

# Praccomp 2022\_R tutorial

## R basic

### Introduction

R as a calculator

```
1 + 100
```

```
2^2
```

```
7/2
```

```
4-1
```

```
2*2
```

```
3+5*2
```

```
(3+5)*2
```

```
2e2
```

```
2/10000
```

```
sin(1)
```

```
log(1)
```

```
exp(0)
```

```
1 == 1
```

```
1 >= 1
```

```
1 < 1
```

```
1 != 2
```

```
x <- 10  
x
```

```
x + x
```

```
x - 5
```

```
x^2
```

```
y <- x+x  
y
```

## Vectorization

```
1:5
```

```
2^(1:5)
```

```
v <- 1:5  
2^v
```

```
log10(v)
```

## Environment

```
ls()
```

```
rm(v)  
ls()
```

## Package Management

```
installed.packages()
```

```
install.packages("vegan", dependencies=TRUE)
```

```
library (vegan)
```

# Project Management

1. Set up management structure (i.e., “data,” “src,” “results,” “doc”)
2. Discuss avoiding redundancy of files
3. Make sure to use version control (i.e., Git)

## Getting help

```
?plot  
help(plot)  
?"<-"  
vignette("FAQ-vegan")  
citation("vegan")
```

## Data and their formats

```
cats <- data.frame(coat = c("calico", "black", "tabby"),  
                  weight = c(2.1,5,3.2),  
                  likes_string = c(1,0,1))
```

```
cats
```

```
getwd()
```

```
write.csv(cats, file = "C:/Users/DNA$TY/Desktop/Grad School/Praccomp/BIOL4800_6220/data/cats.csv")  
cats <- read.csv("C:/Users/DNA$TY/Desktop/Grad School/Praccomp/BIOL4800_6220/data/cats.csv")  
cats$coat
```

```
cats$weight
```

```
cats$weight*10
```

```
log(cats$weight)
```

```
logweight <- log(cats$weight)  
cbind(cats,logweight)
```

```
cats <- cbind(cats,logweight)  
cats
```

```
paste ("My cat is ", cats$coat, ", and it weighs ", cats$weight, " kg. ", sep="")
```

# Data Types

```
typeof(cats$coat)
```

```
typeof(cats$weight)
```

```
typeof(cats$likes_string)
```

```
typeof(3.14159)
```

```
typeof(1i)
```

```
typeof(FALSE)
```

```
typeof(TRUE)
```

```
typeof(T)  
typeof(F)
```

```
typeof(cats)
```

```
class(cats)
```

```
file.show("C:/Users/DNA$TY/Desktop/Grad School/Praccomp/BIOL4800_6220/data/cats.csv")
```

```
cats$likes_string <- as.logical(cats$likes_string)  
typeof(cats$likes_string)
```

```
cats$likes_string
```

```
c("a", "b")
```

```
ab <- c("a", "b")  
ab
```

```
c(ab, "c")
```

```
c(ab, ab)
```

```
1:10
```

```
seq(10)
```

```
z <- seq(10)  
head(z, n=3)
```

```
length(z)
```

```
class(z)
```

```
typeof(z)
```

```
seq(70,100, by=5)
```

## Data Frames

```
cats
```

```
str(cats$coat)
```

```
coats<-c("tabby", "tortoiseshell", "tortoiseshell", "black", "tabby")  
coats
```

```
str(coats)
```

```
factor(coats)
```

```
categories <- factor(coats)  
class(coats)
```

```
class(categories)
```

```
str(categories)
```

## Lists

```
list_example <- list(title= "Numbers", numbers = 1:10, data=TRUE)
```

```
list_example
```

```
another_list <- list(1, "a", TRUE, 1+1i)
another_list
```

```
typeof(list_example)
```

```
class(list_example)
```

```
typeof(cats)
```

```
class(cats)
```

```
data.frame(list_example)
```

```
cats
```

```
cats[,3]
```

```
cats[3,]
```

```
cats[2:3,c(1,3)]
```

## Matrices

```
matrix_example <- matrix(0, ncol=5, nrow=3)
matrix_example
```

```
class(matrix_example)
```

```
typeof(matrix_example)
```

```
str(matrix_example)
```

```
dim(matrix_example)
```

```
ncol(matrix_example)
```

```
nrow(matrix_example)
```

```
class(data.frame(matrix_example))
```

```
df_example <- data.frame(matrix_example)
df_example
```

## Subsetting

```
p <- c(2.3,6.9,4.0,23,1)
p
```

```
names(p) <- c('a','b','c','d','e')
p
```

```
p[1]
```

```
p[2:4]
```

```
p[c(1,5)]
```

```
p[c(1,1,1,3,5,5)]
```

```
p[6]
```

```
p[-3]
```

```
p[c(-1,-5)]
```

```
p[-(2:4)]
```

```
p[c('a','c')]
```

```
p[c(T,F,T,F,T)]
```

```
p[names(p) != 'c']
```

## Factors

```
f <- factor(c('a', 'b', 'c', 'd', 'e'))
f
```

```
f[f == 'a']
```

```
f[1:3]
```

```
f[f %in% c('b', 'c')]
```

```
f[-3]
```

```
f2 <- factor (c('a', 'a', 'b', 'c', 'c'))  
f2
```

```
f2[f2 == 'a']
```

```
f2[f2 %in% c('a', 'c')]
```

## Matrices Resumed

```
set.seed(65)  
m <- matrix(rnorm(6*4), ncol=4, nrow=6)  
m
```

```
m[3:4, c(3,1)]
```

```
m[,c(3,4)]
```

## List Revisited

```
xlist <- list(a= "BIOL48006220", b=seq(1,10, by = 0.5), data = "Grade")  
xlist
```

```
xlist[1:2]
```

```
xlist['a']
```

```
xlist[['a']]
```

```
xlist ['b']
```

```
xlist[['b']]
```

```
xlist$b
```



# Data Frames Revisted

```
gp <- read.csv("C:/Users/DNA$TY/Desktop/Grad School/Praccomp/BIOL4800_6220/data/compt_plastic_wst.csv")
```

```
head(gp, n=10L)
```

```
head(gp[3], n=10L)
```

```
nrow(gp)
```

```
head(gp[["country"]], 10L)
```

```
gp$X2001
```

```
gp[c(1,3), 2:5]
```

```
gp [ which(gp$X1997 <= 100 & gp$X2016 >= 100),]
```

## Conditionals and Flow

```
n <- 10
```

```
if (n <10) {  
  print("n is less than 10")  
} else if (n >10) {  
  print ("n is greater than 10")  
} else {  
  print("n is equal to 10")  
}
```

```
for (n in seq(1,20)) {  
  if (n <10) {  
    print("n is less than 10")  
  } else if (n >10) {  
    print ("n is greater than 10")  
  } else {  
    print("n is equal to 10")  
  }  
}
```

```
g <- 0

while (g <=10) {
  print(paste(g, "is less than or equal to 10"))
  g <- g+1
}
```

## Plotting and Figures

```
install.packages("ggplot2")
```

```
library(ggplot2)
```

```
gp <- read.csv("C:/Users/DNA$TY/Desktop/Grad School/Praccomp/BIOL4800_6220/data/GapminderData/data/gapminder_all.csv")
head(gp)
```

```
ggplot(data=gp, mapping=aes(x=gdpPercap_1952, y=pop_1952)) + geom_point()
```

```
ggplot(data=gp, mapping=aes(x=gdpPercap_2002, y=pop_2002)) + geom_point()
```

```
gapminder <- read.csv("http://raw.githubusercontent.com/swcarpentry/r-novice-gapminder/gh-pages/_episodes_rmd/data/gapminder_data.csv")
head(gapminder)
```

```
ggplot(data=gapminder, mapping=aes(x=gdpPercap, y=lifeExp)) + geom_point()
```

```
ggplot(data=gapminder, mapping=aes(x=year, y=lifeExp, by=country, color=continent)) + geom_line()
```

```
ggplot(data=gapminder, mapping=aes(x=year, y=lifeExp, by=country)) + geom_line(mapping=aes(color=continent)) + geom_point()
```

```
ggplot(data=gapminder, mapping =aes(x=gdpPercap, y=lifeExp)) + geom_point(alpha=0.5) + scale_x_log10()
```

```
ggplot(data=gapminder, mapping=aes(x=gdpPercap, y=lifeExp)) + geom_point(alpha=0.25, color="purple") + scale_x_log10() + geom_smooth(method= lm, color="gold", size=1.25)
```

```
africas <- gapminder[gapminder$continent == "Africa", ]
head(africas)
```

```
ggplot(data=africas, mapping=aes(x=year, y=lifeExp)) +  
  geom_line(color= "red") +  
  facet_wrap( ~ country) +  
  theme(axis.text.x= element_text(angle = 45)) +  
  labs(  
    x = "Year",  
    y = "Life Expectancy",  
    title = "Life Expectancy Over Time in African Countries"  
  )
```

```
AfricanLifeExp <- ggplot(data=africas, mapping=aes(x=year, y=lifeExp)) +  
  geom_line(color = "red") +  
  facet_wrap( ~ country) +  
  theme(axis.text.x= element_text(angle = 45)) +  
  labs(  
    x = "Year",  
    y = "Life Expectancy",  
    title = "Life Expectancy Over Time in African Countries"  
  )
```

```
ggsave(filename = "C:/Users/DNA$TY/Desktop/Grad School/Praccomp/BIOL4800_6220/data/GapminderData/AfricanLifeExp.png", plot = AfricanLifeExp, width = 24, height = 40, dpi = 300, units = "cm")
```

```
pdf(file = "C:/Users/DNA$TY/Desktop/Grad School/Praccomp/BIOL4800_6220/results/AfricanLifeExp.pdf", width = 24, height = 40)  
plot(AfricanLifeExp)  
dev.off()
```

```
write.table(gapminder, file = "C:/Users/DNA$TY/Desktop/Grad School/Praccomp/BIOL4800_6220/data/gapminder_web.csv", sep=",")
```

```
write.csv(africas, file="C:/Users/DNA$TY/Desktop/Grad School/Praccomp/BIOL4800_6220/data/gapminder_web.csv")
```

## Fancy Plots

```
#install.packages(c("ggribes","viridis","hrbrthemes"), dependencies = T)
```

```
library(ggribes)  
library(ggplot2)  
library(viridis)
```

```
library(hrbrthemes)
```

```
# Plot
ggplot(lincoln_weather, aes(x = `Mean Temperature [F]`, y = `Month`, fill = ..x..)) +
  geom_density_ridges_gradient(scale = 3, rel_min_height = 0.01) +
  scale_fill_viridis(name = "Temp. [F]", option = "C") +
  labs(title = 'Temperatures in Lincoln NE in 2016') +
  theme_ipsum() +
  theme(
    legend.position="none",
    panel.spacing = unit(0.1, "lines"),
    strip.text.x = element_text(size = 8)
  )
```

```
spider_data <- read.csv(file="https://wsc.nmbe.ch/resources/species_export_20221101.csv", header
=TRUE)
```

```
spider_data <- read.csv("data/spider_data_20221101.csv")
spider_data
```

```
install.packages(c("ggstatsplot","palmerpenguins","tidyverse"), dependencies=T)
```

```
library(ggstatsplot)
```

```
library (tidyverse)
```

```
data("penguins", package = "palmerpenguins")
penguins
```

```
penguins2 <- drop_na(penguins)
penguins2
```

```
plt <- ggbetweenstats(
  data = penguins,
  x = species,
  y = bill_length_mm
)
```

```
plt
```

```
gapminder2 <- drop(gapminder)
```

```
boxplot(gapminder2$lifeExp ~ gapminder2$continent)
```

```
install.packages("vioplot", dependencies = T)
```

```
library(vioplot)
```

```
with(gapminder2, vioplot(  
  lifeExp~continent, col = "blue"  
))
```

```
plt <- ggbetweenstats(  
  data = gapminder2,  
  x = continent,  
  y = lifeExp  
)
```

```
plt
```

```
plt2 <- plt +  
  theme(  
    axis.ticks = element_blank(),  
    axis.line = element_line(colour = "grey50"),  
    panel.grid = element_line(color = "#b4aea9"),  
    panel.grid.minor = element_blank(),  
    panel.grid.major.x = element_blank(),  
    panel.grid.major.y = element_line(linetype = "dashed"),  
    panel.background = element_rect(fill = "#fbf9f4", color = "#fbf9f4"),  
    plot.background = element_rect(fill = "#fbf9f4", color = "#fbf9f4")  
  ) +  
  labs(  
    x = "Continent",  
    y = "Life Expectancy (years)",  
    title = "Life expectancy of people living on each continent"  
  )  
plt2
```

```
ggsave(  
  filename = "results/gapminder_lifeExpXcontinentweb-violinplot-with-ggstatsplot.png",  
  plot = plt2,  
  width = 8,  
  height = 8,  
  device = "png"  
)
```

```
install.packages("maptools")  
library(maptools)
```

```
data(wrld_simpl)  
afr<-wrld_simpl[wrld_simpl$REGION==2,]  
plot(afr)
```

```
levels(penguins2$species)
```

```
penguin_matrix <- with(penguins2, cbind(bill_length_mm, bill_depth_mm, flipper_length_mm, body_mass_g))  
penguin_matrix
```

```
penguin_pca <- princomp(penguin_matrix, cor=TRUE)  
summary(penguin_pca)
```

```
loadings(penguin_pca)
```

```
biplot(penguin_pca, xlab=penguins2[,2])
```

```
penguin_pca$scores
```

## Statistics

# Additional ways of importing/reading and manipulating data

```
rand <- c(12, 54, 98, 65, 38)  
rand
```

```
sum(rand)
```

```
length(rand)
```

```
avg.rand <- sum(rand)/length(rand)  
avg.rand
```

```
mean(rand)
```

```
sort(rand)
```

```
min(rand)
```

```
max(rand)
```

```
cumsum(rand)
```

```
diff(rand)
```

```
rand [2]
```

```
rand*2
```

```
pedes <- scan()
```

## Analyses

# Univariate Statistics

### 1. Categorical Data

#### a. Barplots

```
beer <- c(3, 4, 1, 1, 3, 4, 3, 3, 1, 3, 2, 1, 2, 1, 2, 3, 2, 3, 1, 1, 1, 1, 4, 3, 1)
length(beer)
```

```
barplot(beer)
```

```
table(beer)
```

```
barplot(table(beer), xlab="Beer", ylab="Frequency")
```

```
barplot(table(beer)/length(beer), xlab="Beer", ylab="Proportion")
```

```
pie(table(beer), main="Beer preference by students")
```

### 2. Numerical Data

#### a. Stem-and-leaf Plots

```
stem(pedes)
```

#### b. Strip chart

```
stripchart(pedes, method="stack")
```

### 3. Measures of center

#### a. Mean

```
mean(pedes)
```

```
mean(gapminder$lifeExp)
```

#### b. Median

```
median(pedes)
```

```
median(gapminder$lifeExp)
```

```
with(gapminder, median(lifeExp))
```

#### c. Mode

```
which(table(pedes) == max(table(pedes)))
```

### 4. Variation

#### a. Range

```
range(pedes)
```

```
diff(range(pedes))
```

#### b. Variance

```
var(pedes)
```

```
sd(pedes)
```

#### c. IQR

```
IQR(pedes)
```

#### d. Z-scores

```
scale(pedes)
```

#### e. Summary

```
summary(pedes)
```

### 5. Plots

#### a. Histograms

```
hist(pedes, breaks = "scott")
```



```
hist(pedes, breaks = "scott", prob=TRUE)
```

```
hist(pedes, breaks = "scott", prob=TRUE)  
lines(density(pedes))
```

```
plot(density(pedes))
```

#### b. Box Plots

```
boxplot(pedes)
```

```
summary(pedes)
```

### ####Bivariate and Multivariate Statistics 1. Plotting and Regression a. Box Plotting

```
spid.gen <- read.csv("C:/Users/DNA$TY/Desktop/Grad School/Praccomp/BIOL4800_6220/data/spider_genitalia.csv")  
spid.gen
```

```
boxplot(spid.gen$left.bulb ~ spid.gen$habitat)
```

#### c. Linear Regression

```
gen.reg <- lm(spid.gen$left.bulb ~ spid.gen$right.bulb)
```

```
summary(gen.reg)
```

```
plot(spid.gen$left.bulb ~ spid.gen$right.bulb)  
abline(gen.reg)
```

#### d. Correlation Coefficients & Spearman Rank Correlation

```
cor.gen <- with(data=spid.gen, cor(left.bulb, right.bulb))  
cor.gen^2
```

```
spearman.cor.gen <- with(data=spid.gen, cor(left.bulb, right.bulb, method="spearman"))  
spearman.cor.gen^2
```

#### e. Residuals

```
residuals(gen.reg)
```

#### f. Transformations

```
plot(spid.gen$left.bulb^2 ~ spid.gen$right.bulb)
```

## 2. Comparing Discrete Treatment Effects a. Chi-squared

```
obs_weighted <- c(4,15,6,15,18,2)
obs_fair <- c(10,10,10,10,10,10)
exp <- c(.16,.17,.16,.17,.17,.17)
chisq.test(obs_weighted, p=exp)
chisq.test(obs_fair, p=exp)
```

### b. T-tests

```
#H0 - true mean is equal to 0, but you can set true mean to other values with mu=
t.test(spid.gen$carapace.length)
```

```
t.test(spid.gen$carapace.length, mu=29)
```

```
habitat.t2 <- t.test(spid.gen$carapace.length~spid.gen$habitat)
habitat.t2
```

```
habitat.tless <- t.test(spid.gen$left.bulb, alternative = "less")
habitat.tless
```

```
habitat.tgreater <- t.test(spid.gen$left.bulb, alternative = "greater")
habitat.tgreater
```

### c. Analysis of Variance (ANOVA)

```
gen.lm <- lm(spid.gen$left.bulb ~ spid.gen$habitat)
summary(gen.lm)
```

```
anova(gen.lm)
```

```
gen.anova <- aov(spid.gen$left.bulb~spid.gen$habitat)
summary(gen.anova)
```

```
gapminder.lifeexpectancy.continent.anova <- aov(gapminder2$lifeExp~gapminder2$continent)
summary(gapminder.lifeexpectancy.continent.anova)
```

```
gapminder.tukey.two.way <- TukeyHSD(gapminder.lifeexpectancy.continent.anova)
gapminder.tukey.two.way
```

### d. Analysis of Co-Variance (ANCOVA)

```
boxplot(spid.gen$left.bulb~spid.gen$habitat)
```

```
gen.ancova <- lm(spid.gen$left.bulb~spid.gen$habitat*spid.gen$carapace.length)
summary(gen.ancova)
```

```
gapminder.lifeexp.continent_GDP.ancova <- lm(gapminder2$lifeExp~gapminder2$continent+gapminder2
$gdpPercap)
summary(gapminder.lifeexpectancy.continent_GDP.ancova)
```

```
TukeyHSD(gapminder.lifeexp.continent_GDP.ancova)
```

#above is tbd

### 3. Principal Component Analysis

```
sp.matrix <- with(spid.gen, cbind(left.bulb,right.bulb,carapace.length,leg4.length))
sp.matrix
```

```
sp.pca <- princomp(sp.matrix, cor=TRUE)
summary(sp.pca)
```

```
loadings(sp.pca)
```

```
biplot(sp.pca)
```

```
install.packages("ggfortify", dependencies = T)
library(ggfortify)
sp.pca.plot <- autoplot(sp.pca,
                        data = spid.gen,
                        colour = 'habitat')
sp.pca.plot
```

```
plot(1:25, rep(0.25,25), pch=1:25, col=1:25, ylim=c(0,6), cex=2, ylab="Line types (lty) 1 to 6",
xlab="Plotting character (pch) 1 to 25 and colours (col) 1 to 8", main="Line types (lty), plotti
ng characters (pch),\nand colors (col) for plot and xyplot", lab=c(25,7,2))
```

```
#additional command line reference for pdf save to local folder below
pdf(./results)

points(1:8, rep(0.5,8), pch=20, col=1:8, cex=3)

abline(h=1:6, lty=1:6, col=1:6, lwd=5)
```

## Looping and Conditionals

### a. If then statements

```
x <- 5
if (x > 0) {
  print ("Positive Number")
} else if (x < 0) {
  print ("Negative Number")
} else
  print ("Zero")
```

```
a <- c(5,7,2,9)
ifelse(a %% 2 == 0, "even", "odd")
```

#### b. For loops

```
z <- c(2,5,3,9,8,11,6)
count <- 0

for (val in z) {
  if(val %% 2 == 0) count= count+1
}
print(count)
```

#### c. While statement

```
i <- 1
while (i<5) {
  print (i)
  i = i+1
}
```

#### d. Interruptions

```
x <- 1:5

for (val in x) {
  if (val == 3) {
    break
  }
  print(val)
}
```

```
x <- 1:5
for (val in x ) {
  if (val==3) {
    next
  }
  print(val)
}
```

```
x <- 1
repeat {
  print(x)
  x= x+1
  if (x == 20) {
    break
  }
}
```

## Randomization and dataset management (short)

### a. data set management

```
B <- matrix(
  c(2,4,3,1,5,87),
  nrow=3,
  ncol=2
)

B
```

```
t(B)
```

```
C <- matrix(
  c(7,4,2),
  nrow = 3,
  ncol = 1
)

C
```

```
BC <- cbind(B,C)
```

```
BC
```

```
c(B)
```

### b. Randomization

```
# rnorm(n,mean,sd)
# sample(x,size, replace = FALSE, prob = NULL)
```

```
ndist <- rnorm(100, 50, 10)
ndist
```

```
mean(ndist)
sd(ndist)
```

```
hist(ndist)
```

```
sample(ndist, 10)
```

```
test <- 1:100  
test
```

```
sample(test, 50, replace = FALSE)
```

```
sample(test, 50, replace = TRUE)
```