Introduction

R language is the GNU arm of S language, which has taken the computational world by storm in the last decade. Starting as a compendium of statistical tools, this language has grown up into a canopy lording over a research analysis environment thereby subsuming many hitherto complicated manoeuvres onto the realms of syntactical simplicity. As this an exponentially expanding field of development with ever exploding information downpour, it would be a near impossible task to frame it onto a short simple foundational discourse. However in the subsequent sections we would try to view the potential and the extent of practicality we would unravel the hidden features of the software through a GUI envelop also apart from the regular console and syntax based one. To get its power more understandable we would visualize its forays into the field of analytics using medium scale examples from marine fisheries data.

- R is “GNU S” — A language and environment for data manipulation, calculation and graphical display.
  - R is similar to the award-winning S system, which was developed at Bell Laboratories by John Chambers et al.,
  - a suite of operators for calculations on arrays, in particular matrices,
  - a large, coherent, integrated collection of intermediate tools for interactive data analysis,
  - graphical facilities for data analysis and display either directly at the computer or on hardcopy
  - a well developed programming language which includes conditionals, loops, user defined recursive functions and input and output facilities.
- The core of R is an interpreted computer language.
  - It allows branching and looping as well as modular programming using functions.
  - Most of the user-visible functions in R are written in R, calling upon a smaller set of internal primitives.
It is possible for the user to interface to procedures written in C, C++ or FORTRAN languages for efficiency, and also to write additional primitives.

**R, S and S-plus - a brief time line**

- **S**: an interactive environment for data analysis developed at Bell Laboratories since 1976
  - 1988 - S2: RA Becker, JM Chambers, A Wilks
  - 1992 - S3: JM Chambers, TJ Hastie
  - 1998 - S4: JM Chambers

- Exclusively licensed by AT&T/Lucent to Insightful Corporation, Seattle WA. Product name: “S-plus”.
- Implementation languages C, Fortran.
- See: [http://cm.bell-labs.com/cm/ms/departments/sia/S/history.html](http://cm.bell-labs.com/cm/ms/departments/sia/S/history.html)
- **R**: initially written by Ross Ihaka and Robert Gentleman at Dep. of Statistics of University of Auckland, New Zealand during 1990s.
- Since 1997: international “R-core” team of ca. 15 people with access to common CVS archive.

**What R does and does not**

- data handling and storage: numeric, textual, matrix algebra, hash tables and regular expressions
- high-level data analytic and statistical functions
- classes (Object Oriented “OO”)
- graphics
- programming language: loops, branching, subroutines

- is not a database, but connects to DBMSs
- has no graphical user interfaces, but connects to Java, TclTko
- language interpreter can be very slow, but allows to call own C/C++ code
- no spreadsheet view of data, but connects to Excel/MsOffice
- no professional / commercial support

**R and statistics**

- Packaging: a crucial infrastructure to efficiently produce, load and keep consistent
software libraries from (many) different sources / authors, which are updated at a best possible refresh rate

- Statistics: most packages deal with statistics and data analysis and there are many conduit and value addition libraries which augment the statistical inference
- State of the art: many statistical researchers provide their methods as R packages

**Statistical Analysis**

Data Analysis and Presentation happen to be the core strength of R software environment and the ease with which this is performed makes the environment as the ultimate winner. Faster computational routines and amenability of access and modification to interim steps and results makes the programming environment a winner.

- The R distribution contains functionality for large number of statistical procedures.
  - linear and generalized linear models
  - nonlinear regression models
  - time series analysis
  - classical parametric and nonparametric tests
  - clustering
  - smoothing
- R also has a large set of functions which provide a flexible graphical environment for creating various kinds of data presentations.

**References For R**

- Classical and modern statistical techniques have been implemented.
  - Some of these are built into the base R environment.
  - Many are supplied as packages. There are about 8 packages supplied
with R (called “standard” packages) and many more are available through the cran family of Internet sites (via http://cran.r-project.org).

- All the R functions have been documented in the form of help pages in an “output independent” form which can be used to create versions for HTML, LATEX, text etc.
  - The document “An Introduction to R” provides a more user-friendly starting point.
  - An “R Language Definition” manual
  - More specialized manuals on data import/export and extending R.

**R installations**

**Getting Started**

To install R on your MAC or PC the starting point has to be [http://www.r-project.org/](http://www.r-project.org/).

Depending on the choice of operating system the installer/ zip file with checksum may be downloaded and verified.

An effort to download R for Windows would have the following sequence of interactions with the portal, whose snapshots are given below:
Its always a good idea to download all the files.
MDI is when the windows will be contained within one large window.
Recent Advances in Fishery Biology techniques for Biodiversity Evaluation and Conservation

This is similar to how Excel is setup. SDI is a single document interface where every item will get its own window. This is similar to how SPSS is set up where it has separate data editor, viewer, and syntax windows. Once you choose which your prefer, click next.

Choosing either html or plain text and clicking is the next step.

The installation may take awhile

To install packages on Windows, clicking on packages and install packages will be the next step.
Scrolling down to country nearest and choosing a "mirror" that is close is the next step.

Scrolling down list until the requisite package is the next step, keeping in mind that R lists things in alphabetical order and by uppercase than lowercase. Once a package is clicked to load, R will install not only the package but all of the packages needed to run the package, including the dependencies.

To actually use the package, one has to go back to the package tab and click on load package.

**Using Help Command**

?solve translates on to giving details of help information about “solve” function whilst help.search or ?? allows searching for help in various ways.
R Commander – A graphical interaction “skin” for R

R provides a powerful and comprehensive system for analysing data and when used in conjunction with the R-commander (a graphical user interface, commonly known as Rcmdr) it also provides one that is easy and intuitive to use. Basically, R provides the engine that carries out the analyses and Rcmdr provides a convenient way for users to input commands. The Rcmdr program enables analysts to access a selection of commonly-used R commands using a simple interface that should be familiar to most computer users. It also serves the important role of helping users to implement R commands and develop their knowledge and expertise in using the command line — an important skill for those wishing to exploit the full power of the program. (http://www.rcommander.com/)

a) Loading R Commander

- Packages -> Install Packages -> Cran Mirror Selection -> Rcmdr
b) Opening R Commander

Open R -> Packages -> Load Packages -> Rcmdr
c) **Loading Data**

Data -> Load data

![Image of R Commander window showing file load options]

---

d) **Active Data selection**

Data -> Active data set -> Select active data set

![Image of R Commander window showing active data set selection]

---
Recent Advances in Fishery Biology techniques for Biodiversity Evaluation and Conservation

**e) Menu driven File edit options**

Script will save it as an R file .R and Output will save it as a text file .txt

**f) Summary of the data**

Statistics -> Summaries

Numerical Summeries – can also provide mean, standard deviation, skewness, kurtosis etc.
g) **Mean, Standard Deviation, Skewness, Kurtosis**

![Image of R Commander for numerical summaries](image1)

![Image of R Commander for data set and output window](image2)

h) **Contingency Tables**

![Image of R Commander for data set and output window](image3)

![Image of R Commander for two-way table](image4)
i) **Correlations in R Commander**

Correlation analysis can be done with R as follows.

**Correlation** is a bivariate analysis that measures the strengths of association between two variables and the direction of the relationship. In terms of the strength of relationship, the value of the correlation coefficient varies between +1 and -1. When the value of the correlation coefficient lies around ± 1, then it is said to be a perfect degree of association between the two variables. As the correlation coefficient value goes towards 0, the relationship between the two variables will be weaker. The direction of the relationship is simply the + (indicating a positive relationship between the variables) or - (indicating a negative relationship between the variables) sign of the correlation. Usually, in statistics, we measure four types of correlations: Pearson Correlation, Kendall rank correlation, Spearman correlation, and the Point-Biserial correlation. The software below allows you to very easily conduct a correlation.

![Correlation in R Commander](image)

j) **Independent T-Test**

The independent t-test, also referred to as an independent-samples t-test, independent measures t-test or unpaired t-test, is used to determine whether the mean of a dependent variable (e.g., weight, anxiety level, salary, reaction time, etc.) is the same in two unrelated, independent groups (e.g., males vs females, employed vs unemployed, under 21 year olds vs those 21 years and older, etc.). Specifically, you use an independent t-test to determine whether the mean difference between two groups is statistically significantly different to zero.
k) **One Way ANOVA**

ANOVA (Analysis of Variance) is a statistical technique that assesses potential differences in a scale-level dependent variable by a nominal-level variable having 2 or more categories. For example, an ANOVA can examine potential differences in IQ scores by Country (US vs. Canada vs. Italy vs. Spain). The ANOVA, developed by Ronald Fisher in 1918, extends the $t$ and the $z$ test which have the problem of only allowing the nominal level variable to have just two categories. This test is also called the Fisher analysis of variance. ANOVAs are used in three ways: one–way Anova, two-way ANOVA, and N-way Multivariate ANOVA.

**One-Way ANOVA**

A one-way ANOVA refers to the number of independent variables—not the number of categories in each variables. A one-way ANOVA has just one independent variable. For example, difference in IQ can be assessed by Country, and County can have 2, 20, or more different Countries in that variable.

The software below allows you to easily conduct an ANOVA.

**Statistics->One Way ANOVA**
I) Factor Analysis

Factor analysis is a technique that is used to reduce a large number of variables into fewer numbers of factors. This technique extracts maximum common variance from all variables and puts them into a common score. As an index of all variables, we can use this score for further analysis. Factor analysis is part of general linear model (GLM) and this method also assumes several assumptions: there is linear relationship, there is no multicollinearity, it includes relevant variables into analysis, and there is true correlation between variables and factors. Several methods are available, but principal component analysis is used most commonly.

Types of factoring:

There are different types of methods used to extract the factor from the data set:

1. **Principal component analysis:** This is the most common method used by researchers. PCA starts extracting the maximum variance and puts them into the first factor. After that, it removes that variance explained by the first factors and then starts extracting maximum variance for the second factor. This process goes to the last factor.

2. **Common factor analysis:** The second most preferred method by researchers, it extracts the common variance and puts them into factors. This method does not include the unique variance of all variables. This method is used in SEM.

3. **Image factoring:** This method is based on correlation matrix. OLS Regression method is used to predict the factor in image factoring.
4. **Maximum likelihood method:** This method also works on correlation metrix but it uses maximum likelihood method to factor.

5. **Other methods of factor analysis:** Alfa factoring outweighs least squares. Weight square is another regression based method which is used for factoring.

Result are shown as follows

```r
> FA <- factanal(~aap+audu+bdal, factors=1, rotation="varimax",
>                scores="none", data=binge.orig)
> FA
library(sem, pos=4)
> library(sem, pos=4)

FA <- factanal(~aap+audu+bdal, factors=1, rotation="varimax",
>                scores="none", data=binge.orig)
FA

Call:
factanal(x = ~aap + audu + bdal, factors = 1, data = binge.orig,     scores = "none", rotation = "varimax")

Residuals:
 aap    audu    bdal
  0.069  0.224  0.096

Loadings:
 Factor1
 aap  0.308
 audu 0.322
 bdal 0.626

Factor1
SS loadings  1.231
Proportion Var 0.410

The degrees of freedom for the model is 0 and the fit was 0
> remove(FA)
> library(sem, pos=4)
```
J) Graphs

Gparhs -> Scatter plot

Gparhs -> Box plot
R Basics

**R is object base**

Types of objects (scalar, vector, matrices and arrays Assignment of objects)

**Building a data frame**

**Operation Symbols**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>Addition</td>
</tr>
<tr>
<td>-</td>
<td>Subtraction</td>
</tr>
<tr>
<td>*</td>
<td>Multiplication</td>
</tr>
<tr>
<td>/</td>
<td>Division</td>
</tr>
<tr>
<td>%%</td>
<td>Modulo (estimates remainder in a division)</td>
</tr>
<tr>
<td>^</td>
<td>Exponential</td>
</tr>
</tbody>
</table>

**R as a Calculator**

1550+2000
## [1] 3550

or various calculations in the same row

2+3; 5*9; 6-6
## [1] 5
## [1] 45
## [1] 0

**As Mathematics**

1+1
## [1] 2

2+2*7
## [1] 16
(2+2)*7
## [1] 28

**As Variables**

x<-2
x
## [1] 2
y<-3
y
## [1] 3
5->z
(x*y)+z
## [1] 11

Numbers in R: NAN and NA
NAN (not a number) NA (missing value) - Basic handling of missing values

Missing values are noise to statistical estimations. We are going to learn a basic command for handling missing values.

x<-c(1,2,3,4,5,6,NA)
mean(x)
## [1] NA
mean(x,na.rm=TRUE)
## [1] 3.5

Objects in R

Objects in R obtain values by assignment.
This is achieved by the gets arrow, <-, and not the equal sign, =.

Objects can be of different kinds.

Built in Functions

R has many built in functions that compute different statistical procedures.

Functions in R are followed by ( ). Inside the parenthesis we write the object (vector, matrix, array, dataframe) to which we want to apply the function.

# Create a sequence of numbers from 32 to 44.
print(seq(32,44))
## [1] 32 33 34 35 36 37 38 39 40 41 42 43 44
# Find mean of numbers from 25 to 82.
print(mean(25:82))
## [1] 53.5
# Find sum of numbers frm 41 to 68.
Recent Advances in Fishery Biology techniques for Biodiversity Evaluation and Conservation

print(sum(41:68))
## [1] 1526

Vectors

Vectors are variables with one or more values of the same type.

A variable with a single value is known as scalar. In R a scalar is a vector of length 1. There are at least three ways to create vectors in R: (a) sequence, (b) concatenation function, and (c) scan function.

Create two vectors of different lengths.
vector1 <- c(5,9,3)
vector2 <- c(10,11,12,13,14,15)

vector1
## [1] 5 9 3
vector2
## [1] 10 11 12 13 14 15

Arrays

Arrays are numeric objects with dimension attributes. The difference between a matrix and an array is that arrays have more than two dimensions.

# Take the above vectors as input to the array.
result <- array(c(vector1,vector2),dim = c(3,3,2))

print(result)
## , , 1
##
## [,1] [,2] [,3]
## [1,] 5 10 13
## [2,] 9 11 14
## [3,] 3 12 15
##
Matrices

A matrix is a two dimensional array. The command colnames

# Elements are arranged sequentially by row.
M <- matrix(c(3:14), nrow = 4, byrow = TRUE)

print(M)

String Characters

In R, string variables are defined by double quotation marks.

letters<-'c("a","b","c")

Subscripts and Indices

Select only one or some of the elements in a vector, a matrix or an array. We can do this by using subscripts in square brackets [ ].

In matrices or dataframes the first subscript refers to the row and the second to the column.

Dataframe

Researchers work mostly with dataframes. With previous knowledge you can built dataframes in R. Also, import dataframes into R.
Recent Advances in Fishery Biology techniques for Biodiversity Evaluation and Conservation

# Create the data frame.
emp.data <- data.frame(
  emp_id = c(1:5),
  emp_name = c("Rick","Dan","Michelle","Ryan","Gary"),
  salary = c(623.3,515.2,611.0,729.0,843.25),
  stringsAsFactors = FALSE
)

# Print the data frame.
print(emp.data)

##   emp_id emp_name salary start_date
## 1      1 Rick 623.30 2012-01-01
## 2      2 Dan 515.20 2013-09-23
## 3      3 Michelle 611.00 2014-11-15
## 4      4 Ryan 729.00 2014-05-11
## 5      5 Gary 843.25 2015-03-27

A journey wading through the amazing summarizing and analytical capabilities of R- a case study

Let the presumed data pertain to landings and standardized effort of a maritime state estimated by ICAR-CMFRI during the interregnum 1997 to 2013

calling file in R

klm <- read.csv("C:/Users/cmfri/Desktop/cpue_spcode_kldata.csv",header=TRUE)

To know header portion of the data set

head(klm)

##   year month species raised nomeff stdcpue
## 1 1997   1   40 20595.35 122.0811 3.634042
## 2 1997   2   40 24201.10 114.3719 4.532246
## 3 1997   3   40 23497.64 255.0315 3.926130
To check the last few rows of the dataset

\texttt{tail(klm)}

\begin{verbatim}
## year month species raised nomeff stdcpue
## 245815 2013 7 4580 0 0.000000 0.000000
## 245816 2013 8 4580 1674 2.059835 1.667304
## 245817 2013 9 4580 0 0.000000 0.000000
## 245818 2013 10 4580 0 0.000000 0.000000
## 245819 2013 11 4580 0 0.000000 0.000000
## 245820 2013 12 4580 0 0.000000 0.000000
\end{verbatim}

to know the observations in the data

\texttt{length(klm)}

\begin{verbatim}
## [1] 6
\end{verbatim}

to know the structure of the dataframe

\texttt{str(klm)}

\begin{verbatim}
## 'data.frame': 245820 obs. of 6 variables:
## $ month: int 1 2 3 4 5 6 7 8 9 10 ...
## $ species: int 40 40 40 40 40 40 40 40 40 ...
## $ raised : num 20595 24201 23498 50177 137626 ...
## $ nomeff : num 122 114 255 155 315 ...
## $ stdcpue: num 3.63 4.53 3.93 6.76 13.81 ...
\end{verbatim}

Descriptive statistics analysis

\texttt{summary(klm)}

\begin{verbatim}
## year month species raised
## Min. :1997 Min. :1.00 Min. : 0 Min. : 0
## 1st Qu.:2001 1st Qu.:3.75 1st Qu.:867 1st Qu.: 0
## Median :2005 Median :6.50 Median :1513 Median : 0
\end{verbatim}
Recent Advances in Fishery Biology techniques for Biodiversity Evaluation and Conservation

##  Mean :2005  Mean : 6.50  Mean :2201  Mean : 42699
##  3rd Qu.:2009  3rd Qu.: 9.25  3rd Qu.:4016  3rd Qu.:0
##  Max. :2013  Max. :12.00  Max. :9999  Max. :71536031
##                                                NA's :30
##      nomeff            stdcpue
##  Min. :     0.0   Min. :   0.000
##  1st Qu.:     0.0   1st Qu.:   0.000
##  Median :     0.0   Median :   0.000
##  Mean :   154.2   Mean :   7.112
##  3rd Qu.:     0.0   3rd Qu.:   0.000
##  Max. :119100.1   Max. :5600.000
##

If further enhanced list of summary statistics information about the data like third and fourth order moments, then the `describe` function of `psych` or `summary` function would come in handy.

```r
library(psych)
describe(klm[,3:6])
```

<table>
<thead>
<tr>
<th>vars</th>
<th>n</th>
<th>mean</th>
<th>sd</th>
<th>median</th>
<th>trimmed</th>
<th>mad</th>
<th>min</th>
</tr>
</thead>
<tbody>
<tr>
<td>species</td>
<td>245820</td>
<td>2201.15</td>
<td>1951.83</td>
<td>1513</td>
<td>1941.16</td>
<td>1257.24</td>
<td>0</td>
</tr>
<tr>
<td>raised</td>
<td>245790</td>
<td>42699.02</td>
<td>719150.48</td>
<td>0</td>
<td>62.52</td>
<td>0.00</td>
<td>0</td>
</tr>
<tr>
<td>nomeff</td>
<td>3 245820</td>
<td>154.25</td>
<td>1543.66</td>
<td>0</td>
<td>0.16</td>
<td>0.00</td>
<td>0</td>
</tr>
<tr>
<td>stdcpue</td>
<td>4 245820</td>
<td>7.11</td>
<td>52.38</td>
<td>0</td>
<td>0.11</td>
<td>0.00</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>max</th>
<th>range</th>
<th>skew</th>
<th>kurtosis</th>
<th>se</th>
</tr>
</thead>
<tbody>
<tr>
<td>species</td>
<td>9999.0</td>
<td>9999.0</td>
<td>1.40</td>
<td>1.91</td>
</tr>
<tr>
<td>raised</td>
<td>71536030.7</td>
<td>71536030.7</td>
<td>44.70</td>
<td>2681.18</td>
</tr>
<tr>
<td>nomeff</td>
<td>119100.1</td>
<td>119100.1</td>
<td>22.83</td>
<td>770.70</td>
</tr>
<tr>
<td>stdcpue</td>
<td>5600.0</td>
<td>5600.0</td>
<td>21.65</td>
<td>971.06</td>
</tr>
</tbody>
</table>

If one wants to study monthly catch grouped information so that an idea about issues like which month (used as a group) would have etched up maximum landings/ catch, then simple literally rooted commands like `describeBy` (`psych`) or `aggregate` would come in handy.
```r
library(psych)
describeBy(klm$raised, klm$month)
##
## Descriptive statistics by group
## group: 1
##    vars     n     mean       sd median trimmed mad min      max    range
## X1    1 20485 41379.48 784622.6      0  146.65   0   0 51193526 51193526
##     skew kurtosis      se
## X1 46.55  2497.42 5482.05
## ————————————————————————————
## group: 2
##    vars     n     mean       sd median trimmed mad min      max    range
## X1    1 20485 32904.06 535506.3      0  113.45   0   0 45468199 45468199
##     skew kurtosis      se
## X1 49.62  3259.68 3741.51
## ————————————————————————————
## group: 3
##    vars     n     mean       sd median trimmed mad min      max    range
## X1    1 20485 39087.37 569052.1      0  162.51   0   0 31762665 31762665
##     skew kurtosis      se
## X1 38.4  1796.15 3975.89
## ————————————————————————————
## group: 4
##    vars     n     mean     sd median trimmed mad min      max    range
## X1    1 20471 33795.18 477389      0   64.13   0   0 31931384 31931384
##     skew kurtosis      se
## X1 42.59  2353.01 3336.59
## ————————————————————————————
```
## Recent Advances in Fishery Biology techniques for Biodiversity Evaluation and Conservation

### group: 5

<table>
<thead>
<tr>
<th>vars</th>
<th>n</th>
<th>mean</th>
<th>sd</th>
<th>median</th>
<th>trimmed</th>
<th>mad</th>
<th>min</th>
<th>max</th>
<th>range</th>
</tr>
</thead>
<tbody>
<tr>
<td>X1</td>
<td>1</td>
<td>20485</td>
<td>37566.67</td>
<td>469275.5</td>
<td>0</td>
<td>96.2</td>
<td>0</td>
<td>30492626</td>
<td>30492626</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>skew</th>
<th>kurtosis</th>
<th>se</th>
</tr>
</thead>
<tbody>
<tr>
<td>33.18</td>
<td>1478.99</td>
<td>3278.76</td>
</tr>
</tbody>
</table>

### group: 6

<table>
<thead>
<tr>
<th>vars</th>
<th>n</th>
<th>mean</th>
<th>sd</th>
<th>median</th>
<th>trimmed</th>
<th>mad</th>
<th>min</th>
<th>max</th>
<th>range</th>
</tr>
</thead>
<tbody>
<tr>
<td>X1</td>
<td>1</td>
<td>20485</td>
<td>34552.2</td>
<td>655525.6</td>
<td>0</td>
<td>30.67</td>
<td>0</td>
<td>65432961</td>
<td>65432961</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>skew</th>
<th>kurtosis</th>
<th>se</th>
</tr>
</thead>
<tbody>
<tr>
<td>61.23</td>
<td>5239.89</td>
<td>4580.07</td>
</tr>
</tbody>
</table>

### group: 7

<table>
<thead>
<tr>
<th>vars</th>
<th>n</th>
<th>mean</th>
<th>sd</th>
<th>median</th>
<th>trimmed</th>
<th>mad</th>
<th>min</th>
<th>max</th>
<th>range</th>
</tr>
</thead>
<tbody>
<tr>
<td>X1</td>
<td>1</td>
<td>20485</td>
<td>32621.2</td>
<td>643003.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>49428947</td>
<td>49428947</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>skew</th>
<th>kurtosis</th>
<th>se</th>
</tr>
</thead>
<tbody>
<tr>
<td>42.19</td>
<td>2362.03</td>
<td>4492.57</td>
</tr>
</tbody>
</table>

### group: 8

<table>
<thead>
<tr>
<th>vars</th>
<th>n</th>
<th>mean</th>
<th>sd</th>
<th>median</th>
<th>trimmed</th>
<th>mad</th>
<th>min</th>
<th>max</th>
<th>range</th>
</tr>
</thead>
<tbody>
<tr>
<td>X1</td>
<td>1</td>
<td>20484</td>
<td>57397.86</td>
<td>713381.8</td>
<td>0</td>
<td>31.03</td>
<td>0</td>
<td>38795185</td>
<td>38795185</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>skew</th>
<th>kurtosis</th>
<th>se</th>
</tr>
</thead>
<tbody>
<tr>
<td>26.21</td>
<td>920.16</td>
<td>4984.42</td>
</tr>
</tbody>
</table>

### group: 9

<table>
<thead>
<tr>
<th>vars</th>
<th>n</th>
<th>mean</th>
<th>sd</th>
<th>median</th>
<th>trimmed</th>
<th>mad</th>
<th>min</th>
<th>max</th>
<th>range</th>
</tr>
</thead>
<tbody>
<tr>
<td>X1</td>
<td>1</td>
<td>20485</td>
<td>55833.65</td>
<td>901880.9</td>
<td>0</td>
<td>34.3</td>
<td>0</td>
<td>71536031</td>
<td>71536031</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>skew</th>
<th>kurtosis</th>
<th>se</th>
</tr>
</thead>
<tbody>
<tr>
<td>41.11</td>
<td>2415.63</td>
<td>6301.32</td>
</tr>
</tbody>
</table>
### group: 10
### vars  n  mean     sd  median  trimmed  mad   min     max    range
### X1  1 20484  57071.88 915432.9   0  89.05  0  0 55973676 55973676
### skew  kurtosis     se
### X1 34.05  1453.38 6396.16

### group: 11
### vars  n  mean     sd  median  trimmed  mad   min     max    range
### X1  1 20485  51210.52 915220   0  133.56  0  0 49127745 49127745
### skew  kurtosis     se
### X1 36.33  1488.92 6394.51

### group: 12
### vars  n  mean     sd  median  trimmed  mad   min     max    range
### X1  1 20471  38960.92 830555.4   0  134.37  0  0 66844967 66844967
### skew  kurtosis     se
### X1   56  3639.25 5804.96

**Selecting subsets of data:**

# to know the whole species entries
\[ t<-klm\$species \]

**`length(t)`**

## [1] 245820

# to know the june species entries
\[ d<-klm\$species[klm\$month=="6"] \]

**`length(d)`**

## [1] 20485

to exclude some data

# exclude june catch and know the entries
\[ e<-klm\$species[klm\$month!="6"] \]
**Recent Advances in Fishery Biology techniques for Biodiversity Evaluation and Conservation**

```r
length(e)
## [1] 225335
correlation of the data
# correlation between catch and effort for the whole period
attach(klm)
cor.test(raised,nomeff,method="pearson")
##
##  Pearson's product-moment correlation
##
## data:  raised and nomeff
## t = 434.94, df = 245790, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## 0.6572472 0.6617152
## sample estimates:
##     cor
## 0.659487
##multiple correlation
##Here we select the oilsardine catch. The oilsardine species code as 362
##we pick all the years monthly oil sardine
sp362<-klm[(klm$species=="362"),]
cordat<-sp362[,4:6]
cor(cordat)
raised  nomeff  stdcpue
raised  1.0000000 0.45713639 0.61135090
nomeff  0.4571364 1.00000000 0.06860281
stdcpue 0.6113509 0.06860281 1.00000000

**Linear regression & ANOVA**
fit <- lm(raised~ year + month + nomeff, data=sp362)
# show results
```
summary(fit)
##
## Call:
## lm(formula = raised ~ year + month + nomeff, data = sp362)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -24406856  -5945766   -838374   4725596  40857882
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)  
## (Intercept) -2.1476e+09  2.787e+08  -7.706 5.93e-13 ***
## year         1.0721e+06  1.389e+05   7.716 5.59e-13 ***
## month        7.9972e+05  1.969e+05   4.062 6.97e-05 ***
## nomeff       3.9973e+02  4.493e+01   8.897 3.44e-16 ***
##
## Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

## Residual standard error: 9689000 on 200 degrees of freedom
## Multiple R-squared:  0.4275, Adjusted R-squared:  0.4189
## F-statistic: 49.78 on 3 and 200 DF,  p-value: < 2.2e-16
#
# model coefficients

coefficients(fit)
## (Intercept) year month nomeff
## -2.147604e+09  1.072090e+06  7.997178e+05  3.997276e+02
#
# CIs for model parameters

confint(fit, level=0.95)
##
## (Intercept) year month nomeff
##  -2.697162e+09 -7.706 5.93e-13 ***
##  1.346082e+06  4.062 6.97e-05 ***
##  8.897 3.44e-16 ***

## Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

## Residual standard error: 9689000 on 200 degrees of freedom
## Multiple R-squared:  0.4275, Adjusted R-squared:  0.4189
## F-statistic: 49.78 on 3 and 200 DF,  p-value: < 2.2e-16
# model coefficients
## nomeff       3.111348e+02  4.883205e+02
# predicted values

### fitted(fit)

<table>
<thead>
<tr>
<th></th>
<th>10609</th>
<th>10610</th>
<th>10611</th>
<th>10612</th>
<th>10613</th>
<th>10614</th>
</tr>
</thead>
<tbody>
<tr>
<td>#</td>
<td>-3789651.96</td>
<td>-75345.54</td>
<td>15111313.36</td>
<td>13412874.31</td>
<td>17168949.26</td>
<td>120681.70</td>
</tr>
<tr>
<td>#</td>
<td>10615</td>
<td>10616</td>
<td>10617</td>
<td>10618</td>
<td>10619</td>
<td>10620</td>
</tr>
<tr>
<td>#</td>
<td>11475956.42</td>
<td>2176177.37</td>
<td>4491241.24</td>
<td>20281254.70</td>
<td>10248865.43</td>
<td>6278101.08</td>
</tr>
<tr>
<td>#</td>
<td>10621</td>
<td>10622</td>
<td>10623</td>
<td>10624</td>
<td>10625</td>
<td>10626</td>
</tr>
<tr>
<td>#</td>
<td>1848628.97</td>
<td>-945019.58</td>
<td>10648970.16</td>
<td>18599757.89</td>
<td>1915100.95</td>
<td>4945529.10</td>
</tr>
<tr>
<td>#</td>
<td>10627</td>
<td>10628</td>
<td>10629</td>
<td>10630</td>
<td>10631</td>
<td>10632</td>
</tr>
<tr>
<td>#</td>
<td>1844457.32</td>
<td>4524979.63</td>
<td>8480021.57</td>
<td>27270345.64</td>
<td>26410785.24</td>
<td>7449598.25</td>
</tr>
<tr>
<td>#</td>
<td>10633</td>
<td>10634</td>
<td>10635</td>
<td>10636</td>
<td>10637</td>
<td>10638</td>
</tr>
<tr>
<td>#</td>
<td>8195286.59</td>
<td>18056830.84</td>
<td>12504031.29</td>
<td>4797286.88</td>
<td>690139.61</td>
<td>7333241.94</td>
</tr>
<tr>
<td>#</td>
<td>10639</td>
<td>10640</td>
<td>10641</td>
<td>10642</td>
<td>10643</td>
<td>10644</td>
</tr>
<tr>
<td>#</td>
<td>9086615.20</td>
<td>12777192.22</td>
<td>16114211.77</td>
<td>21825496.12</td>
<td>23957847.88</td>
<td>30125417.82</td>
</tr>
<tr>
<td>#</td>
<td>10645</td>
<td>10646</td>
<td>10647</td>
<td>10648</td>
<td>10649</td>
<td>10650</td>
</tr>
<tr>
<td>#</td>
<td>16794955.21</td>
<td>8159428.15</td>
<td>18423291.70</td>
<td>38539644.49</td>
<td>22526843.37</td>
<td>15428828.71</td>
</tr>
<tr>
<td>#</td>
<td>10651</td>
<td>10652</td>
<td>10653</td>
<td>10654</td>
<td>10655</td>
<td>10656</td>
</tr>
<tr>
<td>#</td>
<td>19942372.43</td>
<td>8463199.11</td>
<td>16820433.97</td>
<td>16852255.88</td>
<td>19772511.73</td>
<td>16832240.83</td>
</tr>
<tr>
<td>#</td>
<td>10657</td>
<td>10658</td>
<td>10659</td>
<td>10660</td>
<td>10661</td>
<td>10662</td>
</tr>
<tr>
<td>#</td>
<td>6812947.52</td>
<td>2187489.33</td>
<td>3280344.12</td>
<td>24388104.43</td>
<td>18000977.41</td>
<td>15107404.98</td>
</tr>
<tr>
<td>#</td>
<td>10663</td>
<td>10664</td>
<td>10665</td>
<td>10666</td>
<td>10667</td>
<td>10668</td>
</tr>
<tr>
<td>#</td>
<td>11071325.90</td>
<td>8804492.99</td>
<td>11659447.99</td>
<td>15882452.30</td>
<td>13614255.15</td>
<td>14360781.30</td>
</tr>
<tr>
<td>#</td>
<td>10669</td>
<td>10670</td>
<td>10671</td>
<td>10672</td>
<td>10673</td>
<td>10674</td>
</tr>
<tr>
<td>#</td>
<td>4963345.25</td>
<td>3874425.71</td>
<td>8638896.83</td>
<td>15820079.63</td>
<td>9947652.94</td>
<td>10608928.30</td>
</tr>
<tr>
<td>#</td>
<td>10675</td>
<td>10676</td>
<td>10677</td>
<td>10678</td>
<td>10679</td>
<td>10680</td>
</tr>
<tr>
<td>#</td>
<td>11831223.68</td>
<td>10715678.08</td>
<td>18370843.69</td>
<td>18033007.59</td>
<td>24787443.71</td>
<td>20792659.27</td>
</tr>
<tr>
<td>#</td>
<td>10681</td>
<td>10682</td>
<td>10683</td>
<td>10684</td>
<td>10685</td>
<td>10686</td>
</tr>
<tr>
<td>#</td>
<td>10734533.89</td>
<td>14786524.50</td>
<td>23586068.72</td>
<td>15174415.81</td>
<td>14696669.45</td>
<td>21641645.35.88</td>
</tr>
<tr>
<td></td>
<td>26747332.20</td>
<td>27817053.16</td>
<td>27904369.27</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

# residuals
## residuals(fit)

<table>
<thead>
<tr>
<th></th>
<th>10609</th>
<th>10610</th>
<th>10611</th>
<th>10612</th>
<th>10613</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5952459.84</td>
<td>12255563.09</td>
<td>-3371411.14</td>
<td>-4445741.27</td>
<td>-8889076.47</td>
</tr>
<tr>
<td></td>
<td>10614</td>
<td>10615</td>
<td>10616</td>
<td>10617</td>
<td>10618</td>
</tr>
<tr>
<td></td>
<td>986134.71</td>
<td>-5748266.48</td>
<td>-336390.21</td>
<td>2807133.26</td>
<td>1645172.74</td>
</tr>
<tr>
<td></td>
<td>10619</td>
<td>10620</td>
<td>10621</td>
<td>10622</td>
<td>10623</td>
</tr>
<tr>
<td></td>
<td>-3629105.70</td>
<td>-4577842.81</td>
<td>3072907.21</td>
<td>3243308.73</td>
<td>-5672890.07</td>
</tr>
<tr>
<td></td>
<td>10624</td>
<td>10625</td>
<td>10626</td>
<td>10627</td>
<td>10628</td>
</tr>
<tr>
<td></td>
<td>-15696727.40</td>
<td>289232.12</td>
<td>2042122.32</td>
<td>1117366.99</td>
<td>2926082.40</td>
</tr>
<tr>
<td></td>
<td>10629</td>
<td>10630</td>
<td>10631</td>
<td>10632</td>
<td>10633</td>
</tr>
<tr>
<td></td>
<td>5230228.43</td>
<td>-20382271.56</td>
<td>-5264124.44</td>
<td>-5075967.51</td>
<td>1491577.71</td>
</tr>
<tr>
<td></td>
<td>10634</td>
<td>10635</td>
<td>10636</td>
<td>10637</td>
<td>10638</td>
</tr>
<tr>
<td></td>
<td>-9837151.49</td>
<td>-6712232.19</td>
<td>-764792.30</td>
<td>-437886.38</td>
<td>2231690.27</td>
</tr>
<tr>
<td></td>
<td>10639</td>
<td>10640</td>
<td>10641</td>
<td>10642</td>
<td>10643</td>
</tr>
<tr>
<td></td>
<td>-1443831.23</td>
<td>-2440345.04</td>
<td>14926587.99</td>
<td>-6794617.92</td>
<td>2635516.43</td>
</tr>
<tr>
<td></td>
<td>10644</td>
<td>10645</td>
<td>10646</td>
<td>10647</td>
<td>10648</td>
</tr>
<tr>
<td></td>
<td>-17311907.92</td>
<td>-5709093.26</td>
<td>4952910.28</td>
<td>-6048902.56</td>
<td>-6642668.40</td>
</tr>
<tr>
<td></td>
<td>10649</td>
<td>10650</td>
<td>10651</td>
<td>10652</td>
<td>10653</td>
</tr>
<tr>
<td></td>
<td>-9406029.73</td>
<td>11491464.13</td>
<td>29486574.30</td>
<td>2963737.40</td>
<td>3482526.36</td>
</tr>
<tr>
<td></td>
<td>10654</td>
<td>10655</td>
<td>10656</td>
<td>10657</td>
<td>10658</td>
</tr>
<tr>
<td></td>
<td>764926.90</td>
<td>5721591.58</td>
<td>-8014761.85</td>
<td>-334238.52</td>
<td>5160023.79</td>
</tr>
<tr>
<td></td>
<td>10659</td>
<td>10660</td>
<td>10661</td>
<td>10662</td>
<td>10663</td>
</tr>
<tr>
<td></td>
<td>3802703.26</td>
<td>-10108379.25</td>
<td>-2107670.27</td>
<td>-3238790.51</td>
<td>6520269.00</td>
</tr>
<tr>
<td></td>
<td>10664</td>
<td>10665</td>
<td>10666</td>
<td>10667</td>
<td>10668</td>
</tr>
<tr>
<td></td>
<td>6117951.47</td>
<td>3707721.08</td>
<td>4118584.97</td>
<td>744008.66</td>
<td>-2535146.08</td>
</tr>
<tr>
<td></td>
<td>10669</td>
<td>10670</td>
<td>10671</td>
<td>10672</td>
<td>10673</td>
</tr>
<tr>
<td></td>
<td>5587891.61</td>
<td>247621.47</td>
<td>-2882708.00</td>
<td>800991.54</td>
<td>-911955.00</td>
</tr>
</tbody>
</table>

# anova table

anova(fit)

## Analysis of Variance Table

## Response: raised
## Analysis of Variance Table

<table>
<thead>
<tr>
<th>Source</th>
<th>DF</th>
<th>Sum Sq</th>
<th>Mean Sq</th>
<th>F value</th>
<th>Pr(&gt;F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>year</td>
<td>1</td>
<td>4.6080e+15</td>
<td>4.6080e+15</td>
<td>49.083</td>
<td>3.663e-11 ***</td>
</tr>
<tr>
<td>month</td>
<td>1</td>
<td>1.9813e+15</td>
<td>1.9813e+15</td>
<td>21.104</td>
<td>7.689e-06 ***</td>
</tr>
<tr>
<td>nomeff</td>
<td>1</td>
<td>7.4316e+15</td>
<td>7.4316e+15</td>
<td>79.159</td>
<td>3.445e-16 ***</td>
</tr>
<tr>
<td>Residuals</td>
<td>200</td>
<td>1.8776e+16</td>
<td>9.3882e+13</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**Signif. codes:**  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

# Covariance Matrix for Model Parameters

```
vcov(fit)

# (Intercept) year month nomeff
(Intercept) 7.767104e+16 -3.872335e+13 28849322448.9 -1.085409e+09
year -3.872335e+13 1.930661e+10 -132736938.4 5.147853e+05
month 2.884932e+10 -1.327369e+08 38753042588.4 -5.204691e+05
nomeff -1.085409e+09 5.147853e+05 -520469.1 2.018502e+03
```

# Regression Diagnostics

```
influence(fit)

# $hat

# 10609 10610 10611 10612 10613 10614
# 0.042348953 0.032174152 0.030947216 0.024014063 0.027363125 0.031587019
# 10615 10616 10617 10618 10619 10620
# 0.018101845 0.031744185 0.029944584 0.028749417 0.028915850 0.042004060
# 10621 10622 10623 10624 10625 10626
# 0.036951680 0.032836278 0.020628210 0.029105061 0.025090117 0.020127986
# 10627 10628 10629 10630 10631 10632
# 0.028928511 0.025311220 0.021317185 0.041136744 0.038894083 0.038442958
# 10633 10634 10635 10636 10637 10638
# 0.024751425 0.032951924 0.018613317 0.018864207 0.027982400 0.015391058
# 10639 10640 10641 10642 10643 10644
# 0.014401572 0.013346093 0.015061997 0.022355644 0.027879390 0.046154691
# 10645 10646 10647 10648 10649 10650
# 0.031627027 0.018558878 0.023833019 0.112821017 0.025427226 0.010871644
# 10651 10652 10653 10654 10655 10656
```
## coefficients

### (Intercept)

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>10609</td>
<td>2.217824e+07</td>
<td>-1.095925e+04</td>
<td>-1.325088e+04</td>
<td>-3.148198546</td>
<td></td>
</tr>
<tr>
<td>10610</td>
<td>4.411931e+07</td>
<td>-2.183848e+04</td>
<td>-2.228032e+04</td>
<td>-4.498752468</td>
<td></td>
</tr>
<tr>
<td>10611</td>
<td>-1.067489e+07</td>
<td>5.318300e+03</td>
<td>5.379473e+03</td>
<td>-1.436946526</td>
<td></td>
</tr>
<tr>
<td>10612</td>
<td>-1.430707e+07</td>
<td>7.125744e+03</td>
<td>5.005198e+03</td>
<td>-1.244058740</td>
<td></td>
</tr>
<tr>
<td>10613</td>
<td>-2.792623e+07</td>
<td>1.393898e+04</td>
<td>.644383e+03</td>
<td>-3.898604484</td>
<td></td>
</tr>
<tr>
<td>10614</td>
<td>3.637567e+06</td>
<td>-1.803856e+03</td>
<td>-6.792737e+01</td>
<td>-0.548821439</td>
<td></td>
</tr>
<tr>
<td>10615</td>
<td>-1.912700e+07</td>
<td>9.531031e+03</td>
<td>-1.168978e+03</td>
<td>-0.136134257</td>
<td></td>
</tr>
<tr>
<td>10616</td>
<td>-1.236679e+06</td>
<td>6.142401e+02</td>
<td>2.614444e+02</td>
<td>0.182574103</td>
<td></td>
</tr>
<tr>
<td>10617</td>
<td>1.017484e+07</td>
<td>-5.060185e+03</td>
<td>3.311361e+03</td>
<td>-1.300911103</td>
<td></td>
</tr>
<tr>
<td>10618</td>
<td>5.221933e+06</td>
<td>-2.616049e+03</td>
<td>2.285340e+03</td>
<td>0.594874799</td>
<td></td>
</tr>
<tr>
<td>10619</td>
<td>-1.269309e+07</td>
<td>6.332354e+03</td>
<td>-7.146199e+03</td>
<td>0.885644012</td>
<td></td>
</tr>
<tr>
<td>10620</td>
<td>-1.689093e+07</td>
<td>8.416379e+03</td>
<td>-1.142621e+04</td>
<td>2.385068449</td>
<td></td>
</tr>
<tr>
<td>10621</td>
<td>9.988869e+06</td>
<td>-4.931698e+03</td>
<td>-6.845283e+03</td>
<td>-1.449495213</td>
<td></td>
</tr>
<tr>
<td>10622</td>
<td>1.048887e+07</td>
<td>-5.182988e+03</td>
<td>-5.814728e+03</td>
<td>-1.523215775</td>
<td></td>
</tr>
<tr>
<td>10623</td>
<td>-1.631084e+07</td>
<td>8.103095e+03</td>
<td>8.519957e+03</td>
<td>-0.699865368</td>
<td></td>
</tr>
<tr>
<td>10624</td>
<td>-4.218674e+07</td>
<td>2.105372e+04</td>
<td>1.871018e+04</td>
<td>-8.082331986</td>
<td></td>
</tr>
<tr>
<td>10625</td>
<td>9.242638e+05</td>
<td>-4.579190e+02</td>
<td>-1.489350e+02</td>
<td>-0.132336511</td>
<td></td>
</tr>
<tr>
<td>10626</td>
<td>6.358893e+06</td>
<td>-3.155937e+03</td>
<td>-2.504379e+02</td>
<td>-0.691128004</td>
<td></td>
</tr>
<tr>
<td>10627</td>
<td>3.641035e+06</td>
<td>-1.805648e+03</td>
<td>3.989493e+02</td>
<td>-0.629386219</td>
<td></td>
</tr>
<tr>
<td>10628</td>
<td>9.337116e+06</td>
<td>-4.637748e+03</td>
<td>2.201757e+03</td>
<td>-1.355018464</td>
<td></td>
</tr>
</tbody>
</table>

### $sigma$

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>10609</td>
<td>10610</td>
<td>10611</td>
<td>10612</td>
<td>10613</td>
<td>10614</td>
</tr>
<tr>
<td>9704033</td>
<td>9673382</td>
<td>9710573</td>
<td>9708368</td>
<td>9692571</td>
<td>9713348</td>
</tr>
<tr>
<td>10615</td>
<td>10616</td>
<td>10617</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9704033</td>
<td>9673382</td>
<td>9710573</td>
<td>9708368</td>
<td>9692571</td>
<td>9713348</td>
</tr>
<tr>
<td>10618</td>
<td>10619</td>
<td>10620</td>
<td>10621</td>
<td>10622</td>
<td>10623</td>
</tr>
</tbody>
</table>
Recent Advances in Fishery Biology techniques for Biodiversity Evaluation and Conservation

---

## $wt.res

<table>
<thead>
<tr>
<th></th>
<th>5952459.84</th>
<th>12255563.09</th>
<th>-3371411.14</th>
<th>-4445741.27</th>
<th>-8889076.47</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10614</td>
<td>10615</td>
<td>10616</td>
<td>10617</td>
<td>10618</td>
</tr>
<tr>
<td></td>
<td>986134.71</td>
<td>-5748266.48</td>
<td>-336390.21</td>
<td>2807133.26</td>
<td>1645172.74</td>
</tr>
<tr>
<td></td>
<td>10619</td>
<td>10620</td>
<td>10621</td>
<td>10622</td>
<td>10623</td>
</tr>
<tr>
<td></td>
<td>-3629105.70</td>
<td>-4577842.81</td>
<td>3072907.21</td>
<td>3243308.73</td>
<td>-5672890.07</td>
</tr>
<tr>
<td></td>
<td>10624</td>
<td>10625</td>
<td>10626</td>
<td>10627</td>
<td>10628</td>
</tr>
</tbody>
</table>
Plots in R

```r
## scatter plot
sp3621 <- sp362[, c(1:2,4)]
attach(sp3621)
## The following objects are masked from klm:
##
## month, raised, year
plot(year, raised, main = "sardine catch[1997-2013]", xlab = "year", ylab = "catch(kg)")
```
##Histogram

```r
hist(raised, main="Histogram for oilsardine catch[1997-2013]",
lab="catch",
col="green",
breaks=5)
```

![Histogram](image1.png)

##Bar plot

```r
barplot(raised, main="sardine catch Distribution",
xlab="Number of years")
```

![Bar Plot](image2.png)
Boxplot in r

# Boxplot of catch vs month

```r
boxplot(raised~month, data=sp3621, main="Sardine catch ",
lab="months", ylab="catch(kg)", col=rainbow(length(unique(month))))
```

to plot a correlation in r

```r
##we select sardine correlations
cordat<-sp362[,4:6]
library(PerformanceAnalytics)
chart.Correlation(cordat, method="pearson")
```
R for reading NetCDF data

NetCDF files contain one or more variables, which are usually structured as regular N-dimensional arrays. For example, you might have a variable named “Temperature” that is a function of longitude, latitude, and height. NetCDF files also contain dimensions, which describe the extent of the variables’ arrays. In our Temperature example, the dimensions are “longitude”, “latitude”, and “height”. Data can be read from or written to variables in arbitrary hyperslabs (for example, you can read or write all the Temperature values at a given height, or at a given latitude).

The R package ‘ncdf4’ allows reading from, writing to, and creation of netCDF files, either netCDF version 3 or (optionally) netCDF version 4. If you choose to create version 4 output files, be aware that older netcdf software might only be able to read version 3 files.

In fact this package can help extracting details from HDF5 format files too. This package can create NetCDF files from data.frames also. Nc_open() is the function to be used for opening a NetCDF file and for creating a NetCDF file the function is nc_creat(). Once opened the attributes and variable names of the data can be got by using the generic print() command. To get specific variables the function is ncvar_get()

An example:

library(ncdf4)
ncold <- nc_open(“states_population.nc”)
data <- ncvar_get(ncold)
print(“here is the data in the file:”)
print(data)
nc_close( ncold )
The output is given below:
> ncold <- nc_open(“states_population.nc”)
> print(ncold)
File states_population.nc (NC_FORMAT_CLASSIC):
1 variables (excluding dimension variables):
int Pop[StateNo]
units: count
_FillValue: -1
long_name: Population
1 dimensions:
StateNo  Size:50
units: count
long_name: StateNo
1 global attributes:
source: Census 2000 from census bureau web site
>

R in numerical methods

Taking cue from the fact that integration is infinitesimal addition, brutal algorithmic power of R has been put to use to find solutions of definite integrals. The most common function used for this purpose is integrate().

An example:

For the double integral given below

$$\int_{0}^{1} \int_{x}^{1} x \sin(y^2) dy dx$$

A couple of lines as given below would do the job in R environment

```r
integrate(function(x) {
sapply(x, function(x) {
integrate(function(y) x*sin(y^2),x,1)$value
})
},0,1)
```

The output is given below (with error measure)

```r
> integrate(function(x) {
+ sapply(x, function(x) {
+ integrate(function(y) x*sin(y^2),x,1)$value
+ })
+ },0,1)
0.09105548 with absolute error < 1e-15
>
```
References


