

Data Analysis Course Outline

Below are the tentative weekly topics to be covered. Also included are selected R functions that we will learn how to use.

Further, there are examples of some code and its output for each week to give you an rough idea about what these topics are covering.

We will learn all of this as we go, so don't be scared by anything... it get pretty easy once you have a foundation.

This document was constructed entirely within R as an R-markdown (Rmd) file.

The complete code for this document can be found online [HERE](#)

All course documents and data sets can be found on the Course GitHub repository

WEEK 1

- Why use code!?
- INSTALL R / R-studio
- Familiarize R-studio functions and layout
- Where to look for help

```
# Functions covered (parital list):  
help()
```

WEEK 2

- Assigning things to objects
- Get familiar with object types and basic functions
 - values, vectors, lists
 - data frames and matrices
 - boolean, character, numeric, POSIXct
- Accessing elements of objects
- Boolean evaluations
- Data-type conversions

```
# Functions covered (parital list):  
=          <-          ->  
class()  
data.frame()  as.factor()  as.numeric()  as.character()  as.POSIXct()  as.matrix()  
==          <          >          <=          >=  
+          -          *          /  
which()      signif()      ceiling()      floor()        round()  
c()          list()        cbind()       rbind()        sum()          mean()         sd()  
  
vector = c(1,2,3,4,5,6,7,8,9,10) # assign a series of numbers to an object called "vector"  
mean(vector) # calculate the mean of the numbers in that object  
  
## [1] 5.5
```

WEEK 3

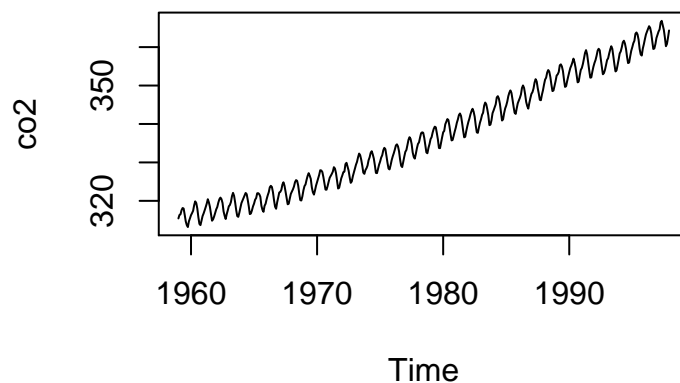
- Importing data
- Useful data formats
- Data structure and attributes
- Summary stats and basic visualizations
- Exploring data
 - Sorting, Transposing, Sampling
 - heatmaps, boxplots, barcharts, scatterplots, histograms

Functions covered (parital list):

<code>read.csv()</code>	<code>read.delim()</code>			
<code>str()</code>	<code>dim()</code>	<code>names()</code>	<code>attributes()</code>	<code>head()</code>
<code>summary()</code>	<code>min()</code>	<code>max()</code>	<code>range()</code>	<code>quantile()</code>
<code>hist()</code>	<code>boxplot()</code>	<code>barplot()</code>	<code>plot()</code>	<code>heatmap()</code>
<code>sample()</code>	<code>t()</code>	<code>sort()</code>	<code>tail()</code>	<code>var()</code>

plot(co2, main = "[CO2] Time Series") # make a simple plot of the data in the object called "co2"

[CO2] Time Series



summary(co2) # generate statistical summaries of those data

##	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
##	313.2	323.5	335.2	337.1	350.3	366.8

WEEK 4

- Finding/Installing/Loading packages
- Extending functionality of R
- Subsetting and manipulating raw data
- Output options

Functions covered (parital list):

<code>install.packages()</code>		
<code>library()</code>		
<code>cor()</code>		
<code>write.table()</code>	<code>sink()</code>	<code>tiff/jpeg/png() / dev.off()</code>

```
data # show original data
```

```
##      Plant      Type Treatment conc uptake
## 1      Qn1      Quebec nonchilled   95  16.0
## 43     Mn1 Mississippi nonchilled   95  10.6
## 44     Mn1 Mississippi nonchilled  175  19.2
## 2      Qn1      Quebec nonchilled  175  30.4
## 3      Qn1      Quebec nonchilled  250  34.8
## 45     Mn1 Mississippi nonchilled  250  26.2
## 4      Qn1      Quebec nonchilled  350  37.2
## 46     Mn1 Mississippi nonchilled  350  30.0
## 5      Qn1      Quebec nonchilled  500  35.3
## 47     Mn1 Mississippi nonchilled  500  30.9
```

```
data[data$Type == "Quebec",] # subset data to only include samples from "Quebec"
```

```
##      Plant  Type Treatment conc uptake
## 1      Qn1 Quebec nonchilled   95  16.0
## 2      Qn1 Quebec nonchilled  175  30.4
## 3      Qn1 Quebec nonchilled  250  34.8
## 4      Qn1 Quebec nonchilled  350  37.2
## 5      Qn1 Quebec nonchilled  500  35.3
```

Skills Test 1:

- * Import data set
 - * Convert elements to new data type
 - * Subset based on values
 - * Calculate summary statistics
 - * Create basic summary figures
 - * Export summary statistics to text file
-

WEEK 5

- Data estimations
 - point estimates
 - interval estimates
- Hypothesis testing / Model fitting
 - t-test (paired/unpaired)
 - chi-square
 - ANOVA
 - LM/GLM

```
# Functions covered (parital list):
```

```
lm()          glm()          aov()
t.test()      chisq.test()
```

```
ANOVA = aov(uptake ~ conc, data = CO2) # define an analysis of variance model CO2 uptake by plants, pre
summary(ANOVA) # show summary ANOVA table and P-value
```

```
##              Df Sum Sq Mean Sq F value    Pr(>F)
## conc           1   2285   2285.0    25.25 2.91e-06 ***
## Residuals     82    7422     90.5
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

WEEK 6

- Experimental design
- Common designs and analysis options
- Quantitative vs qualitative data
- Probability distributions
- Fitting distributions
- Type I and Type II errors
- Post-hoc tests

```
# Packages used (partial list):
```

```
fitdistrplus
```

```
MASS
```

```
# Functions covered (parital list):
```

```
plotdistr()      descdist()
```

```
fitdistr()       denscomp()      cdfcomp()
```

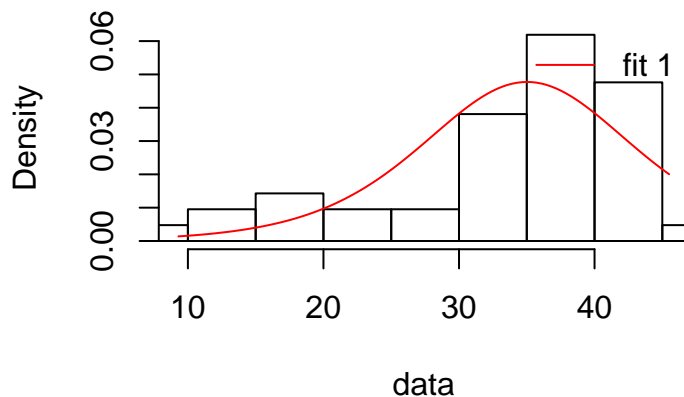
```
TukeyHSD()
```

```
# Fit CO2 uptake data from Quebec samples to a logistic probability distribution
```

```
fit.logistic = fitdistr(CO2$uptake[CO2$Type == "Quebec"], distr = "logis")
```

```
denscomp(fit.logistic) # Plot comparison between logistic distribution and actual data
```

Histogram and theoretical densities



WEEK 7

- Non-parametric alternatives
- Mann-Whitney-Wilcoxin
- Kruskal-Wallace
- Apply functions

```
# Packages used (partial list):
```

```
# Functions covered (parital list):
```

```
wilcox.test()    kruskal.test()
```

```

apply()      sapply()      lapply()      tapply()

data[,4:5] # look at columns 4 and 5 from object called "data"

##      conc uptake
## 1      95   16.0
## 43     95   10.6
## 44    175   19.2
## 2      175   30.4
## 3      250   34.8
## 45     250   26.2
## 4      350   37.2
## 46     350   30.0
## 5      500   35.3
## 47     500   30.9

apply(data[,4:5], 2, sum) # Apply the 'sum' function to those columns

##      conc uptake
## 2740.0  270.6

```

WEEK 8

- Other peoples' data
- Principles of tidy data
- Intuitive manipulations and group functions
 - filter
 - arrange
 - select
 - mutate
 - group_by
 - summarize
 - %>%
- Tidy data transformations
 - gather
 - spread

```

# Packages used (partial list):
dplyr      plyr      tidyr
# Functions covered (partial list):
filter()   arrange()   select()   mutate()
group_by() summarize() %>%
gather()   spread()

# Get specific summary data for defined groups from the object called "data" and save as object called
group.summaries = data %>%
  group_by(Type) %>%
  summarize(Samples = n(), Mean.uptake = mean(uptake), Total.uptake = sum(uptake), StDev.uptake = sd(uptake))

as.data.frame(group.summaries) # display summary info for different locations (groups) as a data frame

##      Type Samples Mean.uptake Total.uptake StDev.uptake
## 1   Quebec      5      30.74      153.7      8.608020

```

```
## 2 Mississippi      5      23.38      116.9      8.501882
```

Skills Test 2:

```
* Import messy data
* Convert to tidy format
* Plot data distribution
* Rearrange and mutate data set
* Summary stats on grouped data
* Test hypothesis / post-hoc tests
```

WEEK 9

- Predicting data
- Intro to ggplot

```
# Packages used (partial list):
ggplot2
```

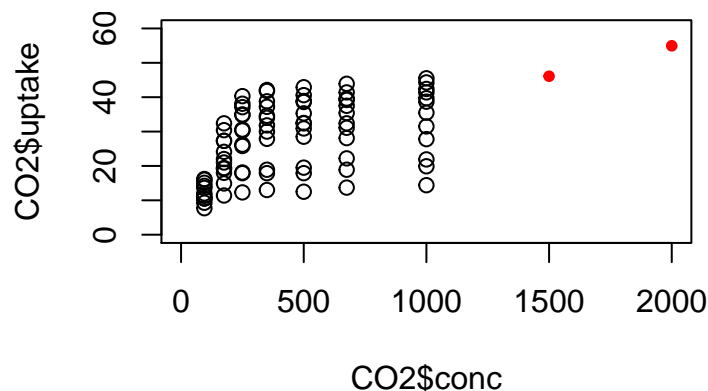
```
# Functions covered (parital list):
predict()
qplot()
ggplot()
  aes()
```

```
library(ggplot2) # load ggplot2 package
```

```
new.data = data.frame(conc = c(1500,2000)) # give new predictor values (CO2 concentration)
predicted = predict(ANOVA, newdata = new.data) # predict plant uptake for those values based on previous
predicted # Look at predictions based on our ANOVA model
```

```
##          1          2
## 46.09617 54.96146
```

```
plot(CO2$conc, CO2$uptake, xlim=c(0,2000), ylim=c(0,60)) # simple plot of CO2 data
points(x=c(1500,2000), y=predicted[1:2], pch=20, col="Red") # Add our predicted uptake values for higher CO2
```



WEEK 10

- Figure generation
- Figure export

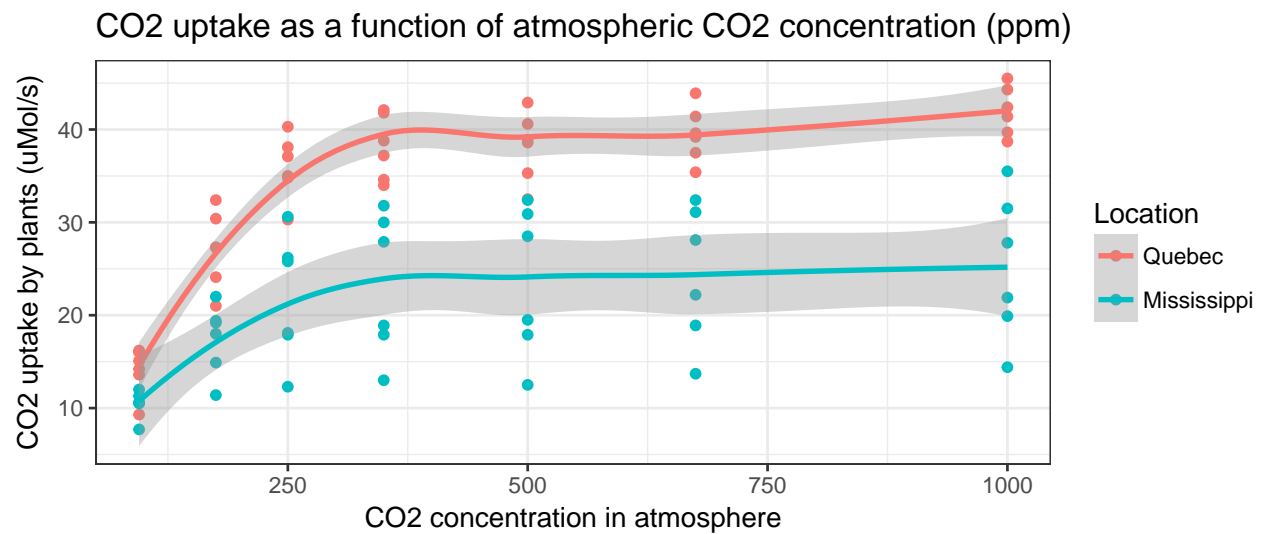
```
# Packages used (partial list):
ggplot2

# Functions covered (parital list):
ggplot()
geom_point()      geom_boxplot()      geom_bar()      geom_violin()
labs()            ggsave()

library(ggplot2) # load ggplot2 package

# Create ggplot and save as abject called "CO2.plot"
CO2.plot = ggplot(CO2, aes(x=conc, y=uptake, col=Type))+
  geom_point() +
  geom_smooth(method = "loess") +
  labs(x="CO2 concentration in atmosphere", y="CO2 uptake by plants (uMol/s)") +
  ggtitle("CO2 uptake as a function of atmospheric CO2 concentration (ppm)") +
  theme_bw() +
  scale_color_discrete(name = "Location")

# Display plot
CO2.plot
```



WEEK 11

- Figure generation continued

```
# Packages used (partial list):
ggplot2
gridExtra

# Functions covered (parital list):
```

```

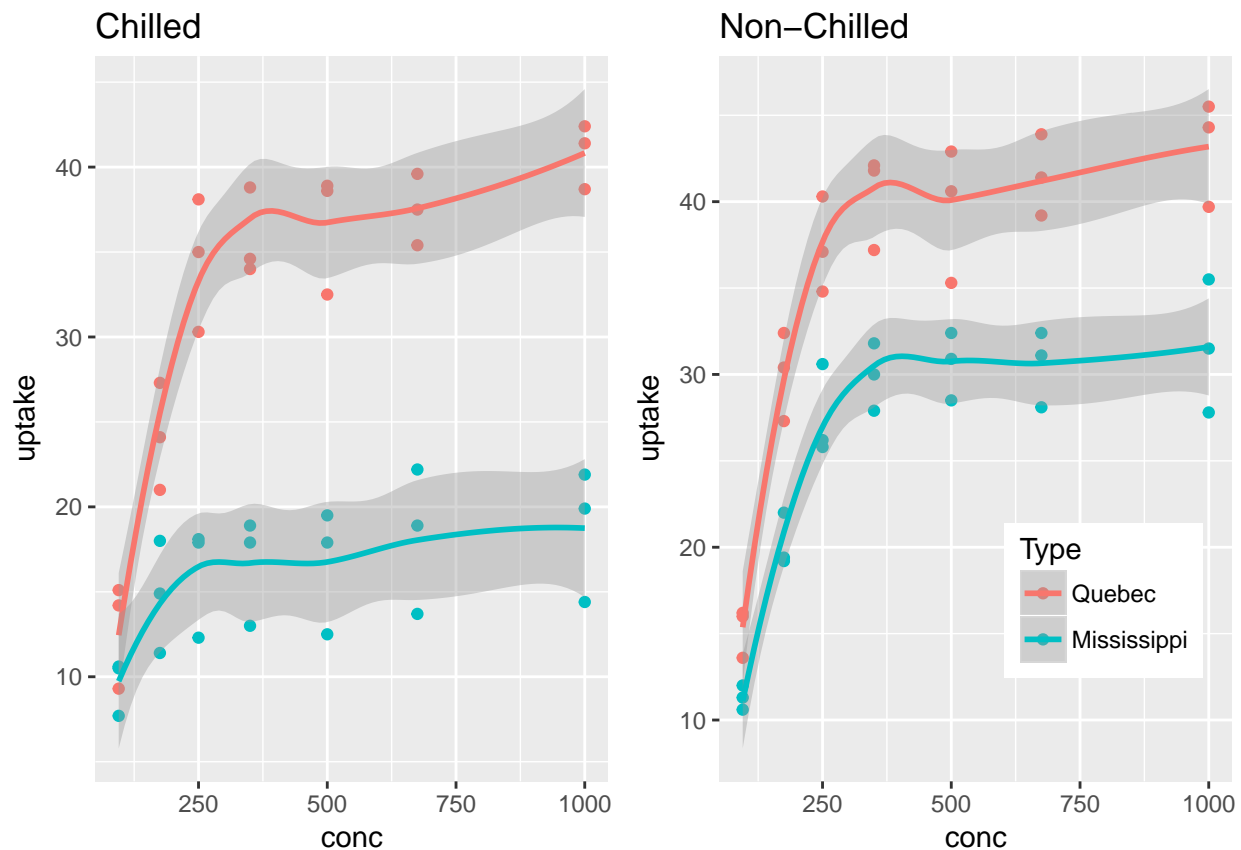
grid.arrange()
ggplot()
scale_*()

# Make two separate plots - One for each temperature treatment
C02.plot.1 = ggplot(C02[C02$Treatment == "chilled",], aes(x=conc, y=uptake, col=Type)) +
  geom_point() +
  geom_smooth(method = "loess") +
  ggtitle("Chilled") +
  theme(legend.position="none")

C02.plot.2 = ggplot(C02[C02$Treatment == "nonchilled",], aes(x=conc, y=uptake, col=Type)) +
  geom_point() +
  geom_smooth(method = "loess") +
  ggtitle("Non-Chilled") +
  theme(legend.position=c(.75,.25))

# Combine the two plots into one image
grid.arrange(C02.plot.1, C02.plot.2, nrow = 1)

```



WEEK 12

- Data standardization / normalization
- Ecology examples

- Ordinations / NMDS
- PermANOVA
- Distance measures
- Diversity measures

Packages used (partial list):

vegan

Functions covered (parital list):

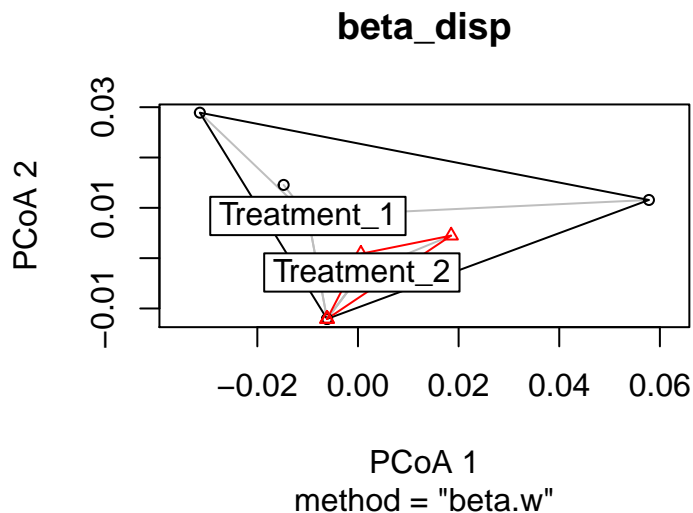
```
decostand()      rrarefy()      dist()          betadiver()
metaMDS()       adonis()       diversity()     betadisper()
```

head(community_matrix)[,1:10] # Take a look at the community composition (observed counts) for differen

```
##      Species_1 Species_2 Species_3 Species_4 Species_5 Species_6
## Sample_1      548      883      480      357        64      832
## Sample_2      218       67      815      726      974      740
## Sample_3        0      306      199      991      933      244
## Sample_4      792      321      405       35      808      863
## Sample_5      560      460       0      863      794      921
## Sample_6       74      766      573      715      853      453
##      Species_7 Species_8 Species_9 Species_10
## Sample_1      942      434       0      735
## Sample_2      410      294      252      895
## Sample_3      624      378      387      665
## Sample_4      793      749      918      223
## Sample_5      175      478      228       94
## Sample_6      984       63      447      952
```

Look at the beta diversity between two randomly-generated communities

```
beta_div = betadiver(community_matrix, method = "w")
beta_disp = betadisper(beta_div, treat)
plot(beta_disp)
```



Skills Test 3:

- * Import data set
- * Fit appropriate model

- * Use model to predict new response values from new predictors
 - * Generate and export plots from data sets
-

WEEK 13

- Importing and manipulating DNA sequence data
 - Bioconductor
 - Sequence data
 - Biostrings
- Phylogenetics examples
 - Sequence alignment
 - Tree building
 - Taxonomic assignment

Packages used (partial list):

```
Bioconductor
ape
biostrings
```

Functions covered (parital list):

To be decided...

Assign a DNA sequence to a special DNASTring object

```
Seq_1 = DNASTring("TCTCTTCGCCGCCGCTGGCCTTGCCGCTGCTGCTCCTCTCGAGTCTCGCCAGGACACCGCATCCTGCCCTGTCACCACTGAGGG")
```

reverseComplement(Seq_1) # Gives reverse-compliment of a DNA sequence

```
## 97-letter "DNASTring" instance
```

```
## seq: CCAGACGTAGTCACCCTCAGTGGTGACAGGGCAG...GCAGCAGCGGCAAGGCCAGCGGCGGCGAAGAGA
```

WEEK 14

- Command-line tools
 - BASH
 - compression
 - grep, sed, find, |, gzip/gunzip, tar, mv, cp, mkdir, etc.
-

WEEK 15

- Data management
- Reporting
- Rmd

Skills Test 4 (final):

- Command-line data access and manipulation
- Writing a script to
 - * import specific data

- * tidy and normalize data
- * subset and group
- * test hypotheses
- * create intuitive plots that include test statistics
- Save script as readable report

This document was constructed entirely within R as an R-markdown (Rmd) file.

The complete code for this document can be found online [HERE](#) at the Course GitHub repository
