

Package: rWSBIM1207 (via r-universe)

September 13, 2024

Type Package

Title Companion Package for WSBIM1207 Course

Version 0.1.19

Description Companion package for the WSBIM1207 course, distributing data and general documentation, and making course administration easier.

License GPL-2

Depends R (>= 3.5.0)

Imports utils, tibble, dplyr

Suggests MSnbase, Biobase, SummarizedExperiment, pRolocdata, curatedTCGAData, ALL, tidyverse, rmarkdown, readr

Encoding UTF-8

LazyData true

RoxygenNote 7.3.1

Repository <https://uclouvain-cbio.r-universe.dev>

RemoteUrl <https://github.com/UCLouvain-CBIO/rWSBIM1207>

RemoteRef HEAD

RemoteSha 3561033e7a99182664a0a20f603382b5539941f0

Contents

apple_mobility	2
beers	3
covid19	4
english_education_files	4
faillites_be	5
GSE96870	6
interro	7
jdf	7
kem	8
load_exam_data	9

mvylnq	10
peptides	10
population	11
road_accidents	11
rWSBIM1207	12
tcga	12
valentine	14

Index	15
--------------	-----------

apple_mobility	<i>Apple Mobility Data</i>
----------------	----------------------------

Description

Apple mobility data, downloaded from <https://www.apple.com/covid19/mobility> on the 18 August 2021.

The following description has been taken from the Apple Mobility Trends Reports page:

The CSV file and charts on this site show a relative volume of directions requests per country/region, sub-region or city compared to a baseline volume on January 13th, 2020. We define our day as midnight-to-midnight, Pacific time. Cities are defined as the greater metropolitan area and their geographic boundaries remain constant across the data set. In many countries/regions, sub-regions, and cities, relative volume has increased since January 13th, consistent with normal, seasonal usage of Apple Maps. Day of week effects are important to normalize as you use this data. Data that is sent from users' devices to the Maps service is associated with random, rotating identifiers so Apple doesn't have a profile of individual movements and searches. Apple Maps has no demographic information about our users, so we can't make any statements about the representativeness of usage against the overall population.

Usage

```
apple_mobility.csv()
```

Source

<https://www.apple.com/covid19/mobility>

Examples

```
apple_mobility.csv()
read.csv(apple_mobility.csv())
```

beers	<i>Beer consumption data</i>
-------	------------------------------

Description

A small data frame describing the beer consumption and demographics of 48 people.

Usage

```
data("beers")
```

Format

A data frame with 48 observations on the following 8 variables.

Record_ID a numeric vector

Work a factor with levels Employed Unemployed

Consumption a numeric vector

Gender a factor with levels Female Male

Age a numeric vector

Day a numeric vector

Month a numeric vector

Year a numeric vector

Examples

```
data(beers)
beers
str(beers)

f <- beers.csv()
basename(f)
beers2 <- read.csv(f, sep = ";")
beers2

identical(beers, beers2)
```

 covid19

Coronavirus (COVID-19) Cases

Description

These data were downloaded from Novel Coronavirus COVID-19 (2019-nCoV) Data Repository by Johns Hopkins CSSE (<https://github.com/CSSEGISandData/COVID-19>):

This is the data repository for the 2019 Novel Coronavirus Visual Dashboard operated by the Johns Hopkins University Center for Systems Science and Engineering (JHU CSSE). Also, Supported by ESRI Living Atlas Team and the Johns Hopkins University Applied Physics Lab (JHU APL).

The 'covid19_cases.csv()', 'covid19_deaths.csv()' and 'covid19_recovered.csv()' functions return the path to a comma-separated file containing confirmed cases, deaths and recovered cases over time for a certain number of countries/regions. See example below for details.

Usage

```
covid19_cases.csv()
covid19_deaths.csv()
covid19_recovered.csv()
```

Examples

```
cv <- readr::read_csv(covid19_cases.csv())
## dates (format: month/day/year)
names(cv)[-1:4]
## Countries/Regions
unique(cv[[2]])
## Province/States
unique(cv[[1]])
```

 english_education_files

TidyTuesday's English Education Data

Description

These two files describe the 'Educational attainment of young people in English towns' from the the [UK Office for National Statistics](#). It was explored in the July 2023 article "[Why do children and young people in smaller towns do better academically than those in larger towns?](#)".

Two files are available:

- `english_education.csv` was prepared as part of the [Tidy Tuesday](#) series on [Educational attainment of young people in English towns](#) (2024-01-23). The page also describes the variables.
- `edu_income_eprivation_and_educational_attainment.csv` was downloaded from the UK Office for National Statistics and converted from xls to csv. The variables in this table are also described on the page linked above.

Usage

```
english_education_files()
```

Value

character(2) with file names.

Author(s)

Laurent Gatto

References

- "Why do children and young people in smaller towns do better academically than those in larger towns?".
- Tidy Tuesday data from 2024-01-23: <https://github.com/rfordatascience/tidytuesday/blob/master/data/2024/2024-01-23/readme.md>

Examples

```
english_education_files()
```

faillites_be

Evolution mensuelle des faillites par NACE

Description

Ce jeu de données repondent l'évolution mensuelle des faillites par NACE - 15 jours de 2005 à 2023, telles que distribuées par statbel, l'office belge de statistique.

Usage

```
faillites_be()
```

Value

character(2) with file names.

Traitement des données

- Les données ont été téléchargées de <https://statbel.fgov.be/fr/open-data/evolution-mensuelle-des-faill>
- Un sous-échantillon de 11544 observation a été sélectionné, de telle manière a garder les entreprises de grandes tailles. Voir le scripts/faillites.R.

Ce package re-distribue deux fichiers

- TF_BANKRUPTCIES_subset.txt.zip: échantillon des données sous format compressé, les valeurs étant séparées pas le caractère | (format identique aux données complètes distribuées par statbel).
- Method_BANKRUPTCIES.xlsx: les métadonnées décrivant les variables, en format xlsx, tel que distribué par statbel.

Examples

```
faillites_be()
```

GSE96870

RNA-seq data from Blackmore et al. 2017

Description

Dataset from *The effect of upper-respiratory infection on transcriptomic changes in the CNS* by Blackmore et al. (2017) (<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5544260/>) used as data for the Bioconductor intro and RNA-seq lessons and WSBIM1207 courses. The two variables are of class SummarizedExperiment.

Usage

```
data("GSE96870_intro")  
data("GSE96870_intro_ranges")
```

Details

For details on how the data was prepared, see <https://github.com/Bioconductor/bioconductor-teaching/tree/master/data/GSE96870>.

Examples

```
data("GSE96870_intro")  
GSE96870_intro  
rowData(GSE96870_intro)  
  
data("GSE96870_intro_ranges")  
GSE96870_intro_ranges  
rowRanges(GSE96870_intro_ranges)
```

interro

Practice datasets

Description

‘interroA.csv’ and ‘interroB.csv’ are two comma-separated spreadsheets that provide made-up data about student test results.

A data frame with 100 observations on the following 8 variables.

id student identifier.

height student heights (in cm).

gender F or M.

X a vector of random data drawn from $N(0, 1)$.

interro1 a numeric vector with test scores.

interro2 a numeric vector with test scores.

interro3 a numeric vector with test scores.

interro4 a numeric vector with test scores.

The ‘interroC’ data contains results for results for 3 new tests for 15 students, including a subset of students in ‘interroA’.

‘interroL’ is a long format containing all results from ‘interroA’ and ‘interroB’.

Examples

```
f <- interroA.csv()
interroA <- read.csv(f)
head(interroA)
```

```
f2 <- interro2.rds()
readRDS(f2)
```

jdf

Data illustrating join operations

Description

A set of tibbles contain information about genes and their protein products used to illustrate join operations.

Usage

```
data("jdf")
```

Source

These data are based on feature variables from the hyperLOPIT2015 data available in the pRolocdata package. The script to generate, join.R, is available in the scripts package directory.

Examples

```
data(jdf)
library("dplyr")
dplyr::full_join(jdf1, jdf2)
dplyr::left_join(jdf6, jdf7)
```

kem

KEM RNA-Seq data

Description

This data is composed of three files, namely kem_counts.tsv, kem_counts2.tsv, containing counts data and the annotation file kem_annot.tsv containing the annotations the KEM samples. Both files are encoded as tab-separated sheets and can be found with the kem.tsv() function.

Credit: the data have been generated by Mr Kevin Missault.

Format

The RNA-Seq count data for 13 ENSEMBL transcripts (ref) and 16 KEM samples is encoded as:

ref ENSEMBL transcript identifiers.

- 'KEM182-01' Expression counts for all genes in sample KEM182-01.
- 'KEM182-02' Expression counts for all genes in sample KEM182-02.
- 'KEM182-03' Expression counts for all genes in sample KEM182-03.
- 'KEM182-04' Expression counts for all genes in sample KEM182-04.
- 'KEM182-05' Expression counts for all genes in sample KEM182-05.
- 'KEM182-06' Expression counts for all genes in sample KEM182-06.
- 'KEM182-07' Expression counts for all genes in sample KEM182-07.
- 'KEM182-08' Expression counts for all genes in sample KEM182-08.
- 'KEM182-09' Expression counts for all genes in sample KEM182-09.
- 'KEM182-10' Expression counts for all genes in sample KEM182-10.
- 'KEM182-11' Expression counts for all genes in sample KEM182-11.
- 'KEM182-12' Expression counts for all genes in sample KEM182-12.
- 'KEM182-13' Expression counts for all genes in sample KEM182-13.
- 'KEM182-14' Expression counts for all genes in sample KEM182-14.
- 'KEM182-15' Expression counts for all genes in sample KEM182-15.
- 'KEM182-16' Expression counts for all genes in sample KEM182-16.

The kem_counts2.tsv file contains counts data for 4774 features.

The annotation contains the following variables for the 16 observations:

sample_id Sample identifier.

jurkat A character (yes or no) defining of the cells a Jurkat cells.

cell_type Jurkat cell type (A or B).

treatment Treatment: either none or stimulated.

Examples

```
kem.tsv()  
kem2.tsv()  
kem3.tsv()
```

load_exam_data	<i>Make and load exam data</i>
----------------	--------------------------------

Description

This function generates data based on a student number and loads it in the users environment.

Usage

```
load_exam_data(noma)
```

Arguments

noma 'character(1)' with the student number. Must be coercible to a 'numeric'.

Value

Invisibly returns 'TRUE'. Used for its side effect to load an object of class 'MSnSet' in the user's global environment.

Examples

```
load_exam_data("0123")  
x0123
```

`mvyIng`*Data from Mulvey et al. 2015*

Description

This is the "mulvey2015" data from Mulvey et al., *Dynamic proteomic profiling of extra-embryonic endoderm differentiation in mouse embryonic stem cells.* *Stem Cell.* (PMID 26059426). See below for more details.

It was extracted from pRolocdata package and converted into a long format.

Usage

```
data("mvyIng")
```

Examples

```
data(mvyIng)
mvyIng
```

`peptides`*A vector of peptide sequences*

Description

A character vector containing peptide sequences.

Usage

```
data("peptides")
```

Source

The peptides were extracted from the 'hyperLOPIT2015ms3r1psm' object in the 'pRolocdata' package using 'unique(as.character(fData(hyperLOPIT2015ms3r1psm)\$Sequence))'.

Examples

```
data(peptides)
head(peptides)
```

population	<i>Belgium population</i>
------------	---------------------------

Description

A set of csv files providing the population sizes in different regions in Belgium from 1970, 1981 and 1991 to 2023.

Usage

```
population_be.csv()
```

Examples

```
population_be.csv()  
read.csv(population_be.csv()[1])
```

road_accidents	<i>Road accidents data</i>
----------------	----------------------------

Description

Number of killed, seriously injured, slightly injured and uninjured victims of road accidents, by age group, type of user, sex and various characteristics of the accident in Belgium in 2022.

Nombre de tués, blessés graves, blessés légers et victimes indemnes d'accidents de la route, par classe d'âges, genre d'utilisateur, sexe et diverses caractéristiques de l'accident en Belgique en 2022.

These are publicly available data downloaded on 16 June 2023 from the Belgian government open data page. For more details on these data, see <https://statbel.fgov.be/fr/themes/mobilite/circulation/accidents-de-la-circulation>

Usage

```
road_accidents_be_2022.rds()  
road_accidents_be_meta.csv()
```

Examples

```
readRDS(road_accidents_be_2022.rds())  
read.csv(road_accidents_be_meta.csv())
```

`rWSBIM1207`*rWSBIM1207: Companion package for WSBIM1207 course*

Description

This package is used to distribute data and general documentation about the WSBIM1207 course, and to make course administration easier. For details about the course and the course material, see <http://bit.ly/WSBIM1207>.

Usage

```
rWSBIM1207version()
```

rWSBIM1207 datasets

The following data sets are available. Consult the respective manual pages for further details.

- tcga clinical and RNA expression data: see `?tcga`.
- beer consumption data: see `?beers`.
- tables to illustrate joins: see `jdf`.

Author(s)

Maintainer: Laurent Gatto <laurent.gatto@uclouvain.be> ([ORCID](#))

Examples

```
## check the package version that is currently installed  
rWSBIM1207version()
```

`tcga`*TCGA data*

Description

The Cancer Genome Atlas (TCGA) is a collaboration between the National Cancer Institute (NCI) and the National Human Genome Research Institute (NHGRI) that has generated multi-omics analyses (genomic, transcriptomic, proteomic and epigenetic) in 33 types of cancer. RNAseq and clinical data analysed here come from LUAD (lung adenocarcinoma) tumors and corresponding patients. TCGA clinical and RNAseq expression data extracted from the `curatedTCGAData` package. See `inst/scripts/tcga.R` for details.

Usage

```
data("expression")
data("clinical1")
data("clinical2")
data("clinical_table_ex1")
```

Format

expression: RNA expression data frame with 570 observations on the following 8 variables.

sampleID a factor
patient a character vector
type a character vector
A1BG a numeric vector
A1CF a numeric vector
A2BP1 a numeric vector
A2LD1 a numeric vector
A2ML1 a numeric vector

clinical1: clinical data for 516 observations on the following 15 variables.

patientID a character vector
tumor_tissue_site a character vector
gender a character vector
age_at_diagnosis a numeric vector
vital_status a numeric vector
days_to_death a numeric vector
days_to_last_followup a numeric vector
pathologic_stage a character vector
pathology_T_stage a character vector
pathology_N_stage a character vector
pathology_M_stage a character vector
smoking_history a character vector
number_pack_years_smoked a numeric vector
year_of_tobacco_smoking_onset a numeric vector
stopped_smoking_year a numeric vector

clinical2: small clinical data with 516 observations on the following 3 variables.

patientID a character vector
gender a character vector
years_at_diagnosis a numeric vector

A clinical summary data with 2 observations on the following 3 variables.

gender a character vector

'current smoker' a numeric vector

'lifelong non-smoker' a numeric vector

In addition, the `clinical1.csv` and `expression.csv` function return the paths to these respective comma-separated value spreadsheets. The `expressions.csv` function returns the path to the expression data split by gene.

Examples

```
data(expression)
data(clinical1)
data(clinical2)
data(clinical_table_ex1)
```

valentine

TidyTuesday's Valentine's Day Consumer Data

Description

These data, originally organized by Suraj Das for a Kaggle dataset, have been cleaned and prepared for the [TidyTuesday series](#).

The data contains three files:

- `historical_spending` contains Valentine's spending from 2010 to 2022.
- `gifts_age` contains information on gifts by age.
- `gifts_gender` contains information on gifts by gender.

For a description of the variables of these table, see the link above.

Note that in these historical surveys, gender was collected as only 'Men' and 'Women', which does not accurately include all genders.

Usage

```
valentine()
```

Value

character(3) of urls to file names.

Author(s)

Laurent Gatto

Examples

```
valentine()
```

Index

* datasets

- apple_mobility, 2
 - beers, 3
 - covid19, 4
 - interro, 7
 - jdf, 7
 - kem, 8
 - mvyIng, 10
 - peptides, 10
 - population, 11
 - road_accidents, 11
 - tcga, 12
- apple_mobility, 2
- beers, 3
- beers2.csv (beers), 3
- clinical1 (tcga), 12
- clinical2 (tcga), 12
- clinical_table_ex1 (tcga), 12
- covid19, 4
- covid19_cases.csv (covid19), 4
- covid19_deaths.csv (covid19), 4
- covid19_recovered.csv (covid19), 4
- english_education
(english_education_files), 4
- english_education_files, 4
- expression (tcga), 12
- expressions.csv (tcga), 12
- faillites (faillites_be), 5
- faillites_be, 5
- gifts_age (valentine), 14
- gifts_gender (valentine), 14
- GSE96870, 6
- GSE96870_intro (GSE96870), 6
- GSE96870_intro_ranges (GSE96870), 6
- historical_spending (valentine), 14
- interro, 7
- interro2.rds (interro), 7
- interroA (interro), 7
- interroB (interro), 7
- interroC (interro), 7
- interroL (interro), 7
- jdf, 7
- jdf1 (jdf), 7
- jdf2 (jdf), 7
- jdf3 (jdf), 7
- jdf4 (jdf), 7
- jdf5 (jdf), 7
- jdf6 (jdf), 7
- jdf7 (jdf), 7
- kem, 8
- kem2.tsv (kem), 8
- kem3.tsv (kem), 8
- kem_annot (kem), 8
- kem_count (kem), 8
- load_exam_data, 9
- mvyIng, 10
- peptides, 10
- population, 11
- population_be.csv (population), 11
- road_accidents, 11
- road_accidents_be_2022.rds
(road_accidents), 11
- road_accidents_be_meta.csv
(road_accidents), 11
- rWSBIM1207, 12
- rWSBIM1207-package (rWSBIM1207), 12
- rWSBIM1207version (rWSBIM1207), 12

tcga, [12](#)

valentin (valentine), [14](#)

valentine, [14](#)