STAT 3355
Introduction to Data Analysis

Lecture 05: R Basics IV

Created by Dr. Qiwei Li
Presented by Dr. Octavious Smiley

Department of Mathematical Sciences
The University of Texas at Dallas

Last Class

- Data in R
  - Basic data modes: numeric, integer, character, factor, logical
  - Basic data classes: data vector, data matrix, data frame

Learning Goals

- Where can I find a dataset?
- Use `R` to read a dataset
- Use `R` to write a dataset

Table of Contents

External Sources

- Typing in datasets is tedious
  - Large size
  - Possible numerous inputting errors
- Data are everywhere
  - Built-in datasets in existing `R` packages
  - Datasets on webpages
  - Your own datasets

R Packages

- R is designed to have a small code kernel, as additional functionalities can take up memory that may be in short supply
- Over $10,000$ R packages are stored in the Comprehensive R Archive Network (CRAN): r-project.org
- To use a dataset in a package named a
  - Install the package via the function `install.packages("a")`
  - Load the package via the function `library(a)`

R Package: SAFARI

- https://cran.r-project.org/web/packages/SAFARI/index.html

SAFARI: Shape Analysis for AI-Reconstructed Images

Provides functionality for image processing and shape analysis in the context of reconstructed medical images generated by deep learning-based methods or standard image processing algorithms and produced from different medical imaging types, such as X-ray, Computational Tomography (CT), Magnetic Resonance Imaging (MRI), and pathology imaging. Specifically, offers tools to segment regions of interest and to extract quantitative shape descriptors for applications in signal processing, statistical analysis and modeling, and machine learning.

Version:	0.1.0
Depends:	R (≥ 3.5.0)
Imports:	caTools, EBImage, graphics, lattice, png
Published:	2021-02-25
Author:	Esteban Fernandez Morales [aut, cre], Qiwei Li [aut]
Maintainer:	Esteban Fernandez Morales <esteban.fernandezmorales at utdallas.edu>
License:	GPL (≥ 3)
URL:	https://github.com/estfernandez/SAFARI
NeedsCompilation:	no
Materials:	README NEWS
CRAN checks:	SAFARI results

Documentation:

Reference manual: SAFARI.pdf

Downloads:

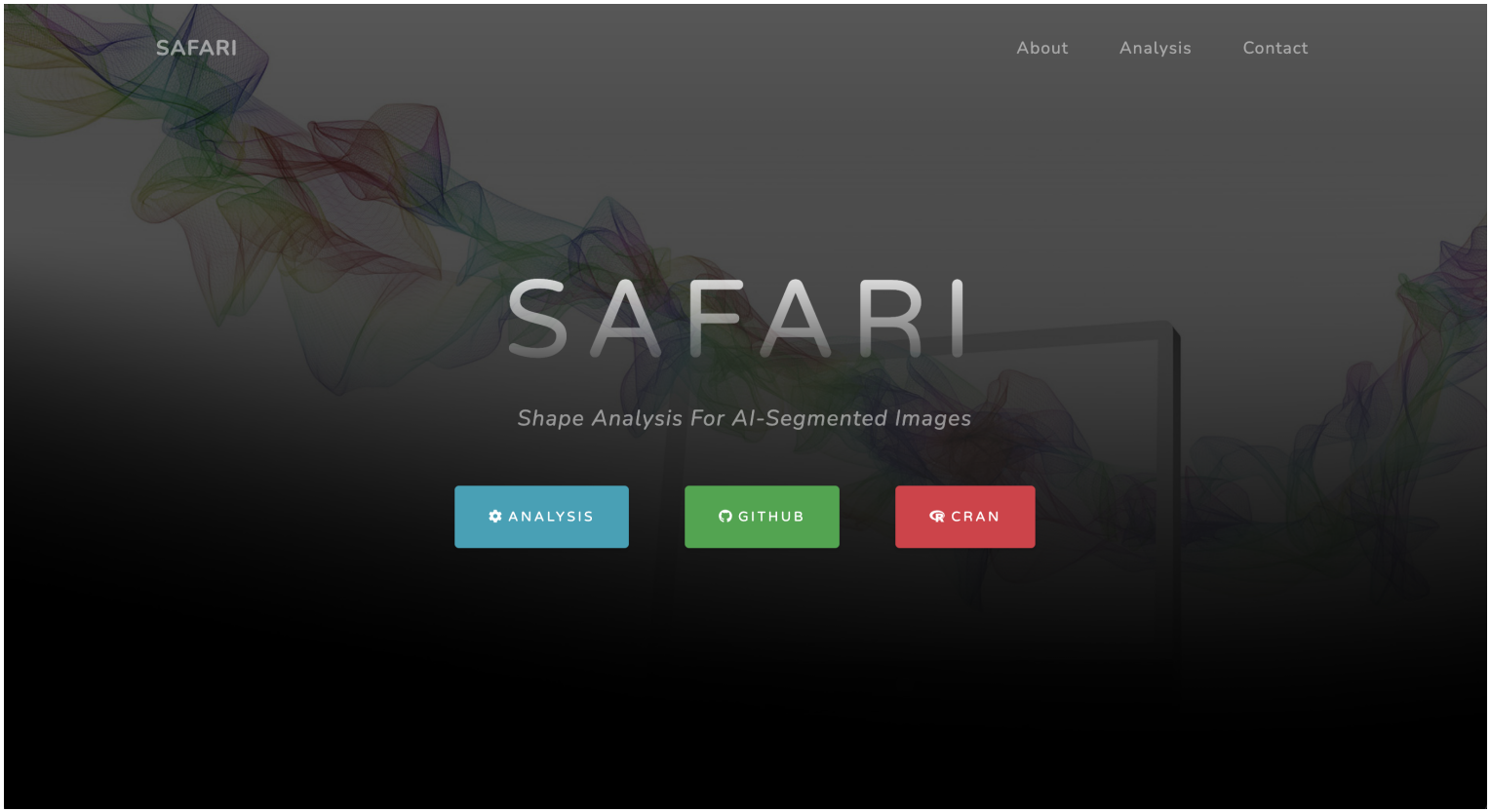
Package source:	SAFARI_0.1.0.tar.gz
Windows binaries:	r-devel: SAFARI_0.1.0.zip, r-release: SAFARI_0.1.0.zip, r-oldrel: SAFARI_0.1.0.zip
macOS binaries:	r-release (arm64): SAFARI_0.1.0.tgz, r-oldrel (arm64): SAFARI_0.1.0.tgz, r-release (x86_64): SAFARI_0.1.0.tgz, r-oldrel (x86_64): SAFARI_0.1.0.tgz

Linking:

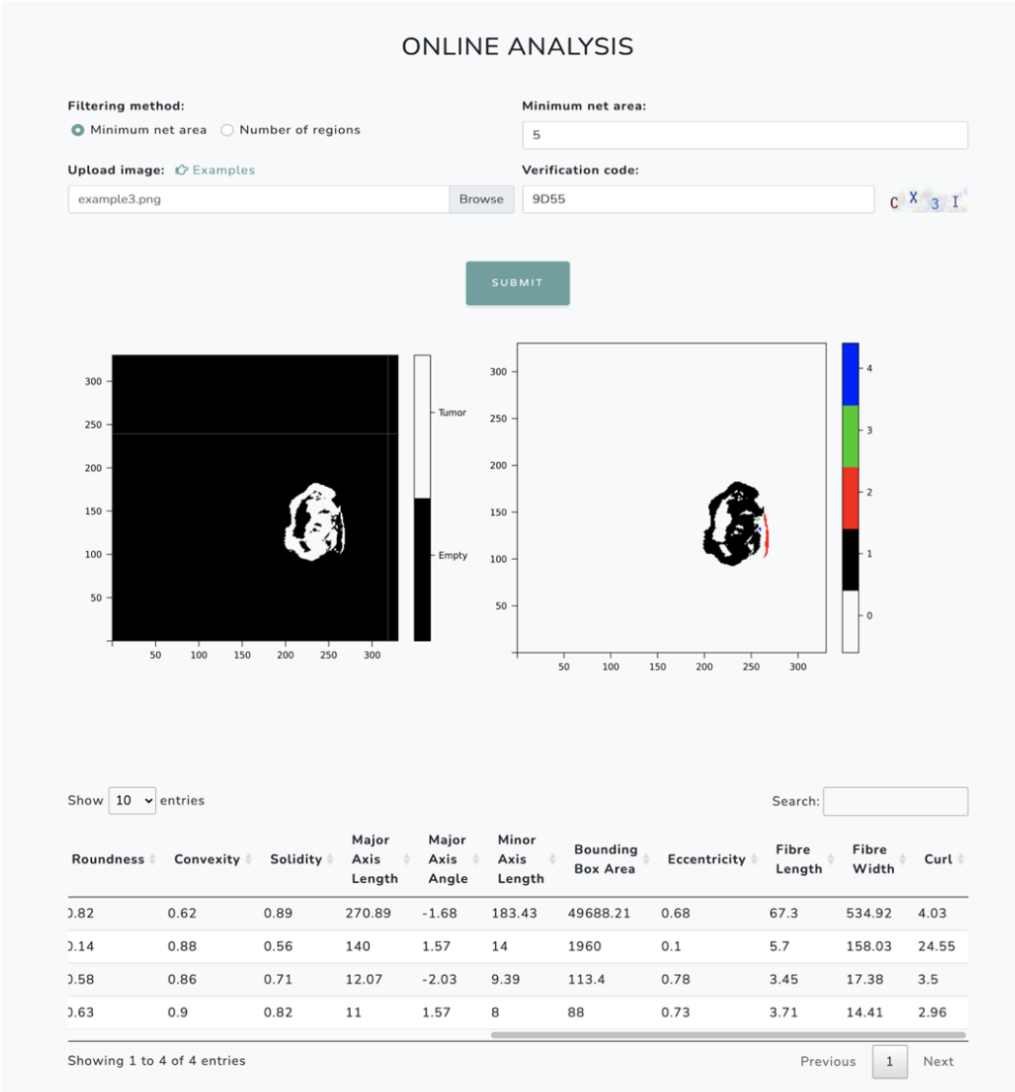
Please use the canonical form https://CRAN.R-project.org/package=SAFARI to link to this page.

R Package: SAFARI

- https://lce.biohpc.swmed.edu/safari/



R Package: SAFARI



R Package: SAFARI

Fernández *et al. BMC Medical Imaging* (2022) 22:129
https://doi.org/10.1186/s12880-022-00849-8

BMC Medical Imaging

SOFTWARE Open Access

SAFARI: shape analysis for AI-segmented images

Esteban Fernández[1], Shengjie Yang[2], Sy Han Chiou[1], Chul Moon[3], Cong Zhang[1], Bo Yao[2], Guanghua Xiao[2*] and Qiwei Li[1*]

Abstract

Background: Recent developments to segment and characterize the regions of interest (ROI) within medical images have led to promising shape analysis studies. However, the procedures to analyze the ROI are arbitrary and vary by study. A tool to translate the ROI to analyzable shape representations and features is greatly needed.

Results: We developed SAFARI (shape analysis for AI-segmented images), an open-source R package with a user-friendly online tool kit for ROI labelling and shape feature extraction of segmented maps, provided by AI-algorithms or manual segmentation. We demonstrated that half of the shape features extracted by SAFARI were significantly associated with survival outcomes in a case study on 143 consecutive patients with stage I–IV lung cancer and another case study on 61 glioblastoma patients.

Conclusions: SAFARI is an efficient and easy-to-use toolkit for segmenting and analyzing ROI in medical images. It can be downloaded from the comprehensive R archive network (CRAN) and accessed at https://lce.biohpc.swmed.edu/safari/.

Keywords: Medical imaging, Machine learning, Shape representations, Shape descriptors

Dataset `mpg`

- In `R` package `ggplot2`
- Fuel economy data from 1999 and 2008 for 38 popular models of car
  - A subset of the fuel economy data that the Environmental Protection Agency (EPA) makes available on fueleconomy.gov
  - Contains only models which had a new release every year between 1999 and 2008
  - $n = 234$ car models and $p = 11$ variables
  - Variables include manufacturer, model name (model), engine displacement in litres (displ), year of manufacture (year), number of cylinders (cyl), type of transmission (trans), drive type (drv), city miles per gallon (cty), highway miles per gallon (hwy), fuel type (fl), and type of car (class)

Dataset mpg

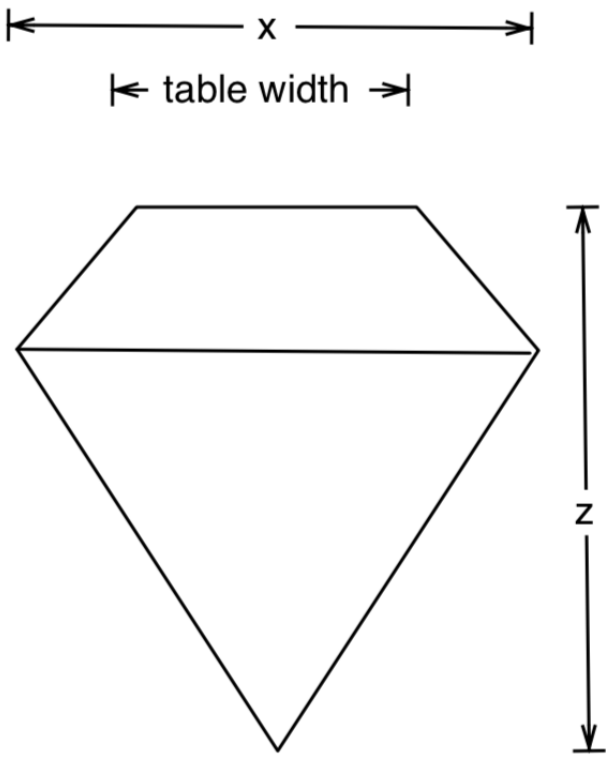
- Examples

```
install.packages("ggplot2")
library(ggplot2)
data(mpg)

# Summary
?mpg
head(mpg)
str(mpg)
summary(mpg)
```

Dataset `diamonds`

- In `R` package `ggplot2`
- Prices of $54,000$ round cut diamonds
  - A dataset containing the prices and other attributes of almost $54,000$ diamonds makes available on diamondse.info
  - $n = 53,940$ diamonds and $p = 10$ variables
  - Variables include price in USD (price), weight (carat), cut quality (cut), color (color), clearness (clarity), length (x), width (y), depth (z), total depth percentage (depth), and width of top relative to widest point (table)

Dataset `diamonds`



depth = z / diameter
table = table width / x * 100

Dataset diamonds

- Examples

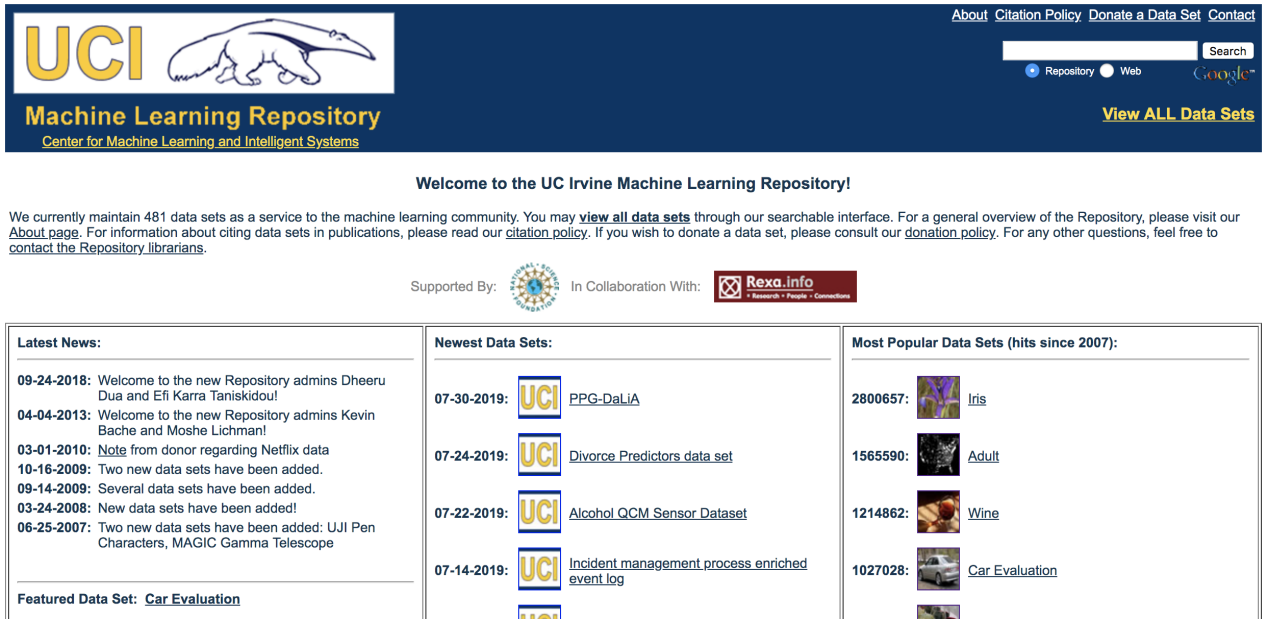
```
install.packages("ggplot2")
library(ggplot2)
data(diamonds)

# Summary
?diamonds
head(diamonds)
str(diamonds)
summary(diamonds)
```

Datasets in `UsingR`

- A collection of datasets to accompany the textbook *Using R for Introductory Statistics (2nd Ed.)* by John Verzani
- About 100 small datasets from different sources
- We will use many datasets in this package to illustrate the data summary concepts in future classes

UCI Machine Learning Repository

- Link: archive.ics.uci.edu/ml/index.php
- Maintain 481 datasets as a service to the machine learning (ML) community
- Many are benchmark datasets to evaluate different ML/statistical methods

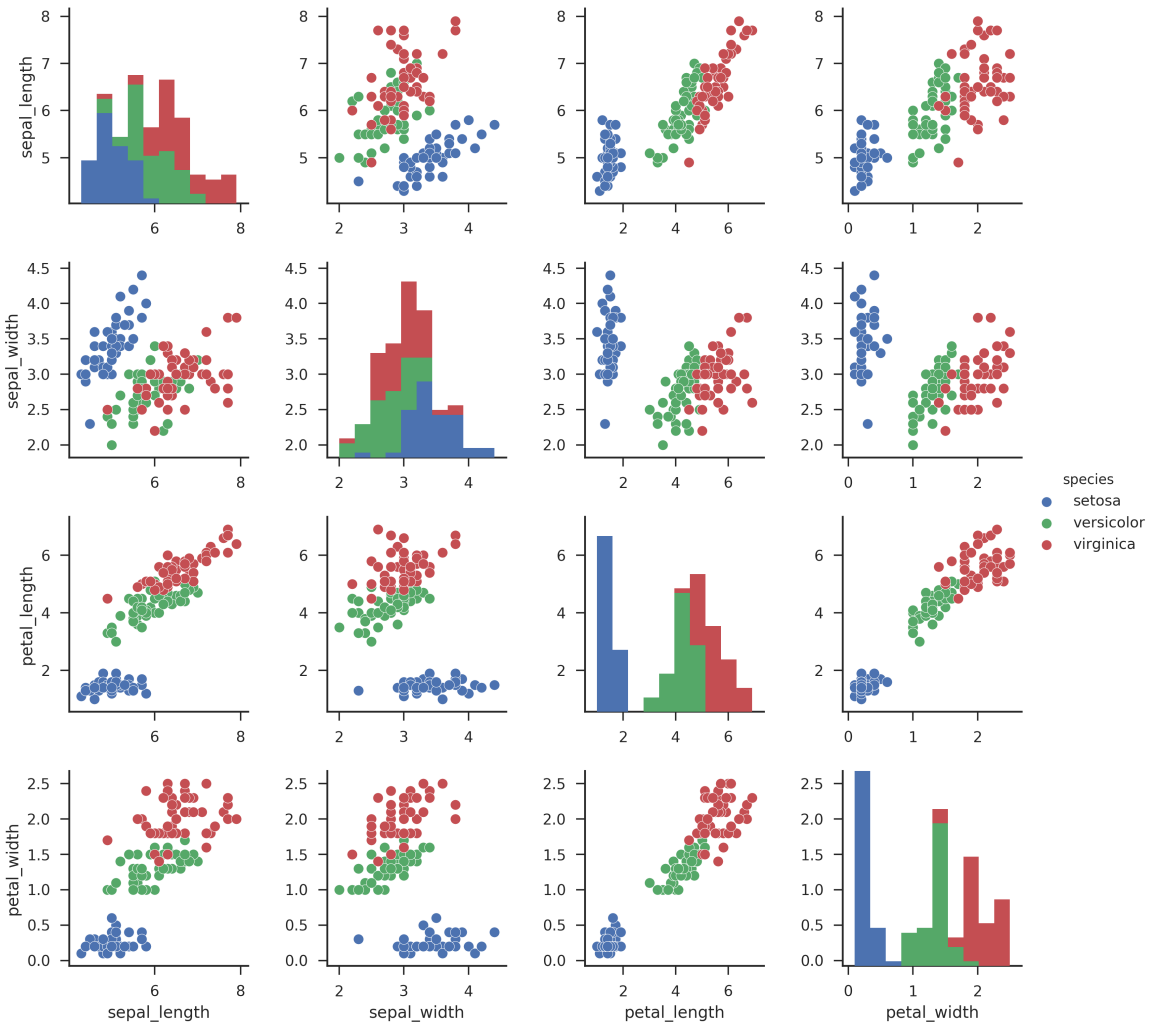


Iris Dataset

- Best known database to be found in the pattern recognition literature, which was collected by Ronald A. Fisher in 1936
  - A dataset containing the sepal and petal size for three types of iris plants: setosa, versicolour, and virginica
  - $n = 150$ iris flowers and $p = 5$ variables
  - Variables include sepal length and width, petal length and width, and type

Iris Dataset

Iris Dataset

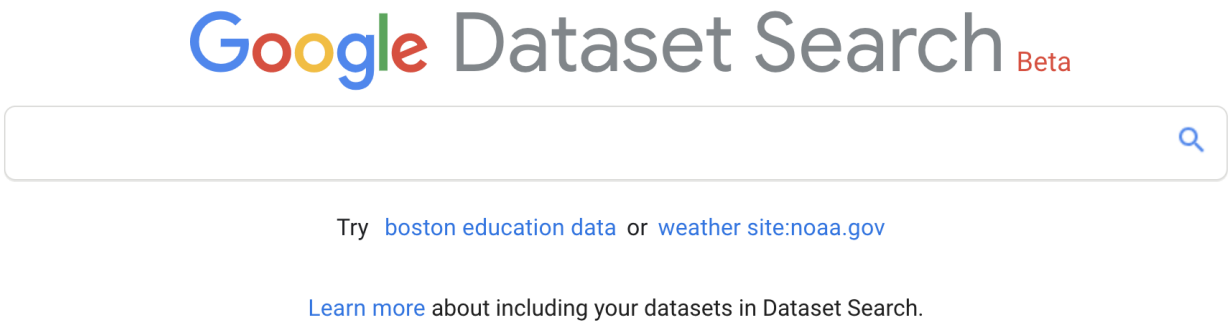


Adult Dataset

- Predict whether income exceeds $\$50,000$ per year based on U.S. census data in 1994
  - A dataset containing the sociodemographic information extracted from the census bureau database found at www.census.gov
  - $n = 32,561$ individuals and $p = 15$ variables
  - Variables include age, workclass, weight, education level, martial status, occupation, relationship, race, sex, capital gain, capital loss, working hours per week, native country, and if the annual income exceeds $\$50,000$

Many More

- Google Dataset Search Engine



Working Directory

- The current working directory of `R` process
- Remember to set your working directory before opening a dataset and working on it
- View your working directory via
  - The function `getwd()`
  - "Go To Working Directory" in the Output window
- Set your working directory via
  - The function `setwd(p)`, where p is a canonical path name separated by / (Mac) or \ (Windows)
  - "Set As Working Directory" in the Output window

Working Directory

- Examples

```
# Without setting the working directory
iris <- read.table("~/Desktop/stat3355/data/
    iris.data")

# Set the working directory
setwd("~/Desktop/stat3355/data/")
iris <- read.table("iris.data")
```

Reading Functions

- `read.table(f, header = FALSE, sep = "", na.strings = "NA", stringsAsFactors = TRUE)`
  - f is a canonical file path
  - `header`: A logical value indicating whether the file contains the names of the variables as its first line
  - `sep`: The field separator character with default value setting to space
  - `na.strings`: A character vector of strings which are to be interpreted as missing values
  - `stringsAsFactors`: A logical value indicating should character vectors be converted to factors

Reading Functions

- `read.csv(f, header = TRUE, sep = ",")`
  - For "comma separated value" files
  - Can be imported to and exported from Microsoft Excel
  - `read.csv2()` for "semicolon separated value" files
- `read.delim(f, header = TRUE, sep = "\t")`
  - For "tab separated value" files
- `read.fwf(f, widths, header = FALSE, sep = "\t")`
  - For a table of fixed width formatted data
  - `widths`: A integer vector that gives widths for multiple variables

Reading Iris Dataset

- Examples

```
iris <- read.table("iris.data")
iris <- read.table("iris.data", sep = ",")
names(iris) <- c("sepal_length", "sepal_
    width", "petal_length", "petal_width", "
    type")

iris <- read.csv("iris.data")
iris <- read.csv("iris.data", header = FALSE
    )
names(iris) <- c("sepal_length", "sepal_
    width", "petal_length", "petal_width", "
    type")

str(iris)
```

Reading Adult Dataset

- Examples

```
adult <- read.table("adult.data")
adult <- read.csv("adult.data", header =
    FALSE)
str(adult)

# Once you know the missing code
adult <- read.csv("adult.data", header =
    FALSE, na.string = " ?")
str(adult)
adult$V2[1:30]

# The percent of missing data for the second
    variable
sum(is.na(adult$V2))/dim(adult)[2]*100
```

Special Values in R

- Missing values `NA`
  - Dataset may be incomplete
  - Could be estimated from the rest of data
  - Most of time, it is not a zero!!!
  - Use the argument `na.rm = TRUE` in most functions that summarize a numeric vector or matrix

Special Values in R

- Null values: `NULL`
  - A reserved value
- Not a number values: `NaN`
  - Arises from arithmetic operations that are undefined, such as $0/0$
- Infinity values: `Inf`
  - The maximum numeric value R can store: 2^{1024}

Other Reading Functions

- Spreadsheets
  - The function `read.xlsx()` provided by the `R` package `xlsx` (with Java)
  - The function `read.xls()` provided by the `R` package `gdata` (with Perl)
  - The function `read.excel()` for both xlsx and xls files provided by the `R` package `readxl`

Other Reading Functions

- Other data formats in Statistics
  - The function `read_sas()` for reading sas7bdat and sas7bcat files (SAS)
  - The function `read_sav()` for reading sav files (SPSS)
  - The function `read_dta()` for reading dta files up to version 15 (Stata)
  - All are provided by the `R` package `haven`

Data Loading Process

- A recommended path to load a dataset in R
  - Download the dataset and save as a txt file
  - Open the txt file with Microsoft Excel (using Text Import Wizard)
  - Add variable names, replace all missing values with blank, and save as a csv file
  - Move the csv file into your working directory
  - Load the data via the function `read.csv(f, stringsAsFactors = FALSE)`
  - Check the dataset with the function `str()`, `summary()`, and `head()`

Data Cleaning Process

- A recommended path to clean a dataset in R
  - Convert uninformative numbers or characters to informative labels
  - For numeric variables: rounding, truncating, or binning if necessary
  - For character variables: turning into factor or logical variables if necessary

Example

- Load and clean the adult dataset by
  - Read the data by using the function `read.table()` with the arguments: `sep = ","`, `header = FALSE`, `na.string = "?"`, `stringsAsFactor = FALSE`
  - Name each variable according to archive.ics.uci.edu/ml/datasets/Adult
  - Turn the last variable into a logical variable, where `TRUE` corresponds to an anuual income greater than $\$50,000$, and `FALSE` otherwise
  - *Turn the education variable into an ordered factor variable with levels: preschool, elementary school (1st – 4th, 5th – 6th), junior high school (7th – 8th), senior high school (9th, 10th, 11th, 12th, HS-grad), undergraduate (Prof-school, Assoc-acdm, Assoc-voc, Some-college, Bachelors), and graduate (Masters and Doctorate)

Example

- Solutions

```
# Load the dataset after moving it to your
    working directory
adult <- read.table("adult.data", sep = ",",
     header = FALSE, na.strings = " ?",
    stringsAsFactors = FALSE);

# Name each variable
names(adult) <- c("age", "workclass", "
    weight", "education", "education_num", "
    marital", "occupation", "relationship", "
    race", "sex", "capital_gain", "capital_
    loss", "hours_per_week", "coutnry", "
    annual_income")
```

Example

- Solutions

```
# Turn the annual_income variable to logical
index_of_those_less_than_50K <- which(adult$
    annual_income == " <=50K")
adult$annual_income[index_of_those_less_than
    _50K] <- FALSE

index_of_those_greater_than_50K <- which(
    adult$annual_income == " >50K")
adult$annual_income[index_of_those_greater_
    than_50K] <- TRUE

adult$annual_income <- as.logical(adult$
    annual_income)
```

Example

- Solutions

```
# Turn the education variable to factor
index_elementary <- which(adult$education %
    in% c(" 1st-4th", " 5th-6th"))
adult$education[index_elementary] <- "
    Elementary_school"

index_junior <- which(adult$education %in% c
    (" 7th-8th"))
adult$education[index_junior] <- "Junior_
    high_school"
```

Example

- Solutions

```
index_senior <- which(adult$education %in% c
    (" 9th", " 10th", " 11th", " 12th", " HS-
    grad"))
adult$education[index_senior] <- "Senior_
    high_school"

index_undergraduate <- which(adult$education
     %in% c(" Prof-school", " Assoc-acdm", "
    Assoc-voc", " Some-college", " Bachelors"
    ))
adult$education[index_undergraduate] <- "
    Undergraduate"
```

Example

- Solutions

```
index_graduate <- which(adult$education %in%
    c(" Masters", " Doctorate"))
adult$education[index_graduate] <- "Graduate
    "

adult$education[which(adult$education == "
    Preschool")] <- "Preschool"

adult$education <- factor(adult$education,
    levels = c("Preschool", "Elementary_
    school", "Junior_high_school", "Senior_
    high_school", "Undergraduate", "Graduate"
    ))
```

Example

- Alternative solutions

```
adult$education_new <- cut(adult$education_
    num, breaks = c(1, 2, 4, 5, 10, 15, 16),
    labels = c("preschool", "elementary", "
    junior", "high", "undergraduate", "
    graduate"))
```

Writing Functions for Formatted Files

- `write.table(a, f, col.names = TRUE, quote = TRUE, sep = "", na = "NA")`
  - a is the variable name of a data frame X
  - f is a canonical file path ending with txt
  - `col.names`: A logical value indicating whether the column names of X are to be written along
  - `quote`: A logical value indicating whether those character or factor columns will be surrounded by double quotes
  - `sep`: The field separator character with default value setting to space
  - `na`: The string to use for missing values in the data

Writing Functions for Formatted Files

- `write.csv(a, f, row.names = FALSE, na = "NA")`
  - a is the variable name of a data frame X
  - f is a canonical file path ending with csv
  - `row.names`: A logical value indicating whether the row names of X are to be written along
  - `na`: The string to use for missing values in the data
  - Each value is separated by a comma

Writing Functions for Objects

- `save(a, f)`
  - a is the variable name of an object
  - Can store multiple objects
  - f is a canonical file path ending with RData
  - RData can be loaded by `R` with the function `load()`
- `save.image(f)`
  - Save all the objects in your working space

Your Turn

- Download the "Adult" dataset on the UCI Machine Learning Repository. The link is https://archive.ics.uci.edu/ml/datasets/Adult. Read the data in its original format (.data) by using the function `read.table()` or `read.csv()` in an appropriate way.
- Transform and save the adult dataset by
  - Change the unit of capital gain and capital loss variables from USD to JPY $(1 : 109)$
  - Only keep the subjects with master or above degree
  - Save the data into a csv file on your desktop

Your Turn

- Solutions

```
# Load data
adult <- read.csv("adult.data", header =
    FALSE)
names(adult) <- c("age", "workclass", "
    fnlwgt", "edu", "edu_num", "martial", "
    occu", "relationship", "race", "sex", "
    gain", "loss", "hours", "native", "income
    ")

# Multiply all entries in the capital_gain
    and capital_loss variables by 109
adult$gain <- adult$gain*109
adult$loss <- adult$loss*109
```

Your Turn

- Solutions

```
# Create a new data frame only contains
    subjects with master or above
index <- which(adult$edu == " Masters" |
    adult$edu == " Doctorate")
adult_new <- adult[index,]

# Write the new data frame into a csv file
write.csv(adult_new, file = "adult_new.csv")
```