Package 'elections'

May 8, 2026

Version 1.0.1

Date 2018-03-02

Title USA Presidential Elections Data

Author Willem M. van der Wal <willem@vanderwalresearch.com>

Maintainer Willem M. van der Wal <willem@vanderwalresearch.com>

Depends R (>= 3.0.0)

Description

This includes a dataset on the outcomes of the USA presidential elections since 1920, and various predictors, as used in <https://www.vanderwalresearch.com/blog/15-elections>.

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2023-12-11 08:22:38 UTC

Contents

eldat 1

Index **4**

`eldat` *Outcomes of the USA presidential elections since 1920, and various predictors*

Description

This is a dataset with the outcomes of the USA presidential elections since 1920. I have used this dataset in my blog describing predictive models for the 2020 election. The data include not only the winner and loser of each election, but also the popular vote margin, turnout and information on the development of the Dow Jones index and the per capita disposable income in the four years before each election. Willem M. van der Wal, PhD (vanderwalresearch.com).

Usage

```
data(eldat)
```

Format

A data frame with observations on the following variables:

`electionyear` Calendar year in which the election was held.

`presel.Date` Date at which the election was held.

`winner` Name of the winner.

`winnerparty` Party of the winner.

`winnerparty.tmin1` Party of the winner, one election earlier.

`winnerparty.tmin2` Party of the winner, two elections earlier.

`winnerparty.tmin3` Party of the winner, three elections earlier.

`winnerparty.tmin4` Party of the winner, four elections earlier.

`runnerup` Name of the runner up.

`runnerupparty` Party of the runner up.

`popvotepercmargin.rep` Popular vote margin (%) of the republican party as compared to the democratic party.

`popvotepercmargin.rep.tmin1` Popular vote margin (%) of the republican party as compared to the democratic party, one election earlier.

`turnoutperc` Turnout (%).

`turnoutperc.tmin1` Turnout (%), one election earlier.

`djia.reldiff` The relative change (%) of the Dow Jones index in the four years before the election.

`dispincome` Per capita disposable income (2009 dollars) in the calendar year of the election.

`dispincchange` Relative change (%) of the per capita disposable income over the four years before the election.

Details

The "tmin..." variables, djia.reldiff and dispincchange could be used as possible predictors in models that predict the outcome of the election.

Author(s)

Willem M. van der Wal <willem@vanderwalresearch.com>, vanderwalresearch.com.

References

I used the following sources for these data: Complete List Of All The Presidents Of The United States, List of Presidents of the United States, List of United States presidential elections by popular vote margin, Dow Jones Industrial Average and Bureau of Economic Analysis - National Data - GDP & Personal Income.

Examples

```
#Example 1: fit model for probability that the winner is a republican,
#using only the outcomes of the last two elections.

#Load data
data(eldat)

#Fit model for probability that the winner is a republican
elmod <- glm(winnerparty == "Rep." ~ winnerparty.tmin1*winnerparty.tmin2,
data = eldat, family = binomial(link = logit))
summary(elmod)
#ok, coefficients clearly illustrate "pendulum" effect,
#don't mind the p-values because of small sample size

#Prediction from elmod, with cutoff 0.5
eldat$p.elmod <- predict.glm(elmod, type = "response") #predicted probability
eldat$pred.elmod <- ifelse(eldat$p.elmod > 0.5, "Rep.", "Dem.") #predicted outcome
with(eldat, table(pred.elmod, winnerparty)) #crosstable
100*sum(with(eldat, winnerparty == pred.elmod))/nrow(eldat) #% correctly predicted
#76% correct
#indicator wrong/right prediction
eldat$ind.elmod <- with(eldat, ifelse(winnerparty == pred.elmod, "OK", "WRONG!"))
#show prediction
eldat[, c("electionyear", "winner", "winnerparty", "pred.elmod", "p.elmod", "ind.elmod")]

#25-fold crossvalidation with 1-24 split
#(leave out one, fit model, predict for the observation left out)
eldat$p.elmod.CV <- NA #predicted cross-validated probability (first fill with NAs)
for(i in 1:25){
tempmod <- glm(winnerparty == "Rep." ~ winnerparty.tmin1*winnerparty.tmin2,
data = eldat[-i,], family = binomial(link = logit)) #fit model on training data
eldat$p.elmod.CV[i] <- predict.glm(tempmod, type = "response", newdata = eldat[i,])[[1]]
#predicted probability for test data
}

#Evaluate the predictions from the crossvalidation
eldat$pred.elmod.CV <- ifelse(eldat$p.elmod.CV > 0.5, "Rep.", "Dem.") #predicted outcome
with(eldat, table(pred.elmod.CV, winnerparty)) #crosstable
100*sum(with(eldat, winnerparty == pred.elmod.CV))/nrow(eldat) #% correctly predicted
#still 76% correct
eldat$ind.elmod.CV <- with(eldat, ifelse(winnerparty == pred.elmod.CV, "OK", "WRONG!"))
eldat[,c("electionyear", "winner", "winnerparty", "pred.elmod.CV", "p.elmod.CV", "ind.elmod.CV")]

#Overview
100*sum(with(eldat, winnerparty == pred.elmod))/nrow(eldat) #Without CV: 76% correct
100*sum(with(eldat, winnerparty == pred.elmod.CV))/nrow(eldat) #With CV: 76% correct
```

Index

∗ **datasets**
eldat, 1

eldat, 1