NYU HEALTH SCIENCES LIBRARY

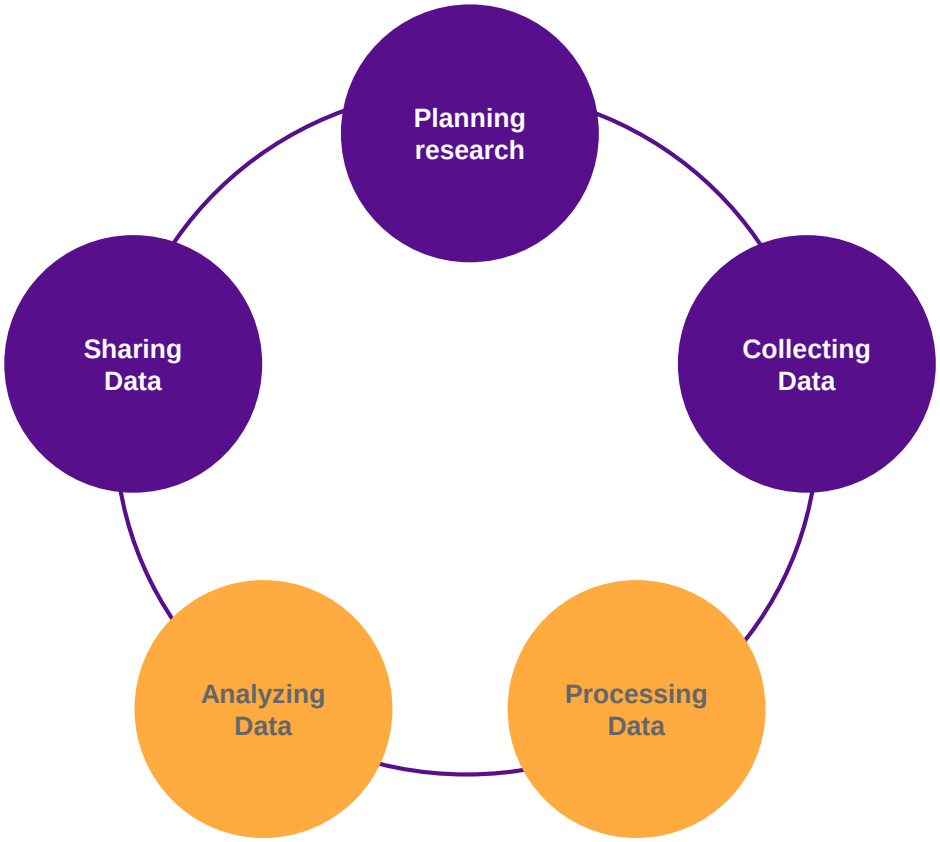
Reproducibility Workshop: Part 2
R Markdown for Reproducible Data Processing and Analysis

Alisa Surkis, PhD, MLS
Assistant Director, Research Data and Metrics
NYU Health Sciences Library

Class Overview

- Why use R?
- Why use R Markdown for Reproducibility?
- Rstudio
- Terminology and Syntax
- Setting up an R Project
- R Basics
- Reproducible Case Study: Data Processing and Analysis

Data Lifecycle

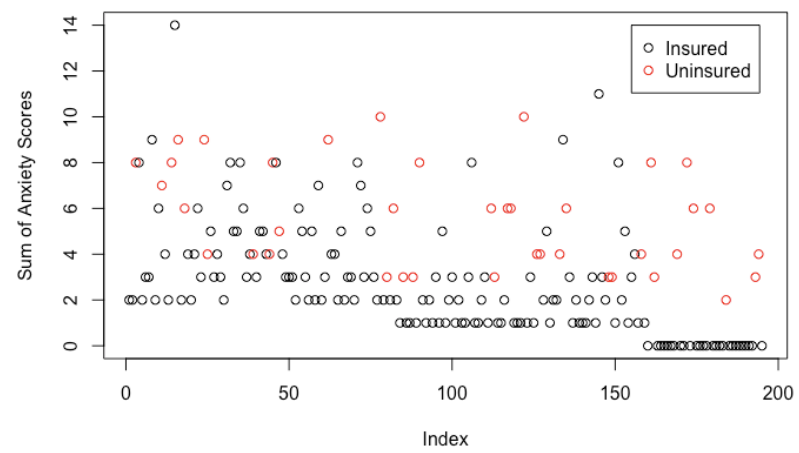


NYU HEALTH SCIENCES LIBRARY

Why use R?

```
notUninsured <- select(filter(AnalysisData, ever_uninsured == 0), anxietySum, difficultyFunctioning)
Uninsured <- select(filter(AnalysisData, ever_uninsured == 1), anxietySum, difficultyFunctioning)
diffAnxiety <- t.test(notUninsured$anxietySum, Uninsured$anxietySum)
diffAnxietyHyp <- ifelse(diffAnxiety$p.value < 0.05, "Rejected","Accepted")
```

```
EHRData$DATE <- mdy(EHRData$DATE)

EHRData <- spread(EHRData, key = description, value=value)

EHRData <- select(EHRData, patient_id="PATIENT", date_of_visit = "DATE", ir
housing_status = "Housing status", HIV_status = "HIV status in {nominal}")
```



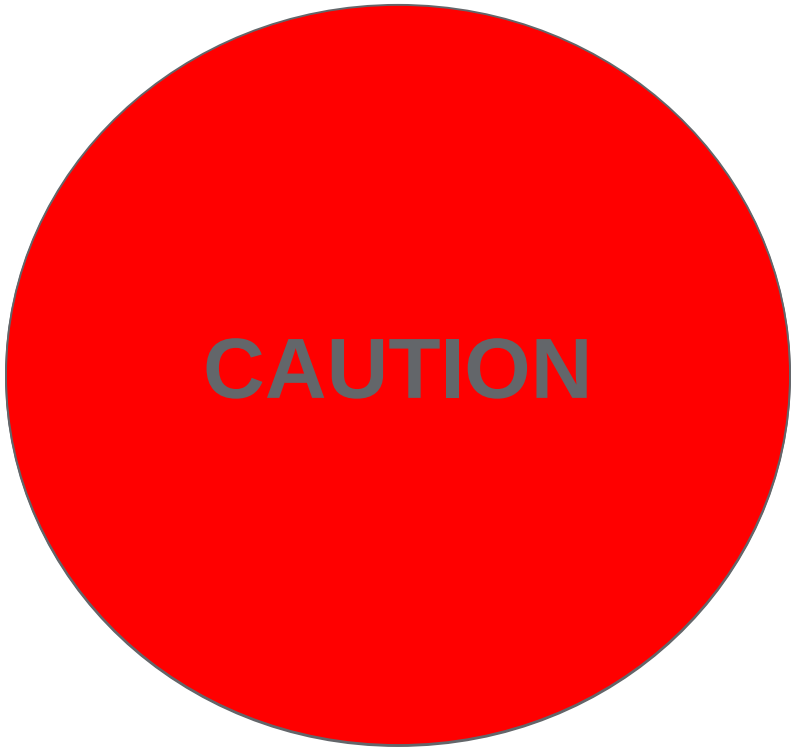
Everything in one places: data processing, analysis, and visualization

NYU HEALTH SCIENCES LIBRARY

Why R?

It’s free!

Large user community contributing functionality



Why R Markdown?

Creates formatted documents that integrate:

- Code -- HOW
- Narrative Text -- WHY
- Output -- WHAT

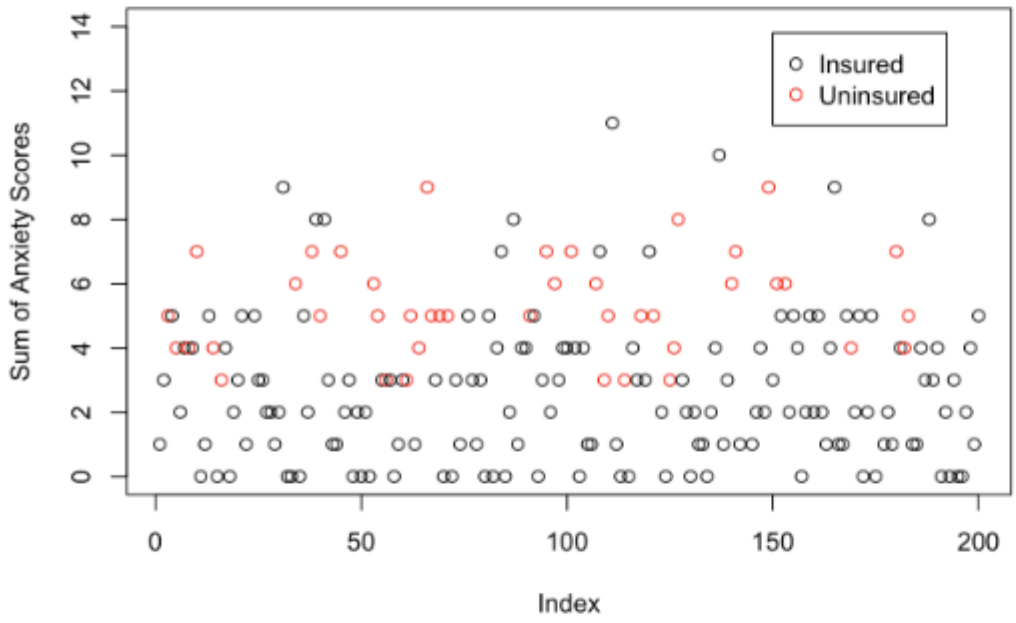
Produces an analysis notebook

```
analysis$sex <- as.factor(analysis$sex)
```

Uninsured analysis

Plot the anxietySum with symbol color determined by if ever uninsured Separate the insured and uninsured and run a t-test

```
plot(analysis$anxietySum, col=analysis$ever_uninsured,ylab="Sum of Anxiety Scores",ylim=c(0,14))
legend(150,13.8,legend=c("Insured","Uninsured"),pch=1,col=1:2)
```




```
notUninsured <- filter(analysis, ever_uninsured == FALSE)
Uninsured <- filter(analysis, ever_uninsured == TRUE)

diffAnxiety <- t.test(notUninsured$anxietySum, Uninsured$anxietySum)
diffAnxietyHyp <- ifelse(diffAnxiety$p.value < 0.05, "Rejected","Accepted")
```

Results of analysis with uninsured vs insured populations

The hypothesis that there is no difference between the insured and uninsured populations in their total anxiety is Rejected with p-value of 9.038641710^{-14}

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins | Project: (None)

PullAllCTSAPubs.R

Source on Save | Run | Source

```r
setwd("~/Google Drive/Work/Publication Classification/Analysis/R Code")

## load library to run entrez calls from R
library(rentrez)
library(XML)

## read in all CTSA grant nums and pull UL1, TL1, and KL2 grant nums
GrantNum <- read.csv("../All CTSA Grant Numbers.csv", stringsAsFactors = FALSE, na.strings="")
UL1 <- GrantNum$UL1[!is.na(GrantNum$UL1)]
TL1 <- GrantNum$TL1[!is.na(GrantNum$TL1)]
KL2 <- GrantNum$KL2[!is.na(GrantNum$KL2)]

## create a single vector with all grant numbers listed
AllGrantNums <- as.vector(append(append(UL1,TL1),KL2))

## pull out the "kernel" of the grant number, e.g. TR000038
kernels <- sapply(AllGrantNums, function(x) substring(x,5,13))

## create a full grant number without a space, e.g. UL1TR000038
fullNums <- gsub(" ","",AllGrantNums)

## create a vector with both the kernel and the full grant to allow
## for searching on different variants of the grant number
variants <- as.vector(append(kernels, fullNums))

publist <- lapply(variants, function(x) entrez_search(db="pubmed", term=x, retmax=5000))
pmids <- lapply(publist, function(x) x$ids)
pmidList <- unique(unlist(pmids))

## pad pmidList with a pmid with no abstract so that will be an even multiple of 500
```

15:1 (Top Level) R Script

Environment | History

Import Dataset | List

Global Environment

Data

GrantNum	178 obs. of 6 variables

Values

AllGrantNums	chr [1:495] "UL1 TR000086" "UL1 RR025750" "UL1 TR001073" "UL1 TR000157" "U…
KL2	chr [1:178] "KL2 TR000088" "KL2 RR025749" "KL2 TR001071" "KL2 TR000158" "K…
TL1	chr [1:139] "TL1 TR000087" "TL1 RR025748" "TL1 TR001072" "TL1 TR000159" "T…
UL1	chr [1:178] "UL1 TR000086" "UL1 RR025750" "UL1 TR001073" "UL1 TR000157" "U…

Console C:/Users/surkia01/Google Drive/Work/Publication Classification/Analysis/R Code/

```
R version 3.4.1 (2017-06-30) -- "Single Candle"
Copyright (C) 2017 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[workspace loaded from ~/.RData]

> setwd("C:/Users/surkia01/Google Drive/Work/Publication Classification/Analysis/R Code")
> library(rentrez)
>
> GrantNum <- read.csv("../All CTSA Grant Numbers.csv", stringsAsFactors = FALSE, na.strings="")
> UL1 <- GrantNum$UL1[!is.na(GrantNum$UL1)]
> TL1 <- GrantNum$TL1[!is.na(GrantNum$TL1)]
> KL2 <- GrantNum$KL2[!is.na(GrantNum$KL2)]
>
> ## create a single vector with all grant numbers listed
> AllGrantNums <- as.vector(append(append(UL1,TL1),KL2))
>
>
```

Files | Plots | Packages | Help | Viewer

R: Compare Installed Packages with CRAN-like Repositories | Find in Topic

update.packages {utils} R Documentation

Compare Installed Packages with CRAN-like Repositories

Description

`old.packages` indicates packages which have a (suitable) later version on the repositories whereas `update.packages` offers to download and install such packages.

`new.packages` looks for (suitable) packages on the repositories that are not already installed, and optionally offers them for installation.

Usage

```
update.packages(lib.loc = NULL, repos = getOption("repos"),
                contriburl = contrib.url(repos, type),
                method, instlib = NULL,
                ask = TRUE, available = NULL,
                oldPkgs = NULL, ..., checkBuilt = FALSE,
                type = getOption("pkgType"))

old.packages(lib.loc = NULL, repos = getOption("repos"),
             contriburl = contrib.url(repos, type),
             instPkgs = installed.packages(lib.loc = lib.loc),
             method, available = NULL, checkBuilt = FALSE,
             type = getOption("pkgType"))

new.packages(lib.loc = NULL, repos = getOption("repos"),
             contriburl = contrib.url(repos, type),
             instPkgs = installed.packages(lib.loc = lib.loc),
             method, available = NULL, ask = FALSE, ...,
```

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins | Project: (None)

PullAllCTSAPubs.R

Source on Save | Run | Source

```r
setwd("~/Google Drive/Work/Publication Classification/Analysis/R Code")

## load library to run entrez calls from R
library(rentrez)
library(XML)

## read in all CTSA grant nums and pull UL1, TL1, and KL2 grant nums
GrantNum <- read.csv("../All CTSA Grant Numbers.csv", stringsAsFactors = FALSE, na.strings="")
UL1 <- GrantNum$UL1[!is.na(GrantNum$UL1)]
TL1 <- GrantNum$TL1[!is.na(GrantNum$TL1)]
KL2 <- GrantNum$KL2[!is.na(GrantNum$KL2)]

## create a single vector with all grant numbers listed
AllGrantNums <- as.vector(append(append(UL1,TL1),KL2))

## pull out the "kernel" of the grant number, e.g. TR000038
kernels <- sapply(AllGrantNums, function(x) substring(x,5,13))

## create a full grant number without a space, e.g. UL1TR000038
fullNums <- gsub(" ","",AllGrantNums)

## create a vector with both the kernel and the full grant to allow
## for searching on different variants of the grant number
variants <- as.vector(append(kernels, fullNums))

publist <- lapply(variants, function(x) entrez_search(db="pubmed", term=x, retmax=5000))
pmids <- lapply(publist, function(x) x$ids)
pmidList <- unique(unlist(pmids))

## pad pmidList with a pmid with no abstract so that will be an even multiple of 500
```

15:1 (Top Level) R Script

Environment | History

Import Dataset | List

Global Environment

Data

GrantNum	178 obs. of 6 variables

Values

AllGrantNums	chr [1:495] "UL1 TR000086" "UL1 RR025750" "UL1 TR001073" "UL1 TR000157" "U…
KL2	chr [1:178] "KL2 TR000088" "KL2 RR025749" "KL2 TR001071" "KL2 TR000158" "K…
TL1	chr [1:139] "TL1 TR000087" "TL1 RR025748" "TL1 TR001072" "TL1 TR000159" "T…
UL1	chr [1:178] "UL1 TR000086" "UL1 RR025750" "UL1 TR001073" "UL1 TR000157" "U…

Console C:/Users/surkia01/Google Drive/Work/Publication Classification/Analysis/R Code/

```
R version 3.4.1 (2017-06-30) -- "Single Candle"
Copyright (C) 2017 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos,
'help.start()' for an HTML bro
Type 'q()' to quit R.

[Workspace loaded from ~/.RDat

> setwd("C:/Users/surkia01/Google Drive/Work/Publication Classification/Analysis/R Code")
> library(rentrez)
>
> GrantNum <- read.csv("../All CTSA Grant Numbers.csv", stringsAsFactors = FALSE, na.strings="")
> UL1 <- GrantNum$UL1[!is.na(GrantNum$UL1)]
> TL1 <- GrantNum$TL1[!is.na(GrantNum$TL1)]
> KL2 <- GrantNum$KL2[!is.na(GrantNum$KL2)]
>
> ## create a single vector with all grant numbers listed
> AllGrantNums <- as.vector(append(append(UL1,TL1),KL2))
>
>
```

Console

Files | Plots | Packages | Help | Viewer

R: Compare Installed Packages with CRAN-like Repositories | Find in Topic

update.packages {utils} R Documentation

Compare Installed Packages with CRAN-like Repositories

Description

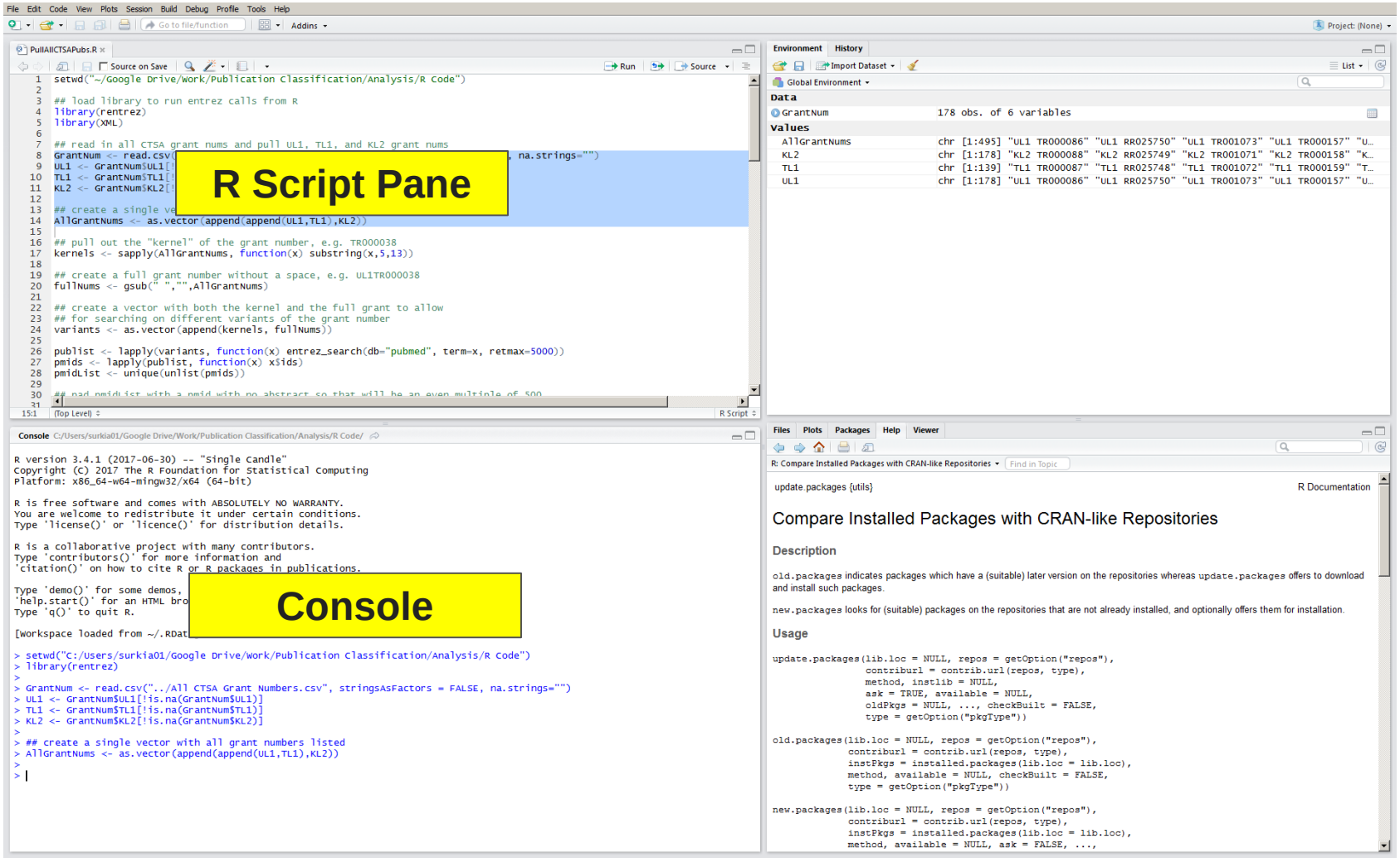
`old.packages` indicates packages which have a (suitable) later version on the repositories whereas `update.packages` offers to download and install such packages.

`new.packages` looks for (suitable) packages on the repositories that are not already installed, and optionally offers them for installation.

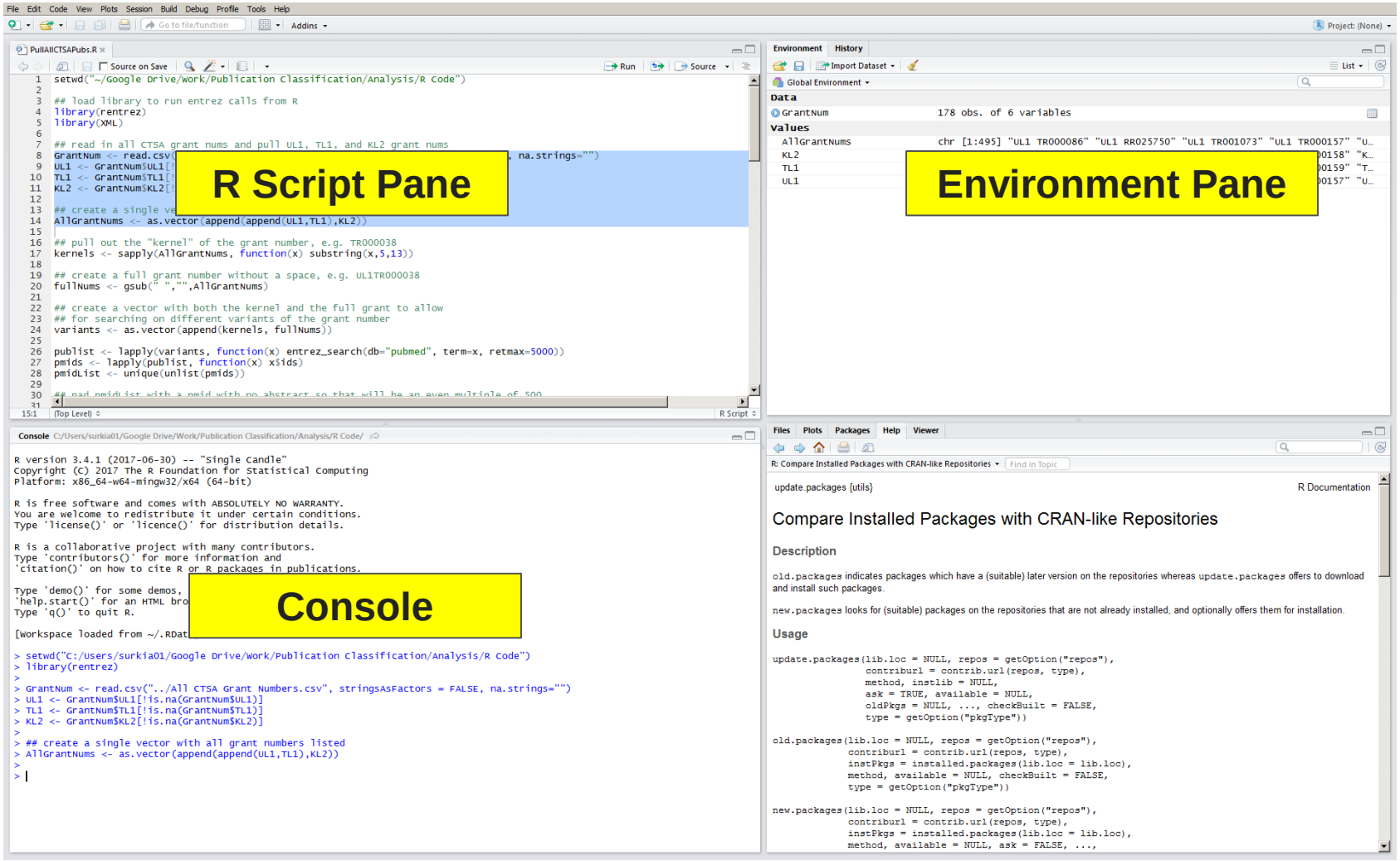
Usage

```r
update.packages(lib.loc = NULL, repos = getOption("repos"),
                contriburl = contrib.url(repos, type),
                method, instlib = NULL,
                ask = TRUE, available = NULL,
                oldPkgs = NULL, ..., checkBuilt = FALSE,
                type = getOption("pkgType"))

old.packages(lib.loc = NULL, repos = getOption("repos"),
             contriburl = contrib.url(repos, type),
             instPkgs = installed.packages(lib.loc = lib.loc),
             method, available = NULL, checkBuilt = FALSE,
             type = getOption("pkgType"))

new.packages(lib.loc = NULL, repos = getOption("repos"),
             contriburl = contrib.url(repos, type),
             instPkgs = installed.packages(lib.loc = lib.loc),
             method, available = NULL, ask = FALSE, ...,
```

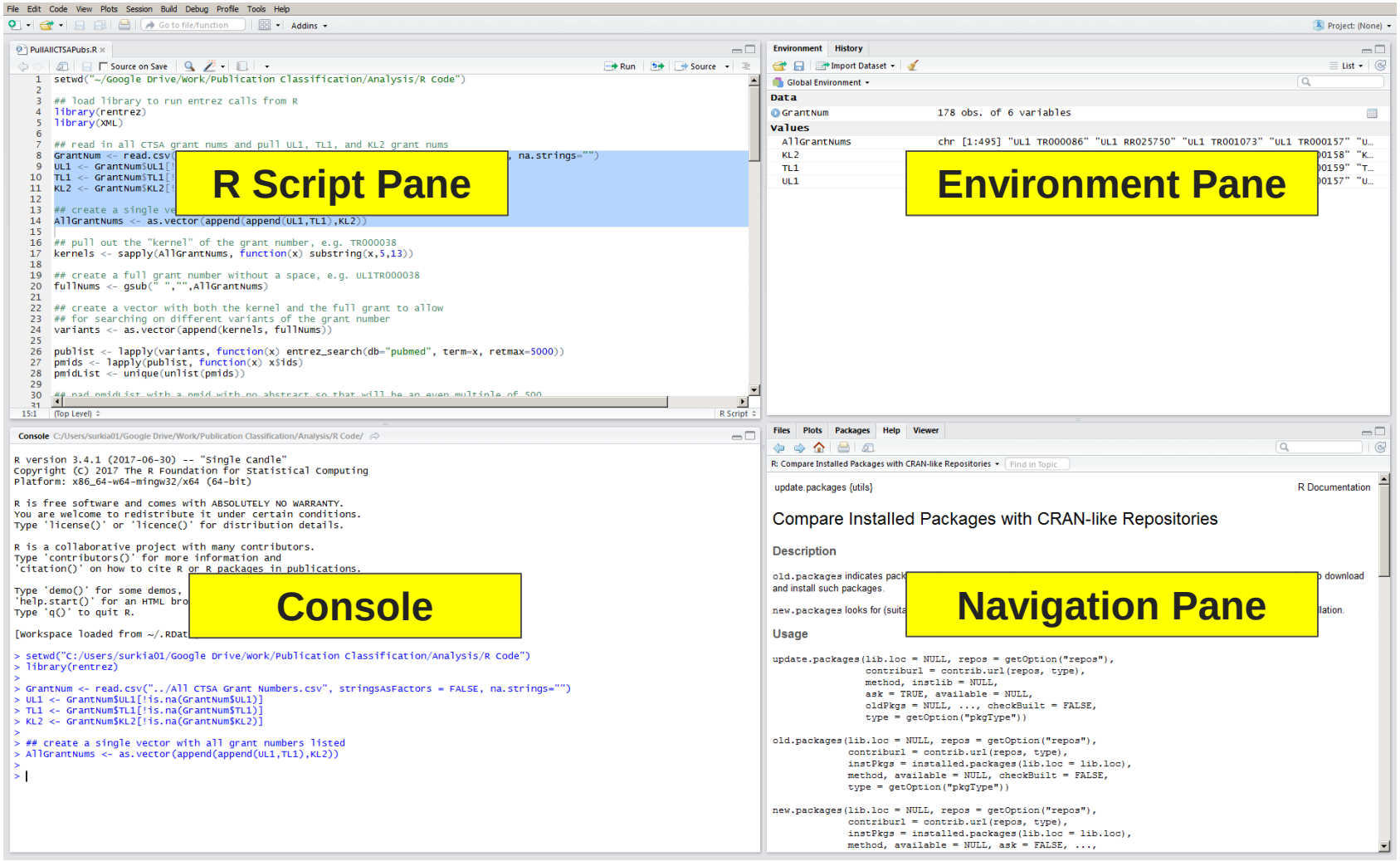
RStudio


File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
PullAllCTSAPubs.R
Source on Save
Run
Source
setwd("~/Google Drive/Work/Publication Classification/Analysis/R Code")
load library to run entrez calls from R
library(rentrez)
library(XML)
read in all CTSA grant nums and pull UL1, TL1, and KL2 grant nums
GrantNum <- read.csv(
, na.strings="")
UL1 <- GrantNum$UL1[!
TL1 <- GrantNum$TL1[!
KL2 <- GrantNum$KL2[!
create a single ve
AllGrantNums <- as.vector(append(append(UL1,TL1),KL2))
pull out the "kernel" of the grant number, e.g. TR000038
kernels <- sapply(AllGrantNums, function(x) substring(x,5,13))
create a full grant number without a space, e.g. UL1TR000038
fullNums <- gsub(" ","",AllGrantNums)
create a vector with both the kernel and the full grant to allow
for searching on different variants of the grant number
variants <- as.vector(append(kernels, fullNums))
publist <- lapply(variants, function(x) entrez_search(db="pubmed", term=x, retmax=5000))
pmids <- lapply(publist, function(x) x$ids)
pmidList <- unique(unlist(pmids))
15:1
(Top Level)
R Script
R Script Pane
Environment History
Import Dataset
List
Global Environment
Data
GrantNum 178 obs. of 6 variables
values
AllGrantNums chr [1:495] "UL1 TR000086" "UL1 RR025750" "UL1 TR001073" "UL1 TR000157" "U...
KL2 chr [1:178] "KL2 TR000088" "KL2 RR025749" "KL2 TR001071" "KL2 TR000158" "K...
TL1 chr [1:139] "TL1 TR000087" "TL1 RR025748" "TL1 TR001072" "TL1 TR000159" "T...
UL1 chr [1:178] "UL1 TR000086" "UL1 RR025750" "UL1 TR001073" "UL1 TR000157" "U...
Console C:/Users/surkia01/Google Drive/Work/Publication Classification/Analysis/R Code/
R version 3.4.1 (2017-06-30) -- "Single Candle"
Copyright (C) 2017 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
Console
> setwd("C:/Users/surkia01/Google Drive/Work/Publication Classification/Analysis/R Code")
> library(rentrez)
> GrantNum <- read.csv("../All CTSA Grant Numbers.csv", stringsAsFactors = FALSE, na.strings="")
> UL1 <- GrantNum$UL1[!is.na(GrantNum$UL1)]
> TL1 <- GrantNum$TL1[!is.na(GrantNum$TL1)]
> KL2 <- GrantNum$KL2[!is.na(GrantNum$KL2)]
> ## create a single vector with all grant numbers listed
> AllGrantNums <- as.vector(append(append(UL1,TL1),KL2))
Files Plots Packages Help Viewer
R: Compare Installed Packages with CRAN-like Repositories
Find in Topic
update.packages {utils}
R Documentation
Compare Installed Packages with CRAN-like Repositories
Description
old.packages indicates packages which have a (suitable) later version on the repositories whereas update.packages offers to download and install such packages.
new.packages looks for (suitable) packages on the repositories that are not already installed, and optionally offers them for installation.
Usage
update.packages(lib.loc = NULL, repos = getOption("repos"),
contriburl = contrib.url(repos, type),
method, instlib = NULL,
ask = TRUE, available = NULL,
oldPkgs = NULL, ..., checkBuilt = FALSE,
type = getOption("pkgType"))
old.packages(lib.loc = NULL, repos = getOption("repos"),
contriburl = contrib.url(repos, type),
instPkgs = installed.packages(lib.loc = lib.loc),
method, available = NULL, checkBuilt = FALSE,
type = getOption("pkgType"))
new.packages(lib.loc = NULL, repos = getOption("repos"),
contriburl = contrib.url(repos, type),
instPkgs = installed.packages(lib.loc = lib.loc),
method, available = NULL, ask = FALSE, ...,

RStudio



RStudio



RStudio

- Integrated development environment for R, which means one window with access to:
  - Variables
  - Help window
  - Plots
  - Scripting
- Convenience -- Tab Completion
- Reproducibility -- Integration with R Markdown

The data frame: backbone of R

risk_summary × | ProcessDataNotebook.Rmd × | EHRData × | REDCapData ×

Filter

	patient_id	age	date_of_test	sex	black	nativeAmerican	asian	pacificIsland	white	height_inches	weight_lbs	temp
1	HHUID0040937	43	2018-01-10	1	0	0	1	0	0	64	164	98
2	HHUID0081981	36	2018-04-01	2	0	0	1	0	0	69	192	97
3	HHUID0030501	33	2017-10-30	1	0	0	1	0	0	63	139	99
4	HHUID0019343	51	2018-04-14	1	1	0	0	0	0	66	181	98
5	HHUID0075313	54	2018-01-21	1	0	0	0	0	1	72	177	98
6	HHUID0083101	45	2017-10-13	1	0	0	0	0	1	67	180	101
7	HHUID0013597	49	2017-11-29	1	1	0	0	0	0	62	146	95
8	HHUID0084797	32	2018-03-25	2	0	0	0	0	1	70	194	97
9	HHUID0023084	38	2018-02-04	2	0	0	0	0	1	75	208	99
10	HHUID0043222	35	2018-02-09	1	0	0	0	0	1	64	151	97
11	HHUID0075792	49	2018-02-06	1	0	0	0	0	1	60	135	98
12	HHUID0085441	51	2018-03-11	1	1	0	0	0	1	69	NA	97
13	HHUID0029155	35	2017-11-26	1	0	0	0	0	1	62	174	98
14	HHUID0029533	52	2018-04-29	1	0	0	0	0	1	69	164	96
15	HHUID0013556	32	2018-04-14	2	0	0	0	0	1	73	216	97
16	HHUID0060753	23	2017-10-18	1	0	0	0	0	1	61	141	98
17	HHUID0040787	39	2018-01-29	1	0	0	0	0	1	70	186	99
18	HHUID0064580	23	2018-05-20	2	1	0	0	0	0	67	209	98
19	HHUID0016042	31	2017-11-14	1	0	0	0	0	1	67	183	98
20	HHUID0067541	55	2018-03-07	2	0	0	1	0	0	73	198	99
21	HHUID0028556	24	2017-09-26	1	0	0	0	0	1	70	186	99
22	HHUID0027191	44	2017-12-30	1	0	0	0	0	1	62	152	97
23	HHUID0010402	42	2017-09-20	2	0	0	0	0	1	67	175	95
24	HHUID0037431	23	2018-01-29	2	0	0	0	0	1	63	167	97
25	HHUID0065432	49	2017-09-23	1	0	0	0	0	1	69	165	98
26	HHUID0019336	36	2017-12-21	1	1	0	0	0	0	70	164	97
27	HHUID0057340	23	2018-04-10	1	0	0	0	0	1	63	155	99

NYU HEALTH SCIENCES LIBRARY

The data frame: backbone of R

Each column is a different **variable**

risk_summary × | ProcessDataNotebook.Rmd × | EHRData × | REDCapData ×

Filter

	patient_id	age	date_of_test	sex	black	nativeAmerican	asian	pacificIsland	white	height_inches	weight_lbs	temp
1	HHUID0040937	43	2018-01-10	1	0	0	1	0	0	64	164	98
2	HHUID0081981	36	2018-04-01	2	0	0	1	0	0	69	192	97
3	HHUID0030501	33	2017-10-30	1	0	0	1	0	0	63	139	99
4	HHUID0019343	51	2018-04-14	1	1	0	0	0	0	66	181	98
5	HHUID0075313	54	2018-01-21	1	0	0	0	0	1	72	177	98
6	HHUID0083101	45	2017-10-13	1	0	0	0	0	1	67	180	101
7	HHUID0013597	49	2017-11-29	1	1	0	0	0	0	62	146	95
8	HHUID0084797	32	2018-03-25	2	0	0	0	0	1	70	194	97
9	HHUID0023084	38	2018-02-04	2	0	0	0	0	1	75	208	99
10	HHUID0043222	35	2018-02-09	1	0	0	0	0	1	64	151	97
11	HHUID0075792	49	2018-02-06	1	0	0	0	0	1	60	135	98
12	HHUID0085441	51	2018-03-11	1	1	0	0	0	1	69	NA	97
13	HHUID0029155	35	2017-11-26	1	0	0	0	0	1	62	174	98
14	HHUID0029533	52	2018-04-29	1	0	0	0	0	1	69	164	96
15	HHUID0013556	32	2018-04-14	2	0	0	0	0	1	73	216	97
16	HHUID0060753	23	2017-10-18	1	0	0	0	0	1	61	141	98
17	HHUID0040787	39	2018-01-29	1	0	0	0	0	1	70	186	99
18	HHUID0064580	23	2018-05-20	2	1	0	0	0	0	67	209	98
19	HHUID0016042	31	2017-11-14	1	0	0	0	0	1	67	183	98
20	HHUID0067541	55	2018-03-07	2	0	0	1	0	0	73	198	99
21	HHUID0028556	24	2017-09-26	1	0	0	0	0	1	70	186	99
22	HHUID0027191	44	2017-12-30	1	0	0	0	0	1	62	152	97
23	HHUID0010402	42	2017-09-20	2	0	0	0	0	1	67	175	95
24	HHUID0037431	23	2018-01-29	2	0	0	0	0	1	63	167	97
25	HHUID0065432	49	2017-09-23	1	0	0	0	0	1	69	165	98
26	HHUID0019336	36	2017-12-21	1	1	0	0	0	0	70	164	97
27	HHUID0057340	23	2018-04-10	1	0	0	0	0	1	63	155	99

NYU HEALTH SCIENCES LIBRARY

The data frame: backbone of R

Each column is a different **variable**

Each row is an **observation**

risk_summary × | ProcessDataNotebook.Rmd × | EHRData × | REDCapData ×

Filter

	patient_id	age	date_of_test	sex	black	nativeAmerican	asian	pacificIsland	white	height_inches	weight_lbs	temp
1	HHUID0040937	43	2018-01-10	1	0	0	1	0	0	64	164	98
2	HHUID0081981	36	2018-04-01	2	0	0	1	0	0	69	192	97
3	HHUID0030501	33	2017-10-30	1	0	0	1	0	0	63	139	99
4	HHUID0019343	51	2018-04-14	1	1	0	0	0	0	66	181	98
5	HHUID0075313	54	2018-01-21	1	0	0	0	0	1	72	177	98
6	HHUID0083101	45	2017-10-13	1	0	0	0	0	1	67	180	101
7	HHUID0013597	49	2017-11-29	1	1	0	0	0	0	62	146	95
8	HHUID0084797	32	2018-03-25	2	0	0	0	0	1	70	194	97
9	HHUID0023084	38	2018-02-04	2	0	0	0	0	1	75	208	99
10	HHUID0043222	35	2018-02-09	1	0	0	0	0	1	64	151	97
11	HHUID0075792	49	2018-02-06	1	0	0	0	0	1	60	135	98
12	HHUID0085441	51	2018-03-11	1	1	0	0	0	1	69	NA	97
13	HHUID0029155	35	2017-11-26	1	0	0	0	0	1	62	174	98
14	HHUID0029533	52	2018-04-29	1	0	0	0	0	1	69	164	96
15	HHUID0013556	32	2018-04-14	2	0	0	0	0	1	73	216	97
16	HHUID0060753	23	2017-10-18	1	0	0	0	0	1	61	141	98
17	HHUID0040787	39	2018-01-29	1	0	0	0	0	1	70	186	99
18	HHUID0064580	23	2018-05-20	2	1	0	0	0	0	67	209	98
19	HHUID0016042	31	2017-11-14	1	0	0	0	0	1	67	183	98
20	HHUID0067541	55	2018-03-07	2	0	0	1	0	0	73	198	99
21	HHUID0028556	24	2017-09-26	1	0	0	0	0	1	70	186	99
22	HHUID0027191	44	2017-12-30	1	0	0	0	0	1	62	152	97
23	HHUID0010402	42	2017-09-20	2	0	0	0	0	1	67	175	95
24	HHUID0037431	23	2018-01-29	2	0	0	0	0	1	63	167	97
25	HHUID0065432	49	2017-09-23	1	0	0	0	0	1	69	165	98
26	HHUID0019336	36	2017-12-21	1	1	0	0	0	0	70	164	97
27	HHUID0057340	23	2018-04-10	1	0	0	0	0	1	63	155	99

NYU HEALTH SCIENCES LIBRARY

The data frame: backbone of R

Each column is a different **variable**

Each row is an **observation**

Like a spreadsheet

risk_summary × | ProcessDataNotebook.Rmd × | EHRData × | REDCapData ×

Filter

	patient_id	age	date_of_test	sex	black	nativeAmerican	asian	pacificIsland	white	height_inches	weight_lbs	temp
1	HHUID0040937	43	2018-01-10	1	0	0	1	0	0	64	164	98
2	HHUID0081981	36	2018-04-01	2	0	0	1	0	0	69	192	97
3	HHUID0030501	33	2017-10-30	1	0	0	1	0	0	63	139	99
4	HHUID0019343	51	2018-04-14	1	1	0	0	0	0	66	181	98
5	HHUID0075313	54	2018-01-21	1	0	0	0	0	1	72	177	98
6	HHUID0083101	45	2017-10-13	1	0	0	0	0	1	67	180	101
7	HHUID0013597	49	2017-11-29	1	1	0	0	0	0	62	146	95
8	HHUID0084797	32	2018-03-25	2	0	0	0	0	1	70	194	97
9	HHUID0023084	38	2018-02-04	2	0	0	0	0	1	75	208	99
10	HHUID0043222	35	2018-02-09	1	0	0	0	0	1	64	151	97
11	HHUID007[illegible]									60	135	98
12	HHUID008[illegible]									69	NA	97
13	HHUID002[illegible]									62	174	98
14	HHUID002[illegible]									69	164	96
15	HHUID0013556	32	2018-04-14	2	0	0	0	0	1	73	216	97
16	HHUID0060753	23	2017-10-18	1	0	0	0	0	1	61	141	98
17	HHUID0040787	39	2018-01-29	1	0	0	0	0	1	70	186	99
18	HHUID0064580	23	2018-05-20	2	1	0	0	0	0	67	209	98
19	HHUID0016042	31	2017-11-14	1	0	0	0	0	1	67	183	98
20	HHUID0067541	55	2018-03-07	2	0	0	1	0	0	73	198	99
21	HHUID0028556	24	2017-09-26	1	0	0	0	0	1	70	186	99
22	HHUID0027191	44	2017-12-30	1	0	0	0	0	1	62	152	97
23	HHUID0010402	42	2017-09-20	2	0	0	0	0	1	67	175	95
24	HHUID0037431	23	2018-01-29	2	0	0	0	0	1	63	167	97
25	HHUID0065432	49	2017-09-23	1	0	0	0	0	1	69	165	98
26	HHUID0019336	36	2017-12-21	1	1	0	0	0	0	70	164	97
27	HHUID0057340	23	2018-04-10	1	0	0	0	0	1	63	155	99

NYU HEALTH SCIENCES LIBRARY

Data types within data frames

- Numeric variable = *numeric* or *integer*

  Ex: 1, 1.5, 200000, 3.14159

Data types within data frames

- Numeric variable = *numeric* or *integer*

  Ex: 1, 1.5, 200000, 3.14159

- Text variable = *character*

  Ex: a, b, hello, 3b

Data types within data frames

- Numeric variable = *numeric* or *integer*

  Ex: 1, 1.5, 200000, 3.14159

- Text variable = *character*

  Ex: a, b, hello, 3b

- Categorical variable = *factor*

  Ex: cat, dog, pig, rhino, horse

Data types within data frames

- Numeric variable = *numeric* or *integer*

  Ex: 1, 1.5, 200000, 3.14159

- Text variable = *character*

  Ex: a, b, hello, 3b

- Categorical variable = *factor*

  Ex: cat, dog, pig, rhino, horse

- Ordered categorical variable = *ordered factor*

  Ex: xsmall, small, medium, large, xlarge

Data types within data frames

- Numeric variable = *numeric* or *integer*

  Ex: 1, 1.5, 200000, 3.14159

- Text variable = *character*

  Ex: a, b, hello, 3b

- Categorical variable = *factor*

  Ex: cat, dog, pig, rhino, horse

- Ordered categorical variable = *ordered factor*

  Ex: xsmall, small, medium, large, xlarge

- True/false = *logical*

  Ex: TRUE, FALSE

Talking to R

Function calls

- how to tell R what to do
- always followed by ()

```
getwd()
```

Functions: Output

```
dat <- read.csv(file = "allFound.csv", header = TRUE)
```

Assignment operator (`<-`): assigns the output of the function to a name that you can then refer back to

Functions: Input

```
dat <- read.csv(file = "allFound.csv", header = TRUE)
```

Arguments: instructions that specify how a function should be run. Not always required, may be more than one, they are separated by commas, and they all go in the ()

Functions: Input

```
dat <- read.csv(file = "allFound.csv", header = TRUE)
```

required

optional

Argument: instructions that specify how a function should be run. Not always required, may be more than one, they are separated by commas, and they all go in the ()

Functions: Output

```
dat <- read.csv(file = "allFound.csv", header = TRUE)
```

Data frame

Data frames: Accessing elements

Each column is a different **variable**

Each row is an **observation**

risk_summary × | ProcessDataNotebook.Rmd × | EHRData × | REDCapData ×

Filter

	patient_id	age	date_of_test	sex	black	nativeAmerican	asian	pacificIsland	white	height_inches	weight_lbs	temp
1	HHUID0040937	43	2018-01-10	1	0	0	1	0	0	64	164	98
2	HHUID0081981	36	2018-04-01	2	0	0	1	0	0	69	192	97
3	HHUID0030501	33	2017-10-30	1	0	0	1	0	0	63	139	99
4	HHUID0019343	51	2018-04-14	1	1	0	0	0	0	66	181	98
5	HHUID0075313	54	2018-01-21	1	0	0	0	0	1	72	177	98
6	HHUID0083101	45	2017-10-13	1	0	0	0	0	1	67	180	101
7	HHUID0013597	49	2017-11-29	1	1	0	0	0	0	62	146	95
8	HHUID0084797	33	2018-03-25	2	0	0	0	0	1	70	194	97
9	HHUID0023084	38	2018-02-04	2	0	0	0	0	1	75	208	99
10	HHUID0043222	35	2018-02-09	1	0	0	0	0	1	64	151	97
11	HHUID0075792	49	2018-02-06	1	0	0	0	0	1	60	135	98
12	HHUID0085441	51	2018-03-11	1	1	0	0	0	1	69	NA	97
13	HHUID0029155	35	2017-11-26	1	0	0	0	0	1	62	174	98
14	HHUID0029533	52	2018-04-29	1	0	0	0	0	1	69	164	96
15	HHUID0013556	32	2018-04-14	2	0	0	0	0	1	73	216	97
16	HHUID0060753	23	2017-10-18	1	0	0	0	0	1	61	141	98
17	HHUID0040787	39	2018-01-29	1	0	0	0	0	1	70	186	99
18	HHUID0064580	23	2018-05-20	2	1	0	0	0	0	67	209	98
19	HHUID0016042	31	2017-11-14	1	0	0	0	0	1	67	183	98
20	HHUID0067541	55	2018-03-07	2	0	0	1	0	0	73	198	99
21	HHUID0028556	24	2017-09-26	1	0	0	0	0	1	70	186	99
22	HHUID0027191	44	2017-12-30	1	0	0	0	0	1	62	152	97
23	HHUID0010402	42	2017-09-20	2	0	0	0	0	1	67	175	95
24	HHUID0037431	23	2018-01-29	2	0	0	0	0	1	63	167	97
25	HHUID0065432	49	2017-09-23	1	0	0	0	0	1	69	165	98
26	HHUID0019336	36	2017-12-21	1	1	0	0	0	0	70	164	97
27	HHUID0057340	23	2018-04-10	1	0	0	0	0	1	63	155	99

NYU HEALTH SCIENCES LIBRARY

Data frames: Accessing elements

Each column is a different **variable**

Each row is an **observation**

dat[rows, columns]

Data frames: Accessing elements

Each column is a different **variable**

Each row is an **observation**

dat[rows, columns]

dat[observations, variables]

Data Frames: Accessing variables

risk_summary | ProcessDataNotebook.Rmd | EHRData | REDCapData

Filter

	patient_id	age	date_of_test	sex	black	nativeAmerican	asian	pacificIsland	white	height_inches	weight_lbs	temp
1	HHUID0040937	43	2018-01-10	1	0	0	1	0	0	64	164	98
2	HHUID0081981	36	2018-04-01	2	0	0	1	0	0	69	192	97
3	HHUID0030501	33	2017-10-30	1	0	0	1	0	0	63	139	99
4	HHUID0019343	51	2018-04-14	1	1	0	0	0	0	66	181	98
5	HHUID0075313	54	2018-01-21	1	0	0	0	0	1	72	177	98
6	HHUID0083101	45	2017-10-13	1	0	0	0	0	1	67	180	101
7	HHUID0013597	49	2017-11-29	1	1	0	0	0	0	62	146	95
8	HHUID0084797	32	2018-03-25	2	0	0	0	0	1	70	194	97
9	HHUID0023084	38	2018-02-04	2	0	0	0	0	1	75	208	99
10	HHUID0043222	35	2018-02-09	1	0	0	0	0	1	64	151	97
11	HHUID0075792	49	2018-02-06	1	0	0	0	0	1	60	135	98
12	HHUID0085441	51	2018-03-11	1	1	0	0	0	1	69	NA	97
13	HHUID0029155	35	2017-11-26	1	0	0	0	0	1	62	174	98
14	HHUID0029533	52	2018-04-29	1	0	0	0	0	1	69	164	96
15	HHUID0013556	32	2018-04-14	2	0	0	0	0	1	73	216	97
16	HHUID0060753	23	2017-10-18	1	0	0	0	0	1	61	141	98
17	HHUID0040787	39	2018-01-29	1	0	0	0	0	1	70	186	99

dat$date_of_test

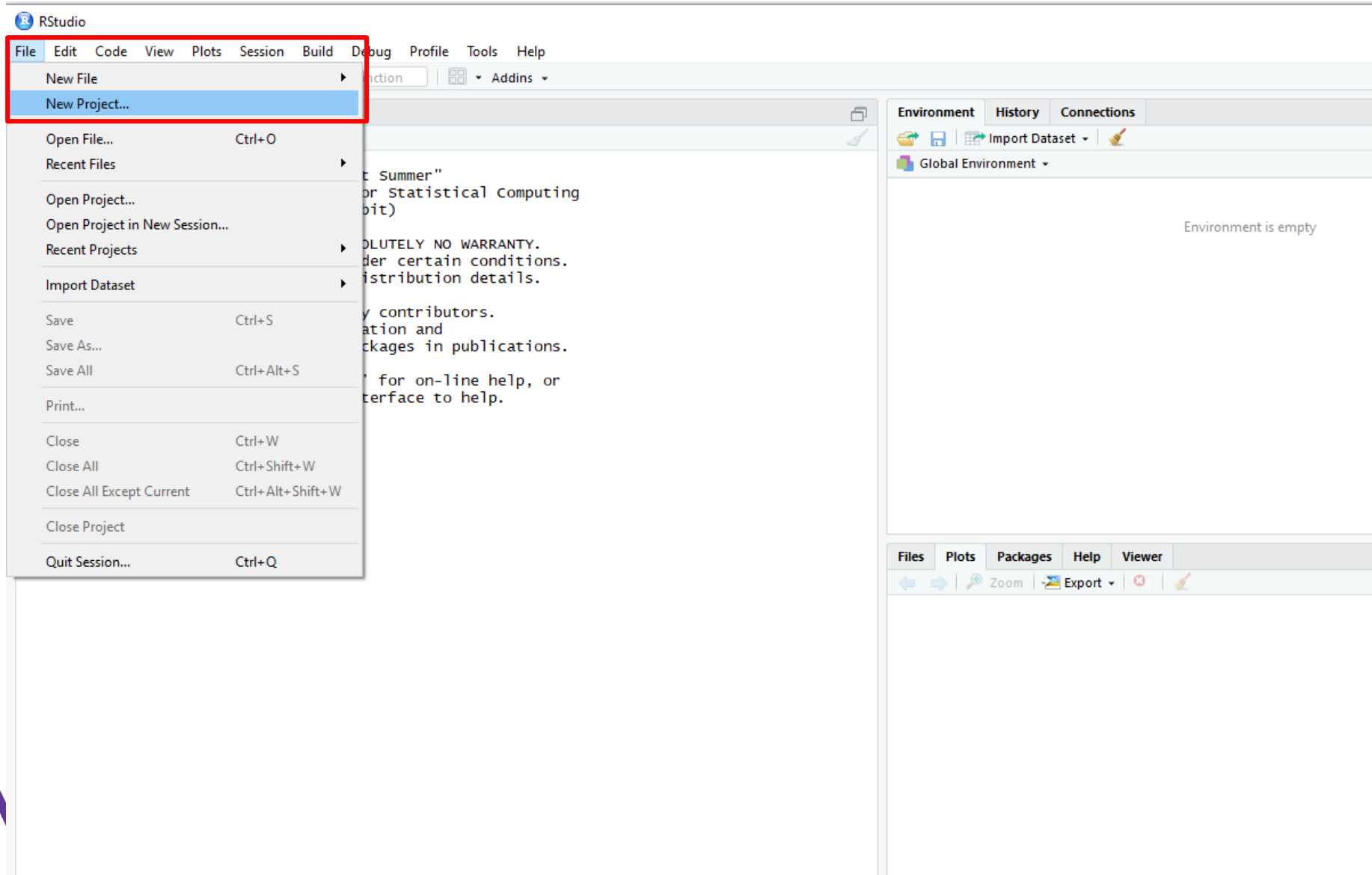
R Projects

- An R Project has its own
  - Working directory
  - Workspace
  - History
  - Source documents
- Convenience of starting where you left off
- Helps with organization

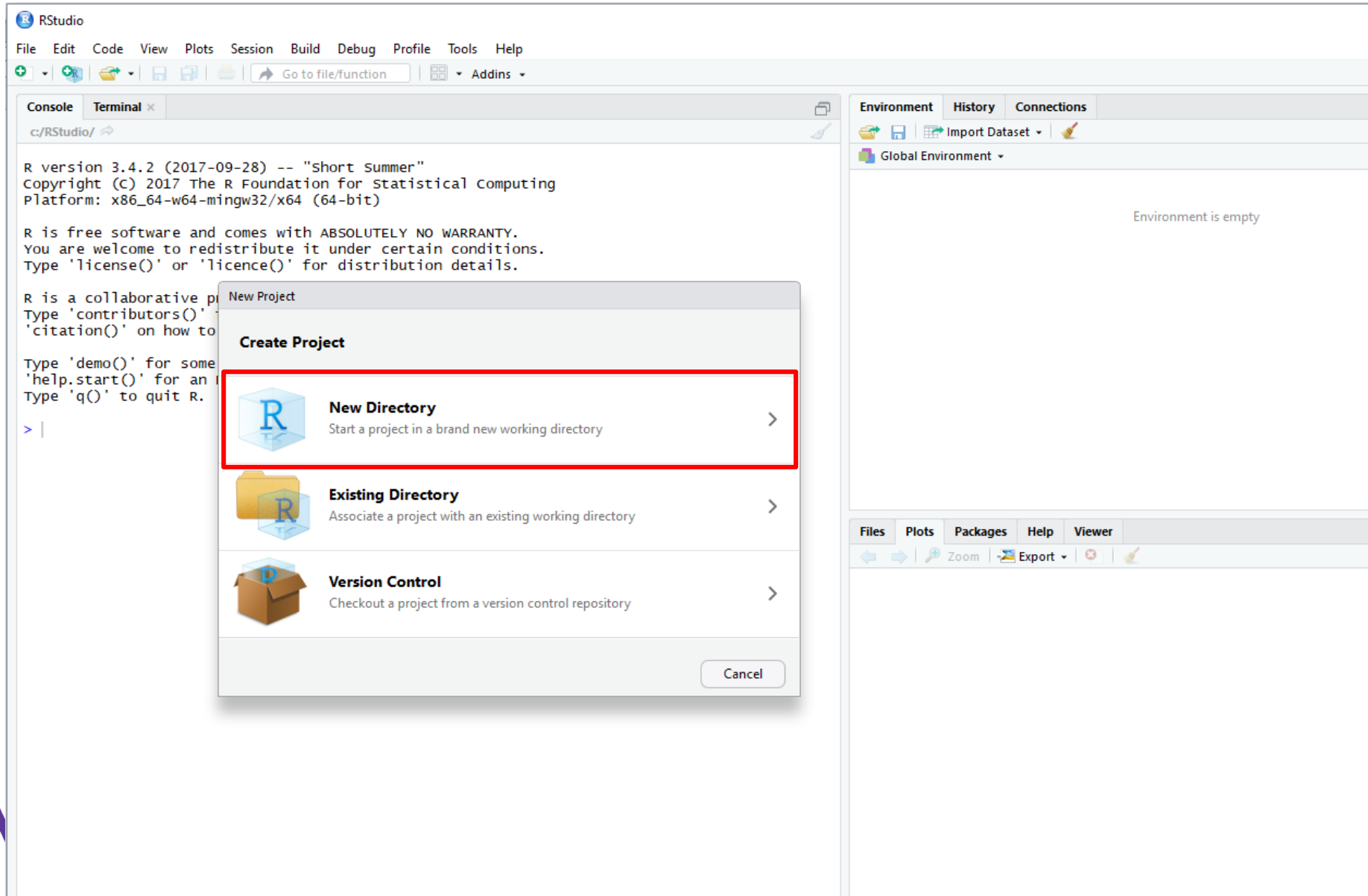
Creating your R Project

- Open R Studio
- Select New Project from the File menu

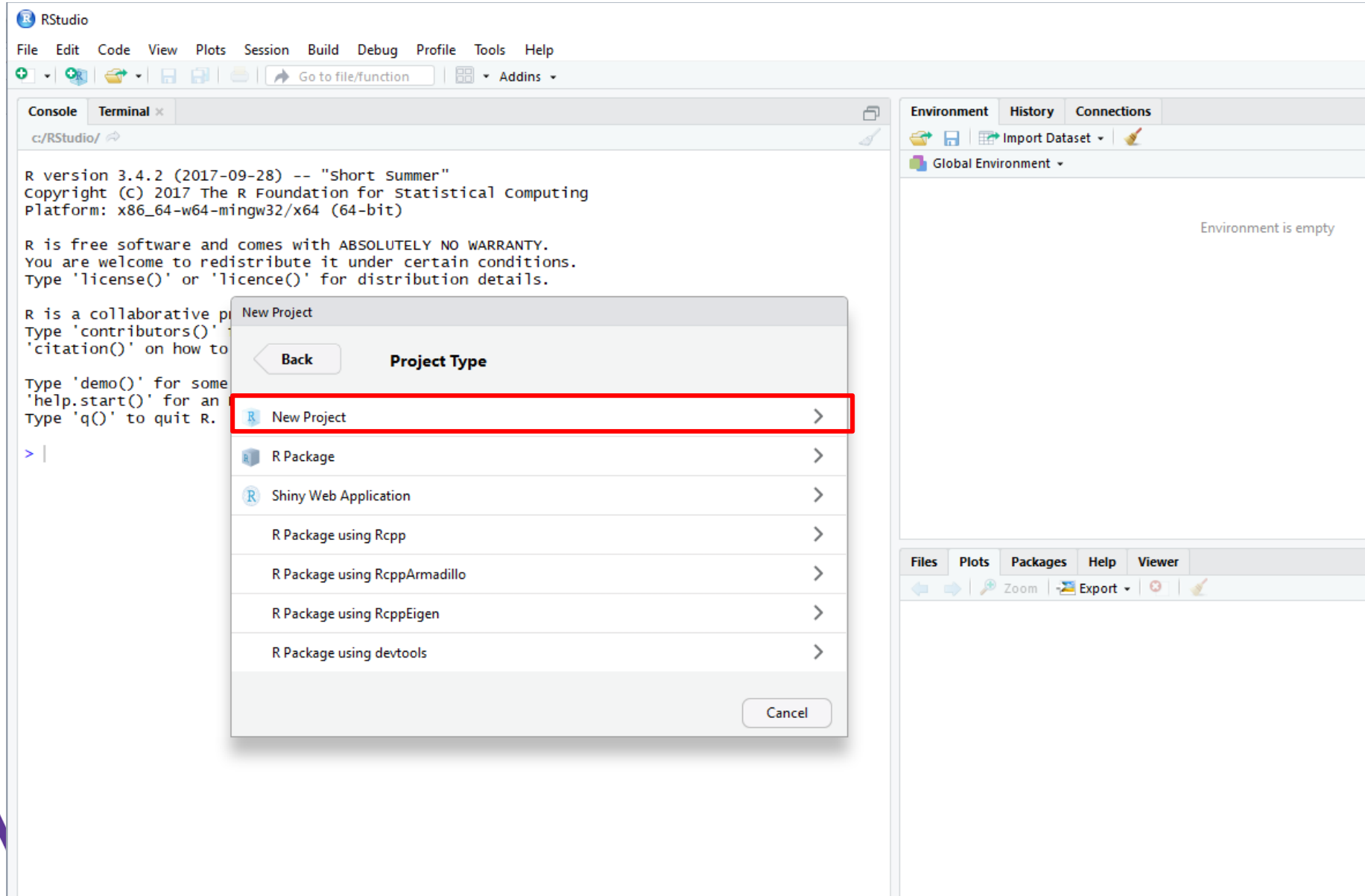
Creating an R Project



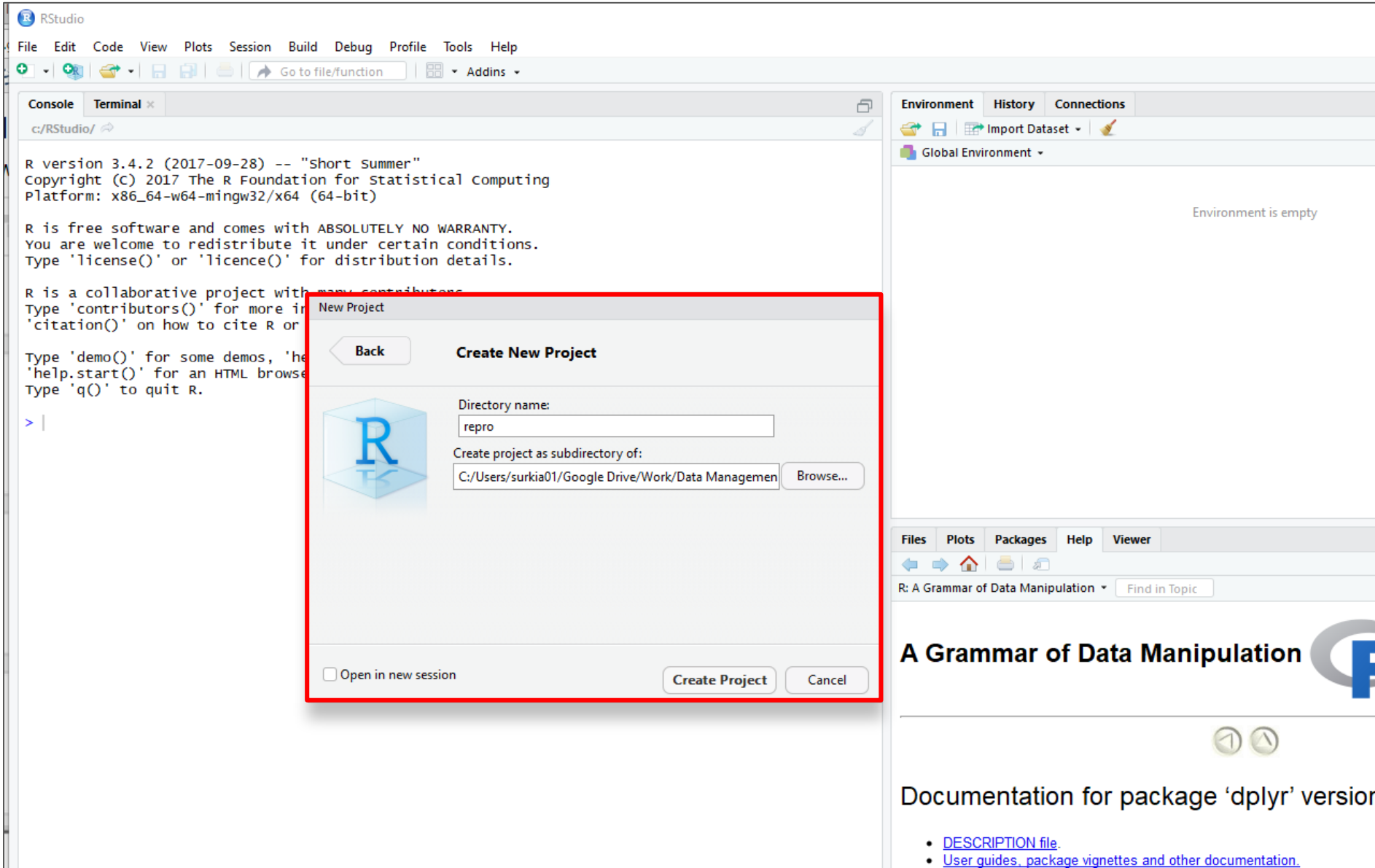
Creating an R Project



Creating an R Project



Creating an R Project



Structuring your project

Create the following folders within your repro folder:

- code
- raw_data
- processed_data
- results

Name	Date modified	Type	Size
.Rproj.user	7/2/2018 3:03 PM	File folder	
code	7/2/2018 3:04 PM	File folder	
processed_data	7/2/2018 3:04 PM	File folder	
raw_data	7/2/2018 3:04 PM	File folder	
results	7/2/2018 3:04 PM	File folder	
desktop.ini	7/2/2018 3:03 PM	Configuration sett...	1 KB
repro.Rproj	7/2/2018 3:03 PM	RPROJ File	1 KB

Data files

Drag and drop the following two files into the code folder that is inside your repro folder:

- EHRData.csv
- REDCapData_Export.csv

Good Enough Practices for Scientific Computing: Data Management

- Save the raw data
- Record all steps for processing data
  - Do all processing within R
- Each project in its own directory
  - Directory named after project
- Each document type in its own sub-directory
- All files with names that reflect their content/function

http://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1005510

GO TO HANDOUT

Reproducibility Checklist

Documentation

❏ Is it clear where to begin? (e.g., can someone picking a project up see where to start running it)

❏ Can you determine which file(s) was/were used as input in a process that produced a derived file?

❏ Is there documentation about every result?

❏ Have you noted the exact version of every external application used in the process?

❏ For analyses that include randomness, have you noted the underlying random seed(s)?

http://ropensci.github.io/reproducibility-guide/sections/checklist/

Reproducibility Checklist

Organization

- ❑ Which is the most recent data file/code?
- ❑ Which folders can I safely delete?
- ❑ Have you stored the raw data behind each plot?
- ❑ Do you run backups on all files associated with your analysis?

http://ropensci.github.io/reproducibility-guide/sections/checklist/

Reproducibility Checklist

Automation

- ❑ Are there lots of manual data manipulation steps are there?
- ❑ Are all custom scripts under version control?
- ❑ Is your writing (content) under version control?

http://ropensci.github.io/reproducibility-guide/sections/checklist/

Reproducibility Checklist

Publication

❏ Have you archived the exact version of every external application used in your process(es)?

❏ Are textual statements connected/linked to the supporting results or data?

❏ Did you archived preprints of resulting papers in a public repository?

❏ Did you release the underlying code at the time of publishing a paper?

❏ Are you providing public access to your scripts, runs, and results?

http://ropensci.github.io/reproducibility-guide/sections/checklist/