

M1-FA1: Fundamentals and Concepts

I. Installing R and RStudio

Using the provided R Programming Tutorial installation guide (R: The R Project for Statistical Computing, n.d.), programs R and RStudio were installed by following the steps below. The first programming tool utilized in the guide was R, whose executable file can be downloaded from [here](#). Clicking the hyperlink should redirect the user to the website seen in the figure below.



As highlighted in the figure, the *download R* hyperlink was clicked to reach the Comprehensive R Archive Network (CRAN) Mirrors page, where the link below the *0-Cloud* option was selected. (The *0-Cloud* mirror provides automatic redirection to servers around the world and is currently sponsored by RStudio.)

CRAN Mirrors	
The Comprehensive R Archive Network is available at the following URLs, please choose a location close to you. Some statistics on the status of the mirrors can be found here: main page , windows release , windows old release .	
If you want to host a new mirror at your institution, please have a look at the CRAN Mirror HOWTO .	
0-Cloud	Automatic redirection to servers worldwide, currently sponsored by Rstudio
Argentina	Universidad Nacional de La Plata
Australia	CSIRO
	AARNET
	School of Mathematics and Statistics, University of Melbourne
	Curtin University
Austria	

After clicking the link, a new page should appear, where different versions of R could be downloaded according to three operating systems: Linux, MacOS, and Windows. Since the operating systems utilized by the members of this group is Windows, the option to download the appropriate operating system was selected.

The Comprehensive R Archive Network

Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- [Download R for Linux \(Debian, Fedora/Redhat, Ubuntu\)](#)
- [Download R for macOS](#)
- [Download R for Windows](#)

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- The latest release (2022-10-31, Innocent and Trusting) [R-4.2.2.tar.gz](#), read [what's new](#) in the latest version.
- Sources of [R alpha and beta releases](#) (daily snapshots, created only in time periods before a planned release).
- Daily snapshots of current patched and development versions are [available here](#). Please read about [new features and bug fixes](#) before filing corresponding feature requests or bug reports.
- Source code of older versions of R is [available here](#).
- Contributed extension [packages](#)

Questions About R

- If you have questions about R like how to download and install the software, or what the license terms are, please read our [answers to frequently asked questions](#) before you send an email.

Next, under *Subdirectories*, the *base* hyperlink was selected because R was being installed into local machines that have not had R installed into them yet.

R for Windows

Subdirectories:

base	Binaries for base distribution. This is what you want to install R for the first time .
contrib	Binaries of contributed CRAN packages (for R >= 3.4.x).
old contrib	Binaries of contributed CRAN packages for outdated versions of R (for R < 3.4.x).
Rtools	Tools to build R and R packages. This is what you want to build your own packages on Windows, or to build R itself.

Please do not submit binaries to CRAN. Package developers might want to contact Uwe Ligges directly in case of questions / suggestions related to Windows binaries.

You may also want to read the [R FAQ](#) and [R for Windows FAQ](#).

Note: CRAN does some checks on these binaries for viruses, but cannot give guarantees. Use the normal precautions with downloaded executables.

Then, to download the executable file that will install R, the highlighted hyperlink was clicked, and the installer was run.

R-4.2.2 for Windows


[Download R-4.2.2 for Windows](#) (76 megabytes, 64 bit)

[README on the Windows binary distribution](#)

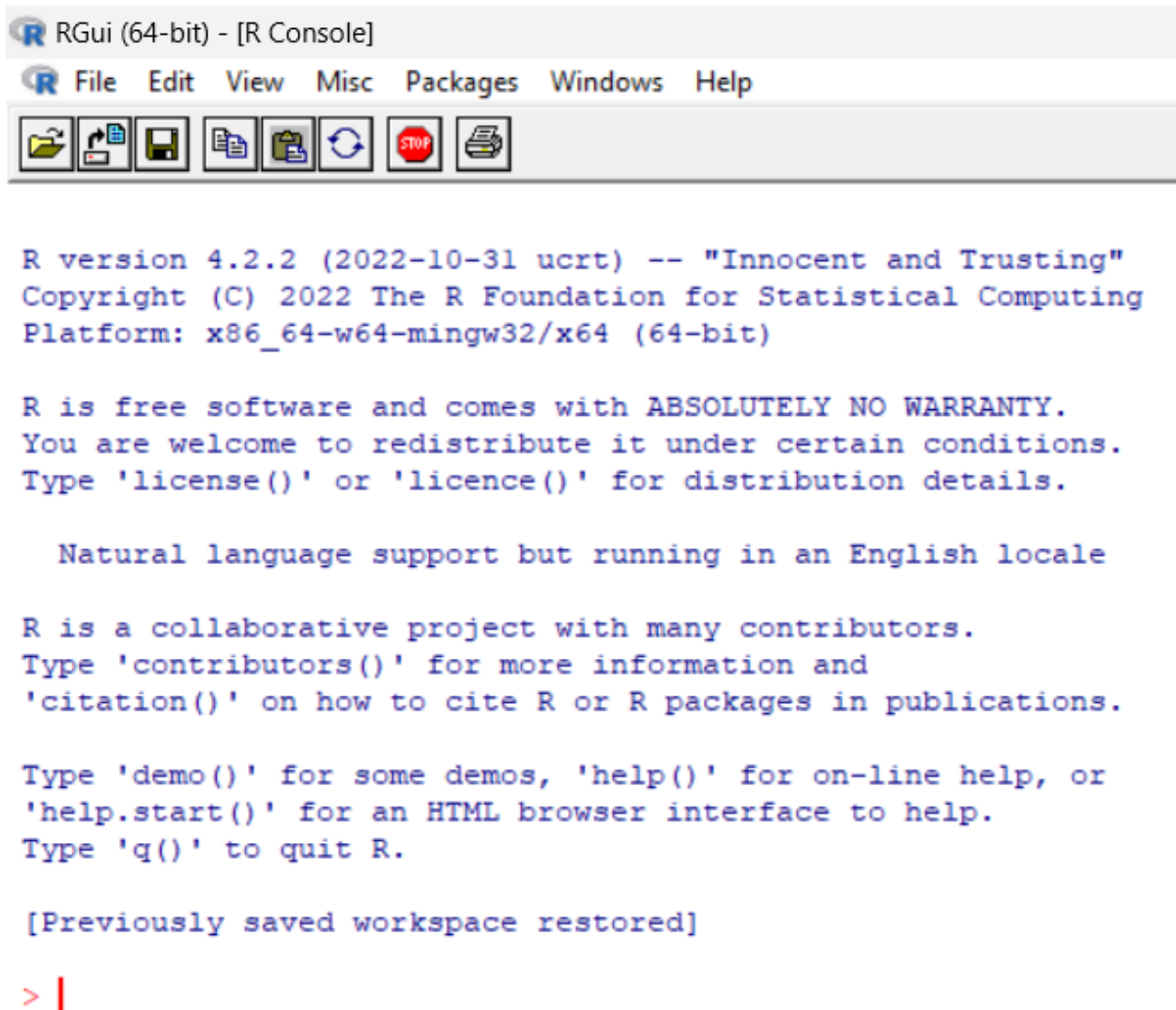
[New features in this version](#)

This build requires UCRT, which is part of Windows since Windows 10 and Windows Server 2016. On older systems, UCRT has to be installed manually from [here](#).

If you want to double-check that the package you have downloaded matches the package distributed by CRAN, you can compare the [md5sum](#) of the .exe to the [fingerprint](#) on the master server.

 R-4.2.2-win	28/02/2023 11:34 am	Application	77,674 KB
---	---------------------	-------------	-----------

R was opened by double-clicking the shortcut that was made upon the application's installation. Selecting the icon should open the R Console, as seen in the figure below.



Afterwards, the members of this group installed the second programming tool, namely RStudio, from this [link](#). Clicking the link should redirect the user to the website, as seen in the figure below. Click **DOWNLOAD RSTUDIO**.



Clicking the button should redirect the user to the downloads page, where they will be given the option to choose between downloading *RStudio Desktop* or *RStudio Server*. For this process, the *RStudio Desktop* was selected by clicking the highlighted button in the figure below.

RStudio Desktop

Find out more about RStudio Desktop and RStudio Desktop Pro below.

DOWNLOAD RSTUDIO



Click **DOWNLOAD RSTUDIO DESKTOP FOR WINDOWS** to download the executable file to install RStudio Desktop for Windows.



PRODUCTS ▾

SOLUTIONS ▾

LEARN & SUPPORT ▾

EXPLORE MORE ▾

PRICING



RStudio Desktop

Used by millions of people weekly, the RStudio integrated development environment (IDE) is a set of tools built to help you be more productive with R and Python.

1: Install R

RStudio requires R 3.3.0+. Choose a version of R that matches your computer's operating system.

2: Install RStudio

DOWNLOAD RSTUDIO DESKTOP FOR WINDOWS

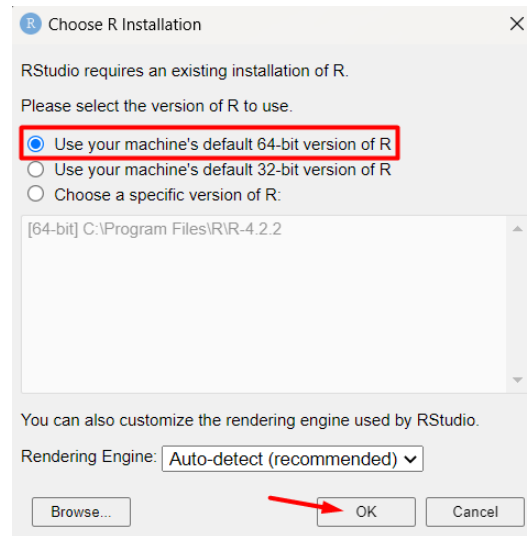
RStudio-2022.12.0-353

28/02/2023 12:04 pm

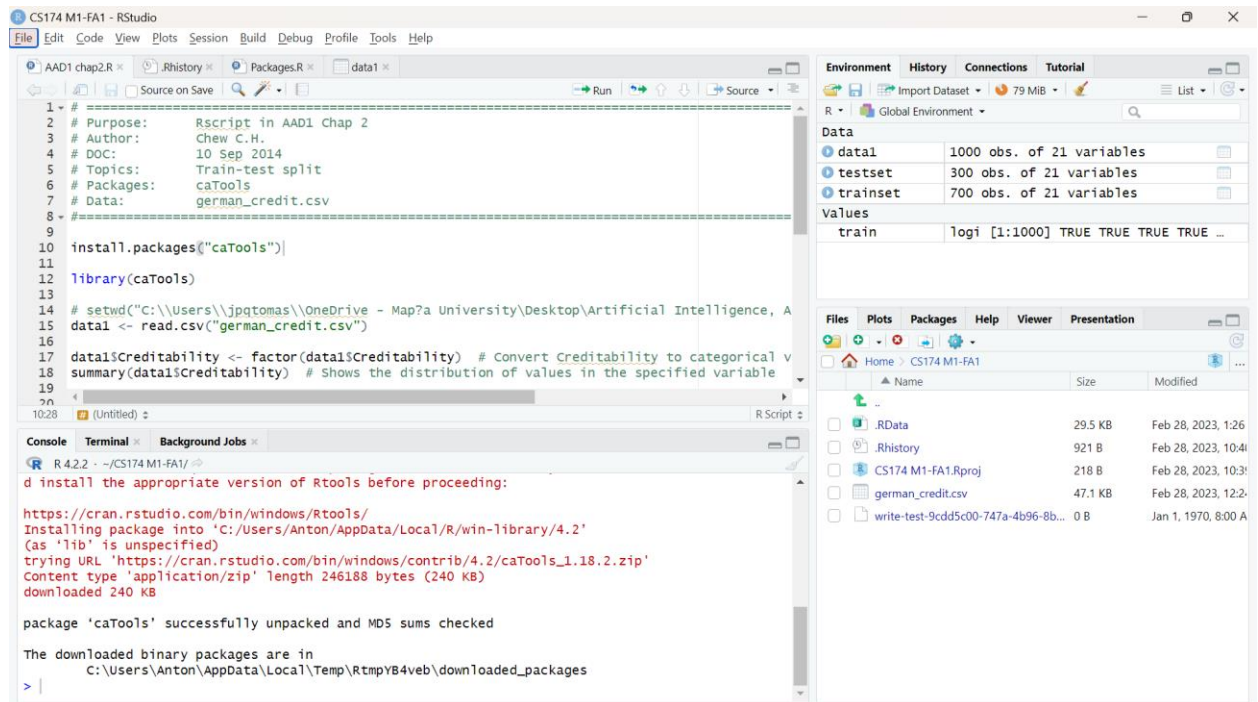
Application

198,016 KB

Next, open RStudio by selecting the rstudio application inside the RStudio folder in the C: drive. The following pop-up window should appear. Select the default option, then click OK.



Since RStudio had already been used by the data scientists, opening RStudio resulted in the figure below. Nevertheless, the same contents and windowpanes should still appear upon initial startup.



II. Running the R Script

Before running the provided R script, the following line of code was added prior to the first line to install the required library package that contains basic utility functions like window statistic functions, LogitBoost classifier, and base64 encoder/decoder, etc. (Jarek Tuszynski jaroslaw.w.tuszynski@saic.com, n.d.)

```
install.packages("caTools")
```

```
> install.packages("caTools")
WARNING: Rtools is required to build R packages but is not currently installed. Please download and install the appropriate version of Rtools before proceeding:
https://cran.rstudio.com/bin/windows/Rtools/
Installing package into 'C:/Users/Anton/AppData/Local/R/win-library/4.2'
(as 'lib' is unspecified)
also installing the dependency 'bitops'

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.2/bitops_1.0-7.zip'
Content type 'application/zip' length 31679 bytes (30 KB)
downloaded 30 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.2/caTools_1.18.2.zip'
Content type 'application/zip' length 246188 bytes (240 KB)
downloaded 240 KB

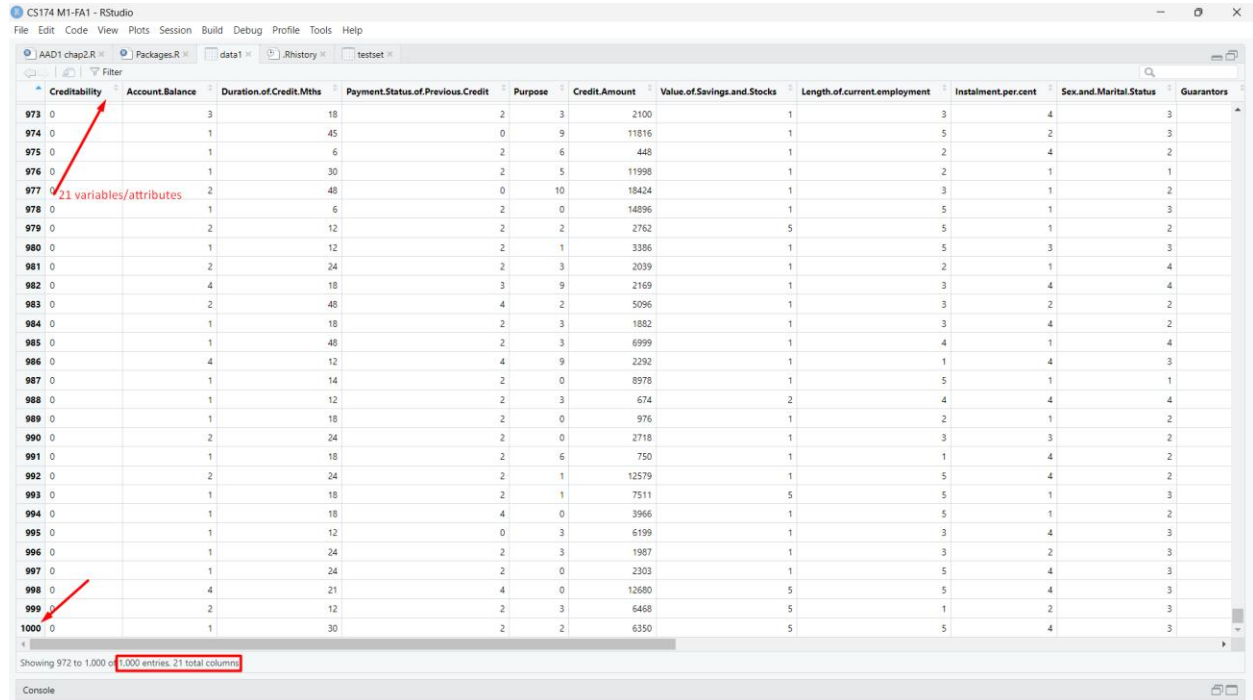
package 'bitops' successfully unpacked and MD5 sums checked
package 'caTools' successfully unpacked and MD5 sums checked
```

After the caTools package has been installed, the rest of the code was run without errors, as seen in the code snippet below.

```
> library(caTools)
> data1 <- read.csv("german_credit.csv")
> data1$Creditability <- factor(data1$Creditability) # Convert Creditability to categorical variable
> summary(data1$Creditability) # Shows the distribution of values in the specified variable
 0      1
300  700
> prop.table(table(data1$Creditability)) # Shows the proportion of values in the specified variable.
 0      1
0.3 0.7
> # set the random number sequence for random split.
> # Reproducible in future with the same number that you choose.
> set.seed(2014)
> # Stratify on Y and randomly split data into train vs test set based on split ratio
> train <- sample.split(Y = data1$Creditability, SplitRatio = 0.7)
> # Get training and test data
> trainset <- subset(data1, train == T)
> testset <- subset(data1, train == F)
> # Check that the proportion of Y is the same in Trainset and Testset
> prop.table(table(trainset$Creditability))
 0      1
0.3 0.7
> prop.table(table(testset$Creditability))
 0      1
0.3 0.7
```


III. Determining the Number of Observable Data

The total number of observable data from the dataset is **twenty-one thousand entries**, with one thousand rows per column. (The dataset contains twenty-one attributes, each of which are listed per column.) A more comprehensive understanding of this description can be found below.



CS174 M1-FA1 - RStudio

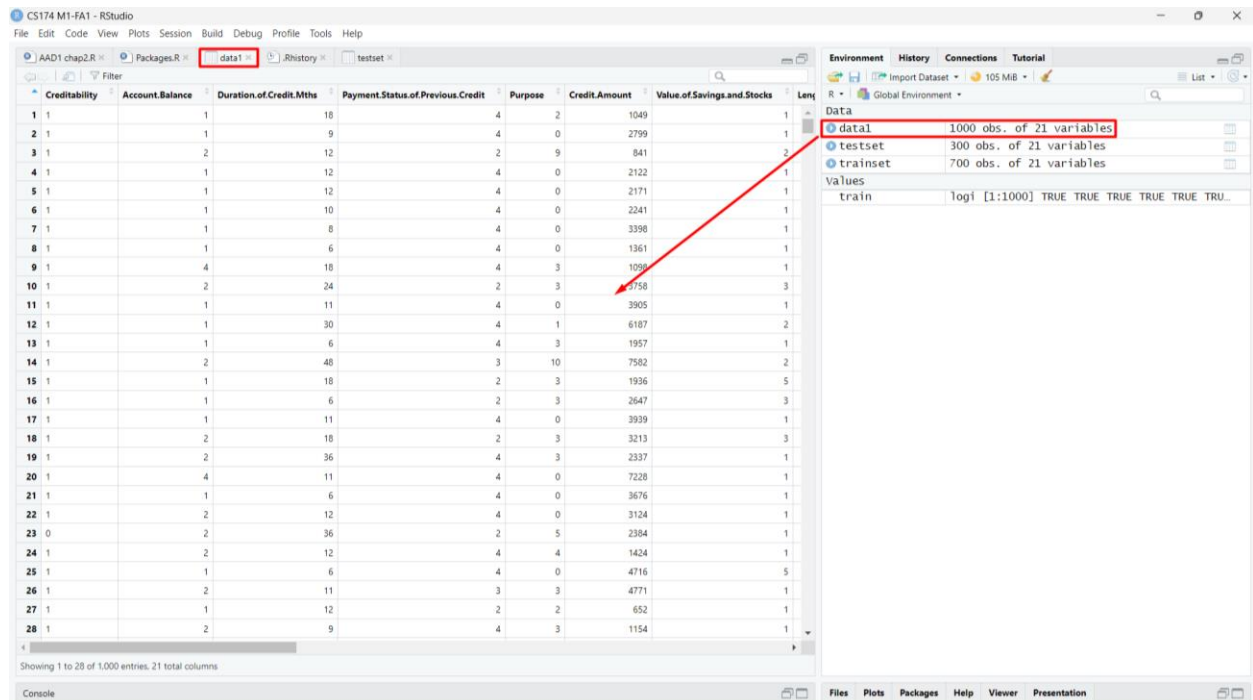
File Edit Code View Plots Session Build Debug Profile Tools Help

AAD1 chap2.R Packages.R data1 Rhistory testset

	Creditability	Account.Balance	Duration.of.Credit.Mths	Payment.Status.of.Previous.Credit	Purpose	Credit.Amount	Value.of.Savings.and.Stocks	Length.of.current.employment	Instalment.per.cent	Sex.and.Marital.Status	Guarantors
973	0		3	18	2	3	2100	1	3	4	3
974	0		1	45	0	9	11816	1	5	2	3
975	0		1	6	2	6	448	1	2	4	2
976	0		1	30	2	5	11998	1	2	1	1
977	0		2	48	0	10	18424	1	3	1	2
978	0		1	6	2	0	14896	1	5	1	3
979	0		2	12	2	2	2762	5	5	1	2
980	0		1	12	2	1	3386	1	5	3	3
981	0		2	24	2	3	2039	1	2	1	4
982	0		4	18	3	9	2169	1	3	4	4
983	0		2	48	4	2	5096	1	3	2	2
984	0		1	18	2	3	1882	1	3	4	2
985	0		1	48	2	3	6999	1	4	1	4
986	0		4	12	4	9	2292	1	1	4	3
987	0		1	14	2	0	8978	1	5	1	1
988	0		1	12	2	3	674	2	4	4	4
989	0		1	18	2	0	976	1	2	1	2
990	0		2	24	2	0	2718	1	3	3	2
991	0		1	18	2	6	750	1	1	4	2
992	0		2	24	2	1	12579	1	5	4	2
993	0		1	18	2	1	7511	5	5	1	3
994	0		1	18	4	0	3966	1	5	1	2
995	0		1	12	0	3	6199	1	3	4	3
996	0		1	24	2	3	1987	1	3	2	3
997	0		1	24	2	0	2303	1	5	4	3
998	0		4	21	4	0	12680	5	5	4	3
999	0		2	12	2	3	6468	5	1	2	3
1000	0		1	30	2	2	6350	5	5	4	3

Showing 972 to 1,000 of 1,000 entries, 21 total columns

Console



CS174 M1-FA1 - RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

AAD1 chap2.R Packages.R data1 Rhistory testset

	Creditability	Account.Balance	Duration.of.Credit.Mths	Payment.Status.of.Previous.Credit	Purpose	Credit.Amount	Value.of.Savings.and.Stocks	Length.of.current.employment	Instalment.per.cent	Sex.and.Marital.Status	Guarantors
1	1		1	18	4	2	1049	1	1		
2	1		1	9	4	0	2799	1			
3	1		2	12	2	9	841	2			
4	1		1	12	4	0	2122	1			
5	1		1	12	4	0	2171	1			
6	1		1	10	4	0	2241	1			
7	1		1	8	4	0	3398	1			
8	1		1	6	4	0	1361	1			
9	1		4	18	4	3	1098	1			
10	1		2	24	2	3	1758	3			
11	1		1	11	4	0	3905	1			
12	1		1	30	4	1	6187	2			
13	1		1	6	4	3	1957	1			
14	1		2	48	3	10	7582	2			
15	1		1	18	2	3	1936	5			
16	1		1	6	2	3	2647	3			
17	1		1	11	4	0	3939	1			
18	1		2	18	2	3	3213	3			
19	1		2	36	4	3	2337	1			
20	1		4	11	4	0	7228	1			
21	1		1	6	4	0	3676	1			
22	1		2	12	4	0	3124	1			
23	0		2	36	2	5	2384	1			
24	1		2	12	4	4	1424	1			
25	1		1	6	4	0	4716	5			
26	1		2	11	3	3	4771	1			
27	1		1	12	2	2	652	1			
28	1		2	9	4	3	1154	1			

Showing 1 to 28 of 1,000 entries, 21 total columns

Console

Environment History Connections Tutorial

R Global Environment

data1 1000 obs. of 21 variables

testset 300 obs. of 21 variables

trainset 700 obs. of 21 variables

Values

train logi [1:1000] TRUE TRUE TRUE TRUE TRUE TRUE...

IV. Determining the Percentage of the Training and Test Set

Before the percentages of the training and testing set were identified, the dataset was first randomly split into training and testing sets based on a split ratio of 7.0. This was accomplished by running the following code snippet:

```
> # Stratify on Y and randomly split data into train vs test set based on  
Split ratio  
> train <- sample.split(Y = data1$Creditability, SplitRatio = 0.7)
```

Note: The line of code above did not yield any output because it merely performed the random splitting of the dataset.

To get the proportions of the training and test sets, the following codes were run. First, Boolean values of True (1) and False (0) were assigned to the trainset and testset, respectively, using the subset() function.

```
> # Get training and test data  
> trainset <- subset(data1, train == T) # T is equivalent to 1  
> testset <- subset(data1, train == F) # F is equivalent to 0
```

According to (Quick-R: Subsetting Data, n.d.), the subset() function is the most efficient way to select variables and observations. Therefore, the lines of code above state that all rows from the dataset with a Boolean value of True or 1 were specified as the trainset and all rows from the dataset with a Boolean value of False or 0 were specified as the testset.

```
> # Check that the proportion of Y is the same in Trainset and Testset  
> prop.table(table(trainset$Creditability))  
  
  0    1  
0.3 0.7  
> prop.table(table(testset$Creditability))  
  
  0    1  
0.3 0.7
```

Then, to check that the split was correctly implemented and that the proportion of Y is the same as the training and testing set, the prop.table() function was run to calculate the value of each table cell as a proportion of all values. (2021) The results reveal that the dataset was divided into proportions of 0.3 and 0.7 for the testing set (Boolean value = 0) and the training set (Boolean value = 1), respectively. Furthermore, the percentages of the training and testing sets can be inferred from the results: **70% of the dataset was proportioned for the training set and 30% of the dataset for the testing set.**

The screenshot below shows a visualization of how the dataset was split into the testset and trainset.

The screenshot shows the RStudio interface. The script editor contains R code for splitting the 'data1' dataset into 'trainset' and 'testset' based on a 0.7 split ratio. The console shows the execution of these commands. The Environment pane on the right displays the objects created: 'data1' (1000 obs. of 21 variables), 'testset' (300 obs. of 21 variables), and 'trainset' (700 obs. of 21 variables). The Files pane shows the project structure.

```
17 data1$Credibility <- factor(data1$Credibility) # Convert Credibility to categorical variable
18 summary(data1$Credibility) # Shows the distribution of values in the specified variable
19
20 prop.table(table(data1$Credibility)) # Shows the proportion of values in the specified variable.
21
22 # set the random number sequence for random split.
23 # Reproducible in future with the same number that you choose.
24 set.seed(2014)
25
26 # Stratify on Y and randomly split data into train vs test set based on Split ratio
27 train <- sample.split(Y = data1$Credibility, SplitRatio = 0.7)
28
29 # Get training and test data
30 trainset <- subset(data1, train == T)
31 testset <- subset(data1, train == F)
32
33 # Check that the proportion of Y is the same in Trainset and Testset
34 prop.table(table(trainset$Credibility))
35
36 prop.table(table(testset$Credibility))
37
38 # ===== END =====
39
40
```

Console output:

```
R 4.2.2 - ~/CS174 M1-FA1/ >
0.3 0.7
> # set the random number sequence for random split.
> # Reproducible in future with the same number that you choose.
> set.seed(2014)
> # Stratify on Y and randomly split data into train vs test set based on Split ratio
> train <- sample.split(Y = data1$Credibility, SplitRatio = 0.7)
> # Get training and test data
> trainset <- subset(data1, train == T)
> testset <- subset(data1, train == F)
> # Check that the proportion of Y is the same in Trainset and Testset
> prop.table(table(trainset$Credibility))
0 1
0.3 0.7
> prop.table(table(testset$Credibility))
0 1
0.3 0.7
```

Environment pane:

Object	Size	Modified
data1	1000 obs. of 21 variables	
testset	300 obs. of 21 variables	
trainset	700 obs. of 21 variables	

Testing and Training Set Proportions

The screenshot shows the RStudio interface. The script editor contains R code for splitting the 'data1' dataset into 'trainset' and 'testset' based on a 0.7 split ratio. The console shows the execution of these commands. The Environment pane on the right displays the objects created: 'data1' (1000 obs. of 21 variables), 'testset' (300 obs. of 21 variables), and 'trainset' (700 obs. of 21 variables). The Files pane shows the project structure.

```
17 data1$Credibility <- factor(data1$Credibility) # Convert Credibility to categorical variable
18 summary(data1$Credibility) # Shows the distribution of values in the specified variable
19
20 prop.table(table(data1$Credibility)) # Shows the proportion of values in the specified variable.
21
22 # set the random number sequence for random split.
23 # Reproducible in future with the same number that you choose.
24 set.seed(2014)
25
26 # Stratify on Y and randomly split data into train vs test set based on Split ratio
27 train <- sample.split(Y = data1$Credibility, SplitRatio = 0.7)
28
29 # Get training and test data
30 trainset <- subset(data1, train == T)
31 testset <- subset(data1, train == F)
32
33 # Check that the proportion of Y is the same in Trainset and Testset
34 prop.table(table(trainset$Credibility))
35
36 prop.table(table(testset$Credibility))
37
38 # ===== END =====
39
40
```

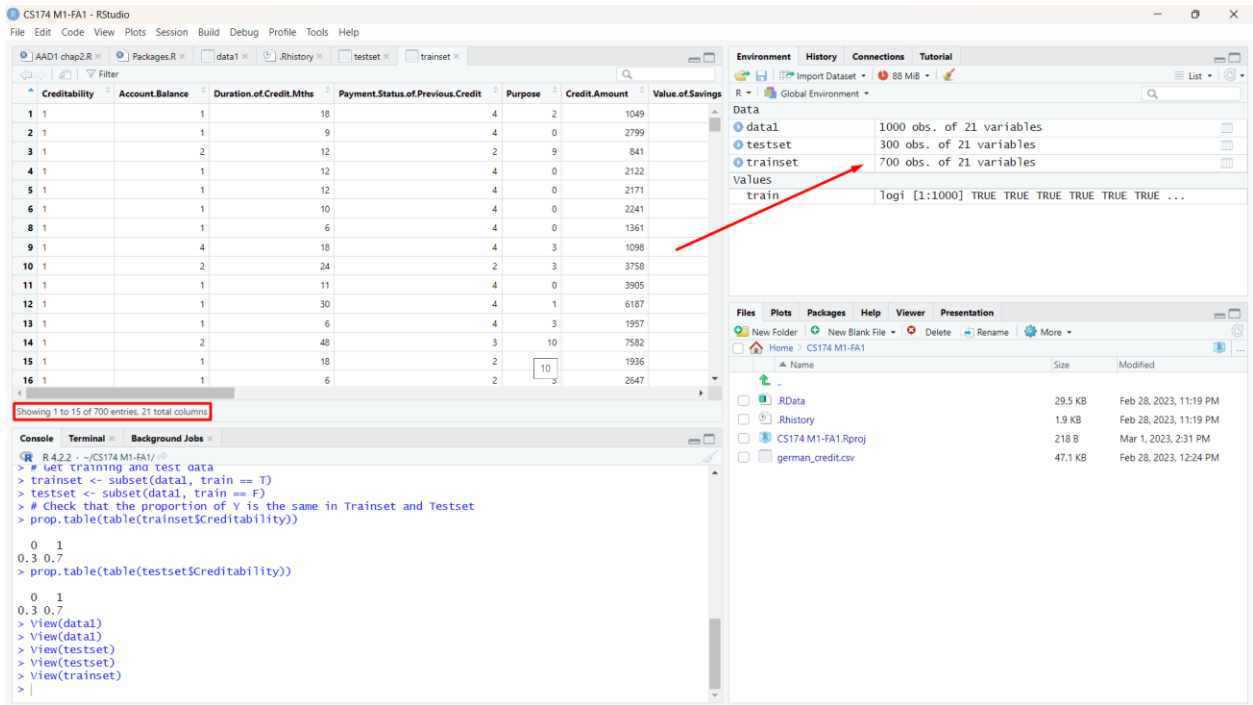
Console output:

```
R 4.2.2 - ~/CS174 M1-FA1/ >
0.3 0.7
> # set the random number sequence for random split.
> # Reproducible in future with the same number that you choose.
> set.seed(2014)
> # Stratify on Y and randomly split data into train vs test set based on Split ratio
> train <- sample.split(Y = data1$Credibility, SplitRatio = 0.7)
> # Get training and test data
> trainset <- subset(data1, train == T)
> testset <- subset(data1, train == F)
> # Check that the proportion of Y is the same in Trainset and Testset
> prop.table(table(trainset$Credibility))
0 1
0.3 0.7
> prop.table(table(testset$Credibility))
0 1
0.3 0.7
> View(data1)
> View(data1)
> View(testset)
> View(testset)
>
```

Environment pane:

Object	Size	Modified
data1	1000 obs. of 21 variables	
testset	300 obs. of 21 variables	
trainset	700 obs. of 21 variables	

Testing Set Proportion (30%)

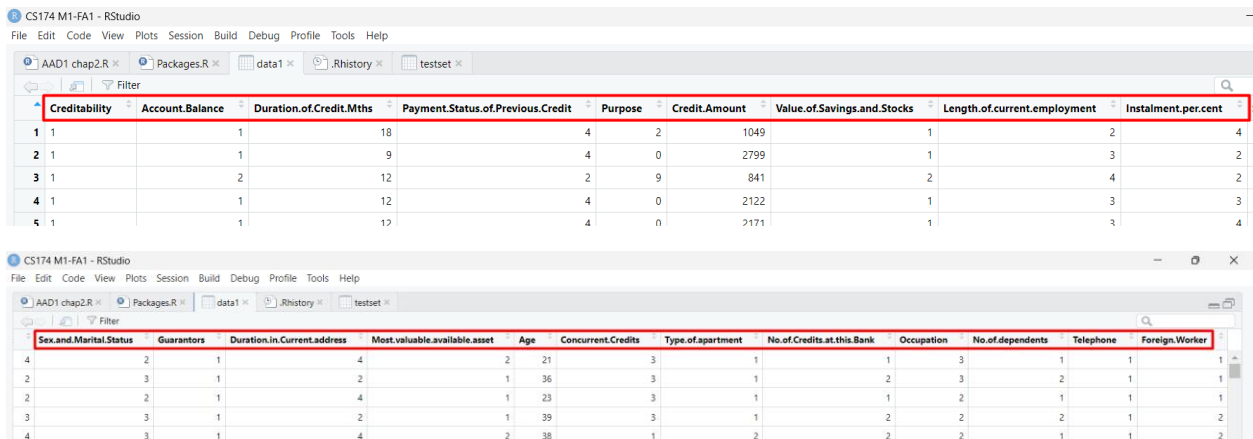


Training Set Proportion (70%)

V. Listing All Variables

The provided dataset (*german_credit.csv*) includes the following variables: *Creditability*, *Account Balance*, *Duration of Credit Mths*, *Payment Status of Previous Credit*, *Purpose*, *Credit Amount*, *Value of Savings and Stocks*, *Length of current employment*, *Instalment per cent*, *Sex and Marital Status*, *Guarantors*, *Duration in Current address*, *Most valuable available asset*, *Age*, *Concurrent Credits*, *Type of apartment*, *No of Credits at this Bank*, *Occupation*, *No of dependents*, *Telephone*, and *Foreign Worker*.

The screenshots below show the dataset variables that were read through RStudio.



VI. Identifying Three Qualitative and Three Quantitative Attributes

Quantitative variables are referred to as “numeric” variables, which are variables that represent a measurable quantity. On the other hand, qualitative variables are sometimes referred to as “categorical” variables, which are those that take on names or labels that fit into categories. (2020) Three quantitative variables from the dataset include *Credit Amount*, *Instalment per cent*, and *Account Balance*. Three qualitative variables from the dataset include *Creditability*, which was converted to a categorical variable in the code, *Purpose*, and *Occupation*.

```
CS174 M1-FA1 - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help

# AAD1 chap2.R
# Packages.R
# data1.R
# Rhistory.R
# testset.R
# trainset.R

# Source on Save
# german_credit.csv

9
10 install.packages("caTools")
11
12 library(caTools)
13
14 # setwd("C:\\Users\\jgptomas\\OneDrive - Map7a University\\Desktop\\Artificial Intelligence, Analytics and Data Science (C")
15 data1 <- read.csv("german_credit.csv")
16
17 data1$Creditability <- factor(data1$Creditability) # Convert Creditability to categorical variable
18 summary(data1$Creditability) # Shows the distribution of values in the specified variable
19
20 prop.table(table(data1$Creditability)) # Shows the proportion of values in the specified variable.
21
22 # set the random number sequence for random split.
23 # Reproducible in future with the same number that you choose.
24 set.seed(2014)
25
26 # Stratify on Y and randomly split data into train vs test set based on Split ratio
27 train <- sample.split(Y = data1$Creditability, SplitRatio = 0.7)
28
29 # Get training and test data
30 trainset <- subset(data1, train == T)
31
3429 (Untitled) R Script
```

```
CS174 M1-FA1 - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help

# AAD1 chap2.R
# Packages.R
# data1.R
# Rhistory.R
# testset.R
# trainset.R

# Source on Save
# german_credit.csv

9
10 install.packages("caTools")
11
12 library(caTools)
13
14 # setwd("C:\\Users\\jgptomas\\OneDrive - Map7a University\\Desktop\\Artificial Intelligence, Analytics and Data Science (C")
15 data1 <- read.csv("german_credit.csv")
16
17 data1$Creditability <- factor(data1$Creditability) # Convert Creditability to categorical variable
18 summary(data1$Creditability) # Shows the distribution of values in the specified variable
19
20 prop.table(table(data1$Creditability)) # Shows the proportion of values in the specified variable.
21
22 # set the random number sequence for random split.
23 # Reproducible in future with the same number that you choose.
24 set.seed(2014)
25
26 # Stratify on Y and randomly split data into train vs test set based on Split ratio
27 train <- sample.split(Y = data1$Creditability, SplitRatio = 0.7)
28
29 # Get training and test data
30 trainset <- subset(data1, train == T)
31
3429 (Untitled) R Script

Environment History Connections Tutorial
R Global Environment
Data
data1 1000 obs. of 21 variables
$ Creditability : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 ...
$ Account.Balance : int 1 1 2 1 1 1 1 1 4 2 ...
$ Duration.of.Credit.Mths : int 18 9 12 12 12 10 8 6 18 24 ...
$ Payment.Status.of.Previous.Credit : int 4 4 2 4 4 4 4 4 2 ...
$ Purpose : int 2 0 9 0 0 0 0 0 3 3 ...
$ Credit.Amount : int 1049 2799 841 2122 2171 2241 3398 1361 1098 3758 ...
$ Value.of.Savings.and.Stocks : int 1 1 2 1 1 1 1 1 3 ...
$ Length.of.current.employment : int 2 3 4 3 3 2 4 2 1 1 ...
$ Instalment.per.cent : int 4 2 2 3 4 1 1 2 4 1 ...
$ Sex.and.Marital.Status : int 2 3 2 3 3 3 3 3 2 2 ...
$ Guarantors : int 1 1 1 1 1 1 1 1 1 1 ...
$ Duration.in.Current.address : int 4 2 4 2 4 3 4 4 4 4 ...
$ Most.valuable.available.asset : int 2 1 1 1 2 1 1 1 3 4 ...
$ Age : int 21 36 23 39 38 48 39 40 65 23 ...
$ Concurrent.Credits : int 3 3 3 3 1 3 3 3 3 3 ...
$ Type.of.apartment : int 1 1 1 1 2 1 2 2 2 1 ...
$ No.of.Credits.at.this.Bank : int 1 2 1 2 2 2 2 1 2 1 ...
$ Occupation : int 3 3 2 2 2 2 2 2 1 1 ...
$ No.of.dependents : int 1 2 1 2 1 2 1 2 1 1 ...
$ Telephone : int 1 1 1 1 1 1 1 1 1 1 ...
$ Foreign.worker : int 1 1 1 2 2 2 2 2 1 1 ...

testset 300 obs. of 21 variables
trainset 700 obs. of 21 variables
Values
train logi [1:1000] TRUE TRUE TRUE TRUE TRUE ...

Console Terminal Background Jobs
R 4.2.2 - ~/CS174 M1-FA1/
0 1
0.3 0.7
> prop.table(table(testset$Creditability))
0 1
0.3 0.7
> View(data1)
> View(data1)
> View(testset)
> View(testset)
> View(trainset)
> View(data1)
> View(data1)
> source("~/30223/CS174/Module 1/M1-FA1/AAD1 chap2.R")
Error in install.packages : Updating loaded packages
>
```

***Credibility* as a categorical (quantitative) variable**

References

- freeCodeCamp.org. (2019, June 6). *R Programming Tutorial - Learn the Basics of Statistical Computing*. YouTube.
<https://www.youtube.com/watch?v=V8eKsto3Ug&feature=youtu.be>
- Jarek Tuszynski jaroslaw.w.tuszynski@saic.com. (n.d.). *caTools-package function - RDocumentation*.
<https://www.rdocumentation.org/packages/caTools/versions/1.17.1/topics/caTools-package>
- Quick-R: Subsetting Data*. (n.d.). <https://www.statmethods.net/management/subset.html>
- R: The R Project for Statistical Computing*. (n.d.). <https://www.r-project.org/>
- Z. (2020, September 18). *Qualitative vs. Quantitative Variables: What's the Difference?* Statology. <https://www.statology.org/qualitative-vs-quantitative-variables/>
- Z. (2021, December 3). *How to Use prop.table() Function in R (With Examples)*. Statology.
<https://www.statology.org/r-prop-table/>