Package: auspol (via r-universe)

August 25, 2024

Title Australian Federal Election Results (2004-2022)

Version 0.0.1.0004

Description Retrieve Australian Federal Election results for House of Representatives and Senate, from 2004 onwards.

URL https://carlosyanez.github.io/auspol/

License MIT + file LICENSE

BugReports https://github.com/carlosyanez/auspol/issues

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Imports arrow, dplyr, forcats, fs, ggpackets, ggplot2, ggalluvial, ggrepel, methods, piggyback, RColorBrewer, rlang, stringr, tibble, tidyr, tidyselect, utils, zip

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Collate 'cache_management.R' 'internal.R' 'house_get_data.R' 'house_preferences_data.R' 'house_preferences_plots.R' 'house_primary_vote_summary_data.R' 'house_primary_vote_summary_plots.R' 'house_results_plots.R' 'lists.R' 'onLoad.R' 'plotting.R'

Config/testthat/edition 3

Repository https://carlosyanez.r-universe.dev

RemoteUrl https://github.com/carlosyanez/auspol

RemoteRef HEAD

RemoteSha a22f740293331d50933b77c85ac4a2ad42061e0a

Contents

`auspol_theme` *Helper function generate colour palette*

Description

Helper function generate colour palette

Usage

```
auspol_theme(
  p,
  type = c("colour", "fill"),
  extra_colours = NULL,
  extra_values = NULL,
  coord_flip = FALSE,
  palette = NULL,
  legend_pos = "none"
)
```

Arguments

p	ggplot object
type	type of scales: c("colour","fill)
extra_colours	named vector additional colour (hex) values
extra_values	vector with all unique combinations (to assign each a colour)
coord_flip	whether to flip coordinate axes
palette	additional colour palette for unnamed parties
legend_pos	legend position

Value

ggplot object

---

data_delete *Helper function to update/download data*

Description

Helper function to update/download data

Usage

```
data_delete(file = NULL)
```

Arguments

file	to delete - defaults to all of them

Value

nothing

data_import *Helper function to update/download data*

Description

Helper function to update/download data

Usage

```
data_import(file)
```

Arguments

file	file to import to the cache

Value

nothing

data_info *Helper function to update/download data*

Description

Helper function to update/download data

Usage

```
data_info()
```

Value

nothing

data_update *Helper function to update/download data*

Description

Helper function to update/download data

Usage

```
data_update(file = NULL)
```

Arguments

file	vectors with file name from repository. By default, downloads all files

Value

nothing

find_cache *Helper function to find cache folder*

Description

Helper function to find cache folder

Usage

```
find_cache()
```

Value

nothing

geom_auspol_bar *Bar chart, customised for this package.*

Description

Bar chart, customised for this package.

Usage

```
geom_auspol_bar(include_labels = TRUE, reference_line = NULL, nudge_x = 4, ...)
```

Arguments

include_labels	Whether to include numeric labels (TRUE by default)
reference_line	Value for reference line. If left empty, no line is added.
nudge_x	distance between label and bar
...	parameters for ggplot2 functions. Label parameters (geom_text_repel()) prefixed with "labels." Reference line parameters (geom_vline()) prefixed with "ref_line."

geom_auspol_line *Line chart, customised for this package.*

Description

Line chart, customised for this package.

Usage

```
geom_auspol_line(include_labels = TRUE, ...)
```

Arguments

include_labels	Whether to include numeric labels (TRUE by default)
...	parameters for ggplot2 functions. Label parameters (geom_text_repel()) prefixed with "labels."

geom_auspol_lollipop *Lollipop or bar chart, custommised for this package.*

Description

Lollipop or bar chart, custommised for this package.

Usage

```
geom_auspol_lollipop(format = "lollipop", include_labels = TRUE, ...)
```

Arguments

format	Output format : "lollipop" (default) or "bar".
include_labels	Whether to include numeric labels (TRUE by default)
...	parameters for ggplot2 geom_segment() (segmnet.prefix), geom_point(), geom_col() and geom_text() (labels. prefix).

get_house_2PF *Two-party preferred flow*

Description

Get flow from primary vote to finalists, for an division on a given election

Usage

```
get_house_2PF(division, year, aggregation = FALSE)
```

Arguments

division	character vector with division names. When left blank, returns all division.
year	number vector with election years. When left blank, returns all years.
aggregation	Whether to present division totals (defaults to FALSE)

Value

dataframe with list of elected MPs

Examples

```
## Not run:
# get primary to finalist flow of preferences for Jagajaga in the 2013 election
get_house_2PF(division="Jagajaga",year=2013,aggregation = TRUE)

## End(Not run)
```

get_house_2PP *Two-party-preferred summary*

Description

Get 2-party preferred party summary (Coalition vs ALP), as calculated by the AEC.

Usage

```
get_house_2PP(
  division = NULL,
  year = NULL,
  state_abb = NULL,
  aggregation = FALSE
)
```

Arguments

division	character vector with division names. When left blank, returns all division.
year	number vector with election years. When left blank, returns all years.
state_abb	vector with state/territory acronym (e.g. NSW,VIC,QLD,etc.)
aggregation	Whether to present division totals (defaults to FALSE)

Value

dataframe with list of elected MPs

Examples

```
## Not run:
get_house_2PP(division = "Indi",
year=2016,
aggregation = TRUE)

## End(Not run)
```

get_house_MPs *Elected MPs*

Description

Retrieve list of elected MPs, filterable by division and year

Usage

```
get_house_MPs(division = NULL, year = NULL)

get_MPs(division = NULL, year = NULL)
```

Arguments

division	character vector with division names. When left blank, returns all division.
year	number vector with election years. When left blank, returns all years.

Value

data frame with list of elected MPs

Examples

```
## Not run:
# Elected MPs in Melbourne and Cooper, 2019 and 2022
get_house_MPs(division = c("Melbourne","Cooper"),
              year = c(2019,2022))

## End(Not run)
```

---

get_house_preferences *Preferences*

Description

Retrieves preference flow, filterable by election and year. Results can be presented by polling place - as retrived from the AEC - or aggregated by electoral division.

Usage

```
get_house_preferences(
  division,
  year,
  polling_places = NULL,
  aggregation = FALSE
)

get_preferences(division, year, polling_places = NULL, aggregation = FALSE)
```

Arguments

division	vector with division names
year	vector with election years
polling_places	list of polling places
aggregation	whether to aggregate by division

Value

dataframe with list of elected MPs

Examples

```
## Not run:
# basic use
get_house_preferences("Wills",2019) |> head(10)
# aggregated version
get_house_preferences("Wills",2019,aggregation = TRUE)
# filtered by polling place
get_house_preferences("Wills",2019, polling_places=c("ABSENT")) |> head(10)


## End(Not run)
```

---

get_house_primary_vote

Primary vote for House Elections

Description

Get primary vote for one or more divisions, for one or more elections. Data can be filtered by state, political party of polling locations. Results can be presented by polling station or aggregated by division.

Usage

```
get_house_primary_vote(
  division = NULL,
  year = NULL,
  state_abb = NULL,
  party_abb = NULL,
  aggregation = FALSE,
  polling_places = NULL
)
```

Arguments

division	character vector with division names. When left blank, returns all division.
year	number vector with election years. When left blank, returns all years.
state_abb	vector with state/territory acronym (e.g. NSW,VIC,QLD,etc.)
party_abb	vector with party abbreviation (e.g. ALP,LIB,NP,GRN,etc.)
aggregation	Whether to present division totals (defaults to FALSE)
polling_places	vector with regex for polling places

Value

sf object with selected polygons

Examples

```
## Not run:
# Primary vote in Brisbane, 2022 election
get_house_primary_vote(division="Brisbane",year=2022)
# Primary vote in Perth and Brisbane in 2019 and 2022 (aggregated)
get_house_primary_vote(division=c("Brisbane","Perth"),year=c(2019,2022),aggregation = TRUE)
# Primary vote for Greens candidates in Tasmania and the Northern Territory, 2019
get_house_primary_vote(state=c("TAS","NT"),year=2019,aggregation = TRUE, party_abb=c("GRN"))

## End(Not run)
```

---

get_house_turnout *Election turn out*

Description

Retrieve election turnout, filterable by division and year

Usage

```
get_house_turnout(division = NULL, year = NULL)
```

Arguments

division	character vector with division names. When left blank, returns all division.
year	number vector with election years. When left blank, returns all years.

Value

data frame turnout numbers

Examples

```
## Not run:
# Turnout in Riverina
 get_house_turnout(division="Riverina",yeat)

## End(Not run)
```

house_2PF_plot *Preferences flow from primary to finalists*

Description

Plot representing flow of preferences from first preferences to candidates in last round. Can be present as alluvial plot or bar chart, showing votes count or percentages.

Usage

```
house_2PF_plot(
  division,
  year,
  var = "Percent",
  extra_colours = NULL,
  plot_format = "bar",
  include_data = FALSE,
  individualise_IND = TRUE
)
```

Arguments

division	Electoral division
year	Election year
var	Variable to be plotted "Percent" (default) or "Transfer Count"
extra_colours	manual mapping of colours for each party, as a named vector.
plot_format	Whether to plot alluvial chart ("alluvial") or a bar chart ("bar", default).
include_data	If set to TRUE, data will be included under «output_var»$source_data (defaults to FALSE)
individualise_IND	If set to TRUE, party abbreviations for each independent candidate will be changed from "IND" to "IND-«candidate's surname»", effectively separating them in party aggregations.

Value

preference flow, ggplot2 object

Examples

```
## Not run:
# Preference flow for Burt, 2022
house_2PF_plot("Burt",2022,plot_format = "alluvial")
# Preference flow for Warringah 2022,
house_2PF_plot("Spence",2013,plot_format = "bar")

## End(Not run)
```

house_2PP_comparison_plot *Two Party-Preferred Comparison*

Description

Plot with two-party preferred values for one of more divisions, for a given year Can be present as alluvial plot or bar chart, showing votes count or percentages.

Usage

```
house_2PP_comparison_plot(
  division = NULL,
  year,
  state = NULL,
  var = "Percentage",
  include_data = TRUE
)
```

Arguments

division	Electoral division
year	Election year
state	If division is left null, use this to select all divisions in one of more states.
var	Variable to be plotted "Percentage" (default) or "Votes"
include_data	If set to TRUE, data will be included under «output_var»$source_data (defaults to FALSE)

Value

ggplot2 object

Examples

```
## Not run:
# Two party preferred plot for Victoria, 2022
house_2PP_comparison_plot(year=2022,state="VIC")

## End(Not run)
```

house_2PP_historical_plot *Two Party-Preferred Comparison*

Description

Plot with two-party preferred values for one of more divisions, for a given year Can be present as alluvial plot or bar chart, showing votes count or percentages.

Usage

```
house_2PP_historical_plot(
  division,
  year = NULL,
  var = "Percentage",
  include_labels = TRUE,
  include_data = TRUE
)
```

Arguments

division	Electoral division
year	Election year
var	Variable to be plotted "Percentage" (default) or "Votes"
include_labels	If set to TRUE, the plot will include each value.
include_data	If set to TRUE, data will be included under «output_var»$source_data (defaults to FALSE)

Value

ggplot2 object

Examples

```
## Not run:
# Plot historical 2PP for Aston
house_2PP_historical_plot(division="Aston")

## End(Not run)
```

house_preference_flow_data *Flow of preferences in a division.*

Description

retrieves data containing preferential voting rounds for a division in a particular election (as published by the AEC). Can be filtered by polling place (including special modes of voting) or it can be presented as an aggregate per division.

Usage

```
house_preference_flow_data(
  division,
  year,
  individualise_IND = TRUE,
  exclude_parties = NULL,
  exclude_rounds = 0
)
```

Arguments

division	division
year	election year
individualise_IND	If set to TRUE, party abbreviations for each independent candidate will be changed from "IND" to "IND-«candidate's surname»", effectively separating them in party aggregations.
exclude_parties	vector with party acronyms to exclude from plot
exclude_rounds	If parties are excluded, include vector indicating from which rounds should them be excluded

Value

list with data frames with results for each round

Examples

```
## Not run:
#get preferences for Wills, 2019
get_house_preferences("Wills",2019)
show results for absent votes only
get_house_preferences("Wills",2019, polling_places=c("ABSENT"),aggregation = FALSE)


## End(Not run)
```

house_preference_flow_plot *Plot House of reps preferences flow*

Description

Plot flow of preferences in a division as an alluvial plot.

Usage

```
house_preference_flow_plot(
  division,
  year,
  var = "Percent",
  exclude_parties = NULL,
  merge_parties = NULL,
  extra_colours = NULL,
  include_data = FALSE
)
```

Arguments

division	Electoral division
year	Election year
var	Variable to be plotted "Percent" (default) or "Preference Count"
exclude_parties	vector with party acronyms to exclude from plot
merge_parties	list of parties to merge in one line following, the format list(NEWCODE=c(code1,code2,etc.))
extra_colours	manual mapping of colours for each party, as a named vector.
include_data	If set to TRUE, output of primary_vote_summary(), will be included under «output_var»$source_data (defaults to FALSE)

Value

preference flow, ggplot2 object

Examples

```
## Not run:
# Preference flow for Wills, 2019
house_preference_flow_plot(division = "Wills",year=2019)
# Preference flow for Warringah 2022,
# excluding two finalists from round 1,
# independent candidate in teal.
house_preference_flow_plot(division = "Warringah",year=2022,

## End(Not run)
```

house_primary_comparison_plot *Plot historical changes in primary vote*

Description

Line chart with historial changes for a division, group of candidates in a party, selected parties, etc.

Usage

```
house_primary_comparison_plot(
  division = NULL,
  year = NULL,
  state = NULL,
  label = "Candidate",
  plotted_variable = "Percentage",
  sort_by_value = TRUE,
  extra_colours = NULL,
  plot_format = "lollipop",
  include_labels = FALSE,
  hor_nudge = 5,
  parties = NULL,
  parties_year = NULL,
  merge_parties = NULL,
  include_others = FALSE,
  include_informal = FALSE,
  individualise_IND = TRUE,
  include_data = TRUE,
  data = NULL
)
```

Arguments

division	Name of ONE electoral division
year	numeric vector with election years (from 2004), defaults to all.
state	Code for one state
label	How to label the results, either by Candidate Name ("Name",default), Party Name ("PartyNm") or Party abbreviation ("PartyAb")
plotted_variable	Variable to plot, out of "OrdinaryVotes", "Percentage" (default) and Percentage_with_Informal
sort_by_value	Whether to sort results by descending order (TRUE by default)
extra_colours	manual mapping of colours for each party, as a named vector.
plot_format	Whether to plot lollipop chart ("lollipop", default) or a bar chart.
include_labels	If set to TRUE, the plot will include each value.

hor_nudge	if labels are included, separation from chart/dot
parties	which parties to include in the summary. All (default), a vector of strings with the party acronyms (see list_parties()), or a number indicating the top n parties from a certain year.
parties_year	If *parties* has is NULL or a number, this indicates if the selection needs to be from a certain year (.e.g only select the historical data for the three top parties in 2012)
merge_parties	list of parties to merge in one line following, the format list(NEWCODE=c(code1,code2,etc.))
include_others	Boolean used along *parties* to included the remaining votes in one "Other" category.
include_informal	Boolean to add informal votes in addition to the party selection. Informal votes will be included if no parties are selected, or the top n parties are selected, and it happens to be in the top n - even if this flag is set to false.
individualise_IND	If set to TRUE, party abbreviations for each independent candidate will be changed from "IND" to "IND-«candidate's surname»", effectively separating them in party aggregations.
include_data	If set to TRUE, output of house_primary_vote_summary(), will be included under «output_var»$source_data (defaults to FALSE)
data	Alternative, instead of providing a parameters, it is possible to provide the data frame with the data to plot, following the format from the output of house_primary_vote_summary().

Value

ggplot2 object

Examples

```
## Not run:
# Compare primary voting in Kooyong in 2022,  lollipop chart (default)
house_primary_comparison_plot(division = "Kooyong",
                              year=2022,
                              individualise_IND = TRUE)
# Liberal Primary vote in Tasmania in 2022, bar chart
house_primary_comparison_plot(state="TAS",
                              year=2022,
                              parties=c("LP"),
                              plot_format = "bar")


## End(Not run)
```

house_primary_historic_plot *Plot primary vote history*

Description

Plot historical primary vote results for a division or group of divisions, being able to select and aggregate political parties. Can plot either percentages or absolute number of ordinary votes.

Usage

```
house_primary_historic_plot(
  division = NULL,
  plotted_variable = "Percentage",
  parties = NULL,
  parties_year = NULL,
  merge_parties = NULL,
  include_others = FALSE,
  include_informal = FALSE,
  individualise_IND = FALSE,
  extra_colours = NULL,
  include_labels = FALSE,
  year = NULL,
  include_data = FALSE,
  include_text_tooltip = FALSE,
  data = NULL
)
```

Arguments

division	named vector with division names
plotted_variable	Variable to plot, out of "OrdinaryVotes", "Percentage" (default) and Percentage_with_Informal
parties	which parties to include in the summary. All (default), a vector of strings with the party acronyms (see list_parties()), or a number indicating the top n parties from a certain year.
parties_year	If *parties* has is NULL or a number, this indicates if the selection needs to be from a certain year (.e.g only select the historical data for the three top parties in 2012).
merge_parties	list of parties to merge in one line following, the format list(NEWCODE=c(code1,code2,etc.))
include_others	Boolean used along *parties* to included the remaining votes in one "Other" category.
include_informal	Boolean to add informal votes in addition to the party selection. Informal votes will be included if no parties are selected, or the top n parties are selected, and it happens to be in the top n - even if this flag is set to false.

individualise_IND
: If set to TRUE, party abbreviations for each independent candidate will be changed from "IND" to "IND-«candidate's surname»", effectively separating them in party aggregations.

extra_colours
: manual mapping of colours for each party, as a named vector.

include_labels
: If set to TRUE, the plot will include each value.

year
: numeric vector with election years (from 2004), defaults to all.

include_data
: If set to TRUE, output of house_primary_vote_summary(), will be included under «output_var»$source_data (defaults to FALSE)

include_text_tooltip
: Flag to include tooltip for plotly mapped as text in ggplot

data
: Alternative, instead of providing a parameters, it is possible to provide the data frame with the data to plot, folowing the format from the output of house_primary_vote_summary().

Value

ggplot2 object

Examples

```
## Not run:
# Plot historic primary voting in Canberra, top 3 parties
house_primary_historic_plot("Canberra", parties =3,


## End(Not run)
```

---

house_primary_vote_summary

Helper function to download data

Description

Helper function to download data

Usage

```
house_primary_vote_summary(
  division = NULL,
  state = NULL,
  year = NULL,
  parties = NULL,
  parties_year = NULL,
  include_others = FALSE,
  merge_parties = NULL,
  include_informal = FALSE,
```

```
  include_names = TRUE,
  individualise_IND = FALSE,
  wide_format = NULL
)
```

Arguments

division	vector with names of electoral divisions (e.g. "Banks", "Wills","Indi")
state	if divisions are not provide, provide a vector with state initials e.g. c("NT","TAS")
year	numeric vector with election years (from 2004), defaults to all.
parties	which parties to include in the summary. All (default), a vector of strings with the party acronyms (see list_parties()), or a number indicating the top n parties from a certain year.
parties_year	If *parties* has is NULL or a number, this indicates if the selection needs to be from a certain year (.e.g only select the historical data for the three top parties in 2012)
include_others	Boolean used along *parties* to included the remaining votes in one "Other" category.
merge_parties	list of parties to merge in one line following, the format list(NEWCODE=c(code1,code2,etc.))
include_informal	Boolean to add informal votes in addition to the party selection. Informal votes will be included if no parties are selected, or the top n parties are selected, and it happens to be in the top n - even if this flag is set to false.
include_names	whether to include the candidates name and surname in the extract (TRUE by default).
individualise_IND	If set to TRUE, party abbreviations for each independent candidate will be changed from "IND" to "IND-«candidate's surname»", effectively separating them in party aggregations.
wide_format	Whether to present the result in long format, like the AEC's source, or a year-by-year summary. Options include NULL (no summarisation, default), "OrdinaryVotes" (absolute numbers), "Percentage_with_Informal" and "Percentage" (which is the percentage counted on elections).

Value

dataframe

Examples

```
## Not run:
# Get primary for Kooyong in 2022
house_primary_vote(division="Kooyong",year=2022)
# Get historic primary for Liberals and Labor in Kooyong
house_primary_vote(division="Kooyong",parties=c("LP","ALP")
#Get primary vote for all National candidates in WA, 2022
house_primary_vote(state="WA",year=2022,parties=c(NP))
```

```
## End(Not run)
```

`house_results_historic` *Historic Resuls*

Description

Plot seats by party across time. Parties can be filtered and grouped by coalitions.

Usage

```
house_results_historic(
  individualise_IND = FALSE,
  merge_parties = NULL,
  parties = NULL,
  include_others = FALSE,
  include_labels = TRUE,
  extra_colours = NULL,
  include_data = FALSE
)
```

Arguments

`individualise_IND`	If set to TRUE, party abbreviations for each independent candidate will be changed from "IND" to "IND-«candidate's surname»", effectively separating them in party aggregations.
`merge_parties`	list of parties to merge in one line following, the format list(NEWCODE=c(code1,code2,etc.))
`parties`	List of political party abbreviations to filter on. If merge_parties is used, those names can be included too.
`include_others`	Boolean used along *parties* to included the remaining votes in one "Other" category.
`include_labels`	If set to TRUE, the plot will include each value.
`extra_colours`	manual mapping of colours for each party, as a named vector.
`include_data`	If set to TRUE, data will be included under «output_var»$source_data (defaults to FALSE)

Value

preference flow, ggplot2 object

Examples

```
## Not run:
# Historic results, focusing showing tally for Coalition, ALP, Greens - others merged together
house_results_historic(merge_parties = list(COAL=c("CLP","LP","LNP","NP")),
                       parties =c("COAL","ALP","GRN"),
                       include_other=TRUE))

## End(Not run)
```

`house_results_tally` *Election Tally*

Description

Plot party totals for a given election. Can aggregate parties into groups, amongst other filters.

Usage

```
house_results_tally(
  year,
  individualise_IND = FALSE,
  merge_parties = NULL,
  add_majority_line = TRUE,
  include_labels = FALSE,
  extra_colours = NULL,
  include_data = FALSE
)
```

Arguments

`year`	Election year
`individualise_IND`	If set to TRUE, party abbreviations for each independent candidate will be changed from "IND" to "IND-«candidate's surname»", effectively separating them in party aggregations.
`merge_parties`	list of parties to merge in one line following, the format list(NEWCODE=c(code1,code2,etc.))
`add_majority_line`	add line representing 50% +1 of the seats
`include_labels`	If set to TRUE, the plot will include each value.
`extra_colours`	manual mapping of colours for each party, as a named vector.
`include_data`	If set to TRUE, data will be included under «output_var»$source_data (defaults to FALSE)

Value

preference flow, ggplot2 object

Examples

```
## Not run:
# Basic example
house_results_tally(2013)
# Coalition votes put together
house_results_tally(2013, merge_parties = list(COAL=c("CLP","LP","LNP","NP")))

## End(Not run)
```

`list_divisions` *Get list of divisions*

Description

get list of all the Australian Federal electoral divisions, being able to filter by any attribute. Covers all divisions from the 2004 Election.

Usage

```
list_divisions(filters = NULL)
```

Arguments

`filters`	*(optional)* list() with filters in the form list(Column="Value")

Value

data frame with lists of divisions

Examples

```
## Not run:
# Get list of all divisions
list_divisions()

#Get list containing only Wills and Melbourne
list_divisions(filters=list(DivisionNm=c("Wills","Melbourne")))

## End(Not run)
```

list_parties *List all political parties.*

Description

Lists all political parties that have participated from the 2004 Election onwards. Parties are presented as recorded by the AEC. List can be filtered by party names matching a regular expression.

Usage

```
list_parties(filters = NULL, party_regex = NULL)
```

Arguments

filters	(optional) list() with filters in the form list(Column="Value").
party_regex	additional filter for party names, taking a regular expression.

Value

data frame with lists of divisions

Examples

```
## Not run:
# Get list of all registered political parties
list_parties()
#
# Get list of all parties whose name start with "Australia"
list_parties(party_regex="^Australia")

## End(Not run)
```

list_polling_places *List all polling stations*

Description

Retrieve list of all polling station that been used from 2044 onwards. Names as recorded by the AEC. List can be filtered by state, division names and regular expressions matching their names.

Usage

```
list_polling_places(filters = NULL)
```

Arguments

filters	(optional) list() with filters in the form list(Column="Value")

Value

data frame with lists of polling stations

Examples

```
## Not run:
# Get list of all registered parties
list_parties()
# Get list of polling places in the division of Hasluck
list_parties(list)


## End(Not run)
```

`list_years` *Get election years.*

Description

Very simple function listing the election years included in this package.

Usage

```
list_years()
```

Value

vector with years

Examples

```
## Not run:
# Get list of all divisions
list_years()

## End(Not run)
```

manage_colours *Helper function generate colour palette*

Description

Helper function generate colour palette

Usage

```
manage_colours(extra_colours = NULL, extra_values = NULL, palette = NULL)
```

Arguments

extra_colours	named vector additional colour (hex) values
extra_values	vector with all unique combinations (to assign each a colour)
palette	palette to replace brewer.pal

Value

named vector with colours

party_colours *Named vector with common party colours, with option to add custom/additional values*

Description

Named vector with common party colours, with option to add custom/additional values

Usage

```
party_colours(extra = NULL)
```

Arguments

extra	named vector additional colour (hex) values

Value

named vector

Index