

MACHINE LEARNING - THE BASICS IN R

Jan-Philipp Kolb

31 Mai, 2019

INTRODUCTION ROUND

PLEASE TELL US SHORTLY...

- ▶ Where are you from? What are you studying/working?
- ▶ What is your experience level in R/other programming languages?
- ▶ What are your expectations of this course?
- ▶ Where do you think you can use Machine Learning in the future?

PRELIMINARIES

- ▶ This topic is huge - we concentrate on presenting the applications in R
- ▶ Usually we have big differences in knowledge and abilities of the participants - please tell, if it is too fast or slow.
- ▶ We have many **exercises** because at the end you can only learn on your own
- ▶ We have many **examples** - try them!
- ▶ If there are questions - always ask
- ▶ R is more fun together - ask your neighbor

CONTENT OF THIS SECTION

- ▶ The first section is about laying the foundations in R. We will need all things covered later on.

TOPICS SECTION:

- ▶ Why R is a good choice
- ▶ Constraints of R-usage
- ▶ R is modular
- ▶ Import and export of data

WHY R IS A GOOD CHOICE . . .

- ▶ ... because it is an **open source language**
- ▶ ... outstanding graphs - **graphics, graphics, graphics**
- ▶ ... relates to other languages - **R can be used in combination with other programs** - e.g. **data linking**
- ▶ ... R can be used **for automation**
- ▶ ... Vast Community - **you can use the intelligence of other people ;-)** and new statistical methodologies are implemented quite fast
- ▶ Because R can be combined with other programs like PostgreSQL or Python

CONSTRAINTS

NEWER MODULES IN PYTHON

- ▶ Machine learning is a field that changes rapidly.
- ▶ Some new tools are first developed in Python.
- ▶ The package `reticulate` offers the possibility to use these modules from an R environment.
- ▶ Good news - Python is also Open Source

BIG DATA

- ▶ Especially if you work with web data, you quickly have to deal with large amounts of data.
- ▶ Therefore one must fall back on databases and parallelization strategies, which can be used in R.

R IS MODULAR

INSTALL PACKAGES FROM CRAN SERVER

```
install.packages("lme4")
```

INSTALL PACKAGES FROM BIOCONDUCTOR SERVER

```
source("https://bioconductor.org/biocLite.R")  
biocLite(c("GenomicFeatures", "AnnotationDbi"))
```

INSTALL PACKAGES FROM GITHUB

```
install.packages("devtools")  
library(devtools)  
  
devtools::install_github("koalaverse/vip")
```

TASK VIEW MACHINE LEARNING

CRAN Task View: Machine Learning & Statistical Learning

Maintainer: Torsten Hothorn

Contact: Torsten.Hothorn at R-project.org

Version: 2018-08-05

URL: <https://CRAN.R-project.org/view=MachineLearning>

Several add-on packages implement ideas and methods developed at the borderline between computer science and statistics - this field of research is usually referred to as machine learning. The packages can be roughly structured into the following topics:

- *Neural Networks and Deep Learning* : Single-hidden-layer neural network are implemented in package [nnet](#) (shipped with base R). Package [RSNNs](#) offers an interface to the Stuttgart Neural Network Simulator (SNNS). [nnt](#) implements recurrent neural networks. Packages implementing deep learning flavours of neural networks include [deepnet](#) (feed-forward neural network, restricted Boltzmann machine, deep belief network, stacked autoencoders), [RcppDL](#) (denoising autoencoder, stacked denoising autoencoder, restricted Boltzmann machine, deep belief network) and [h2o](#) (feed-forward neural network, deep autoencoders). An interface to [tensorflow](#) is available in [tensorflow](#).

INSTALL ALL PACKAGES OF A TASK VIEW

```
install.packages("ctv")  
ctv::install.views("MachineLearning")
```

EXERCISE: FIND R-PACKAGES

Go to <https://cran.r-project.org/> and search for packages that can be used:

- 1) to reduce overfitting
- 2) for regression trees
- 3) for gradient boosting
- 4) for neural networks
- 5) for clustering

PREPARATION - PACKAGES

```
library(dplyr)
```

Introduction to dplyr

When working with data you must:

- Figure out what you want to do.
- Describe those tasks in the form of a computer program.
- Execute the program.

The dplyr package makes these steps fast and easy:

- By constraining your options, it helps you think about your data manipulation challenges.
- It provides simple “verbs”, functions that correspond to the most common data manipulation tasks, to help you translate your thoughts into code.
- It uses efficient backends, so you spend less time waiting for the computer.

THE PACKAGE MAGRITTR

```
library(magrittr)
```

%>%

magrittr

Ceci n'est pas un pipe.

IMPORT .CSV DATA

THE READ.CSV COMMAND

- Use `read.csv2` for German data

```
?read.csv
```

```
?read.csv2
```

USING A PATH TO IMPORT DATA

```
path1<-"https://raw.githubusercontent.com/"  
path2<- "thomaspernet/data_csv_r/master/data/"  
dname <- "titanic_csv.csv"  
titanic <- read.csv(paste0(path1,path2,dname))
```

SAVE THE DATASET

```
save(titanic,file="../data/titanic.RData")
```

THE TITANIC DATASET

pclass	survived	name	sex	age	sibsp	parch	ticket	fare	cabin	embarked	home.dest
1	1	Allen, Miss. Elisabeth...	female	29.0000	0	0	24160	211.3375	B5	S	St Louis, MO
1	1	Allison, Master. Harold...	male	0.9167	1	2	113781	151.5500	C22 C26	S	Montreal, PQ / Chesterville, ON
1	0	Allison, Miss. Helen...	female	2.0000	1	2	113781	151.5500	C22 C26	S	Montreal, PQ / Chesterville, ON
1	0	Allison, Mr. Hudson...	male	30.0000	1	2	113781	151.5500	C22 C26	S	Montreal, PQ / Chesterville, ON
1	0	Allison, Mrs. Hudson...	female	25.0000	1	2	113781	151.5500	C22 C26	S	Montreal, PQ / Chesterville, ON
1	1	Anderson, Mr. Harry	male	48.0000	0	0	19952	26.5500	E12	S	New York, NY
1	1	Andrews, Miss. Katerin...	female	63.0000	1	0	13502	77.9583	D7	S	Hudson, NY
1	0	Andrews, Mr. Thomas...	male	39.0000	0	0	112050	0.0000	A36	S	Belfast, NI
1	1	Appleton, Mrs. Edward...	female	53.0000	2	0	11769	51.4792	C101	S	Bayside, Queens, NY

THE FUNCTION `SCAN` TO IMPORT DATA

- `scan` has an easy way to distinguish comments from data

?scan

EXAMPLE DATASET

```
cat("TITLE extra line", "# a comment", "2 3 5 7", "11 13 17",  
    file = "../data/ex.data", sep = "\n")
```

IMPORT DATA AND SKIP THE FIRST LINE

```
pp<-scan("../data/ex.data",skip=1,quiet=TRUE)
```

```
pp <- scan("../data/ex.data",comment.char="#", skip = 1,  
           quiet = TRUE)
```

THE DOWNLOAD THE DATA FROM UCI.

```
path1 <- "http://archive.ics.uci.edu/ml/"
path2 <- "machine-learning-databases/00243/"
dname <- 'yacht_hydrodynamics.data'

url<- paste0(path1,path2,dname)
Yacht_Data <- readr::read_table(file = url)
```


BUILT IN DATASETS

- ▶ A sample dataset is often provided to demonstrate the functionality of a package.
- ▶ These records can be loaded using the data command.

```
data(iris)
```

- ▶ There is also a **RStudio Add-In** that helps to find a built-in dataset.

```
install.packages("datasets.load")
```

HELP PAGE FOR BUILT IN DATASETS

?kyphosis

kyphosis {rpart}

R Documentation

Data on Children who have had Corrective Spinal Surgery

Description

The `kyphosis` data frame has 81 rows and 4 columns. representing data on children who have had corrective spinal surgery

Usage

```
kyphosis
```

Format

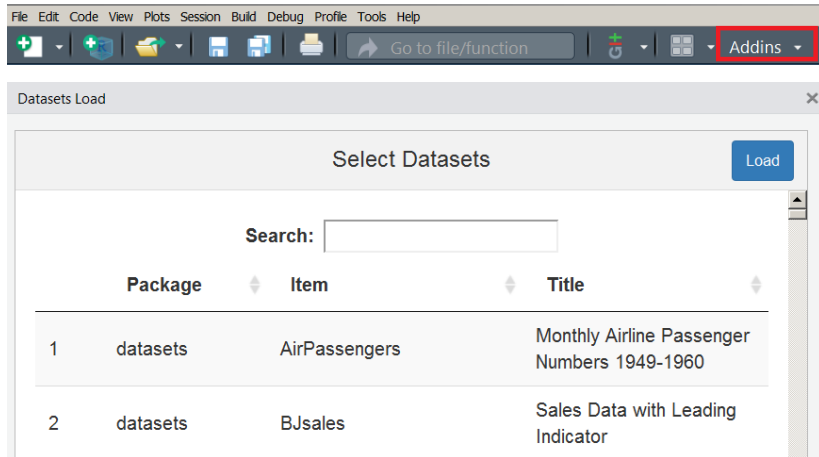
This data frame contains the following columns:

`Kyphosis`

a factor with levels `absent present` indicating if a kyphosis (a type of deformation) was present after the operation.

EXCURSUS RSTUDIO ADDINS

- In the upper right corner there is a button Addins

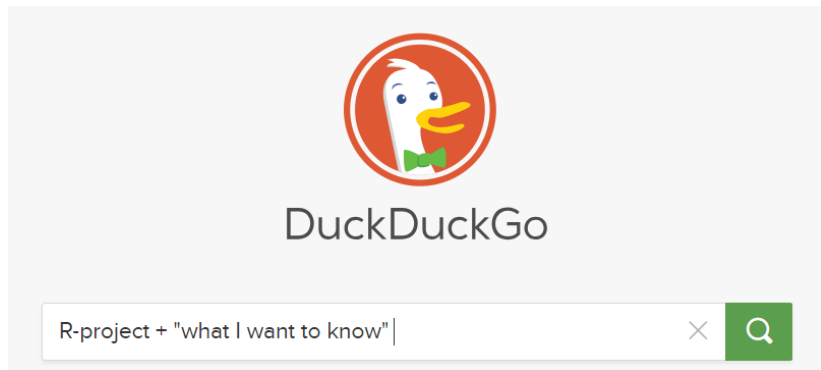


HOW TO GET HELP

- ▶ I use **duckduckgo**:

R-project + "what I want to know"

- ▶ this works of course for all search engines!



EXERCISE: LOAD BUILT-IN DATA

LOAD THE THE BUILT-IN DATASET **SWISS**

- 1) How many observations and variables are available?
- 2) What is the scale level of the variables?

INTERACTIVE DATA TABLE

- 3) Create an interactive data table

THE R-PACKAGE DATA.TABLE

GET AN OVERVIEW

```
data(airquality)
```

```
head(airquality)
```

##	Ozone	Solar.R	Wind	Temp	Month	Day
## 1	41	190	7.4	67	5	1
## 2	36	118	8.0	72	5	2
## 3	12	149	12.6	74	5	3
## 4	18	313	11.5	62	5	4
## 5	NA	NA	14.3	56	5	5
## 6	28	NA	14.9	66	5	6

OVERVIEW WITH DATA.TABLE

```
library(data.table)
(airq <- data.table(airquality))
```

##		Ozone	Solar.R	Wind	Temp	Month	Day
##	1:	41	190	7.4	67	5	1
##	2:	36	118	8.0	72	5	2
##	3:	12	149	12.6	74	5	3
##	4:	18	313	11.5	62	5	4
##	5:	NA	NA	14.3	56	5	5
##	---						
##	149:	30	193	6.9	70	9	26
##	150:	NA	145	13.2	77	9	27
##	151:	14	191	14.3	75	9	28
##	152:	18	131	8.0	76	9	29
##	153:	20	223	11.5	68	9	30

COLUMN (AND ROW) NAMES OF AIRQ

```
colnames(airq)
```

```
## [1] "Ozone" "Solar.R" "Wind" "Temp" "Month" "Day"
```

```
rownames(airq)
```

```
## [1] "1" "2" "3" "4" "5" "6" "7" "8" "9" "10" "11"
## [12] "12" "13" "14" "15" "16" "17" "18" "19" "20" "21" "22"
## [23] "23" "24" "25" "26" "27" "28" "29" "30" "31" "32" "33"
## [34] "34" "35" "36" "37" "38" "39" "40" "41" "42" "43" "44"
## [45] "45" "46" "47" "48" "49" "50" "51" "52" "53" "54" "55"
## [56] "56" "57" "58" "59" "60" "61" "62" "63" "64" "65" "66"
## [67] "67" "68" "69" "70" "71" "72" "73" "74" "75" "76" "77"
## [78] "78" "79" "80" "81" "82" "83" "84" "85" "86" "87" "88"
## [89] "89" "90" "91" "92" "93" "94" "95" "96" "97" "98" "99"
## [100] "100" "101" "102" "103" "104" "105" "106" "107" "108" "109" "110"
## [111] "111" "112" "113" "114" "115" "116" "117" "118" "119" "120" "121"
## [122] "122" "123" "124" "125" "126" "127" "128" "129" "130" "131" "132"
## [133] "133" "134" "135" "136" "137" "138" "139" "140" "141" "142" "143"
## [144] "144" "145" "146" "147" "148" "149" "150" "151" "152" "153" "154"
## [155] "155" "156" "157" "158" "159" "160" "161" "162" "163" "164" "165"
## [166] "166" "167" "168" "169" "170" "171" "172" "173" "174" "175" "176"
## [177] "177" "178" "179" "180" "181" "182" "183" "184" "185" "186" "187"
## [188] "188" "189" "190" "191" "192" "193" "194" "195" "196" "197" "198"
## [199] "199" "200" "201" "202" "203" "204" "205" "206" "207" "208" "209"
## [210] "210" "211" "212" "213" "214" "215" "216" "217" "218" "219" "220"
## [221] "221" "222" "223" "224" "225" "226" "227" "228" "229" "230" "231"
## [232] "232" "233" "234" "235" "236" "237" "238" "239" "240" "241" "242"
## [243] "243" "244" "245" "246" "247" "248" "249" "250" "251" "252" "253"
## [254] "254" "255" "256" "257" "258" "259" "260" "261" "262" "263" "264"
## [265] "265" "266" "267" "268" "269" "270" "271" "272" "273" "274" "275"
## [276] "276" "277" "278" "279" "280" "281" "282" "283" "284" "285" "286"
## [287] "287" "288" "289" "290" "291" "292" "293" "294" "295" "296" "297"
## [298] "298" "299" "300" "301" "302" "303" "304" "305" "306" "307" "308"
## [309] "309" "310" "311" "312" "313" "314" "315" "316" "317" "318" "319"
## [320] "320" "321" "322" "323" "324" "325" "326" "327" "328" "329" "330"
## [331] "331" "332" "333" "334" "335" "336" "337" "338" "339" "340" "341"
## [342] "342" "343" "344" "345" "346" "347" "348" "349" "350" "351" "352"
## [353] "353" "354" "355" "356" "357" "358" "359" "360" "361" "362" "363"
## [364] "364" "365" "366" "367" "368" "369" "370" "371" "372" "373" "374"
## [375] "375" "376" "377" "378" "379" "380" "381" "382" "383" "384" "385"
## [386] "386" "387" "388" "389" "390" "391" "392" "393" "394" "395" "396"
## [397] "397" "398" "399" "400" "401" "402" "403" "404" "405" "406" "407"
## [408] "408" "409" "410" "411" "412" "413" "414" "415" "416" "417" "418"
## [419] "419" "420" "421" "422" "423" "424" "425" "426" "427" "428" "429"
## [430] "430" "431" "432" "433" "434" "435" "436" "437" "438" "439" "440"
## [441] "441" "442" "443" "444" "445" "446" "447" "448" "449" "450" "451"
## [452] "452" "453" "454" "455" "456" "457" "458" "459" "460" "461" "462"
## [463] "463" "464" "465" "466" "467" "468" "469" "470" "471" "472" "473"
## [474] "474" "475" "476" "477" "478" "479" "480" "481" "482" "483" "484"
## [485] "485" "486" "487" "488" "489" "490" "491" "492" "493" "494" "495"
## [496] "496" "497" "498" "499" "500" "501" "502" "503" "504" "505" "506"
## [507] "507" "508" "509" "510" "511" "512" "513" "514" "515" "516" "517"
## [518] "518" "519" "520" "521" "522" "523" "524" "525" "526" "527" "528"
## [529] "529" "530" "531" "532" "533" "534" "535" "536" "537" "538" "539"
## [540] "540" "541" "542" "543" "544" "545" "546" "547" "548" "549" "550"
## [551] "551" "552" "553" "554" "555" "556" "557" "558" "559" "560" "561"
## [562] "562" "563" "564" "565" "566" "567" "568" "569" "570" "571" "572"
## [573] "573" "574" "575" "576" "577" "578" "579" "580" "581" "582" "583"
## [584] "584" "585" "586" "587" "588" "589" "590" "591" "592" "593" "594"
## [595] "595" "596" "597" "598" "599" "600" "601" "602" "603" "604" "605"
## [606] "606" "607" "608" "609" "610" "611" "612" "613" "614" "615" "616"
## [617] "617" "618" "619" "620" "621" "622" "623" "624" "625" "626" "627"
## [628] "628" "629" "630" "631" "632" "633" "634" "635" "636" "637" "638"
## [639] "639" "640" "641" "642" "643" "644" "645" "646" "647" "648" "649"
## [650] "650" "651" "652" "653" "654" "655" "656" "657" "658" "659" "660"
## [661] "661" "662" "663" "664" "665" "666" "667" "668" "669" "670" "671"
## [672] "672" "673" "674" "675" "676" "677" "678" "679" "680" "681" "682"
## [683] "683" "684" "685" "686" "687" "688" "689" "690" "691" "692" "693"
## [694] "694" "695" "696" "697" "698" "699" "700" "701" "702" "703" "704"
## [705] "705" "706" "707" "708" "709" "710" "711" "712" "713" "714" "715"
## [716] "716" "717" "718" "719" "720" "721" "722" "723" "724" "725" "726"
## [727] "727" "728" "729" "730" "731" "732" "733" "734" "735" "736" "737"
## [738] "738" "739" "740" "741" "742" "743" "744" "745" "746" "747" "748"
## [749] "749" "750" "751" "752" "753" "754" "755" "756" "757" "758" "759"
## [760] "760" "761" "762" "763" "764" "765" "766" "767" "768" "769" "770"
## [771] "771" "772" "773" "774" "775" "776" "777" "778" "779" "780" "781"
## [782] "782" "783" "784" "785" "786" "787" "788" "789" "790" "791" "792"
## [793] "793" "794" "795" "796" "797" "798" "799" "800" "801" "802" "803"
## [804] "804" "805" "806" "807" "808" "809" "810" "811" "812" "813" "814"
## [815] "815" "816" "817" "818" "819" "820" "821" "822" "823" "824" "825"
## [826] "826" "827" "828" "829" "830" "831" "832" "833" "834" "835" "836"
## [837] "837" "838" "839" "840" "841" "842" "843" "844" "845" "846" "847"
## [848] "848" "849" "850" "851" "852" "853" "854" "855" "856" "857" "858"
## [859] "859" "860" "861" "862" "863" "864" "865" "866" "867" "868" "869"
## [870] "870" "871" "872" "873" "874" "875" "876" "877" "878" "879" "880"
## [881] "881" "882" "883" "884" "885" "886" "887" "888" "889" "890" "891"
## [892] "892" "893" "894" "895" "896" "897" "898" "899" "900" "901" "902"
## [903] "903" "904" "905" "906" "907" "908" "909" "910" "911" "912" "913"
## [914] "914" "915" "916" "917" "918" "919" "920" "921" "922" "923" "924"
## [925] "925" "926" "927" "928" "929" "930" "931" "932" "933" "934" "935"
## [936] "936" "937" "938" "939" "940" "941" "942" "943" "944" "945" "946"
## [947] "947" "948" "949" "950" "951" "952" "953" "954" "955" "956" "957"
## [958] "958" "959" "960" "961" "962" "963" "964" "965" "966" "967" "968"
## [969] "969" "970" "971" "972" "973" "974" "975" "976" "977" "978" "979"
## [980] "980" "981" "982" "983" "984" "985" "986" "987" "988" "989" "990"
## [991] "991" "992" "993" "994" "995" "996" "997" "998" "999" "1000" "1001"
## [1002] "1002" "1003" "1004" "1005" "1006" "1007" "1008" "1009" "1010" "1011" "1012"
## [1013] "1013" "1014" "1015" "1016" "1017" "1018" "1019" "1020" "1021" "1022" "1023"
## [1024] "1024" "1025" "1026" "1027" "1028" "1029" "1030" "1031" "1032" "1033" "1034"
## [1035] "1035" "1036" "1037" "1038" "1039" "1040" "1041" "1042" "1043" "1044" "1045"
## [1046] "1046" "1047" "1048" "1049" "1050" "1051" "1052" "1053" "1054" "1055" "1056"
## [1057] "1057" "1058" "1059" "1060" "1061" "1062" "1063" "1064" "1065" "1066" "1067"
## [1068] "1068" "1069" "1070" "1071" "1072" "1073" "1074" "1075" "1076" "1077" "1078"
## [1079] "1079" "1080" "1081" "1082" "1083" "1084" "1085" "1086" "1087" "1088" "1089"
## [1090] "1090" "1091" "1092" "1093" "1094" "1095" "1096" "1097" "1098" "1099" "1100"
## [1101] "1101" "1102" "1103" "1104" "1105" "1106" "1107" "1108" "1109" "1110" "1111"
## [1112] "1112" "1113" "1114" "1115" "1116" "1117" "1118" "1119" "1120" "1121" "1122"
## [1123] "1123" "1124" "1125" "1126" "1127" "1128" "1129" "1130" "1131" "1132" "1133"
## [1134] "1134" "1135" "1136" "1137" "1138" "1139" "1140" "1141" "1142" "1143" "1144"
## [1145] "1145" "1146" "1147" "1148" "1149" "1150" "1151" "1152" "1153" "1154" "1155"
## [1156] "1156" "1157" "1158" "1159" "1160" "1161" "1162" "1163" "1164" "1165" "1166"
## [1167] "1167" "1168" "1169" "1170" "1171" "1172" "1173" "1174" "1175" "1176" "1177"
## [1178] "1178" "1179" "1180" "1181" "1182" "1183" "1184" "1185" "1186" "1187" "1188"
## [1189] "1189" "1190" "1191" "1192" "1193" "1194" "1195" "1196" "1197" "1198" "1199"
## [1200] "1200" "1201" "1202" "1203" "1204" "1205" "1206" "1207" "1208" "1209" "1210"
## [1211] "1211" "1212" "1213" "1214" "1215" "1216" "1217" "1218" "1219" "1220" "1221"
## [1222] "1222" "1223" "1224" "1225" "1226" "1227" "1228" "1229" "1230" "1231" "1232"
## [1233] "1233" "1234" "1235" "1236" "1237" "1238" "1239" "1240" "1241" "1242" "1243"
## [1244] "1244" "1245" "1246" "1247" "1248" "1249" "1250" "1251" "1252" "1253" "1254"
## [1255] "1255" "1256" "1257" "1258" "1259" "1260" "1261" "1262" "1263" "1264" "1265"
## [1266] "1266" "1267" "1268" "1269" "1270" "1271" "1272" "1273" "1274" "1275" "1276"
## [1277] "1277" "1278" "1279" "1280" "1281" "1282" "1283" "1284" "1285" "1286" "1287"
## [1288] "1288" "1289" "1290" "1291" "1292" "1293" "1294" "1295" "1296" "1297" "1298"
## [1299] "1299" "1300" "1301" "1302" "1303" "1304" "1305" "1306" "1307" "1308" "1309"
## [1310] "1310" "1311" "1312" "1313" "1314" "1315" "1316" "1317" "1318" "1319" "1320"
## [1321] "1321" "1322" "1323" "1324" "1325" "1326" "1327" "1328" "1329" "1330" "1331"
## [1332] "1332" "1333" "1334" "1335" "1336" "1337" "1338" "1339" "1340" "1341" "1342"
## [1343] "1343" "1344" "1345" "1346" "1347" "1348" "1349" "1350" "1351" "1352" "1353"
## [1354] "1354" "1355" "1356" "1357" "1358" "1359" "1360" "1361" "1362" "1363" "1364"
## [1365] "1365" "1366" "1367" "1368" "1369" "1370" "1371" "1372" "1373" "1374" "1375"
## [1376] "1376" "1377" "1378" "1379" "1380" "1381" "1382" "1383" "1384" "1385" "1386"
## [1387] "1387" "1388" "1389" "1390" "1391" "1392" "1393" "1394" "1395" "1396" "1397"
## [1398] "1398" "1399" "1400" "1401" "1402" "1403" "1404" "1405" "1406" "1407" "1408"
## [1409] "1409" "1410" "1411" "1412" "1413" "1414" "1415" "1416" "1417" "1418" "1419"
## [1420] "1420" "1421" "1422" "1423" "1424" "1425" "1426" "1427" "1428" "1429" "1430"
## [1431] "1431" "1432" "1433" "1434" "1435" "1436" "1437" "1438" "1439" "1440" "1441"
## [1442] "1442" "1443" "1444" "1445" "1446" "1447" "1448" "1449" "1450" "1451" "1452"
## [1453] "1453" "1454" "1455" "1456" "1457" "1458" "1459" "1460" "1461" "1462" "1463"
## [1464] "1464" "1465" "1466" "1467" "1468" "1469" "1470" "1471" "1472" "1473" "1474"
## [1475] "1475" "1476" "1477" "1478" "1479" "1480" "1481" "1482" "1483" "1484" "1485"
## [1486] "1486" "1487" "1488" "1489" "1490" "1491" "1492" "1493" "1494" "1495" "1496"
## [1497] "1497" "1498" "1499" "1500" "1501" "1502" "1503" "1504" "1505" "1506" "1507"
## [1508] "1508" "1509" "1510" "1511" "1512" "1513" "1514" "1515" "1516" "1517" "1518"
## [1519] "1519" "1520" "1521" "1522" "1523" "1524" "1525" "1526" "1527" "1528" "1529"
## [1530] "1530" "1531" "1532" "1533" "1534" "1535" "1536" "1537" "1538" "1539" "1540"
## [1541] "1541" "1542" "1543" "1544" "1545" "1546" "1547" "1548" "1549" "1550" "1551"
## [1552] "1552" "1553" "1554" "1555" "1556" "1557" "1558" "1559" "1560" "1561" "1562"
## [1563] "1563" "1564" "1565" "1566" "1567" "1568" "1569" "1570" "1571" "1572" "1573"
## [1574] "1574" "1575" "1576" "1577" "1578" "1579" "1580" "1581" "1582" "1583" "1584"
## [1585] "1585" "1586" "1587" "1588" "1589" "1590" "1591" "1592" "1593" "1594" "1595"
## [1596] "1596" "1597" "1598" "1599" "1600" "1601" "1602" "1603" "1604" "1605" "1606"
## [1607] "1607" "1608" "1609" "1610" "1611" "1612" "1613" "1614" "1615" "1616" "1617"
## [1618] "1618" "1619" "1620" "1621" "1622" "1623" "1624" "1625" "1626" "1627" "1628"
## [1629] "1629" "1630" "1631" "1632" "1633" "1634" "1635" "1636" "1637" "1638" "1639"
## [1640] "1640" "1641" "1642" "1643" "1644" "1645" "1646" "1647" "1648" "1649" "1650"
## [1651] "1651" "1652" "1653" "1654" "1655" "1656" "1657" "1658" "1659" "1660" "1661"
## [1662] "1662" "1663" "1664" "1665" "1666" "1667" "1668" "1669" "1670" "1671" "1672"
## [1673] "1673" "1674" "1675" "1676" "1677" "1678" "1679" "1680" "1681" "1682" "1683"
## [1684] "1684" "1685" "1686" "1687" "1688" "1689" "1690" "1691" "1692" "1693" "1694"
## [1695] "1695" "1696" "1697" "1698" "1699" "1700" "1701" "1702" "1703" "1704" "1705"
## [1706] "1706" "1707" "1708" "1709" "1710" "1711" "1712" "1713" "1714" "1715" "1716"
## [1717] "1717" "1718" "1719" "1720" "1721" "1722" "1723" "1724" "1725" "1726" "1727"
## [1728] "1728" "1729" "1730" "1731" "1732" "1733" "1734" "1735" "1736" "1737" "1738"
## [1739] "1739" "1740" "1741" "1742" "1743" "1744" "1745" "1746" "1747" "1748" "1749"
## [1750] "1750" "1751" "1752" "1753" "1754" "1755" "1756" "1757" "1758" "1759" "1760"
## [1761] "1761" "1762" "1763" "1764" "1765" "1766" "1767" "1768" "1769" "1770" "1771"
## [1772] "1772" "1773" "1774" "1775" "1776" "1777" "1778" "1779" "1780" "1781" "1782"
## [1783] "1783" "1784" "1785" "1786" "1787" "1788" "1789" "1790" "1791" "1792" "1793"
## [1794] "1794" "1795" "1796" "1797" "1798" "1799" "1800" "1801" "1802" "1803" "1804"
## [1805] "1805" "1806" "1807" "1808" "1809" "1810" "1811" "1812" "1813" "1814" "1815"
## [1816] "1816" "1817" "1818" "1819" "1820" "1821" "1822" "1823" "1824" "1825" "1826"
## [1827] "1827" "1828" "1829" "1830" "1831" "1832" "1833" "1834" "1835" "1836" "1837"
## [1838] "1838" "1839" "1840" "1841" "1842" "1843" "1844" "1845" "1846" "1847" "1848"
## [1849] "1849" "1850" "1851" "1852" "1853" "1854" "1855" "1856" "1857" "1858" "1859"
## [1860] "1860" "1861" "1862" "1863" "1864" "1865" "1866" "1867" "1868" "1869" "1870"
## [1871] "1871" "1872" "1873" "1874" "1875" "1876" "1877" "1878" "1879" "1880" "1881"
## [1882] "1882" "1883" "1884" "1885" "1886" "1887" "1888" "1889" "1890" "1891" "1892"
## [1893] "1893" "1894" "1895" "1896" "1897" "1898" "1899" "1900" "1901" "1902" "1903"
## [1904] "1904" "1905" "1906" "1907" "1908" "1909" "1910" "1911" "1912" "1913" "1914"
## [1915] "1915" "1916" "1917" "1918" "1919" "1920" "1921" "1922" "1923" "1924" "1925"
## [1926] "1926" "1927" "1928" "1929" "1930" "1931" "1932" "1933" "1934" "1935" "1936"
## [1937] "1937" "1938" "1939" "1940" "1941" "1942" "1943" "1944" "1945" "1946" "1947"
## [1948] "1948" "1949" "1950" "1951" "1952" "1953" "1954" "1955" "1956" "1957" "1958"
## [1959] "1959" "1960" "1961" "1962" "1963" "1964" "1965" "1966" "1967" "1968" "1969"
## [1970] "1970" "1971" "1972" "1973" "1974" "1975" "1976" "1977" "1978" "1979" "1980"
## [1981] "1981" "1982" "1983" "1984" "1985" "1986" "1987" "1988" "1989" "1990" "199
```


EXERCISE: RANDOM NUMBERS

- 1) Draw 8 random numbers from the uniform distribution and save them in a vector `x`
- 2) Compute the logarithm of `x`, return suitably lagged and iterated differences,
- 3) compute the exponential function and round the result

```
## [1] 0.5 2.2 0.8 1.7 1.0 0.4 1.4
```

THE PIPE OPERATOR

```
library(magrittr)

# Perform the same computations on `x` as above
x %>% log() %>%
  diff() %>%
  exp() %>%
  round(1)

## [1] 0.5 2.2 0.8 1.7 1.0 0.4 1.4
```

HOW TO DEAL WITH MISSING VALUES

```
?na.omit
```

```
airq
```

##		Ozone	Solar.R	Wind	Temp	Month	Day
##	1:	41	190	7.4	67	5	1
##	2:	36	118	8.0	72	5	2
##	3:	12	149	12.6	74	5	3
##	4:	18	313	11.5	62	5	4
##	5:	NA	NA	14.3	56	5	5
##	---						
##	149:	30	193	6.9	70	9	26
##	150:	NA	145	13.2	77	9	27
##	151:	14	191	14.3	75	9	28
##	152:	18	131	8.0	76	9	29
##	153:	20	223	11.5	68	9	30

THE COMMAND `na.omit`

```
na.omit(airq)
```

##		Ozone	Solar.R	Wind	Temp	Month	Day
##	1:	41	190	7.4	67	5	1
##	2:	36	118	8.0	72	5	2
##	3:	12	149	12.6	74	5	3
##	4:	18	313	11.5	62	5	4
##	5:	23	299	8.6	65	5	7
##	---						
##	107:	14	20	16.6	63	9	25
##	108:	30	193	6.9	70	9	26
##	109:	14	191	14.3	75	9	28
##	110:	18	131	8.0	76	9	29
##	111:	20	223	11.5	68	9	30

CLEAN THE TITANIC DATA SET

```
clean_titanic <- titanic %>%  
  mutate(pclass=factor(pclass,levels = c(1, 2, 3),  
                        labels=c('Upper','Middle','Lower')),  
         survived = factor(survived,levels = c(0, 1),  
                           labels=c('No', 'Yes')) %>%  
na.omit()
```

MUTATE(PCLASS = FACTOR(...:

- ▶ Add label to the variable pclass.
- ▶ 1 becomes Upper, 2 becomes Middle and 3 becomes lower

FACTOR(SURVIVED,...:

- ▶ Add label to the variable survived.
- ▶ 1 Becomes No and 2 becomes Yes
- ▶ na.omit(): Remove the NA observations

GET AN OVERVIEW OF THE DATA

```
glimpse(clean_titanic)
```

```
## Observations: 1,045
```

```
## Variables: 13
```

```
## $ X      <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13,
```

```
## $ pclass <fct> Upper, Upper, Upper, Upper, Upper, Upper, U
```

```
## $ survived <fct> Yes, Yes, No, No, No, Yes, Yes, No, Yes, No
```

```
## $ name    <fct> "Allen, Miss. Elisabeth Walton", "Allison,
```

```
## $ sex     <fct> female, male, female, male, female, male, f
```

```
## $ age     <dbl> 29.0000, 0.9167, 2.0000, 30.0000, 25.0000,
```

```
## $ sibsp   <int> 0, 1, 1, 1, 1, 0, 1, 0, 2, 0, 1, 1, 0, 0, 0
```

```
## $ parch   <int> 0, 2, 2, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
```

```
## $ ticket  <fct> 24160, 113781, 113781, 113781, 113781, 1995
```

```
## $ fare    <dbl> 211.3375, 151.5500, 151.5500, 151.5500, 151
```

```
## $ cabin   <fct> B5, C22 C26, C22 C26, C22 C26, C22 C26, E12
```

```
## $ embarked <fct> S, S, S, S, S, S, S, S, S, C, C, C, C, S, S
```

```
## $ home.dest <fct> "St Louis, MO", "Montreal, PQ / Chestervill
```

WITH GATHER FROM WIDE TO LONG FORMAT

```
library(dplyr)
library(tidyr)
# From http://stackoverflow.com/questions/1181060
stocks <- tibble(
  time = as.Date('2009-01-01') + 0:9,
  X = rnorm(10, 0, 1),
  Y = rnorm(10, 0, 2),
  Z = rnorm(10, 0, 4)
)

gather(stocks, "stock", "price", -time)

## # A tibble: 30 x 3
##   time      stock price
##   <date>    <chr> <dbl>
## 1 2009-01-01 X      0.0632
## 2 2009-01-02 X     -1.54
## 3 2009-01-03 X      0.381
## 4 2009-01-04 X      1.00
```

THE COMMAN EXPAND.GRID

```
expand.grid(letters[1:4],5:3,LETTERS[1:2])
```

##	Var1	Var2	Var3
## 1	a	5	A
## 2	b	5	A
## 3	c	5	A
## 4	d	5	A
## 5	a	4	A
## 6	b	4	A
## 7	c	4	A
## 8	d	4	A
## 9	a	3	A
## 10	b	3	A
## 11	c	3	A
## 12	d	3	A
## 13	a	5	B
## 14	b	5	B
## 15	c	5	B

EXAMPLE DATA - HOUSING VALUES IN SUBURBS OF BOSTON

```
library(MASS)
bdat <- Boston
```

	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	lstat	medv
1	0.00632	18.0	2.31	0	0.5380	6.575	65.2	4.0900	1	296	15.3	396.90	4.98	24.0
2	0.02731	0.0	7.07	0	0.4690	6.421	78.9	4.9671	2	242	17.8	396.90	9.14	21.6
3	0.02729	0.0	7.07	0	0.4690	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7
4	0.03237	0.0	2.18	0	0.4580	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4
5	0.06905	0.0	2.18	0	0.4580	7.147	54.2	6.0622	3	222	18.7	396.90	5.33	36.2
6	0.02985	0.0	2.18	0	0.4580	6.430	58.7	6.0622	3	222	18.7	394.12	5.21	28.7
7	0.08829	12.5	7.87	0	0.5240	6.012	66.6	5.5605	5	311	15.2	395.60	12.43	22.9
8	0.14455	12.5	7.87	0	0.5240	6.172	96.1	5.9505	5	311	15.2	396.90	19.15	27.1
9	0.21124	12.5	7.87	0	0.5240	5.631	100.0	6.0821	5	311	15.2	386.63	29.93	16.5
10	0.17004	12.5	7.87	0	0.5240	6.004	85.9	6.5921	5	311	15.2	386.71	17.10	18.9
11	0.22489	12.5	7.87	0	0.5240	6.377	94.3	6.3467	5	311	15.2	392.52	20.45	15.0
12	0.11747	12.5	7.87	0	0.5240	6.009	82.9	6.2267	5	311	15.2	396.90	13.27	18.9
13	0.09378	12.5	7.87	0	0.5240	5.889	39.0	5.4509	5	311	15.2	390.50	15.71	21.7
14	0.62976	0.0	8.14	0	0.5380	5.949	61.8	4.7075	4	307	21.0	396.90	8.26	20.4

NORMALIZE YOUR DATA

COMPUTE MAXIMUM AND MINIMUM PER COLUMN

```
maxs <- apply(bdat, 2, max)
mins <- apply(bdat, 2, min)
```

SCALE - SCALING AND CENTERING OF MATRIX-LIKE OBJECTS

```
scaled <- as.data.frame(scale(bdat, center = mins,  
                             scale = maxs - mins))
```

THE SCALED DATA

	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black
1	0.000000e+00	0.180	0.06781525	0	0.31481481	0.5775053	0.64160659	0.26920314	0.00000000	0.208015267	0.2872340	1.0000000
2	2.359225e-04	0.000	0.24230205	0	0.17283951	0.5479977	0.78269825	0.34896198	0.04347826	0.104961832	0.5531915	1.0000000
3	2.356977e-04	0.000	0.24230205	0	0.17283951	0.6943859	0.59938208	0.34896198	0.04347826	0.104961832	0.5531915	0.9897373
4	2.927957e-04	0.000	0.06304985	0	0.15020576	0.6585553	0.44181256	0.44854459	0.08695652	0.066793893	0.6489362	0.9942761
5	7.050701e-04	0.000	0.06304985	0	0.15020576	0.6871048	0.52832132	0.44854459	0.08695652	0.066793893	0.6489362	1.0000000
6	2.644715e-04	0.000	0.06304985	0	0.15020576	0.5497222	0.57466529	0.44854459	0.08695652	0.066793893	0.6489362	0.9929901
7	9.213230e-04	0.125	0.27162757	0	0.28600823	0.4696302	0.65602472	0.40292264	0.17391304	0.236641221	0.2765957	0.9967220
8	1.553672e-03	0.125	0.27162757	0	0.28600823	0.5002874	0.95983522	0.43838718	0.17391304	0.236641221	0.2765957	1.0000000
9	2.303251e-03	0.125	0.27162757	0	0.28600823	0.3966277	1.00000000	0.45035419	0.17391304	0.236641221	0.2765957	0.9741036
10	1.840173e-03	0.125	0.27162757	0	0.28600823	0.4680973	0.85478888	0.49673090	0.17391304	0.236641221	0.2765957	0.9743053
11	2.456674e-03	0.125	0.27162757	0	0.28600823	0.5395670	0.94129763	0.47441552	0.17391304	0.236641221	0.2765957	0.9889556
12	1.249299e-03	0.125	0.27162757	0	0.28600823	0.4690554	0.82389289	0.46350335	0.17391304	0.236641221	0.2765957	1.0000000
13	9.830293e-04	0.125	0.27162757	0	0.28600823	0.4460625	0.37178167	0.39295620	0.17391304	0.236641221	0.2765957	0.9838620

THE COMMAND `SAMPLE`

- ▶ We can use this command to draw a sample.
- ▶ We need the command later to split our dataset into a test and a training dataset.

```
sample(1:10,3,replace=T)
```

```
## [1] 9 9 4
```

```
sample(1:10,3,replace=T)
```

```
## [1] 3 4 1
```

```
nrow <- nrow(bdat)*.2  
ind <- sample(1:nrow(bdat),nrow)  
bdat_test <- bdat[ind,]  
bdat_train <- bdat[-ind,]
```

ALTERNATIVE TO SPLIT DATASET

- ▶ Y - Vector of data labels. If there are only a few labels (as is expected) than relative ratio of data in both subsets will be the same.
- ▶ SplitRatio - Splitting ratio

```
split <- caTools::sample.split(Y = bdat$lstat, SplitRatio = .8)
Train <- bdat[split,]
Test <- bdat[!split,]
nrow(Train);nrow(Test)

## [1] 404

## [1] 102
```

SET A SEED

- ▶ `set.seed` is the recommended way to specify seeds.
- ▶ If we set a seed, we get the same result for random events.
- ▶ This function is mainly required for simulations.

```
set.seed(234)
sample(1:10,3,replace=T)
```

```
## [1] 1 2 2
```

```
set.seed(234)
sample(1:10,3,replace=T)
```

```
## [1] 1 2 2
```

TIME MEASUREMENT

```
start_time <- Sys.time()
ab <- runif(10000000)
end_time <- Sys.time()

end_time - start_time

## Time difference of 0.8230469 secs
```

HOW MANY CORES ARE AVAILABLE

```
library(doParallel)  
detectCores()  
  
## [1] 4
```


MAKE CLUSTER

```
cl <- makeCluster(detectCores())
registerDoParallel(cl)

start_time <- Sys.time()
ab <- runif(10000000)
end_time <- Sys.time()

end_time - start_time
## Time difference of 0.5770328 secs

stopCluster(cl)

?parallel::makeCluster
```

RESOURCES

- ▶ **Course materials for the Data Science Specialization**
- ▶ Data wrangling - `dplyr` **vignette** -
- ▶ The usage of pipes - `magrittr` **vignette**
- ▶ Gareth James et al (2013) **An Introduction to Statistical Learning**