

Package ‘OIdata’

July 6, 2012

Type Package

Title Data sets and supplements (OpenIntro)

Version 1.0

Date 2011-11-03

Author Andrew P Bray and David M Diez

Maintainer Andrew P Bray <andrew@openintro.org>

Description A collection of data sets from several sources that may be useful for teaching, practice, or other purposes. Functions have also been included to assist in the retrieval of table data from websites or in visualizing sample data.

License GPL-2 | GPL-3

LazyLoad yes

Depends RCurl,maps,R (>= 2.10)

Imports RCurl,maps

URL <http://www.openintro.org>

Collate 'OIdata-internal.R''getTables.R''mapvar.R''print.htmlTables.R''processTable.R''rowScrape.R''scrapeTable.R''stripTags.R'

R topics documented:

OIdata-package	2
birds	3
esi	4
getTables	6
ipo	6
london_boroughs	8
mapvar	9
military	10
murders	11
piracy	12
processTable	13
sp500	14
state	15
teacher	17
ukDemo	18

Index**19**

OIdata-package

*Data sets and supplements (OpenIntro)***Description**

A collection of data sets from several sources that may be useful for teaching, practice, or other purposes. Functions have also been included to assist in the retrieval of table data from websites or in visualizing sample data.

Details

Package:	OIdata
Type:	Package
Version:	1.0
Date:	2011-11-03
License:	GPL-2 GPL-3
LazyLoad:	yes

Author(s)

Andrew P Bray and David M Diez

Maintainer: Andrew P Bray <andrew@openintro.org>

References

OpenIntro openintro.org

Examples

```
data(military)
(tabFM <- table(military$branch, military$gender) /
  matrix(rep(table(military$branch), 2), ncol = 2))
barplot(tabFM[,1], main = "proportion female")

data(piracy)
pipa <- piracy[piracy$chamber == "senate",]
keep <- pipa$money_pro > 0 & pipa$money_con > 0
for_pipa <- pipa$stance[keep] == "yes"
col <- ifelse(for_pipa, 2, 1)
pch <- ifelse(for_pipa, 20, 1)
plot(pipa$money_pro[keep], pipa$money_con[keep],
  log="xy", col=col, pch=pch)
```

birds

*Aircraft-Wildlife Collisions***Description**

A collection of all collisions between aircraft in wildlife that were reported to the US Federal Aviation Administration between 1990 and 1997, with details on the circumstances of the collision.

Usage

```
data(birds)
```

Format

A data frame with 19302 observations on the following 17 variables.

`opid` Three letter identification code for the operator (carrier) of the aircraft.

`operator` Name of the aircraft operator.

`atype` Make and model of aircraft.

`remarks` Verbal remarks regarding the collision.

`phase_of_flight` Phase of the flight during which the collision occurred: Approach, Climb, Descent, En Route, Landing Roll, Parked, Take-off run, Taxi.

`ac_mass` Mass of the aircraft classified as 2250 kg or less (1), 2251-5700 kg (2), 5701-27000 kg (3), 27001-272000 kg (4), above 272000 kg (5).

`num_engs` Number of engines on the aircraft.

`date` Date of the collision (MM/DD/YYYY).

`time_of_day` Light conditions: Dawn, Day, Dusk, Night.

`state` Two letter abbreviation of the US state in which the collision occurred.

`height` Feet above ground level.

`speed` Knots (indicated air speed).

`effect` Effect on flight: Aborted Take-off, Engine Shut Down, None, Other, Precautionary Landing.

`sky` Type of cloud cover, if any: No Cloud, Overcast, Some Cloud.

`species` Common name for bird or other wildlife.

`birds_seen` Number of birds/wildlife seen by pilot: 1, 2-10, 11-100, Over 100.

`birds_struck` Number of birds/wildlife struck: 0, 1, 2-10, 11-100, Over 100.

Details

The FAA National Wildlife Strike Database contains strike reports that are voluntarily reported to the FAA by pilots, airlines, airports and others. Current research indicates that only about 20% of strikes are reported. Wildlife strike reporting is not uniform as some organizations have more robust voluntary reporting procedures. Because of variations in reporting, users are cautioned that the comparisons between individual airports or airlines may be misleading.

Source

<http://opendata.socrata.com/Government/1990-1999-Reported-Aircraft-Wildlife-Bird-Strikes/inea-i3an>. Original date from faa.gov. Retrieval date: Feb 4, 2012.

References

OpenIntro, openintro.org

Examples

```
data(birds)
table(birds$phase_of_flight)
phase.ord <- sort(table(birds$phase_of_flight), decreasing = TRUE)
par(mar=c(6, 4, 4, 2) + 0.1)
barplot(phase.ord, las = 2)
summary(birds$height)

mosaicplot(birds$phase_of_flight ~ birds$effect, las = 2)
```

esi

Environmental Sustainability Index 2005

Description

This data set comes from the 2005 Environmental Sustainability Index: Benchmarking National Environmental Stewardship. Countries are given an overall sustainability score as well as scores in each of several different environmental areas.

ESI and Component scores are presented as standard normal percentiles. Indicator scores are in the form of z-scores. See Appendix A of the report for information on the methodology and Appendix C for more detail on original data sources.

Usage

```
data(esi)
```

Format

A data frame with 146 observations on the following 29 variables.

```
code ISO3 country code.
country Country.
esi Environmental Sustainability Index.
system ESI core component: systems
stress ESI core component: stresses
vulner ESI core component: vulnerability
cap ESI core component: capacity
global ESI core component: global stewardship
sys_air Air quality.
sys_bio Biodiversity.
sys_lan Land.
sys_wql Water quality.
sys_wqn Water quantity.
```

str_air Reducing air pollution.
str_eco Reducing ecosystem stress.
str_pop Reducing population pressure.
str_was Reducing waste and consumption pressures.
str_wat Reducing water stress.
str_nrm Natural resource management.
vul_heal Environmental health.
vul_sus Basic human sustenance.
vul_dis Exposure to natural disasters.
cap_gov Environmental governance.
cap_eff Eco-efficiency.
cap_pri Private sector responsiveness.
cap_st Science and technology.
glo_col Participation in international collaboration efforts.
glo_ghg Greenhouse gas emissions.
glo_tbp Reducing transboundary environmental pressures.

Details

For more information on how each of the indices were calculated, see the documentation linked below.

Source

ESI Component Indicators. *2005 Environmental Sustainability Index: Benchmarking National Environmental Stewardship*, Yale Center for Environmental Law and Policy, Yale University & Center for International Earth Science Information Network (CIESIN), Columbia University

In collaboration with: World Economic Forum, Geneva, Switzerland Joint Research Centre of the European Commission, Ispra, Italy

Available at <http://www.yale.edu/esi/>

References

Esty, Daniel C., Marc Levy, Tanja Srebotnjak, and Alexander de Sherbinin (2005). *2005 Environmental Sustainability Index: Benchmarking National Environmental Stewardship*. New Haven: Yale Center for Environmental Law and Policy

Examples

```
data(esi)

pch <- ifelse(esi$code == "USA", 19, 1)
plot(esi$cap_st, esi$glo_col, pch=pch)
plot(esi$vulner, esi$cap, pch=pch)
```

getTables	<i>Get the tables from a web page</i>
-----------	---------------------------------------

Description

Retrieve each of the tables in a web page. Tables that have tables nested within them will be ignored.

Usage

```
getTables(URL)
```

Arguments

URL	A character string of a URL, usually beginning with <code>http://</code>
-----	--------------------------------------------------------------------------

Value

A list, where each item is a table in the webpage.

Author(s)

David Diez, Andrew Bray

See Also

[processTable](#), [getURL](#)

Examples

```
## Not run:
URL    <- "http://en.wikipedia.org/wiki/Poverty_in_the_United_States"
Tables <- getTables(URL)
thisOne <- which(sapply(Tables, "[", 1, 1) == "Persons in Family Unit")
if(length(thisOne) > 0){
  Tables[[thisOne[1]]]
}

## End(Not run)
```

ipo	<i>Facebook, Google, and LinkedIn IPO filings</i>
-----	---------------------------------------------------

Description

On Feb 1st, 2011, Facebook Inc. filed an S-1 form with the Securities and Exchange Commission as part of their initial public offering (IPO). This dataset includes the text of that document as well as text from the IPOs of two competing companies: Google and LinkedIn.

Usage

```
data(ipo)
```

Format

The format is a list of three character vectors. Each vector contains the line-by-line text of the IPO Prospectus of Facebook, Google, and LinkedIn, respectively.

Details

Each of the three prospectuses is encoded in UTF-8 format and contains some non-word characters related to the layout of the original documents. For analysis on the words, it is recommended that the data be processed with packages such as `tm` and `stringr`. See example below.

Source

All IPO prospectuses are available from www.sec.gov: [Facebook](#), [Google](#), [LinkedIn](#).

References

<http://blogs.wsj.com/totalreturn/2012/02/06/mark-zuckerberg-ceo-for-life/>

Credit to Qian Liu at the [Wealthfront Blog](#) for the data links and wordcloud example below.

Examples

```
data(ipo)
## Not run:
# install.packages("tm")
# install.packages("wordcloud")
library(tm)
library(wordcloud)

# pre-process data
corp <- Corpus(VectorSource(ipo), readerControl=list(language="en"))
corp <- tm_map(corp, removePunctuation)
corp <- tm_map(corp, tolower)
corp <- tm_map(corp, removeNumbers)
corp <- tm_map(corp, function(x)removeWords(x,stopwords()))
f <- corp[1] # facebook
g <- corp[2] # google
l <- corp[3] # linkedin

tmat <- TermDocumentMatrix(f)
m <- as.matrix(tmat)
freq <- rowSums(m)
words <- rownames(m)
words.ord <- sort.int(freq, decreasing = T, index.return = F)
barplot(words.ord[1:15], las = 2)

wordcloud(words, freq, min.freq = 100, col='blue')

tmat <- TermDocumentMatrix(c(f, g))
m <- as.matrix(tmat)
comparison.cloud(m, max.words = 100)

## End(Not run)
```

london_boroughs	<i>London Borough Boundaries</i>
-----------------	----------------------------------

Description

This dataset contains the coordinates of the boundaries of all 32 boroughs of the Greater London area.

Usage

```
data(london_boroughs)
```

Format

A data frame with 45341 observations on the following 3 variables.

name Name of the borough.

x The "easting" component of the coordinate, see details.

y The "northing" component of the coordinate, see details.

Details

Map data was made available through the Ordnance Survey Open Data initiative. The data use the **National Grid** coordinate system, based upon eastings (x) and northings (y) instead of longitude and latitude.

The name variable covers all 32 boroughs in Greater London: Barking & Dagenham, Barnet, Bexley, Brent, Bromley, Camden, Croydon, Ealing, Enfield, Greenwich, Hackney, Hammersmith & Fulham, Haringey, Harrow, Havering, Hillingdon, Hounslow, Islington, Kensington & Chelsea, Kingston, Lambeth, Lewisham, Merton, Newham, Redbridge, Richmond, Southwark, Sutton, Tower Hamlets, Waltham Forest, Wandsworth, Westminster

Source

<https://www.ordnancesurvey.co.uk/opendatadownload/products.html>

Contains Ordnance Survey data, Crown copyright and database right [2012], used under the **Open Data License**.

References

OpenIntro, openintro.org

Examples

```
data(london_boroughs)
## Not run:
# install.packages("ggplot2")
# install.packages("RColorBrewer")
library(ggplot2)
library(RColorBrewer)
data(murders)
LB      <- london_boroughs
mtab    <- table(murders$borough)
```



```

LB$nmurders <- rep(mtab, rle(as.character(LB$name))$lengths)
p <- ggplot()
p +
  geom_polygon(data=LB, aes(x=x, y=y, group = name, fill = nmurders),
               colour="white" ) +
  scale_fill_gradientn(colours = brewer.pal(7, "Blues"),
                       limits=range(LB$nmurders))

## End(Not run)

```

mapvar

Map a state-level numerical variable

Description

Generate a map of a state-level numerical variable

Usage

```
mapvar(val, abbr, col = 1:3, Legend="")
```

Arguments

val	A vector of numerical values.
abbr	The state abbreviations that correspond to each of the numerical values.
col	A numerical vector of length 3 with positive entries. The numbers correspond to weightings of red, green, and blue in the final map.
Legend	Legend name as a character string.

Details

Relies heavily on the [map](#) function. See the unemployment example for sample code that may help explain how the mapvar function works.

Value

A data frame of the [state.fips](#) data set together with a standardized form of the values.

Author(s)

David Diez

References

OpenIntro, openintro.org

See Also

[state](#)

Examples

```
data(state)
mapvar(state$unempl, state$abbr, 3:1, Legend="Unemployment (%)")

mapvar(state$soc_sec, state$abbr, 1:3, Legend="Social Security (%)")

DC      <- state$state == "District of Columbia"
murder  <- state$murder
murder[DC] <- NA
mapvar(murder, state$abbr, c(3,2,2), Legend="Murders per 100k")

mapvar(state$nuclear, state$abbr, c(2,3,2), Legend="Nuclear Energy (%)")
```

military

*US Military Demographics***Description**

This dataset contains demographic information on every member of the US armed forces including gender, race, and rank.

Usage

```
data(military)
```

Format

A data frame with 1414593 observations on the following 6 variables.

`grade` The status of the service member as enlisted officer or warrant officer.

`branch` The branch of the armed forces: air force, army, marine corps, navy.

`gender` Whether the service member is female or male.

`race` The race identified by the service member: `ami/aln` (american indian/alaskan native), `asian`, `black`, `multi` (multi-ethnic), `p/i` (pacific islander), `unk` (unknown), or `white`.

`hisp` Whether a service member identifies with being hispanic (TRUE) or not (FALSE).

`rank` The numeric rank of the service member (higher number indicates higher rank).

Details

The branches covered by this data set include the Army, Navy, Air Force, and Marine Corps. Demographic information on the Coast Guard is contained in the original data set but has not been included here.

Source

Data provided by the Department of Defense and made available at data.gov. Retrieval date: Feb 20, 2012.

References

OpenIntro, openintro.org

Examples

```

data(military)
cond1 <- military$grade == "officer"
cond2 <- military$branch == "army"
cond3 <- military$gender == "female"
tabF <- table(military$rank[cond1 & cond2 & cond3])
barplot((tabF/sum(tabF)), main = "female army officer ranks")
tabM <- table(military$rank[cond1 & cond2 & !cond3])
barplot(tabM/sum(tabM), main = "male army officer ranks")

(tabFM <- table(military$branch, military$gender) /
  matrix(rep(table(military$branch), 2), ncol = 2))
barplot(tabFM[,1], main = "proportion female")

```

murders

*London Murders, 2006-2011***Description**

This dataset contains the victim name, age, and location of every murder recorded in the Greater London area by the Metropolitan Police from January 1, 2006 to September 7, 2011.

Usage

```
data(murders)
```

Format

A data frame with 838 observations on the following 5 variables.

forename First name(s) of the victim.

age Age of the victim.

date Date of the murder (MM/DD/YY).

year Year of the murder.

borough The London borough in which the murder took place. See the Details section for a list of all the boroughs.

Details

To visualize this data set using a map, see the [london_boroughs](#) dataset, which contains the latitude and longitude of polygons that define the boundaries of the 32 boroughs of Greater London.

The borough variable covers all 32 boroughs in Greater London: Barking & Dagenham, Barnet, Bexley, Brent, Bromley, Camden, Croydon, Ealing, Enfield, Greenwich, Hackney, Hammersmith & Fulham, Haringey, Harrow, Havering, Hillingdon, Hounslow, Islington, Kensington & Chelsea, Kingston, Lambeth, Lewisham, Merton, Newham, Redbridge, Richmond, Southwark, Sutton, Tower Hamlets, Waltham Forest, Wandsworth, Westminster

Source

http://www.met.police.uk/foi/pdfs/disclosure_2011/september/2011080003610.pdf

References

Inspired by [The Guardian Datablog](#).

Examples

```
data(murders)
date1 <- as.Date(murders$date, "%m/%d/%y")
date2 <- julian(date1, origin = as.Date("2006-01-02"))
date_cut <- cut(date2, seq(0, 2073, 90))
barplot(table(date_cut), xaxt = "n",
        xlab = "date from 01/2006 - 09/2011",
        ylab = "n deaths per 90 days")

## Not run:
# install.packages("ggplot2")
# install.packages("RColorBrewer")
library(ggplot2)
library(RColorBrewer)
data(london_boroughs)
LB <- london_boroughs
mtab <- table(murders$borough)
LB$nmurders <- rep(mtab, rle(as.character(LB$name))$lengths)
p <- ggplot()
p +
  geom_polygon(data=LB, aes(x=x, y=y, group = name, fill = nmurders),
              colour="white" ) +
  scale_fill_gradientn(colours = brewer.pal(7, "Blues"),
                      limits=range(LB$nmurders))

## End(Not run)
```

piracy

Piracy and PIPA/SOPA

Description

This data set contains observations on all 100 US Senators and 434 of the 325 US Congressional Representatives related to their support of anti-piracy legislation that was introduced at the end of 2011.

Usage

```
data(piracy)
```

Format

A data frame with 534 observations on the following 8 variables.

name Name of legislator.

party Party affiliation as democrat (D), Republican (R), or Independent (I).

state Two letter state abbreviation.

money_pro Amount of money in dollars contributed to the legislator's campaign in 2010 by groups generally thought to be supportive of PIPA/SOPA: movie and TV studios, record labels.

money_con Amount of money in dollars contributed to the legislator's campaign in 2010 by groups generally thought to be opposed to PIPA/SOPA: computer and internet companies.

years Number of years of service in Congress.

stance Degree of support for PIPA/SOPA with levels Leaning No, No, Undecided, Unknown, Yes

chamber Whether the legislator is a member of either the house or senate.

Details

The Stop Online Piracy Act (SOPA) and the Protect Intellectual Property Act (PIPA) were two bills introduced in the US House of Representatives and the US Senate, respectively, to curtail copyright infringement. The bill was controversial because there were concerns the bill limited free speech rights. ProPublica, the independent and non-profit news organization, compiled this data set to compare the stance of legislators towards the bills with the amount of campaign funds that they received from groups considered to be supportive of or in opposition to the legislation.

For more background on the legislation and the formulation of money_pro and money_con, read the documentation on ProPublica, linked below.

Source

<http://projects.propublica.org/sopa/>

The list may be slightly out of date since many politician's perspectives on the legislation were in flux at the time of data collection.

References

OpenIntro, openintro.org

Examples

```
data(piracy)

pipa <- piracy[piracy$chamber == "senate",]
tab <- aggregate(money_pro ~ stance, data = pipa, mean)
barplot(tab$money_pro, names.arg = tab$stance)
boxplot(money_pro ~ stance, data = pipa)
boxplot(money_con ~ stance, data = pipa)

keep <- pipa$money_pro > 0 & pipa$money_con > 0
for_pipa <- pipa$stance[keep] == "yes"
col <- ifelse(for_pipa, 2, 1)
pch <- ifelse(for_pipa, 20, 1)
plot(pipa$money_pro[keep], pipa$money_con[keep],
     log="xy", col=col, pch=pch)
```

processTable

Process selected columns of a table into numbers

Description

Given a table where some columns represent numeric values but are cluttered with info like dollar signs, percent signs, among other symbols, this function cleans up those columns and converts the entire table into a data frame.

Usage

```
processTable(Table, num = NULL, Names = NULL)
```

Arguments

Table	A table of character entries, usually an item from the list of tables returned by getTables .
num	Column numbers for the numerical columns.
Names	Column names to add to the final data frame.

Value

A data frame.

Author(s)

David M Diez

References

OpenIntro (<http://www.openintro.org/>)

See Also

[getTables](#)

Examples

```
## Not run:
URL    <- "http://en.wikipedia.org/wiki/Poverty_in_the_United_States"
Tables <- getTables(URL)
thisOne <- which(sapply(Tables, "[", 1, 1) == "Persons in Family Unit")
if(length(thisOne) > 0){
  temp <- Tables[[thisOne[1]]]
  processTable(temp[-c(1),], 1:4)
}
# warning may be produced for one entry
# "each additional person adds"

## End(Not run)
```

sp500

Financial information for 50 S&P 500 companies

Description

Fifty companies were randomly sampled from the 500 companies in the S&P 500, and their financial information was collected on March 8, 2012.

Usage

```
data(sp500)
```

Format

A data frame with 50 observations on the following 12 variables.

`market_cap` Total value of all company shares, in millions of dollars.

`stock` The name of the stock (e.g. AAPL for Apple).

`ent_value` Enterprise value, which is an alternative to market cap that also accounts for things like cash and debt, in millions of dollars.

`trail_pe` The market cap divided by the earnings (profits) over the last year.

`forward_pe` The market cap divided by the forecasted earnings (profits) over the next year.

`ev_over_rev` Enterprise value divided by the company's revenue.

`profit_margin` Percent of earnings that are profits.

`revenue` Revenue, in millions of dollars.

`growth` Quarterly revenue growth (year over year), in millions of dollars.

`earn_before` Earnings before interest, taxes, depreciation, and amortization, in millions of dollars.

`cash` Total cash, in millions of dollars.

`debt` Total debt, in millions of dollars.

Source

Yahoo! Finance, finance.yahoo.com, collected March 8, 2012.

References

OpenIntro, openintro.org

Examples

```
data(sp500)
plot(sp500$ent_value, sp500$earn_before)

plot(sp500$ev_over_rev, sp500$forward_pe)

plot(sp500$ent_value, sp500$earn_before, log="xy")

plot(sp500$ev_over_rev, sp500$forward_pe, log="xy")
```

state	<i>State-level data</i>
-------	-------------------------

Description

Information about each state collected from both the official US Census website and from various other sources.

Usage

```
data(state)
```

Format

A data frame with 51 observations on the following 23 variables.

state State name.
 abbr State abbreviation (e.g. "MN").
 fips FIPS code.
 pop2010 Population in 2010.
 pop2000 Population in 2000.
 homeownership Homeownership rate.
 multiunit Percent of living units that are in multi-unit structures.
 income Average income per capita.
 med_income Median household income.
 poverty Poverty rate.
 fed_spend Federal spending per capita.
 land_area Land area.
 smoke Percent of population that smokes.
 murder Murders per 100,000 people.
 robbery Robberies per 100,000.
 agg_assault Aggravated assaults per 100,000.
 larceny Larcenies per 100,000.
 motor_theft Vehicle theft per 100,000.
 soc_sec Percent of individuals collecting social security.
 nuclear Percent of power coming from nuclear sources.
 coal Percent of power coming from coal sources.
 tr_deaths Traffic deaths per 100,000.
 tr_deaths_no_alc Traffic deaths per 100,000 where alcohol was not a factor.
 unempl Unemployment rate (February 2012, preliminary).

Source

Please note that we have not validated the Infochimps data. The other data were collected directly from the corresponding websites.

US Census website (pop2010, pop2000, homeownership, multiunits, income, med_income, poverty, fed_spend, land_area), <http://quickfacts.census.gov/qfd/index.html>

Infochimps (murder, robbery, agg_assault, larcent, motor_theft, soc_sec, coal, nuclear), <http://www.infochimps.com> (see linked pages and corresponding downloadable data sets for original source information)

National Highway Traffic Safety Administration (tr_deaths, tr_deaths_no_alc), <http://www-fars.nhtsa.dot.gov/>

Bureau of Labor Statistics (unempl), <http://www.bls.gov/web/laus/laumstrk.htm>

References

OpenIntro, openintro.org

Examples

```
data(state)
mapvar(state$unempl, state$abbr, 3:1, Legend="Unemployment (%)")

mapvar(state$soc_sec, state$abbr, 1:3, Legend="Social Security (%)")

DC      <- state$state == "District of Columbia"
murder  <- state$murder
murder[DC] <- NA
mapvar(murder, state$abbr, c(3,2,2), Legend="Murders per 100k")

mapvar(state$nuclear, state$abbr, c(2,3,2), Legend="Nuclear Energy (%)")
```

teacher

Teacher Salaries in St. Louis, Michigan

Description

This data set contains teacher salaries from 2009-2010 for 71 teachers employed by the St. Louis Public School in Michigan, as well as several covariates.

Usage

```
data(teacher)
```

Format

A data frame with 71 observations on the following 8 variables.

id Identification code for each teacher, assigned randomly.

degree Highest educational degree attained: BA (bachelor's degree) or MA (master's degree).

fte Full-time enrollment status: full-time 1 or part-time 0.5.

years Number of years employed by the school district.

base Base annual salary, in dollars.

fica Amount paid into Social Security and Medicare per year through the Federal Insurance Contribution Act (FICA), in dollars.

retirement Amount paid into the retirement fund of the teacher per year, in dollars.

total Total annual salary of the teacher, resulting from the sum of base salary + fica + retirement, in dollars.

Source

Posted on opendata.socrata.com by Jeff Kowalski.
Original source: <http://stlouis.edzone.net>

References

OpenIntro, openintro.org

Examples

```
data(teacher)
boxplot(base ~ degree, teacher)

plot(base ~ years, teacher, pch = 16)
cond1 <- teacher$degree == "MA"
points(teacher$years[cond1], teacher$base[cond1],
       pch = 16, col = "orange")
```

ukDemo

*United Kingdom Demographic Data***Description**

This data set comes from the Guardian's Data Blog and includes five financial demographic variables.

Usage

```
data(ukDemo)
```

Format

A data frame with 12 observations on the following 6 variables.

region Region in the United Kingdom

debt Average regional debt, not including mortgages, in pounds

unemployment Percent unemployment

house Average house price, in pounds

pay Average hourly pay, in pounds

rpi Retail price index, which is standardized to 100 for the entire UK, and lower index scores correspond to lower prices

Source

The data was described in the Guardian Data Blog:

<http://www.guardian.co.uk/news/datablog/interactive/2011/oct/27/debt-money-expert-facts>

Posted: Thursday, October 27, 2011

Retrieved: Tuesday, November 1, 2011

References

Guardian Data Blog

Examples

```
data(ukDemo)
plot(ukDemo$pay, ukDemo$rpi)
```

Index

- *Topic **Congress**
 - piracy, [12](#)
- *Topic **London**
 - london_boroughs, [8](#)
 - murders, [11](#)
- *Topic **Michigan**
 - teacher, [17](#)
- *Topic **PIPA**
 - piracy, [12](#)
- *Topic **SOPA**
 - piracy, [12](#)
- *Topic **SP 500**
 - sp500, [14](#)
- *Topic **United States**
 - mapvar, [9](#)
 - state, [15](#)
- *Topic **airplane**
 - birds, [3](#)
- *Topic **bird**
 - birds, [3](#)
- *Topic **borough**
 - london_boroughs, [8](#)
- *Topic **clean data**
 - processTable, [13](#)
- *Topic **copyright infringement**
 - piracy, [12](#)
- *Topic **corpus**
 - ipo, [6](#)
- *Topic **country**
 - esi, [4](#)
- *Topic **crime**
 - murders, [11](#)
 - state, [15](#)
- *Topic **data frame**
 - processTable, [13](#)
- *Topic **data processing**
 - processTable, [13](#)
- *Topic **datasets**
 - birds, [3](#)
 - esi, [4](#)
 - ipo, [6](#)
 - london_boroughs, [8](#)
 - military, [10](#)
 - murders, [11](#)
 - piracy, [12](#)
 - sp500, [14](#)
 - state, [15](#)
 - teacher, [17](#)
 - ukDemo, [18](#)
- *Topic **degree**
 - teacher, [17](#)
- *Topic **demographics**
 - military, [10](#)
- *Topic **education**
 - teacher, [17](#)
- *Topic **efficiency**
 - esi, [4](#)
- *Topic **energy**
 - esi, [4](#)
 - state, [15](#)
- *Topic **environment**
 - esi, [4](#)
- *Topic **financial**
 - sp500, [14](#)
- *Topic **flight**
 - birds, [3](#)
- *Topic **ipo**
 - ipo, [6](#)
- *Topic **legislation**
 - piracy, [12](#)
- *Topic **map**
 - london_boroughs, [8](#)
 - mapvar, [9](#)
 - murders, [11](#)
- *Topic **military**
 - military, [10](#)
- *Topic **money**
 - sp500, [14](#)
- *Topic **murder**
 - murders, [11](#)
- *Topic **package**
 - OIdata-package, [2](#)
- *Topic **regular expressions**
 - processTable, [13](#)
- *Topic **salary**
 - teacher, [17](#)

- *Topic **scrape**
 - getTables, 6
- *Topic **state**
 - state, 15
- *Topic **stocks**
 - sp500, 14
- *Topic **sustainability**
 - esi, 4
- *Topic **teacher**
 - teacher, 17
- *Topic **text mining**
 - ipo, 6
- *Topic **web**
 - getTables, 6
- *Topic **wildlife**
 - birds, 3

birds, 3

esi, 4

getTables, 6, 14

getURL, 6

ipo, 6

london_boroughs, 8, 11

map, 9

mapvar, 9

military, 10

murders, 11

OIdata (OIdata-package), 2

OIdata-package, 2

piracy, 12

print.htmlTables (getTables), 6

processTable, 6, 13

rowScrape (getTables), 6

scrapeTable (getTables), 6

sp500, 14

state, 9, 15

state.fips, 9

stripTags (getTables), 6

teacher, 17

ukDemo, 18