
axis(side = 3, lwd.ticks = 2, las = 1, labels = F)
axis(side = 4, lwd.ticks = 2, las = 1, labels = F)
box(lwd = 2)
mtext("Mean Richness", side =1, line = 3, cex = 1.5)
mtext("Aggregate Stability", side =2, line = 3.5, cex = 1.5)
mtext(expression(paste("(CV"^"-1"," Total Abundance)")),
      side = 2, line = 2.25, cex = 0.8)

div.stab.lm <- lm(portal.div.stab$stability ~ portal.div.stab$mean.rich)
abline(div.stab.lm)
r2 <- bquote(italic(R)^2 == .(format(
  summary(div.stab.lm)$adj.r.square, digits = 3)))
text(3.25, 3.75, cex = 1.5, labels = r2)
```



Question 8:

a. Which plot type has the highest stability in total abundance? How is stability of total abundance measured with the function you learned? How does this measure of stability relate to the coefficient of variation?

b. In your own words, describe the concept of synchrony

c. Interpret the results from the biodiversity-stability relationships you analyzed.

> ***Answer 8a***: The plot type with the highest stability in total abundance is Control. The stability is the inverse of taking the variance over the mean value of a community. ***Answer 8b***: Synchrony is a measure of independence in population density fluctuation. ***Answer 8c***: It shows that as mean richness increases, the aggregate stability postivity increases.

SYNTHESIS

Compare and contrast the core concepts from temporal and spatial diversity (e.g., autocorrelation, scale, variability, etc.). Identify a few of the major challenges associated with studying biodiversity through time and across space.

> ***Answer***: It seems that the biggest challenges assoicated with studying biodiversity through time and space is the issue of independence. Making sure to account for the fact that measurements of samples throughout small space or short time point are not independent to one another. For example: partitioning out the effect of enviroment and spatial autocorrleation (spatial diversity), or factoring in stationarity assumptions, variability, and lag time (temporal diversity). Generally, it seems that in both temperal and spatial diversity, partitioning out the autocorrelation of time

and space and considering the variability and seasonality is important for determining species turnover (beta diversity).