Assignment: Temporal Diversity

Katie Beidler; Z620: Quantitative Biodiversity, Indiana University

14 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.

```
clr = function() {
  ENV = globalenv()
  ll = ls(envir = ENV)
  ll = ll[ll != "clr"]
  rm(list = ll, envir = ENV)
}
```

```
getwd()
setwd("/Users/bhbeidler/GitHub/QB2017_Beidler/Week5-Temporal")

package.list = c('vegan', 'tidyr', 'dplyr', 'codyn', 'ggplot2',
'cowplot', 'MullerPlot', 'RColorBrewer', 'reshape2', 'lubridate',
'TTR', 'xtable', 'multcomp', 'pander', 'png', 'grid', 'tseries', 'nlme', 'forecast', 'lsmeans')
for (package in package.list) {
if (!require(package, character.only = TRUE, quietly = TRUE)) {
install.packages(package, repos='http://cran.us.r-project.org')
library(package, character.only = TRUE) }
}
```

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.

```
portal = read.table("data/combined.csv", sep = ",", header = TRUE)
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 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 ...
```

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

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

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 in 'portal' and 5 plot exclosure treatments ***Answer 1b***: There are 48 species in the 'portal' dataset

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).

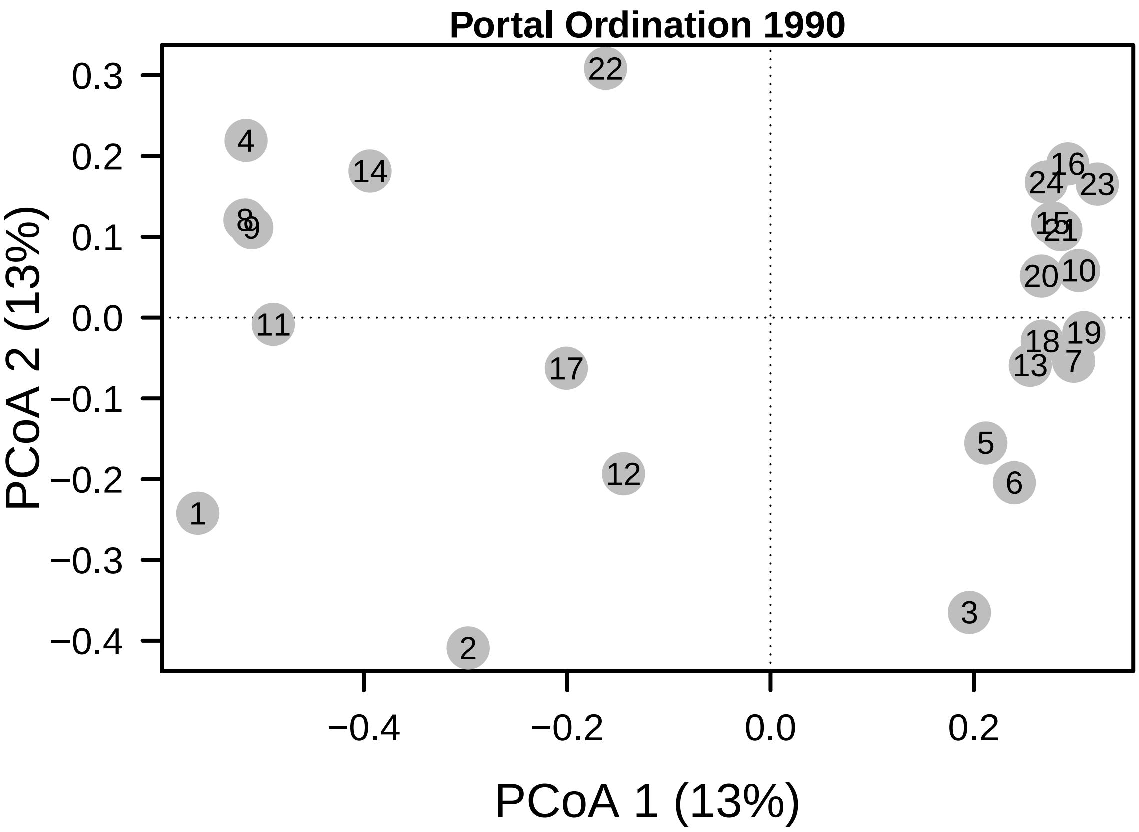
```r
# 1. Create a site by species matrix for any year of your choosing
portal_90_sbys = portal %>%
                 filter(year == 1990) %>%
                 group_by(plot_id) %>%
                 count(species_id) %>%
                 spread(key = species_id, value = n, fill = 0)


#2. Create a vector of plot_type for sites in the site-by-species matrix.
portal_90_sbyex = portal %>%
                    filter(year == 1990) %>%
                    group_by(plot_id, plot_type) %>%
                    count(species_id) %>%
                    spread(key = species_id, value = n, fill = 0)
portal_90_sbyex  = portal_90_sbyex[ ,-1]

ptype = portal_90_sbyex$plot_type
#3. Analyze alpha diversity (e.g., Shannon/Simpson) across the sites for that year.
diversity(portal_90_sbys, index = "shannon")
```

```
##         1         2         3         4         5         6         7
## 1.4259807 1.5715993 2.0057095 0.4633228 1.8585794 2.0918316 1.0221082
##         8         9        10        11        12        13        14
## 0.8179112 0.9194273 1.0194536 1.3371851 2.1579890 1.7833181 1.1351565
##        15        16        17        18        19        20        21
## 1.7464857 0.8760058 1.8253491 2.0266380 1.6608797 1.8586870 1.3840100
##        22        23        24
## 1.3852651 1.3438213 1.4298799
```

```r
diversity(portal_90_sbys, index = "simp")
```

```
##         1         2         3         4         5         6         7
## 0.6700139 0.6687733 0.8286734 0.2181120 0.7878788 0.8478002 0.5589849
##         8         9        10        11        12        13        14
## 0.3988715 0.4443213 0.5693297 0.6406250 0.8654514 0.8059808 0.5479106
##        15        16        17        18        19        20        21
## 0.7784352 0.5408163 0.8021875 0.8361082 0.7823844 0.7862426 0.6934156
##        22        23        24
## 0.7074911 0.6568998 0.6597531
```

```r
#4. Create a PCoA ordination of your site-by-species matrix.
# Ordination

# construct a resemblance matrix based on Bray-Curtis Distance for sbys 1990
portal_90_sbys.db = vegdist(portal_90_sbys, method = "bray")
# Perform a Principal Coordinates Analysis to visualize beta-diversity
portal_90_sbys.pcoa = cmdscale(portal_90_sbys.db, eig = TRUE, k = 3)

# Calculate the variation explained by the first three axes in your ordination
explainvar = round(portal_90_sbys.pcoa$eig[1] / sum(portal_90_sbys.pcoa$eig), 3) * 100
explainvar = round(portal_90_sbys.pcoa$eig[2] / sum(portal_90_sbys.pcoa$eig), 3) * 100
explainvar = round(portal_90_sbys.pcoa$eig[3] / sum(portal_90_sbys.pcoa$eig), 3) * 100
sum.eig = sum(explainvar, explainvar, explainvar)

# plot the PCoA ordination
# Define Plot Parameters
par(mar = c(5, 5, 1, 2) + 0.1)
# Initiate Plot
plot(portal_90_sbys.pcoa $points[ ,1], portal_90_sbys.pcoa $points[ ,2],
xlab = paste("PCoA 1 (", explainvar, "%)", sep = ""),
ylab = paste("PCoA 2 (", explainvar, "%)", sep = ""),
main = "Portal Ordination 1990",pch = 16, cex = 2.0, type = "n", cex.lab = 1.5, cex.axis = 1.2, axes = 
# 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(portal_90_sbys.pcoa $points[ ,1], portal_90_sbys.pcoa $points[ ,2],
pch = 16, cex = 3, bg = "gray", col = "gray")
text(portal_90_sbys.pcoa$points[ ,1], portal_90_sbys.pcoa$points[ ,2],
labels = row.names(portal_90_sbys.pcoa$points))
```



```
#5. Using the hypothesis testing tools you learned in the beta-diversity module, test the hypothesis th

PORTALPERMA.90 = adonis(portal_90_sbys  ~ ptype, method = "bray", permutations = 999)
PORTALPERMA.90
```

```
##
## Call:
## adonis(formula = portal_90_sbys ~ ptype, permutations = 999,      method = "bray")
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## ptype      4    2.8656 0.71640  5.8206 0.55064  0.001 ***
## Residuals 19    2.3385 0.12308         0.44936
## Total     23    5.2041                 1.00000
## ---
## 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: Diveristy appears to differ among sites. The PCoA plot shows that the majority of conrtol sites are grouped on the left side of the plot. ***Answer 2b***: Plot treatment type is a significant predictor of site dissimilarity (P=0.001, R2 = 0.55)

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.

```
# 1. Create a time-by-species matrix that includes year, month, and plot_id for a site other than plot_
# Make a date vector that contains year, month, and day
portal = unite(portal, col = date, c(year, month, day), sep = "-", remove = FALSE)

# Make a taxon vector that contains genus and species names
portal = unite(portal, col = taxon, c(genus, species), sep = "_", remove = FALSE)

time.by.species = portal %>%
                  group_by(year, plot_id) %>%
                  count(taxon) %>%
                  spread(key = taxon, value = n, fill = 0)

filter(time.by.species, year == 1984)
```

```
## Source: local data frame [24 x 50]
## Groups: year, plot_id [24]
##
##     year plot_id Ammodramus_savannarum Ammospermophilus_harrisi
##    <int>   <int>                 <dbl>                    <dbl>
## 1   1984       1                     0                        0
## 2   1984       2                     0                        0
## 3   1984       3                     0                        0
## 4   1984       4                     0                        0
## 5   1984       5                     0                        0
## 6   1984       6                     0                        0
## 7   1984       7                     0                        0
## 8   1984       8                     0                        0
## 9   1984       9                     0                        1
## 10  1984      10                     0                        0
## # ... with 14 more rows, and 46 more variables:
## #   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>
```

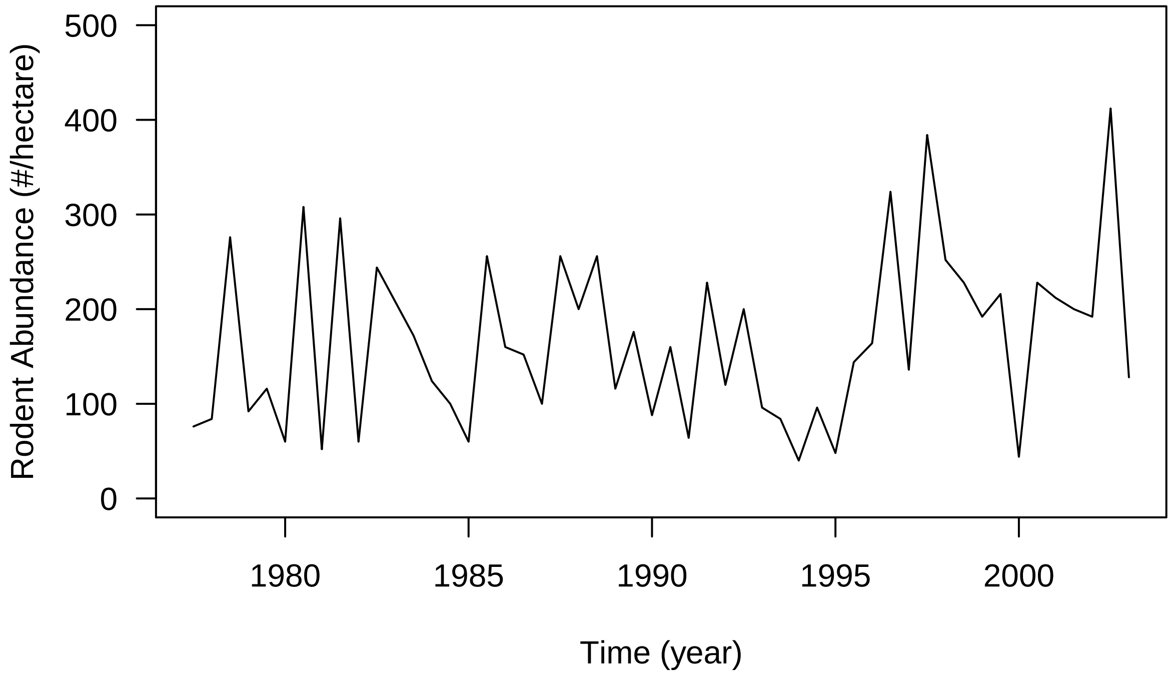
```r
filter(time.by.species, plot_id == 5)
```

```
## Source: local data frame [26 x 50]
## Groups: year, plot_id [26]
##
##     year plot_id Ammodramus_savannarum Ammospermophilus_harrisi
##    <int>   <int>                 <dbl>                    <dbl>
## 1   1977       5                     0                        0
## 2   1978       5                     0                        0
## 3   1979       5                     0                        0
## 4   1980       5                     0                        0
## 5   1981       5                     0                        0
## 6   1982       5                     0                        0
## 7   1983       5                     0                        0
## 8   1984       5                     0                        0
## 9   1985       5                     0                        0
## 10  1986       5                     0                        0
## # ... with 16 more rows, and 46 more variables:
## #   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>
```

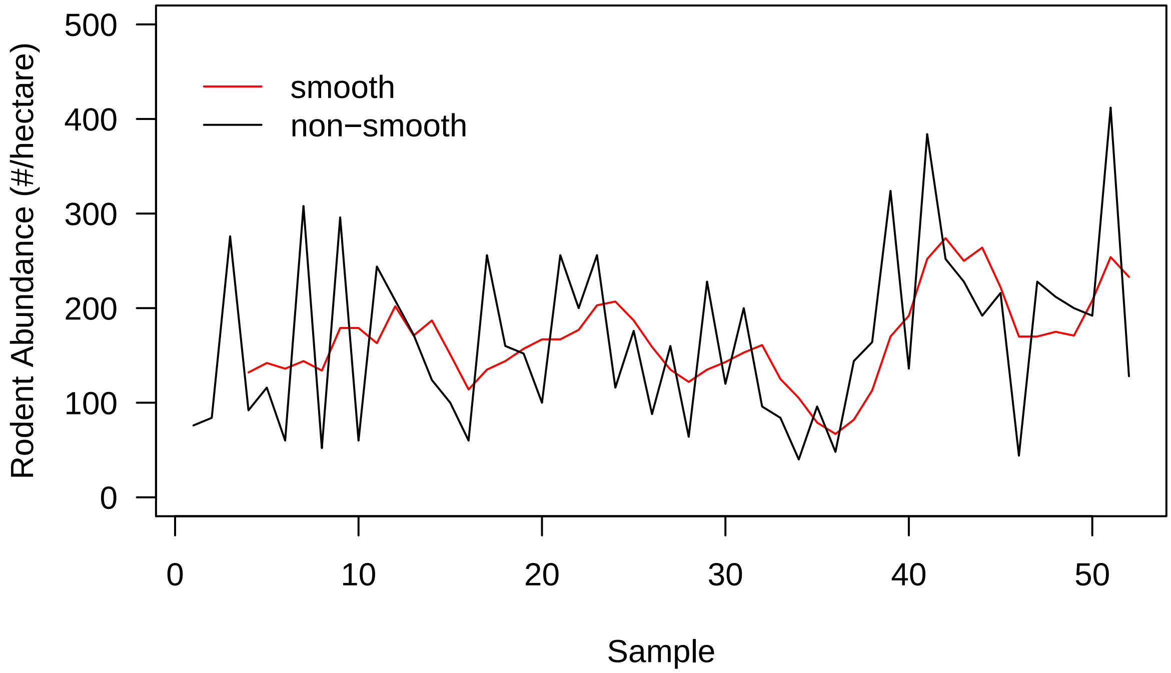
```
time.by.species = as.data.frame(time.by.species)

# 2. Examine per-hectare rodent abundance using simple moving average smoothing.
# Create a time-by-species matrix that includes year, month, and plot_id
time.by.spec.2 = filter(portal, taxa=="Rodent") %>%
                 group_by(year, month, plot_id) %>%
                 count(taxon)
# Create a 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 the 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
```

```
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, ylin
```



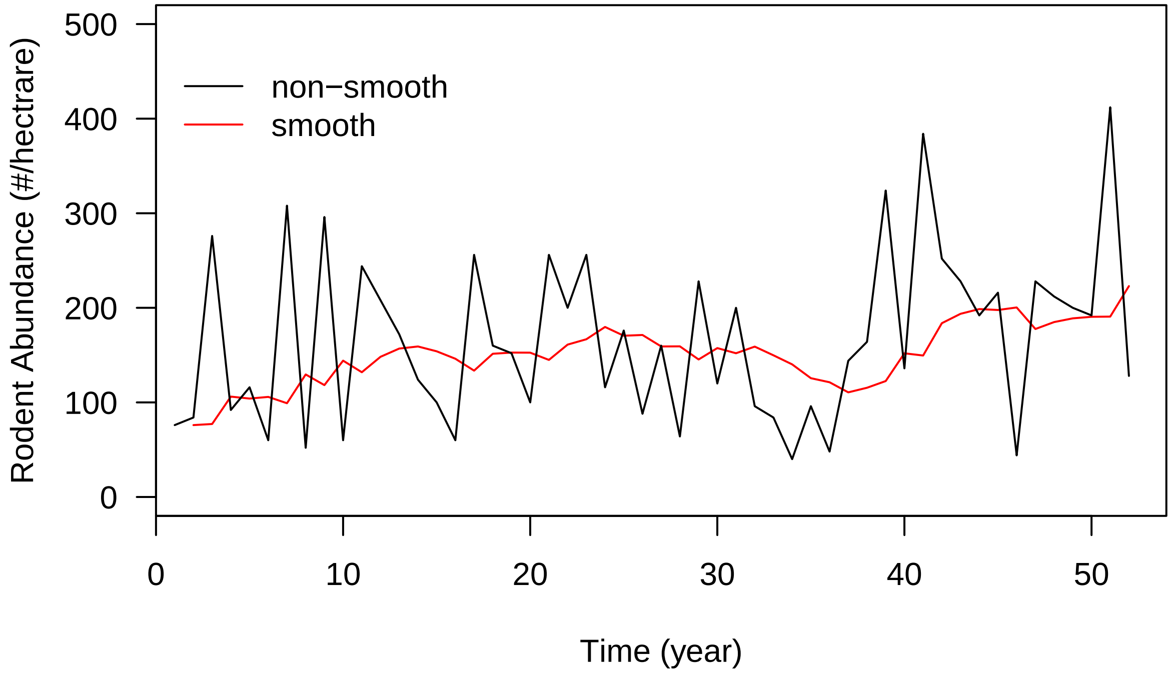
```
# Adding the moving average
abund.sm = SMA(abund$nn, n = 4)
plot(abund.sm, type = "l", col = "red", ylab = "Rodent Abundance (#/hectare)",
xlab = "Sample", las = 1, ylim = c(0, 500))
lines(abund$nn, col = "black")
legend(0, 475, col = c("red", "black"), lty = c(1,1),
c("smooth", "non-smooth"), bty = "n", cex = 1)
```



```
# The Holt-Winters filtering technique is commonly used for exponential smoothing
abund.hw = HoltWinters(abund$nn, beta = FALSE, gamma = FALSE)
abund.hw$fitted
```

```
## Time Series:
## Start = 2
## End = 52
## Frequency = 1
##         xhat     level
##  2  76.00000  76.00000
##  3  77.16537  77.16537
##  4 106.12982 106.12982
##  5 104.07152 104.07152
##  6 105.80915 105.80915
##  7  99.13608  99.13608
##  8 129.56151 129.56151
##  9 118.26305 118.26305
## 10 144.15418 144.15418
## 11 131.89535 131.89535
## 12 148.22575 148.22575
## 13 156.93313 156.93313
## 14 159.12794 159.12794
## 15 154.01081 154.01081
## 16 146.14300 146.14300
## 17 133.59446 133.59446
## 18 151.42540 151.42540
## 19 152.67448 152.67448
## 20 152.57622 152.57622
## 21 144.91739 144.91739
## 22 161.09891 161.09891
## 23 166.76568 166.76568
## 24 179.76454 179.76454
## 25 170.47589 170.47589
## 26 171.28059 171.28059
## 27 159.14902 159.14902
## 28 159.27298 159.27298
## 29 145.39446 145.39446
## 30 157.42770 157.42770
## 31 151.97557 151.97557
## 32 158.97134 158.97134
## 33 149.79824 149.79824
## 34 140.21334 140.21334
## 35 125.61515 125.61515
## 36 121.30108 121.30108
## 37 110.62323 110.62323
## 38 115.48526 115.48526
## 39 122.55246 122.55246
## 40 151.89754 151.89754
## 41 149.58173 149.58173
## 42 183.72969 183.72969
## 43 193.67470 193.67470
## 44 198.67490 198.67490
## 45 197.70256 197.70256
## 46 200.36797 200.36797
## 47 177.58968 177.58968
## 48 184.93300 184.93300
## 49 188.87588 188.87588
## 50 190.49634 190.49634
## 51 190.71538 190.71538
## 52 222.95015 222.95015
```

```
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)
```



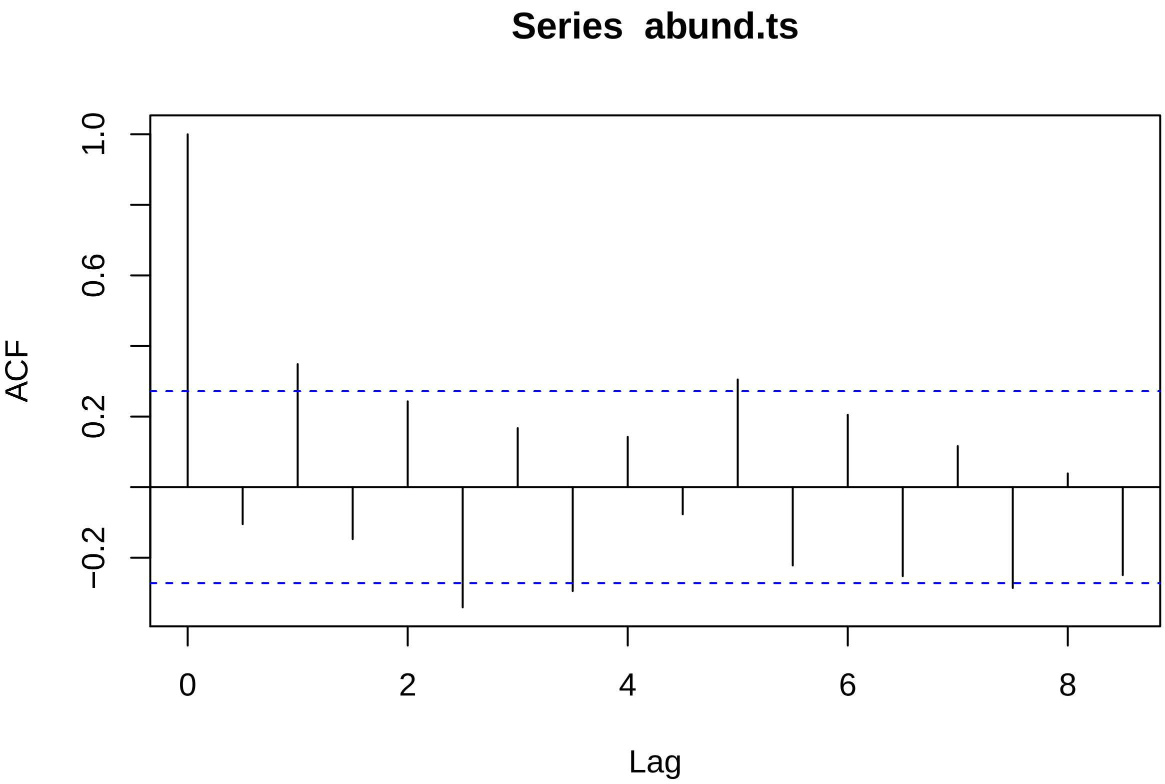
```
# 3. Test whether your data meets the assumption of stationarity.
adf.raw = adf.test(abund.ts, alternative = "stationary")
adf.raw$p.value
```

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

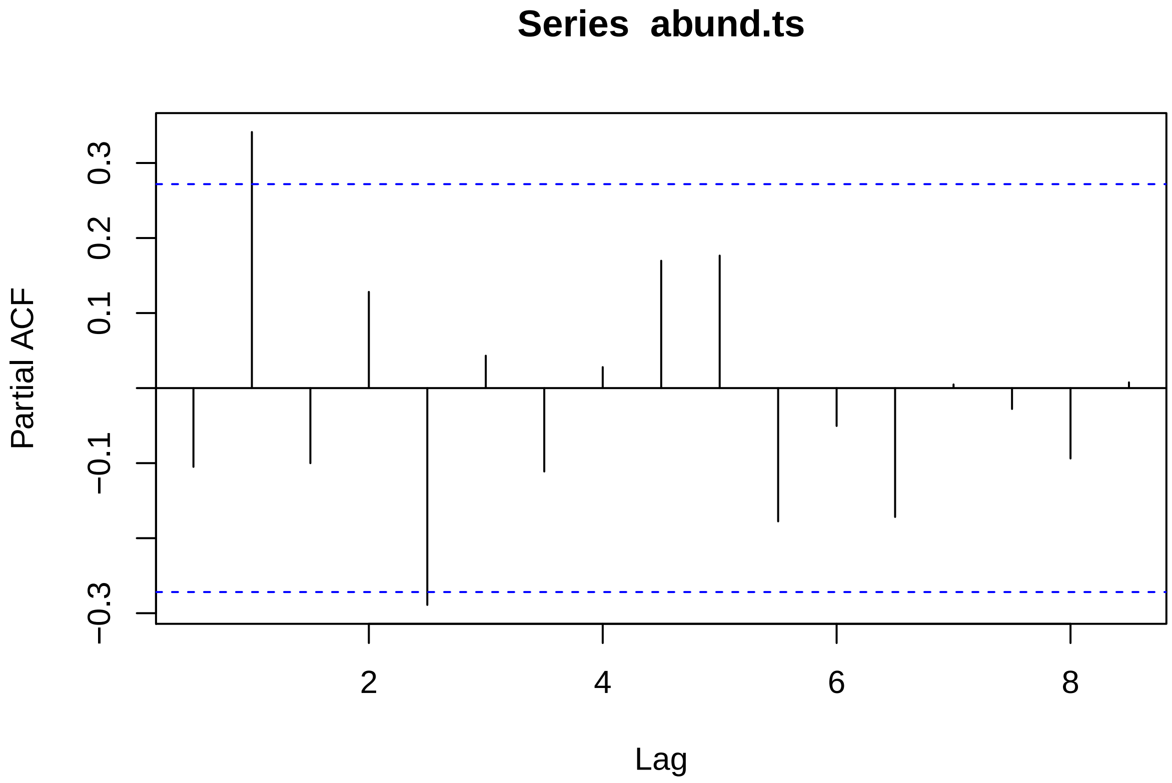
```
abund.ts.diff = diff(abund.ts)
adf.diff = adf.test(abund.ts.diff, alternative = "stationary")
adf.diff$p.value
```

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

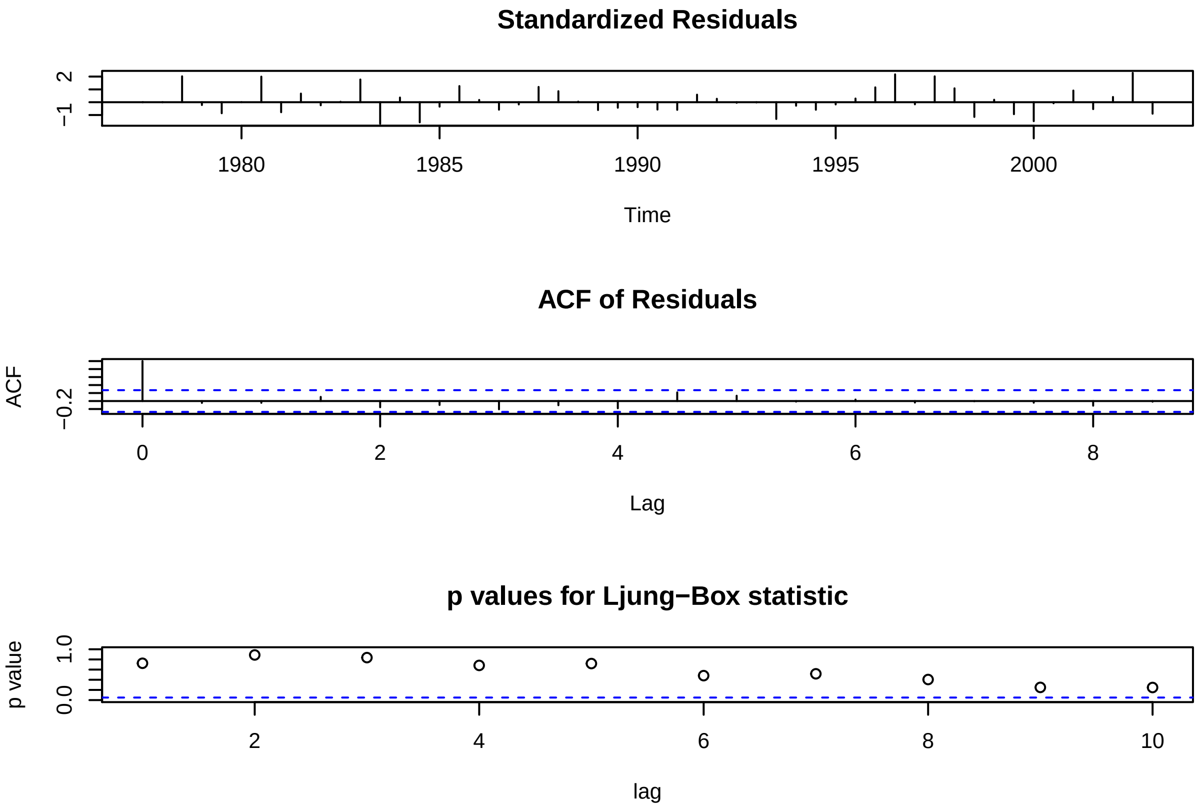
```
#  5. Examine and plot time lags using the partial autocorrelation function (PACF) and autocorrelation
acf(abund.ts)
```



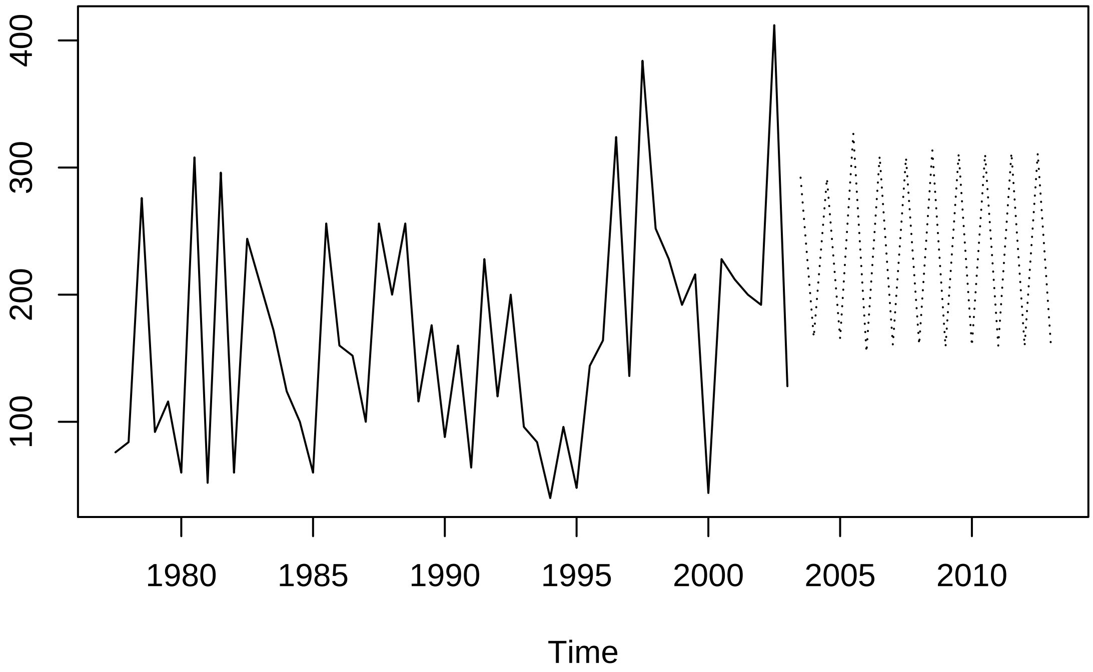
```
pacf(abund.ts)
```



```
# 6. Use the tools outlined in the Handout to create an ARMA model.
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)
```



```
pred.arm = predict(abund.arm, n.ahead = 20)
ts.plot(abund.ts, pred.arm$pred, lty = c(1,3))
```



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:The Dickey-Fuller test indicates that our data does not meet the assumption of stationarity (P>0.05). This violation implies that the mean, variance and covariance of the time series is affected by time (i.e. changes in rodent abundance over time are affected by seasonal variation in precipitation).

Answer 3b:The ACF function allows us to look at the lags in our time series analysis, specifically the lags of the forecast errors in relation to the moving average component of the ARMA model. The ACF tells us about the correlations between lag intervals in the time series. The correlations should decrease with increasing time lag- meaning that past values have less of an effect on current values. We are also using the partial autocorrelation function (PACF) in addition to the ACF function. In contrast to the ACF, the PACF tells us about lags that can be addressed with the autoregressive component of the ARMA model.

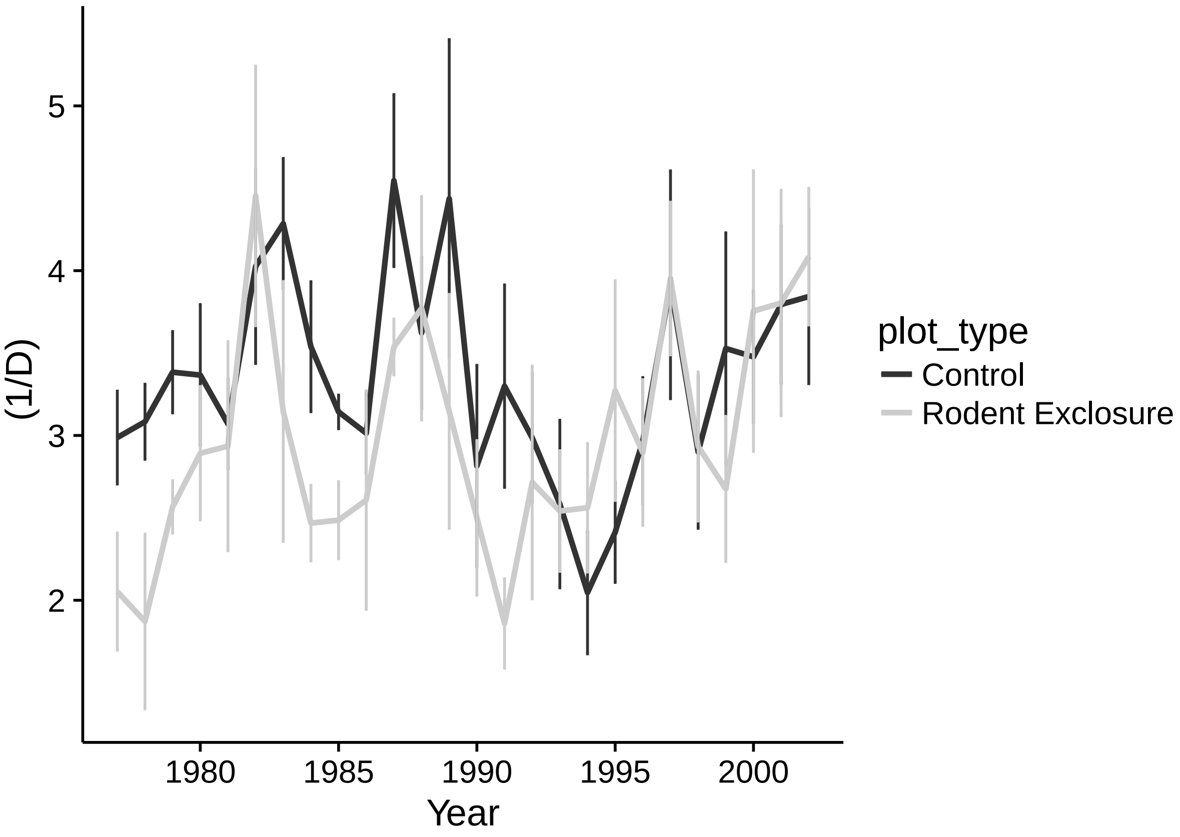
Answer 3c:The ACF tells us that there is a significant positive correlation at time lag 2- indicative of the annual precipitation cycle (rainy vs. non rainy season). Differencing the data allowed us to meet the assumptions of stationarity and therefore we can use a autoregressive integrated moving average (ARIMA) model to to remove the effect of time lag 2. Looking at the diagnostic plots for the ARIMA model, we have removed the seasonal trend in abundances due to precipitation. Forecasting rodent abundances in the future for the study plot, we can see that abundances will fluctuate around 250 and abundances will be less variable than in previous years.

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.

```r
# Construct time-by-species matrix
time.by.species = portal %>%
                  group_by(year, plot_id,plot_type) %>%
                  count(taxon) %>%
                  spread(key = taxon, value = n, fill = 0)
# 2. Calculate the inverse of Simpson's diversity for each year, and plot it as a function of year for

# Calculating Simpsons diversity index
insimp = diversity(time.by.species[,-c(1:3)], "invsimpson")

# Create data frame with experimental design and diversity data
div.all = data.frame(time.by.species[,1:3,], insimp)
# Rename column
names(div.all)[4] = "inv.simp"
# Pull out two of the five Portal treatments
div.treat = div.all[which(div.all$plot_type ==
"Control" | div.all$plot_type == "Rodent Exclosure"), ]

div.treat.plot = group_by(div.treat, plot_type, year) %>%
  summarise(mean = mean(inv.simp), sd = sd(inv.simp),n = n(),sem = sd/sqrt(n))
# avg. diversity per group
# stand. dev. per group
# num. obs. per group
# calc. std. err. mean.
div.plot = ggplot(div.treat.plot, aes(x = year, y = mean, color = plot_type)) + geom_line(size = 1, sho
                  geom_errorbar(aes(ymin = mean - sem, ymax = mean + sem), width = .1) +  xlim(1977, 20
plot(div.plot)
```



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

# AR(1) covariance structure
div.rm1 = lme(inv.simp ~ plot_type * year, random = ~ 1 | plot_id, correlation = corAR1(form = ~ 1 | pl
summary(div.rm1) # Obtain F-test
```

```
## Linear mixed-effects model fit by REML
##  Data: div.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: inv.simp ~ 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
```

```
anova(div.rm1)
```

```
##                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
```

```
# Make cleaner ANOVA table
set.caption("RMANOVA for Portal Using AR(1) covariance Structure")
pander(anova(div.rm1))
```

Table 1: RMANOVA for Portal Using AR(1) covariance Structure

	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

```
# Use `lsmeans` package for time-corrected marginal means
lsmeans(div.rm1, ~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
```

```
# compound symmetery covariance structure
div.rm2 = lme(inv.simp ~ plot_type * year, random = ~ 1 | plot_id, correlation = corCompSymm(form = ~ 1
summary(div.rm2) # Obtain F-test
```

```
## Linear mixed-effects model fit by REML
##   Data: div.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: inv.simp ~ 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
```

```
anova(div.rm2)
```

```
##                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
```

```
# Make cleaner ANOVA table
set.caption("RMANOVA for Portal Using compound symmetry covariance Structure")
pander(anova(div.rm2))
```

Table 2: RMANOVA for Portal Using compound symmetry covariance Structure

	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(div.rm2, ~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
```

```r
# unstructured covariance structure
div.rm3 = lme(inv.simp ~ plot_type *year, random = ~ 1 | plot_id, data = div.treat)
#Look at detailed output
summary(div.rm3) # Obtain F-test
```

```
## Linear mixed-effects model fit by REML
##  Data: div.treat
##        AIC      BIC    logLik
##   1213.368 1236.601 -600.6842
##
## Random effects:
##  Formula: ~1 | plot_id
##         (Intercept) Residual
## StdDev:   0.7104222 1.214275
##
## Fixed effects: inv.simp ~ 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
anova(div.rm3)
```

```
##                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 Using unstructured covariance Structure")
pander(anova(div.rm3))
```

Table 3: RMANOVA for Portal Using unstructured covariance Structure

	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(div.rm3, ~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
```

```r
# Compare the AICs
AIC(div.rm1, div.rm2, div.rm3)
```

```
##         df      AIC
## div.rm1  7 1153.217
## div.rm2  7 1215.368
## div.rm3  6 1213.368
```

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

> ***Answer 4a***: The RM-ANOVA allows you to test for the main effects of treatment, time and a treatment x time interaction by accounting for the non-independence of observations or the fact that time points are not independent of one another and that repeated measurements have been taken for the different plots.
>
> ***Answer 4b***: There is no obvious trend for inverse of Simpson's diversity (1/D) over time. 1/D fluctuates over time and tends to lower in rodent exlosure experiments in the early part of the study (1977-1980).
>
> ***Answer 4c***:The result of the F-test tells us that year, plot type and the plot type by year interaction did not have a significant effect on rodent abundance.
>
> ***Answer 4d***: The ARMA1 or the AR1 covariance structure was the best choice in this repeated measures experiment. The AR1 model had the lowest AIC value and differed from the other covariance structures. The chosen covariance structure of the random effect depends on the

experimental design (Split plots, repeated measures, multi-site trials, hierarchical linear models etc). The covariance structure can affect the interpretation of the data by influencing the results of the F-tests and potentially inflating Type I Error rates or you may falsely reject the (true) null hypothesis. For the compound symmetry and unstructured variance structures the plot type x year interaction was moderately significant (P = 0.052).

6) TEMPORAL BETA DIVERSITY

Turnover

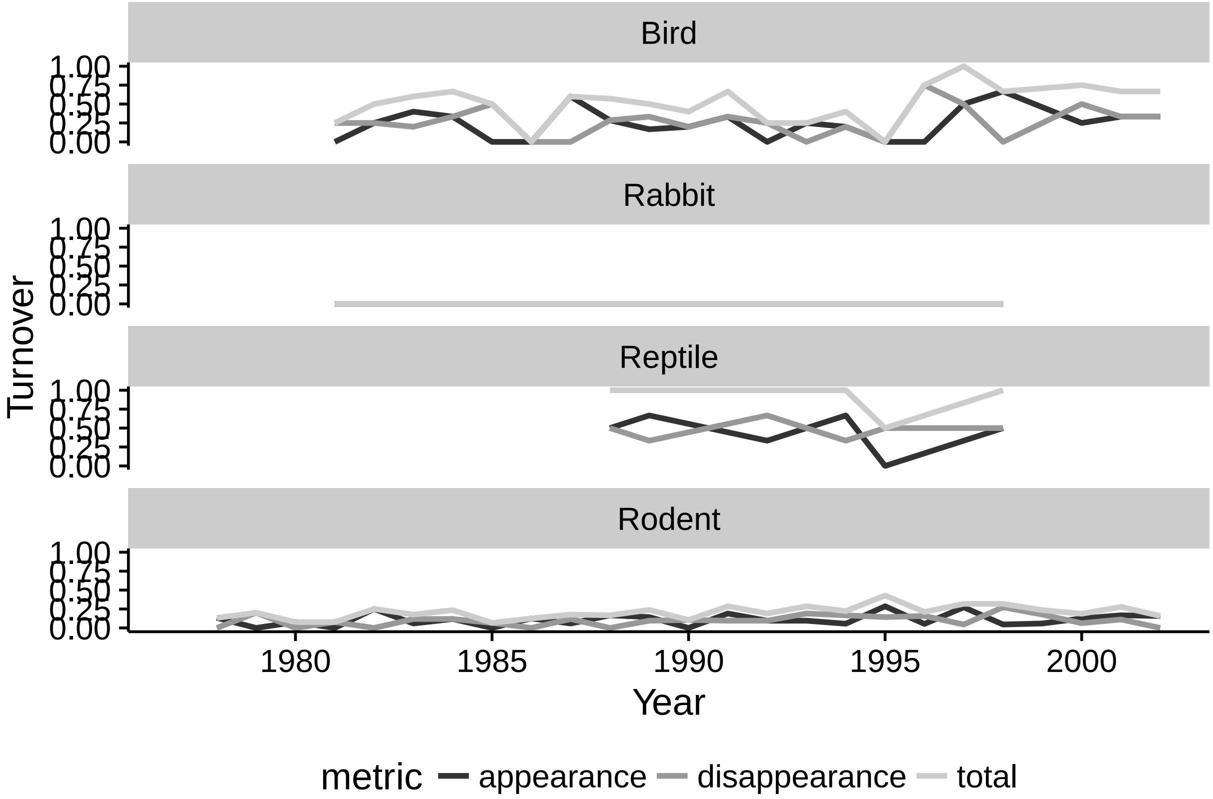
In the R code chunk below, do the following:

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

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

portal.turnover = full_join(portal.total, portal.disappearance) %>%
                  full_join(portal.appearance)
```

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

```
portal.turnover = gather(portal.turnover, key = metric, value = turnover, total, appearance, disappearan
View(portal.turnover)
# 3. Visualize turnover within each group
turn.plot = ggplot(portal.turnover, aes(x = year, y = turnover, color = metric)) +
  geom_line(size = 1, show.legend = T) + facet_wrap(~taxa, ncol = 1) +
  xlim(1977, 2002) +
  xlab("Year") +
  ylab("Turnover") +
  theme(legend.position = "bottom") +
  scale_color_grey()
plot(turn.plot)
```



```
# Low turnover is indicative of a stable community and high turnover is indicative of a dynamic communi
```

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?

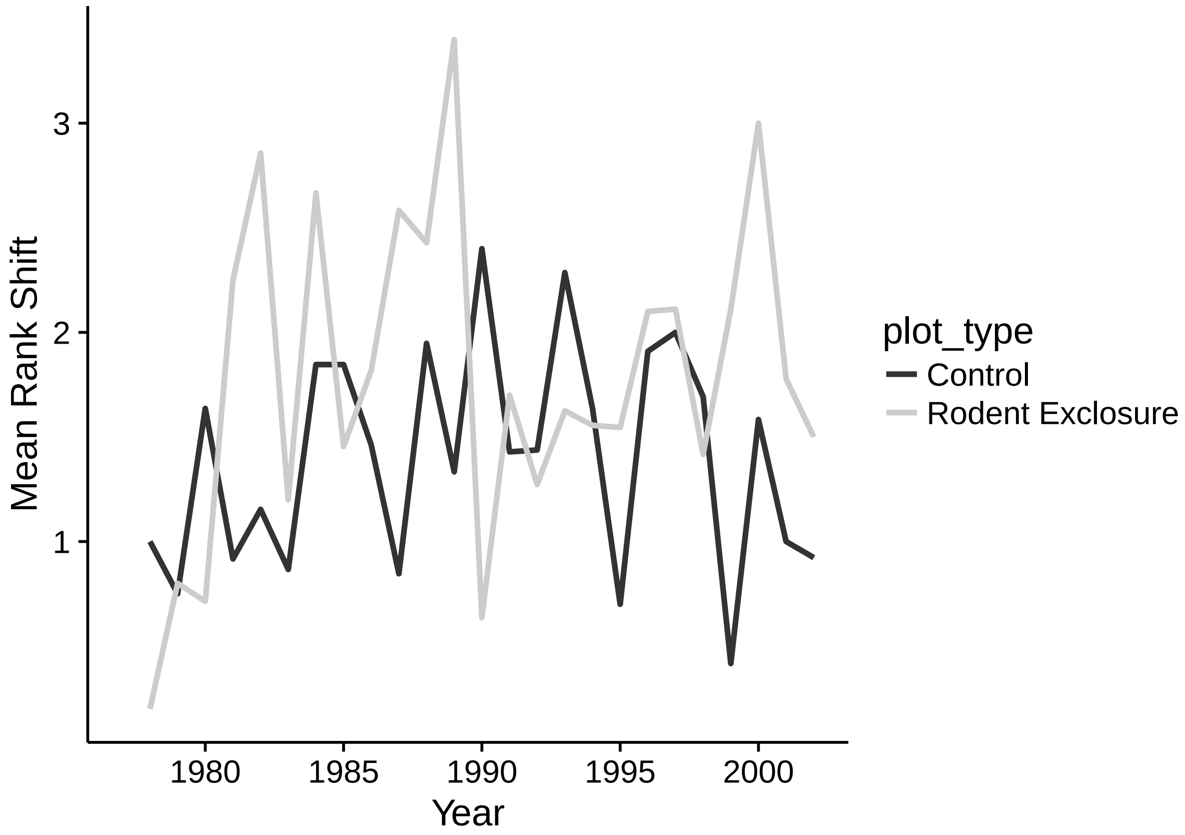
> ***Answer 5a***:Temporal turnover is a measure of change in species composition over time for one location. Spatial turnover is a measure of change in species composition between two locations for one time point (i.e. species gain or lost with cumulative area).
>
> ***Answer 5b***: Rodent communities are the least variable. While, Bird and reptile communities tend to be more 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 the two treatments we analyzed earlier
portal.species.abunds = portal %>%
                        group_by(year, plot_type) %>%
                        count(taxon)

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
xlim(1977, 2002) +
xlab("Year") +
ylab("Mean Rank Shift") + scale_color_grey()
plot(rankshift.plot)
```



```
portal.rankshift %>%
  group_by (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***: The change in the rank shift tells you whether there are changes in commonness or rarity of taxa. The higher the MRS index the greater the change.

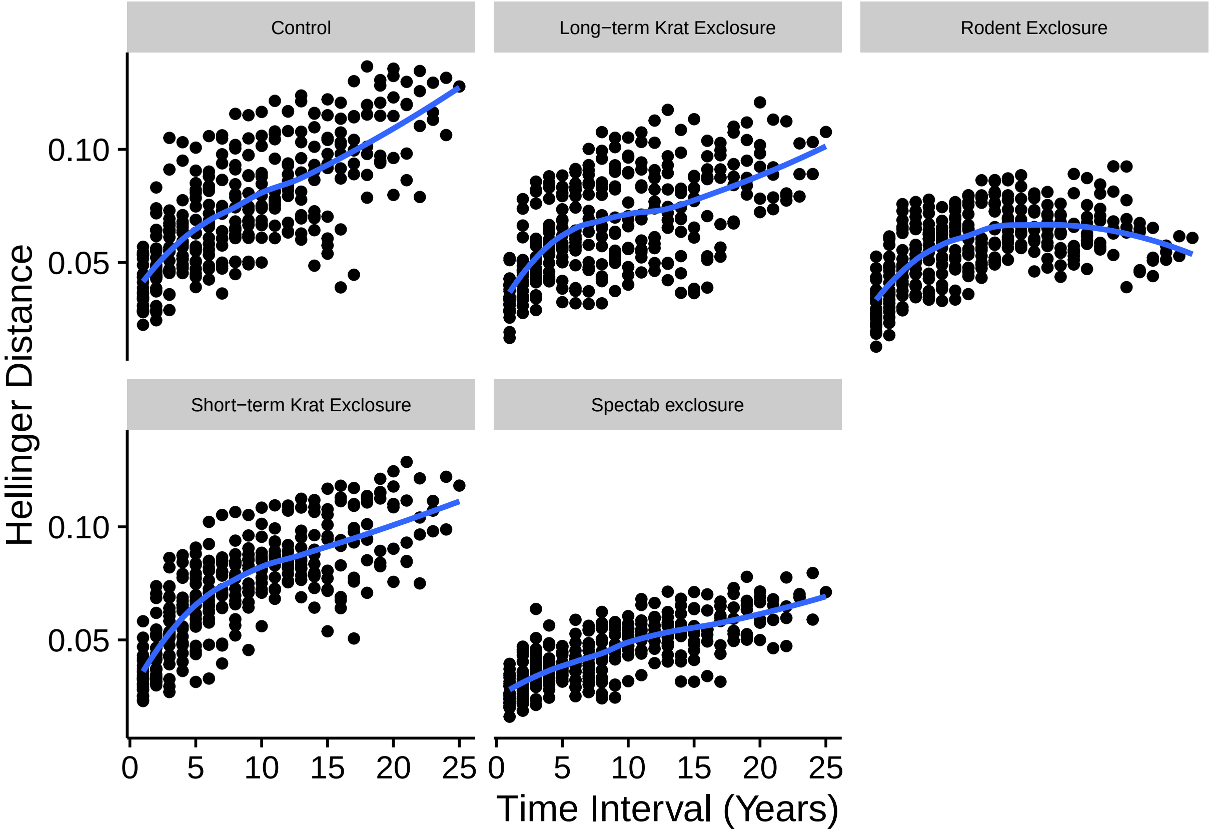
> ***Answer 6b***: The plot shows higher MRS values for Rodent exclosures on average. However, the MRS values are also more variable for the rodent exclosures (CV = 0.44 compared to a CV = 0.38 for the control plots).

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.

```
# In order to calculate relative abundances, we need total abundances # Let's add a column for total ab
# We will 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))

# Now Apply the Hellinger transformation
portal.hellinger.transf = portal.species.abunds %>%
mutate(hellinger.transf = sqrt(n / tot.abund))

# The mutate function creates a new column "hellinger.transf"
# by taking the square root of species relative abundance
# We can use this 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***: You are calculating a rate change interval- from which- you can tell how much a community is diverging over time and the rate at which the divergence is happening.

> ***Answer 7b***: Hellinger distances tend to increase over time for all of the exclosures except for the rodent exclosures. This decrease in abundance in rodent exclosures may be due to a change in competition dynamics. In the Rodent-Exclosure plots seed-eating desert rodents were excluded. H: Interspecific competition for seeds increased among smaller species in the Rodent Exclosure plots, resulting in decreased abundances over time.

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.

```
# 1. Using total abundance as your focal variable, calculate stability (i.e., 1/CV) and synchrony for e
portal.stab = community_stability(df = as.data.frame(portal.species.abunds), time.var = "year",
abundance.var = "n",replicate.var = "plot_type")
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

```
# Here, we will calculate two measures of community-wide synchrony that range from -1 for perfect asynch

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"
```

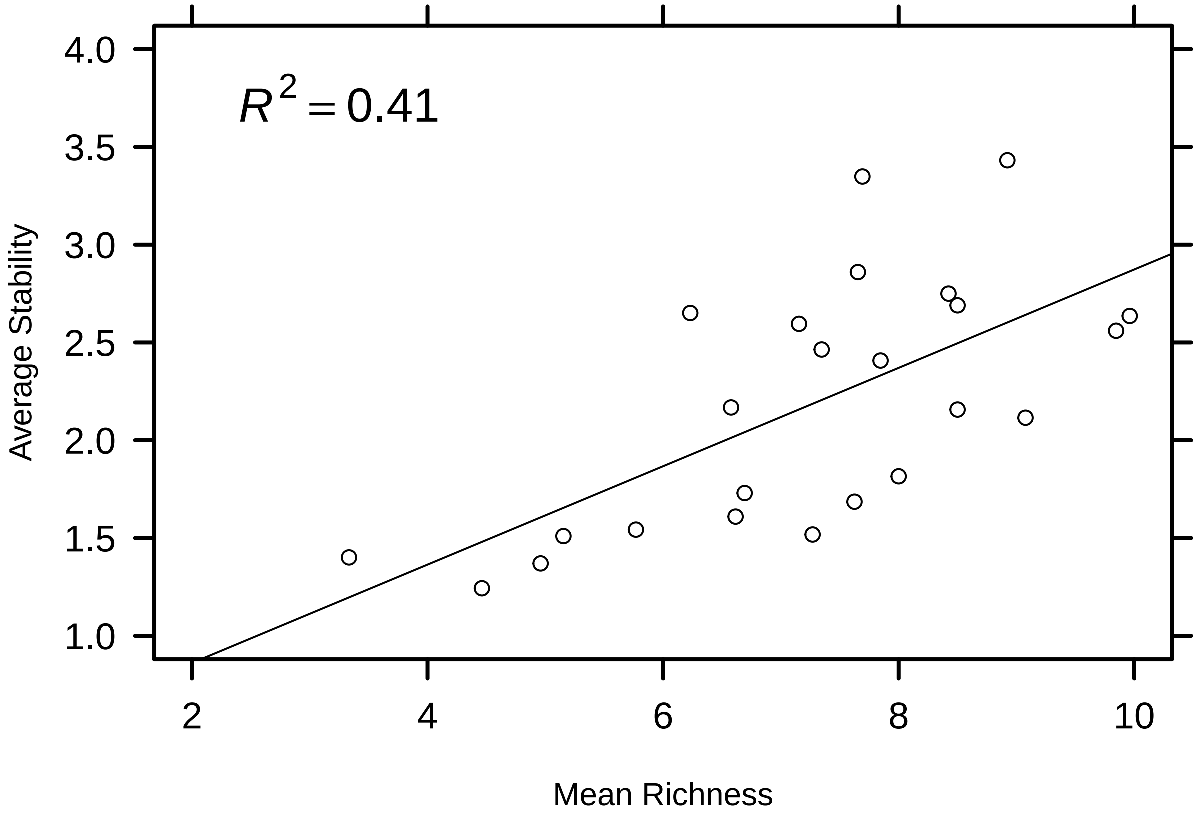
plot_type	loreau	gross
Control	0.1869	0.1263
Long-term Krat Exclosure	0.1578	0.07418
Rodent Exclosure	0.187	0.1423
Short-term Krat Exclosure	0.08773	0.001546
Spectab exclosure	0.1542	0.1393

```
# 2. Test for a biodiversity-stability relationship by regressing community stability on mean richness.
# Recall, earlier we calculated richness in each plot type in each year # Let's group only by plot_id
# Then, we we summarise average annual richness in each plot type
richness = as.data.frame(rowSums(time.by.species[,-c(1:3)]>0))
rich.all = data.frame(time.by.species[,1:3,], richness)
names(rich.all)[4] = "richness"
```

```r
rich.treat = rich.all[which(rich.all$plot_type == "Control"| rich.all$plot_type == "Rodent Exclosure"),
portal.mean.rich.plot = rich.all %>%
                        group_by(plot_id) %>%
                        summarise(mean.rich = mean(richness))
# Let's take a look at how 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"
```

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

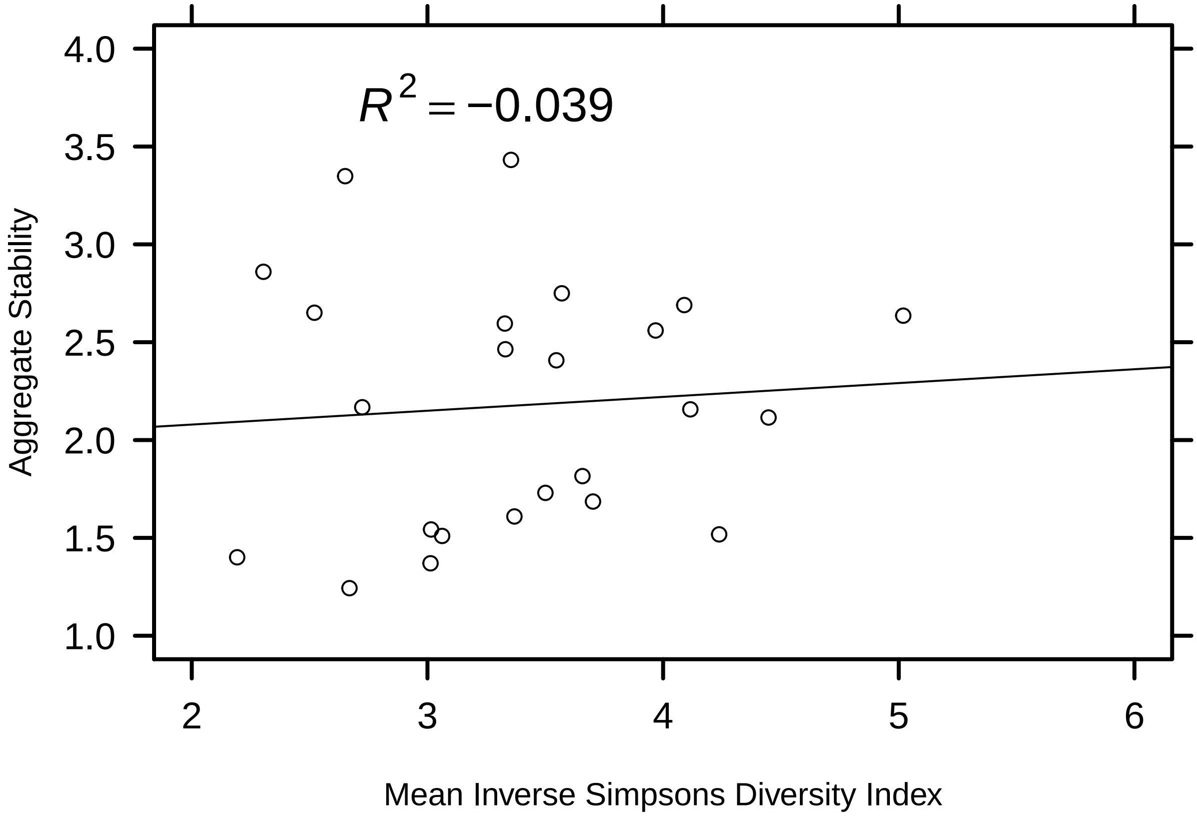


```
# 3. Test for a biodiversity-stability relationship by regressing community stability on mean inverse S

# Recall, earlier we calculated 1/D in each plot type in each year
# Let's group only by plot_id
# Then, we we summarise average annual richness in each plot type
portal.mean.div.plot = div.all %>%
                        group_by(plot_id) %>%
                        summarise(mean.div = mean(inv.simp))
# Let's take a look at how stability metrics relate to mean 1/D
portal.plot.abunds = as.data.frame( group_by(portal, year, plot_id) %>% count(taxon))
portal.stab.plot.D = community_stability(df = portal.plot.abunds, time.var = "year",abundance.var = "n"
# Join richness and stability
portal.div.stab.D = portal.mean.div.plot %>%
                  inner_join(portal.stab.plot.D)
```

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

```
# Plot the relationship
par(mar = c(5,5,1,1))
plot(portal.div.stab.D$stability ~ portal.div.stab.D$mean.div,
     xlab = "Mean Inverse Simpsons Diversity Index", ylab = "Aggregate Stability", yaxt = "n", xaxt = "
xlim = c(2,6), ylim = c(1,4))
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)
```

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



```
Sys.setenv("DISPLAY"=":0")
```

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: Control plots had the highest stability in total abundance. The stability of total abundance is measured as the inverse of the coefficient of variation (1/CV) and is equal to the mean divided by the standard deviation. Higher CVs indicate lower stability and lower CVs indicate greater stability.

Answer 8b: Synchrony is a measure of whether or not population densities are in phase or in sync with one another, meaning that they fluctuate in the same manner in response to a change in the environment. When species display strong synchrony their densities positively covary. ***Answer 8c***: Richness increases with aggregate stability (total abundance; R2 = 0.41). However, mean inverse Simpson's diversity slightly decreases with aggregate stability (R2 = -0.039). It is generally thought that more diverse ecosystems are more stable. The portal data do not seem to be consistent with this expectation from biodiversity-stability theory.

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.

> ***Synthesis***:In order to analyze how biodiversity changes across time and space, you must consider the possibility that some samples in a dataset may not be independent - because samples taken relatively close together in space or time are potentially redundant as a result of autocorrelation. Accounting for repeated measurements on the same variable is one of the challenges associated with comparing diversity across time or space. Both spatial and temporal autocorrelation can be detected through diagnostic plots. Variograms show the distance at which you can compare diversity between sites. A way of assessing temporal autocorrelation or the assumption of stationarity, is to look at time lag plots using the ACF or PACF function. Lags that are significantly correlated indicate that there may be some temporal pattern to your data- for instance a lag at two, indicated a seasonal signal (rainy vs. dry season). If there is a significant correlation at a lag, corrective measures should be taken which involve transforming or differencing the data to get rid of the trend. Both temporal and spatial analyses involve concepts of scale. Spatial extent includes the entire area under investigation, whereas, spatial grain is the smallest sampling unit (1 meter by 1 meter plots for example). Temporal extent is the entire duration of the study (1 year) and temporal grain refers to the smallest observational unit (minutes). Another challenge is defining the most appropriate scale (grain and extent) for representing diversity in your system. There is often a tradeoff in temporal and spatial resolution. Observations collected at high temporal resolution (every hour) will often be limited to a few locations. Whereas, studies that collect data at a high spatial resolution- are usually only done at one point in time.