

FPA2019 Lecture 2: R basics (R, Rstudio, Rmarkdown)

Toshihide Kitakado

March 13-15, 2019

Lecture 2. How to use R via Rstudio

Before starting the seminar, please visit the following website and download the applications which we will use in the seminar.

R

First, please visit the following web site and download “R 3.5.xx for Windows” for example:

<https://cran.r-project.org/>

Download R-3.5.2 for Windows. The R-project for statistical c...

<https://cran.r-project.org/bin/windows/base/>

R-3.5.2 for Windows (32/64 bit)

[Download R 3.5.2 for Windows](#) (79 megabytes, 32/64 bit)

[Installation and other instructions](#)

[New features in this version](#)

If you want to double-check that the package you have downloaded matches the package distributed by CRAN, you can compare the [md5sum](#) of the .exe to the [fingerprint](#) on the master server. You will need a version of md5sum for windows: both [graphical](#) and [command line](#) versions are available.

Frequently asked questions

- [Does R run under my version of Windows?](#)
- [How do I update packages in my previous version of R?](#)
- [Should I run 32-bit or 64-bit R?](#)

Please see the [R FAQ](#) for general information about R and the [R Windows FAQ](#) for Windows-specific information.

Other builds

- Patches to this release are incorporated in the [r-patched snapshot build](#).
- A build of the development version (which will eventually become the next major release of R) is available in the [r-devel snapshot build](#).
- [Previous releases](#)

Note to webmasters: A stable link which will redirect to the current Windows binary release is <https://CRAN.MIRROR>/bin/windows/base/release.htm>.

%

Rstudio

Then please visit the web site below and download the free version of “RStudio Desktop”:

<https://www.rstudio.com/products/rstudio/download/>

Download RStudio - RStudio

<https://www.rstudio.com/products/rstudio/download/>



Products

Resources

Pricing

About Us

Blogs



Choose Your Version of RStudio

RStudio is a set of integrated tools designed to help you be more productive with R. It includes a console, syntax-highlighting editor that supports direct code execution, and a variety of robust tools for plotting, viewing history, debugging and managing your workspace. [Learn More](#) about RStudio features.



	RStudio Desktop Open Source License	RStudio Desktop Commercial License	RStudio Server Open Source License	RStudio Server Pro Commercial License	RStudio Server I RStudio Conn Commercial Lice
	FREE	\$995 per year	FREE	\$9,995 per year	\$29,995 pe year
	DOWNLOAD Learn More	BUY Learn More	DOWNLOAD Learn More	DOWNLOAD Learn More	TALK Learn More
Integrated Tools for R	●	●	●	●	●
Priority Support		●		●	●
Access via Web Browser			●	●	●

%

Rmarkdown

I will explain how you can create a memo/report of your analysis with R in a easy way with Rmarkdown.
I'm writing this handout using it.

Required libraries

Please install the following packages in R by using a command before the seminar.

```
install.packages("package name")
```

```
install.packages("knitr")  
install.packages("rmarkdown")  
install.packages("kableExtra")
```

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

```
install.packages("mgcv")  
install.packages("fields")  
install.packages("ggmap")  
install.packages("marmap")  
install.packages("mapdata")
```

```
install.packages("shiny")
```

Then you can call those libraries.

```
library(knitr)  
library(rmarkdown)  
library(kableExtra)
```

```
library(ggplot2)  
library(gridExtra)
```

```
library(mgcv)  
library(fields)  
library(ggmap)  
library(marmap)  
library(mapdata)
```

```
library(shiny)
```

```
#May not be used (for TK-memo)  
#library(boot) # for regression diagnosis  
#library(MASS)  
#library(MuMIn)  
#library(dplyr)
```

Getting R sessions started with basic arithmetics and handling the list

```
a <- c(1,2,3,4,5)
a
```

```
[1] 1 2 3 4 5
```

```
sum(a)
```

```
[1] 15
```

```
mean(a)
```

```
[1] 3
```

```
a+10
```

```
[1] 11 12 13 14 15
```

```
5*a
```

```
[1] 5 10 15 20 25
```

```
a^2
```

```
[1] 1 4 9 16 25
```

```
log(a)
```

```
[1] 0.0000000 0.6931472 1.0986123 1.3862944 1.6094379
```

```
a[3]
```

```
[1] 3
```

```
a[c(1,2,3)]
```

```
[1] 1 2 3
```

```
a[-c(2,4)]
```

```
[1] 1 3 5
```

```
b <- seq(10,50,10)
```

```
a+b
```

```
[1] 11 22 33 44 55
```

```
a*b
```

```
[1] 10 40 90 160 250
```

```
b/a
```

```
[1] 10 10 10 10 10
```

For loop

```
ss <- 0
for(i in 1:10){
  ss <- ss + i
}
ss
```

```
[1] 55
```

```
# Just in case, this is done simply as  
a<- seq(1,10)  
a
```

```
[1] 1 2 3 4 5 6 7 8 9 10
```

```
sum(a)
```

```
[1] 55
```

Simple analysis for body proportion of horse mackerel

We will conduct a very simple analysis for body proportion of Japanese fish species with statistical testing and visual presentation. We first read a data file named as "Horsemackerel.csv".

```
# Read the data file  
Data <- read.csv("Data/Horsemackerel.csv", header=T)  
head(Data)
```

	Species	TL	FL	SL	HL	BD	ED
1	Maaaji	297	259	235	65	59	15.4
2	Maaaji	285	263	255	65	61	15.2
3	Maaaji	285	255	263	65	63	15.1
4	Maaaji	319	275	269	71	63	16.6
5	Maaaji	297	259	235	65	59	15.1
6	Maaaji	194	172	160	45	42	13.6

```
# Extract the data column and create a new object "proportion"  
Data$BD
```

```
[1] 59 61 63 63 59 42 42 50 46 44 40 53 34 38 38 36 36 36 34 38 37 37 40  
[24] 36 34 38 37 37 40 38 31 30 36 31 39 37 36 41 31 30 36 31 37 41 45 37  
[47] 38 31 30 36 31 37 41 45 37 38 31 30 36 31 37 41 45 37 38 39 31 30 36  
[70] 31 39 37 36 41 31 30 36 31 37 41 45 37 38 39 39 37 37 36 36 35  
[ reached getOption("max.print") -- omitted 85 entries ]
```

```
Data$SL
```

```
[1] 235 255 263 269 235 160 174 182 171 168 152 203 137 149 150 145 146  
[18] 145 136 148 137 145 155 145 136 148 137 145 155 145 118 122 138 133  
[35] 142 129 134 152 137 143 157 150 148 150 175 166 161 137 143 157 150  
[52] 148 150 175 166 161 137 143 157 150 148 150 175 166 161 141 118 122  
[69] 138 133 142 129 134 152 137 143 157 150 148 150 175 166 161 205 213  
[86] 215 221 189 193 197  
[ reached getOption("max.print") -- omitted 85 entries ]
```

```
Proportion <- Data$BD/Data$SL
```

```
# Add "Prop" into the current data set "Data"  
Data <- cbind(Data, Prop=Proportion)  
attach(Data)
```

```
# Extracting data
```

```
SP <- as.numeric(Species)
```

```
SPname <- levels(Species)
```

SP

```
[1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  
[36] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  
[71] 2 2 2 2 2 2 2 2 2 2 2 2 2 4 4 4 4 4 4 4  
 [ reached getOption("max.print") -- omitted 85 entries ]
```

SPname

```
[1] "Kaiwari" "Maaji" "Maruaji" "Moro"
```

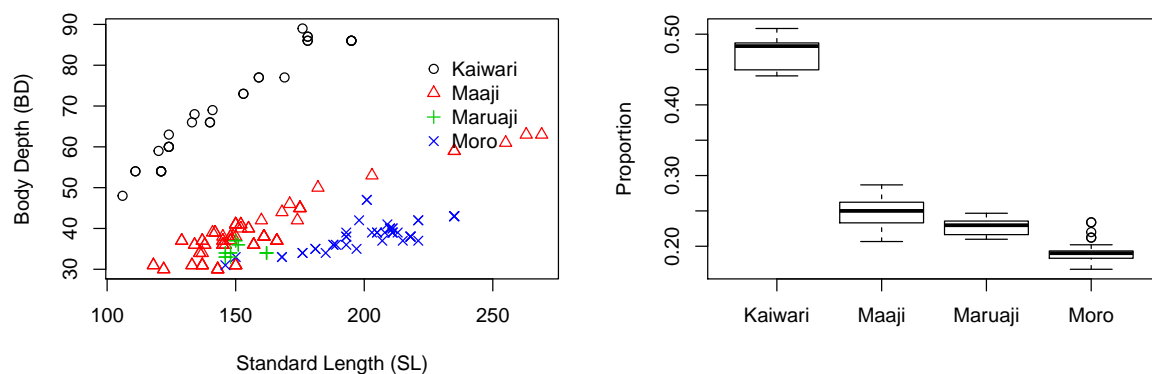
```
# Scatter plot and boxplot
```

```
par(mfrow=c(1,2)) #Splitting screen
```

```
plot(BD~SL, pch=SP, col=SP, xlab="Standard Length (SL)", ylab="Body Depth (BD)")
```

```
legend(220, 85, legend=levels(Species), pch=1:4, col=1:4, bty="n")
```

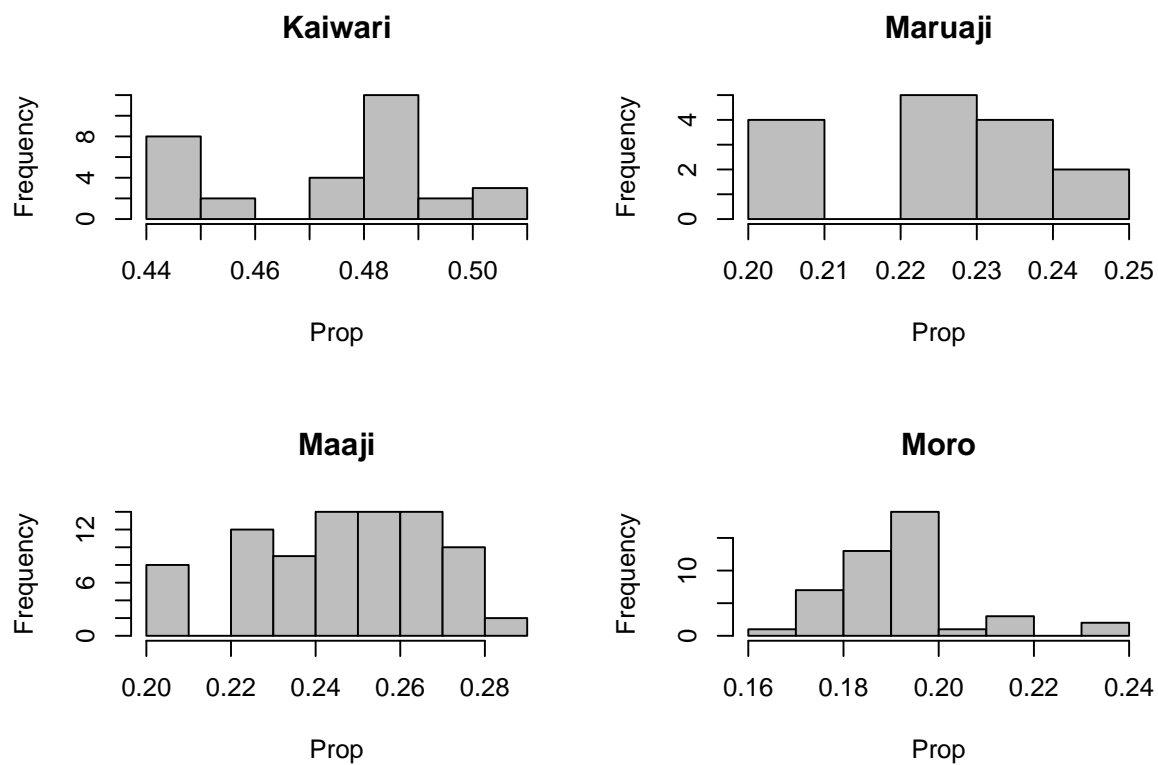
```
boxplot(Prop~Species, ylab="Proportion")
```



```
# Histogram of proportion by species
```

```
par(mfcol=c(2,2))
```

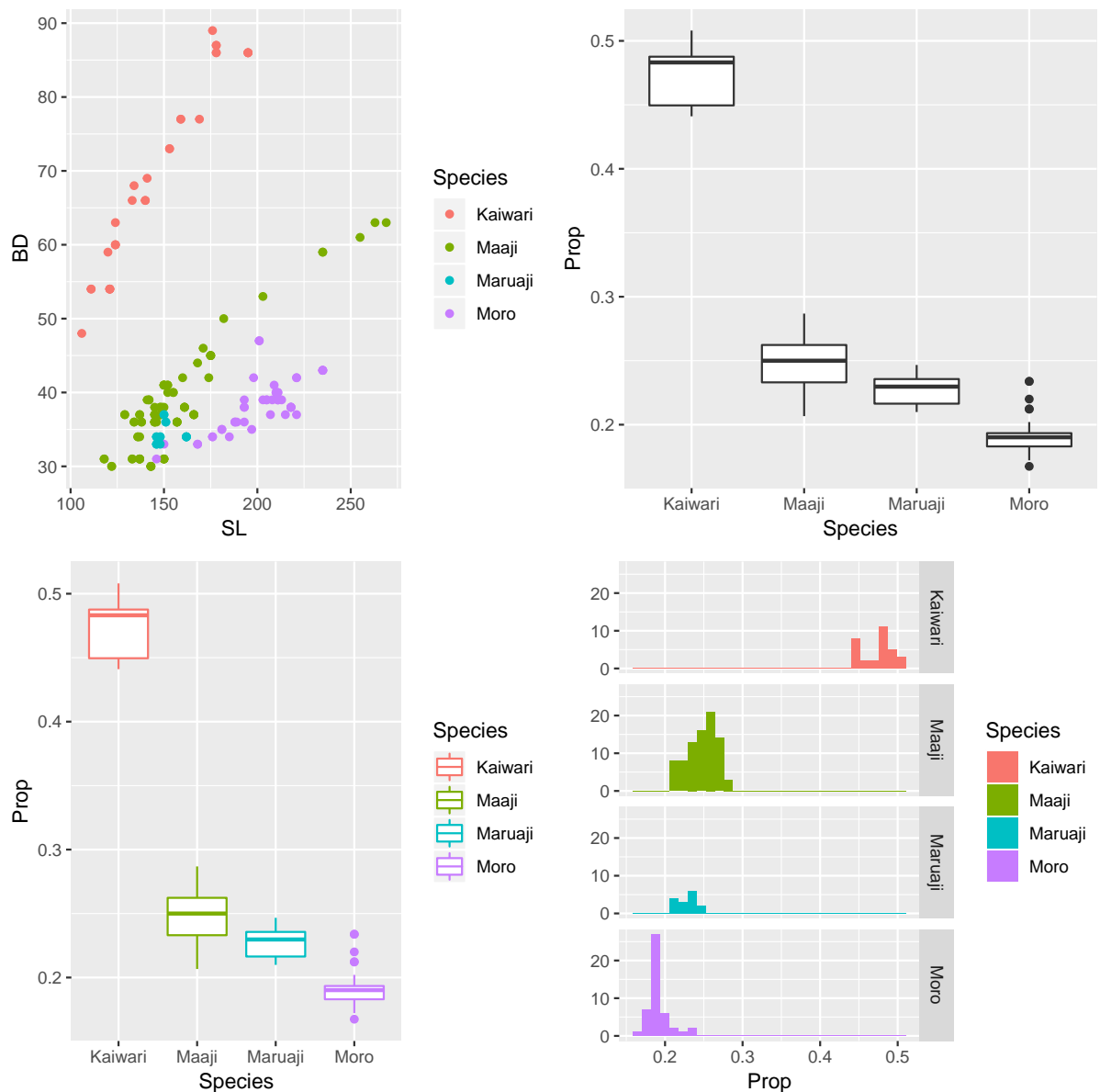
```
for(i in 1:length(SPname)) hist(Prop[SP==i], xlab="Prop", main=paste(SPname[i]), col="gray")
```



More colourful visual presentation by “ggplot”

A library “ggplot2” provides us beautiful figures, but it requires a bit training of syntax, so I shall use a quicker version of command on ggplot.

```
plot1 <- qplot(SL, BD, col=Species)
plot2 <- qplot(Species, Prop, geom="boxplot")
plot3 <- qplot(Species, Prop, geom="boxplot", col=Species)
plot4 <- qplot(Prop, facets=Species~., fill=Species, data=Data)
grid.arrange(plot1, plot2, plot3, plot4, nrow=2, ncol=2)
```



Analysis

Summary statistics

```
y <- split(Prop, Species)
y$Kaiwari
```

```
[1] 0.4893617 0.5074627 0.5056818 0.5080645 0.4916667 0.4556213 0.4528302
[8] 0.4962406 0.4714286 0.4771242 0.4838710 0.4887640 0.4842767 0.4838710
[15] 0.4842767 0.4887640 0.4714286 0.4771242 0.4838710 0.4864865 0.4831461
[22] 0.4410256 0.4462810 0.4410256 0.4462810 0.4410256 0.4462810 0.4831461
[29] 0.4864865 0.4410256 0.4462810
```

```
length(y$Kaiwari)
```

```
[1] 31
```

```
sapply(y, length)
```


Kaiwari	Maaaji	Maruaji	Moro
31	83	15	46

```
lapply(y, length)
```

```
$Kaiwari
[1] 31
```

```
$Maaaji
[1] 83
```

```
$Maruaji
[1] 15
```

```
$Moro
[1] 46
```

```
Ns <- sapply(y, length)
Mean <- sapply(y, mean)
SD <- sapply(y, sd)
data.frame(Nsample=Ns, Mean, SD)
```

	Nsample	Mean	SD
Kaiwari	31	0.4738781	0.02159171
Maaaji	83	0.2477849	0.02045780
Maruaji	15	0.2273268	0.01283770
Moro	46	0.1908011	0.01381693

Example of one sample test (Null hypothesis, $H_0: \mu=0.2$)

```
lapply(y, t.test, mu=0.2)
```

```
$Kaiwari
```

One Sample t-test

```
data: X[[i]]
t = 70.624, df = 30, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0.2
95 percent confidence interval:
 0.4659582 0.4817980
sample estimates:
mean of x
0.4738781
```

```
$Maaaji
```

One Sample t-test

```
data: X[[i]]
t = 21.28, df = 82, p-value < 2.2e-16
```

```

alternative hypothesis: true mean is not equal to 0.2
95 percent confidence interval:
 0.2433179 0.2522520
sample estimates:
mean of x
0.2477849

```

\$Maruaji

One Sample t-test

```

data: X[[i]]
t = 8.2442, df = 14, p-value = 9.634e-07
alternative hypothesis: true mean is not equal to 0.2
95 percent confidence interval:
 0.2202175 0.2344360
sample estimates:
mean of x
0.2273268

```

\$Moro

One Sample t-test

```

data: X[[i]]
t = -4.5155, df = 45, p-value = 4.527e-05
alternative hypothesis: true mean is not equal to 0.2
95 percent confidence interval:
 0.1866980 0.1949042
sample estimates:
mean of x
0.1908011

```

```
res.t.test <- lapply(y, t.test, mu=0.2)
```

Two sample test (if the mean is equal between two species or not)

```

# variance same?
var.test(y[[2]],y[[3]])

```

F test to compare two variances

```

data: y[[2]] and y[[3]]
F = 2.5395, num df = 82, denom df = 14, p-value = 0.05378
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.9838916 5.1560229
sample estimates:

```

```
ratio of variances
      2.539471
```

```
# t-test
t.test(y[[2]],y[[3]], var.equal=T)
```

Two Sample t-test

```
data: y[[2]] and y[[3]]
t = 3.7332, df = 96, p-value = 0.0003205
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.009580294 0.031336083
sample estimates:
mean of x mean of y
0.2477849 0.2273268

t.test(y[[2]],y[[3]])
```

Welch Two Sample t-test

```
data: y[[2]] and y[[3]]
t = 5.1098, df = 28.765, p-value = 1.91e-05
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.01226681 0.02864956
sample estimates:
mean of x mean of y
0.2477849 0.2273268
```

Analysis of variance (test if all the mean are equal or not)

Under an assumption of “common variance” across species,

```
res.lm <- lm(Prop~Species)
summary(res.lm)
```

Call:

```
lm(formula = Prop ~ Species)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.041118	-0.012597	0.002215	0.011178	0.043030

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.473878	0.003342	141.80	<2e-16 ***
SpeciesMaaji	-0.226093	0.003917	-57.73	<2e-16 ***
SpeciesMaruaji	-0.246551	0.005852	-42.13	<2e-16 ***
SpeciesMoro	-0.283077	0.004324	-65.47	<2e-16 ***

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

```
Residual standard error: 0.01861 on 171 degrees of freedom
Multiple R-squared:  0.9653,    Adjusted R-squared:  0.9647
F-statistic: 1584 on 3 and 171 DF,  p-value: < 2.2e-16
```

```
res.lm <- lm(Prop~Species-1)
summary(res.lm)
```

```
Call:
lm(formula = Prop ~ Species - 1)
```

```
Residuals:
      Min       1Q   Median       3Q      Max
-0.041118 -0.012597  0.002215  0.011178  0.043030
```

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
SpeciesKaiwari 0.473878   0.003342  141.80 <2e-16 ***
SpeciesMaaji   0.247785   0.002042  121.32 <2e-16 ***
SpeciesMaruaji 0.227327   0.004804   47.32 <2e-16 ***
SpeciesMoro    0.190801   0.002743   69.55 <2e-16 ***
```

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

```
Residual standard error: 0.01861 on 171 degrees of freedom
Multiple R-squared:  0.9959,    Adjusted R-squared:  0.9958
F-statistic: 1.048e+04 on 4 and 171 DF,  p-value: < 2.2e-16
```

```
res.anova1 <- anova(res.lm)
res.anova1
```

Analysis of Variance Table

```
Response: Prop
      Df Sum Sq Mean Sq F value    Pr(>F)
Species    4 14.5072   3.6268  10476 < 2.2e-16 ***
Residuals 171  0.0592   0.0003
```

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

In case of “heterogeneity in variance”,

```
res.bartlett <- bartlett.test(Prop~Species)
res.bartlett
```

Bartlett test of homogeneity of variances

```
data: Prop by Species
Bartlett's K-squared = 12.515, df = 3, p-value = 0.005813
```

```
res.anova2 <- oneway.test(Prop~Species, var.equal = FALSE)
res.anova2
```

One-way analysis of means (not assuming equal variances)

data: Prop and Species

F = 1373.1, num df = 3.00, denom df = 54.97, p-value < 2.2e-16

Pairwise t-test

```
pairwise.t.test(Prop,Species)
```

Pairwise comparisons using t tests with pooled SD

data: Prop and Species

	Kaiwari	Maaji	Maruaji
Maaji	< 2e-16	-	-
Maruaji	< 2e-16	0.00013	-
Moro	< 2e-16	< 2e-16	9.8e-10

P value adjustment method: holm

```
pairwise.t.test(Prop,Species, pool.sd = FALSE)
```

Pairwise comparisons using t tests with non-pooled SD

data: Prop and Species

	Kaiwari	Maaji	Maruaji
Maaji	< 2e-16	-	-
Maruaji	< 2e-16	1.9e-05	-
Moro	< 2e-16	< 2e-16	1.9e-09

P value adjustment method: holm

```
pairwise.t.test(Prop,Species, pool.sd = FALSE, p.adj = "bonf")
```

Pairwise comparisons using t tests with non-pooled SD

data: Prop and Species

	Kaiwari	Maaji	Maruaji
Maaji	< 2e-16	-	-
Maruaji	< 2e-16	0.00011	-
Moro	< 2e-16	< 2e-16	5.7e-09

P value adjustment method: bonferroni