Package 'dogesr'

June 25, 2025

Contents

---

data.doges   _Load data into the environment_

Description

Load `data.doges` into the environment

Usage

```
data("doges")
```

Value

A dataframe with a row for every doge and doge marriage, and the columns

- `Doge` Full name of the doge.
- `Dogaressa` Full name of the dogaressa (wife of the doge).
- `Doge.raw` Full entry copied from the Wikipedia, original format; includes years of rule.
- `Dogaressa.raw` Full entry copied from the Wikipedia, original format. Years of marriage are include when known; in other cases, they are simply the same as the years of ruling.
- `Century, Start, End, Years` Century where the office of the doge took place, years it started and ended, and how many years it lasted, parsed from `Doge.raw`.
- `Family.doge, Family.dogaressa` Normalized names of the patrician family the doge and dogaressa belonged. The second is null if it was not a patrician family (usual in the first centuries).
- `Family.mother` The family name of the mother of the doge, extracted generally from the Wikipedia

Examples

```
library(dogesr)
data("doges")
# A summary of the duration of the doges ruling
summary(data.doges$Years)

# The families that actually "made doge"
unique(data.doges$Family.doge)

# Families that had either doge or dogaresse
unique( c(data.doges$Family.doge,data.doges$Family.dogaressa))
```

---

doge.families   *Load the list of families that became doges, and their numbers*

Description

Load doge.families into the environment

Usage

```
data("families")
```

Value

A table with two columns:

- Family.doge Name of the family

- n Number of times this family "made" doge

Examples

```
library(dogesr)
data("doge.families")

# How many times did the Dandolos became doge?
doge.families[ doge.families$Family.doge == "Dandolo",]$n

# How many families were doges?
length(doge.families$Family.doge )
```

---

doges.marriages.sn   *Data on doges' (and parents) matrimonial links*

Description

An 'igraph' object that includes doges' marriages, as well as their fathers', when available.

Usage

```
data(doges.marriages)
```

Value

An 'igraph' objects, with vertices corresponding to dogi/dogaresse families, edges (links) corresponding to recorded marriages eigher by doges or by their parents, as shown in the sources. Plese note that this dataset includes self-loops, corresponding to doges that married within their own family.

Every node has two attributes

- `family.type` Traditional "type" of the family, depending on how it became noble.
- `family.type.color` Color codes for every type of family, for easy plotting

Note

Data originally from the Wikipedia

Author(s)

J. J. Merelo

References

There are no references for Rd macro \insertAllCites on this help page.

See Also

link{doges}

Examples

```
library(dogesr)
library(igraph)
data(doges.marriages)

# All families linked to the Contarinis
incident(doges.marriages.sn,as.numeric(V(doges.marriages.sn)["Contarini"]))

# Plot graph, with colors depending on the type of family
plot(doges.marriages.sn,vertex.color=V(doges.marriages.sn)$family.type.color)

# Show self-loops in the dataset
print(E(doges.marriages.sn)[which_loop(doges.marriages.sn)])
```

---

doges.years   *Sub-dataset with the list of doges, their family and when it happened.*

Description

A dataset with doge data and the years their reign started and ended.

Usage

```
data("doges.years")
```

Details

It's essentially the original dataset doges Merelo-Guervós JJ (2022). "What is a good doge? Analyzing the patrician social network of the Republic of Venice." University of Granada. doi:10.48550/ARXIV.2209.07334, https://arxiv.org/abs/2209.07334., minus data referring to dogaresse, and leaves just data for the doges, eliminating also the "raw" columns.

Value

A dataframe with the columns Doge, Century, Start, End, Family, Years

Note

Data originally from the Wikipedia

Author(s)

J. J. Merelo

References

There are no references for Rd macro \insertAllCites on this help page.

See Also

```
link{doges}
```

Examples

```
library(dogesr)
data("doges.years")
summary(doges.years$Years)
```

---

family.colors   *Pre-assigned colors for every type of family*

Description

Load `family.colors` into the environment, assigning a color to every type of family.

Usage

```
data("family.colors")
```

Value

List that assigns a fixed color for every type of family

- Key: family type as in `family.type`.
- Value: color chosen more or less related to the type.

Examples

```
library(dogesr)
data("family.types")
data("family.colors")

# Which color corresponds to the type of the Dandolo family?
family.colors[[family.types[["Dandolo"]]]]
```

---

family.types   *Load data for Venetian family types into the environment*

Description

Load `family.types` into the environment

Usage

```
data("families")
```

Value

A list with every noble family in the republic of Venice, organized as

- Key: family name.
- Value: type of family: Estinte, Vecchie, Apostoliche, Evangeliche, Ducali, Nuove, Nuovis-sime, Soldo; this last name is not standard, and simply describe those who paid to be included into the Maggior Consiglio. This describes how they accessed nobility.

Main design decision here is that this can be used as external index for the type of family.

Examples

```
library(dogesr)
data("families")

# Which type was the Dandolo family?
family.types[["Dandolo"]]

# Which families bought their way into the nobility
family.types == "Soldo"

# The families that actually "made doge"
unique(data.doges$Family.doge)

# And their types
family.types[unique(data.doges$Family.doge)]

# Families that had either doge or dogaresse
unique( c(data.doges$Family.doge,data.doges$Family.dogaressa))
```

---

marriage.graph *Data on doges' matrimonial links*

Description

Matrimonial links as an `igraph` object; it includes doges' marriages, as well as their fathers', when available.

Usage

```
marriage.graph()
```

Details

The result of calling the function is an unfiltered `igraph` object, which you can use to plot the doges social network. Nodes are families, or "casate", and links indicate a wedding has taken place between the two families. Some attributes are added to the nodes, with a color code for easy plotting.

Value

A list that can be used as an igraph object, with noble family names as vertices and edges indicating links

Note

Data originally from the Wikipedia

Author(s)

J. J. Merelo

References

There are no references for Rd macro \insertAllCites on this help page.

See Also

link{doges}

Examples

```
library(dogesr)
library(igraph)
all.matrimonial.links <- marriage.graph()

# All families linked to the Contarinis
incident(all.matrimonial.links,as.numeric(V(all.matrimonial.links)["Contarini"]))
```

---

marriage.graph.slice   *Data on doges' matrimonial links*

Description

Partial graph of doge marriages as an igraph object; it includes doges' marriages, as well as their fathers', when available, but only for the period comprised between the two doges indicated

Usage

```
marriage.graph.slice(...,from="Orso",to="Ludovico Manin")
```

Arguments

...	Not really used
from	A doge name, the first by default
to	Another doge name, the last by default

Details

The result of calling the function is an igraph object, which you can use to plot the doges social network. Nodes are families, or "casate", and links indicate a wedding has taken place between the two families. The doges not comprised between the two slices, or the beginning or end and the indicated last or first, are not included.

Value

A list that can be used as an `igraph` object, with noble family names as vertices and edges indicating links

Note

Data originally from the Wikipedia

Author(s)

J. J. Merelo

References

There are no references for Rd macro \insertAllCites on this help page.

See Also

`link{doges}` `link{marriage.graph}`

Examples

```
library(dogesr)
library(igraph)

# From that one to the end
from.sagredo <- marriage.graph.slice( from="Nicolo Sagredo")

# From the first one to Sagredo
to.sagredo <- marriage.graph.slice( to="Nicolo Sagredo")

# From the first to the last Contarini
from.sagredo <- marriage.graph.slice(
                from="Domenico I Contarini",
                to="Luigi Contarini"
              )
```

Index