Package 'hockeystick'

August 28, 2026

Title Download and Visualize Essential Global Heating Data

Version 1.0.0

Description Provides easy access to essential climate change datasets to non-climate experts. Users can download the latest raw data from authoritative sources and view it via predefined 'ggplot2' charts. Datasets include atmospheric CO2, methane, emissions, instrumental and proxy temperature records, CMIP6 projections, sea levels, Arctic/Antarctic sea-ice, Hurricanes, Wildfires, and Paleoclimate data. Sources include: NOAA Mauna Loa Laboratory <https://gml.noaa.gov/ccgg/trends/data.html>, Global Carbon Project <https://www.globalcarbonproject.org/carbonbudget/>, NASA GISTEMP <https://data.giss.nasa.gov/gistemp/>, National Snow and Sea Ice Data Center <https://nsidc.org/home>, CSIRO <https://research.csiro.au/slrwavescoast/sea-level/measurements-and-data/sea-level-data/>, NOAA Laboratory for Satellite Altimetry <https://www.star.nesdis.noaa.gov/socd/lsa/SeaLevelRise/> and HURDAT Atlantic Hurricane Database <https://www.aoml.noaa.gov/hrd/hurdat/Data_Storm.html>, Vostok Paleo carbon dioxide and temperature data: <doi:10.3334/CDIAC/ATG.009>.

License MIT + file LICENSE

Encoding UTF-8

URL https://cortinah.github.io/hockeystick/, https://github.com/cortinah/hockeystick

BugReports https://github.com/cortinah/hockeystick/issues

Imports ggplot2, lubridate, tools, readr, dplyr, tidyr, patchwork, scales, rvest, tibble, treemapify, RColorBrewer, jsonlite, readxl

Depends R (>= 4.1.0)

Suggests rmarkdown, knitr, spelling, viridisLite, quarto, terra, geodata, svglite, testthat (>= 3.0.0)

Config/testthat/edition 3

Language en-US

VignetteBuilder quarto

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Hernando Cortina [aut, cre] (ORCID:
<https://orcid.org/0000-0001-6790-4870>)

Maintainer Hernando Cortina <hch@alum.mit.edu>

Repository CRAN

Date/Publication 2026-08-28 09:50:16 UTC

Contents

climate_grid . . . 3
emissions_map . . . 4
file_info_ . . . 5
getsize . . . 5
get_carbon . . . 6
get_cmip6 . . . 7
get_cmip6_anom . . . 9
get_dailytemp . . . 10
get_dailytempcop . . . 12
get_emissions . . . 13
get_fires_area . . . 14
get_fires_emissions . . . 16
get_hurricanes . . . 17
get_icecurves . . . 19
get_methane . . . 20
get_paleo . . . 21
get_seaice . . . 23
get_sealevel . . . 24
get_temp . . . 26
get_temp2k . . . 27
hockeystick_cache . . . 28
hscache_path . . . 30
merge_carbontemp . . . 30
plot_carbon . . . 31
plot_carbontemp . . . 32
plot_cmip6 . . . 33
plot_cmip6_anom . . . 34
plot_dailytemp . . . 36
plot_emissions . . . 37
plot_emissions_with_land . . . 39
plot_fires_area . . . 40
plot_fires_emissions . . . 41
plot_hurricanes . . . 43
plot_hurricane_nrg . . . 44
plot_icecurves . . . 45
plot_methane . . . 46
plot_paleo . . . 47

plot_seaice 48
plot_sealevel 49
plot_temp 50
plot_temp2k 51
plot_temp_monthly 52
plot_temp_scatter 53
print.hockeystick_cache_info 54
warming_stripes 55

Index **57**

`climate_grid` *Download and plot essential climate data*

Description

Plots a 2x2 grid of carbon, temperature, sea ice, and sea level charts.

Usage

```
climate_grid(print = TRUE)
```

Arguments

`print`	(boolean) Display climate grid ggplot2 chart, defaults to TRUE. Use FALSE to not display chart.

Details

`warming_stripes` invisibly returns a ggplot2 object with 2x2 grid of carbon, temperature, sea ice, and sea level charts from `get_carbon`, `get_temp`, `get_seaice`, and `get_sealevel`. By default the chart is also displayed. Users may further modify the output ggplot2 chart.

Value

Invisibly returns a ggplot2 object with climate grid

Author(s)

Hernando Cortina, <hch@alum.mit.edu>

Examples

```
# Draw grid

grid <- climate_grid()
```

emissions_map *Download and plot essential climate data*

Description

Plots a treemap of cumulative co2 emissions by country since 1900.

Usage

```
emissions_map(
  dataset = get_emissions(),
  print = TRUE,
  since = 1900,
  number = "all",
  title = substitute(paste(since, "-", to, " Cumulative " * CO[2] *
    " Emissions by Country"), list(since = since, to =
    as.character(dataset[nrow(dataset), 2])))
)
```

Arguments

dataset	Name of the tibble generated by get_emissions
print	(boolean) Display emissions treemap, defaults to TRUE. Use FALSE to not display chart.
since	(numeric) Start year for cumulative emissions, defaults to 1900 if omitted
number	(numeric) Number of countries to display in treemap, defaults to all if omitted
title	(string) Manually specify chart title

Details

emissions_map invisibly returns a ggplot2 object with cumulative emissions treemap by country since 1900 from get_emissions. By default the chart is also displayed. Users may further modify the output ggplot2 chart.

Value

Invisibly returns a ggplot2 object with emissions treemap

Author(s)

Hernando Cortina, <hch@alum.mit.edu>

Examples

```
# Draw treemap

co2map <- emissions_map()

co2map <- emissions_map(since=2000, number=20, title="Top 20 Cumulative Emitters Since 2000")
```

`file_info_` *Display cached file info*

Description

Internal function

Usage

```
file_info_(x)
```

Arguments

x	filenames

`getsize` *Get rounded size of file in kB*

Description

Internal function

Usage

```
getsize(x)
```

Arguments

x	filenames

`get_carbon` *Download and plot essential climate data*

Description

Retrieves atmospheric carbon dioxide measurements from National Oceanic and Atmospheric Administration Earth System Research Laboratories monitoring laboratory in Mauna Loa, Hawaii. https://gml.noaa.gov/ccgg/trends/data.html

Usage

```
get_carbon(use_cache = TRUE, write_cache = getOption("hs_write_cache"))
```

Arguments

`use_cache`	(boolean) Return cached data if available, defaults to TRUE. Use FALSE to fetch updated data.
`write_cache`	(boolean) Write data to cache, defaults to FALSE. Use TRUE to write data to cache for later use. Can also be set using options(hs_write_cache=TRUE)

Details

`get_carbon` invisibly returns a tibble with NOAA's monthly average carbon dioxide measurement. The returned object includes date, year, month, average, trend, std dev, and uncertainty columns. Trend is NOAA's published trend. Please refer to above website for details.

Value

Invisibly returns a tibble with the monthly carbon dioxide series

Author(s)

Hernando Cortina, <hch@alum.mit.edu>

References

Dr. Pieter Tans, NOAA/GML https://gml.noaa.gov/ccgg/trends/ and Dr. Ralph Keeling, Scripps Institution of Oceanography https://scrippsco2.ucsd.edu/.

C.D. Keeling, R.B. Bacastow, A.E. Bainbridge, C.A. Ekdahl, P.R. Guenther, and L.S. Waterman, (1976), Atmospheric carbon dioxide variations at Mauna Loa Observatory, Hawaii, *Tellus*, vol. 28, 538-551

Examples

```
# Fetch from cache if available, otherwise download:
maunaloa <- get_carbon()
#
# Force fetch from source:
maunaloa <- get_carbon(use_cache=FALSE)
#
# Review cache contents and last update dates:
hockeystick_cache_details()
#
# Plot output using package's built-in ggplot2 settings
plot_carbon(maunaloa)
```

`get_cmip6` *Download and plot essential climate data*

Description

Retrieves downscaled CMIP6 climate projection data from WorldClim (https://www.worldclim.org/data/cmip6/). Data are monthly or bioclimatic variables projected by a selected global climate model (GCM) under an SSP scenario for one of four 20-year periods (2021-2100), at 10 arc-minute resolution.

Usage

```
get_cmip6(
  var = "bioc",
  bio = 1,
  ssp = "585",
  period = "2081-2100",
  model = "MPI-ESM1-2-HR",
  res = "10m",
  use_cache = TRUE,
  write_cache = getOption("hs_write_cache")
)
```

Arguments

`var`	(string) Climate variable. One of `"bioc"` (bioclimatic variables, the default), `"tmin"`, `"tmax"`, or `"prec"` (precipitation).
`bio`	(numeric) If `var = "bioc"`, which bioclimatic variable to return (1-19). Defaults to 1 (Annual Mean Temperature). The variables are: 1 = Annual Mean Temperature; 2 = Mean Diurnal Range; 3 = Isothermality (BIO2/BIO7); 4 = Temperature Seasonality (standard deviation); 5 = Max Temperature of Warmest Month; 6 = Min Temperature of Coldest Month; 7 = Temperature Annual Range (BIO5-BIO6); 8 = Mean Temperature of Wettest Quarter; 9 = Mean Temperature of

	Driest Quarter; 10 = Mean Temperature of Warmest Quarter; 11 = Mean Temperature of Coldest Quarter; 12 = Annual Precipitation; 13 = Precipitation of Wettest Month; 14 = Precipitation of Driest Month; 15 = Precipitation Seasonality (coefficient of variation); 16 = Precipitation of Wettest Quarter; 17 = Precipitation of Driest Quarter; 18 = Precipitation of Warmest Quarter; 19 = Precipitation of Coldest Quarter. Variables 1-11 are in degrees C (seasonality variables 4 and 7 as applicable), 12-19 in mm. For the monthly variables (`var` = `"prec"`, `"tmin"`, or `"tmax"`), `bio` instead selects the calendar month, 1-12 (1 = January).
`ssp`	(string) Shared Socioeconomic Pathway scenario. One of "126", "245", "370", or "585" (the default). SSP5-8.5 is the high-emissions upper bound and shows the strongest projected change; SSP2-4.5 is a middle-of-the-road scenario roughly consistent with current policies ("current trends continue"). SSP1-2.6 assumes strong mitigation, and SSP3-7.0 is a high fragmented-world path.
`period`	(string) Projection period. One of "2021-2040", "2041-2060", "2061-2080", or "2081-2100" (the default).
`model`	(string) CMIP6 global climate model. Defaults to "MPI-ESM1-2-HR". Availability of variable/scenario combinations varies by model; the function checks the WorldClim index and reports what is available.
`res`	(string) Resolution: "10m" (~18 km, default), "5m" (~9 km) or "2.5m" (~5 km). Finer resolutions are considerably larger downloads.
`use_cache`	(boolean) Return cached data if available, defaults to TRUE.
`write_cache`	(boolean) Write data to cache, defaults to FALSE.

Details

`get_cmip6` downloads a WorldClim CMIP6 GeoTIFF, aggregates it to a one-degree grid (averaging), and returns a tibble suitable for mapping. Requires the `terra` package (installed automatically when needed).

Value

Invisibly returns a tibble with columns `lon`, `lat`, and `value`: the projected values aggregated to a one-degree grid.

Author(s)

Hernando Cortina, <hch@alum.mit.edu>

References

- WorldClim CMIP6 future climate data: https://www.worldclim.org/data/cmip6/
- Fick, S.E. & Hijmans, R.J. (2017) WorldClim 2: new 1km spatial resolution climate surfaces for global land areas. *Int. J. Climatol.* 37: 4302-4315.

Examples

```
# Projected annual mean temperature under SSP585, 2081-2100:
proj <- get_cmip6(var='bioc', bio=1)
plot_cmip6(proj)
#
# Projected precipitation change under a moderate scenario:
prec <- get_cmip6(var='prec', ssp=245, period='2041-2060', use_cache=FALSE)
```

`get_cmip6_anom` *Download and plot essential climate data*

Description

Computes expected climate change as the difference between a CMIP6 projection retrieved from WorldClim and its 1970-2000 reference-period baseline, on a common one-degree grid.

Usage

```
get_cmip6_anom(
  var = "bioc",
  bio = 1,
  ssp = "585",
  period = "2081-2100",
  model = "MPI-ESM1-2-HR",
  res = "10m",
  use_cache = TRUE,
  write_cache = getOption("hs_write_cache")
)
```

Arguments

`var`, `bio`, `ssp`, `period`, `model`, `res`	Passed through to `get_cmip6`.
`use_cache`	(boolean) Return cached anomaly data if available, defaults to TRUE.
`write_cache`	(boolean) Write data to cache, defaults to FALSE.

Details

`get_cmip6_anom` downloads the projection and the corresponding WorldClim v2.1 baseline, aggregates both to a one-degree grid, and subtracts cell by cell. Differencing removes most of the model's systematic bias, leaving the expected change. Polar regions can show spurious values where the downscaled grids differ in land-sea masking.

Value

Invisibly returns a tibble with columns `lon`, `lat`, and `delta`: the projected change versus the 1970-2000 baseline for each cell. For temperature variables `delta` is the absolute change in degrees C; for precipitation variables (var "prec", or bioc 12-19) it is the percent change, capped at +/-100\ baseline is near zero.

Author(s)

Hernando Cortina, <hch@alum.mit.edu>

References

See `get_cmip6`.

Examples

```
anom <- get_cmip6_anom()
plot_cmip6_anom(anom)
```

`get_dailytemp` *Download and plot essential climate data*

Description

Retrieves the daily air or sea-surface temperature data since 1940 from ClimateReanalyzer.org Source is University of Maine Climate Change Institute. https://climatereanalyzer.org/clim/t2_daily/

Usage

```
get_dailytemp(
  use_cache = TRUE,
  write_cache = getOption("hs_write_cache"),
  region = "W",
  mean_start = if (region %in% c("WS", "NS", "ws", "ns")) 1982 else 1979,
  mean_end = 2000
)
```

Arguments

`use_cache`	(boolean) Return cached data if available, defaults to TRUE. Use FALSE to fetch updated data.
`write_cache`	(boolean) Write data to cache, defaults to FALSE. Use TRUE to write data to cache for later use. Can also be set using options(hs_write_cache=TRUE)

region	(string) Region selection, defaults to world air temperature. Options are: World Air "W", Northern Hemisphere Air "NW", Southern Hemisphere Air "SW", Tropics Air "TR", Arctic Air "AR", Antarctica Air "AN", World Sea Surface "WS", and North Atlantic Sea Surface "NS".
mean_start	(numeric) Start year for historic mean, defaults to 1979.
mean_end	(numeric) End year for historic mean, defaults to 2000.

Value

Invisibly returns a tibble with the daily 2-meter air or sea surface temperatures since 1940 as well as historic mean by day-of-year and current anomaly versus mean.

`get_dailytemp` invisibly returns a tibble with the daily temperatures since 1940 as well as mean by day-of-year and anomaly. Default to world data, but region can be selected among six options.

Region options include world air (default), Northern Hemisphere air, Southern Hemisphere air, Tropics air, Arctic air, Antarctic air, World sea surface and North Atlantic sea surface and is stored in attribute of output. The historic daily mean-by-day period defaults to 1979-2000. This range can be optionally modified.

Data are updated daily. For day-of-year mean removes observations from February 29 on leap years.

Author(s)

Hernando Cortina, <hch@alum.mit.edu>

References

- ClimateReanalyzer.org: https://climatereanalyzer.org/clim/t2_daily/
  Notes: daily mean surface air temperature (2-meter height) estimates from the ECMWF Reanalysis version 5 (ERA5) for the period January 1940 to present. ERA5 is a state-of-the-art numerical climate/weather modeling framework that ingests surface, radiosonde, and satellite observations to estimate the state of the atmosphere through time. ERA5 files have a horizontal grid resolution of 0.25° x 0.25° (about 31km x 31km at 45°N). Each daily temperature represents an average across all model gridcells within the defined latitude/longitude bounds for the selected domain. The means are area-weighted to account for the convergence of longitude lines at the poles

Examples

```
# Fetch temp anomaly from cache if available:
dailytemps <- get_dailytemp()
#
# Force cache refresh:
dailytemps <- get_dailytemp(use_cache=FALSE)
#
# Review cache contents and last update dates:
hockeystick_cache_details()
#
# Plot output using package's built-in ggplot2 settings
plot_dailytemp(dailytemps)
```

```
# Change region to Arctic
arctictemp <- get_dailytemp(region='AR', use_cache=FALSE)
```

`get_dailytempcop` *Download and plot essential climate data*

Description

Retrieves the daily air temperature since 1940 from the EU Copernicus Service https://cds.climate.copernicus.eu/#!/home

Usage

```
get_dailytempcop(
  use_cache = TRUE,
  write_cache = getOption("hs_write_cache"),
  region = "W"
)
```

Arguments

`use_cache`	(boolean) Return cached data if available, defaults to TRUE. Use FALSE to fetch updated data.
`write_cache`	(boolean) Write data to cache, defaults to FALSE. Use TRUE to write data to cache for later use. Can also be set using options(hs_write_cache=TRUE)
`region`	(string) Region selection, defaults to world air temperature. Options are: World Air "W".

Value

Invisibly returns a tibble with the daily 2-meter air temperatures since 1940 as well as historic mean by day-of-year and current anomaly versus mean.

`get_dailytempcop` invisibly returns a tibble with the daily temperatures since 1940 as well as mean by day-of-year and anomaly.

Region options include world air (default). The historic daily mean-by-day period defaults to 1991-2020.

Data are updated daily.

Author(s)

Hernando Cortina, <hch@alum.mit.edu>

References

- Copernicus: https://cds.climate.copernicus.eu/#!/home

  Notes: daily mean surface air temperature (2-meter height) estimates from the ECMWF Reanalysis version 5 (ERA5) for the period January 1940 to present. ERA5 is a state-of-the-art numerical climate/weather modeling framework that ingests surface, radiosonde, and satellite observations to estimate the state of the atmosphere through time. ERA5 files have a horizontal grid resolution of 0.25° x 0.25° (about 31km x 31km at 45°N). Each daily temperature represents an average across all model gridcells within the defined latitude/longitude bounds for the selected domain. The means are area-weighted to account for the convergence of longitude lines at the poles

Examples

```
# Fetch temp anomaly from cache if available:
dailytemps <- get_dailytempcop()
#
# Force cache refresh:
dailytemps <- get_dailytempcop(use_cache=FALSE)
#
# Review cache contents and last update dates:
hockeystick_cache_details()
#
# Plot output using package's built-in ggplot2 settings
plot_dailytemp(dailytemps)
```

---

get_emissions *Download and plot essential climate data*

Description

Retrieves Global Carbon Project (GCP) annual global carbon dioxide emissions since 1750 from Our World In Data repository https://github.com/owid/co2-data

Usage

```
get_emissions(use_cache = TRUE, write_cache = getOption("hs_write_cache"))
```

Arguments

use_cache	(boolean) Return cached data if available, defaults to TRUE. Use FALSE to fetch updated data.
write_cache	(boolean) Write data to cache, defaults to FALSE. Use TRUE to write data to cache for later use. Can also be set using options(hs_write_cache=TRUE)

Details

get_emissions invisibly returns a tibble with GCP's annual carbon dioxide emissions from fossil fuels in aggregate and for every nation. The returned object includes ISO code, country, year, co2 emissions, growth rates, per capita, and decompositions by industry and gas type. Please refer to above website for details.

Value

Invisibly returns a tibble with annual carbon dioxide emissions

Author(s)

Hernando Cortina, <hch@alum.mit.edu>

References

https://www.globalcarbonproject.org/carbonbudget/

Friedlingstein, P. et al (2020), Global Carbon Budget 2020, *Earth System Science Data*, vol. 12, 3269-3340 doi:10.5194/essd1232692020

Examples

```
# Fetch from cache if available:
emissions <- get_emissions()
#
# Force cache refresh:
emissions <- get_emissions(use_cache=FALSE)
#
# Review cache contents and last update dates:
hockeystick_cache_details()
#
# Plot output using package's built-in ggplot2 settings
plot_emissions(emissions)
```

---

get_fires_area *Download and plot essential climate data*

Description

Retrieves global wildfire burnt area and fire count data from the Global Wildfire Information System (GWIS), operated by the Copernicus Emergency Management Service. https://gwis.jrc.ec.europa.eu/apps/gwis.statistics/seasonaltrend

Usage

```
get_fires_area(
  place = "WORLD",
  year = as.numeric(format(Sys.Date(), "%Y")),
  use_cache = TRUE,
  write_cache = getOption("hs_write_cache")
)
```

Arguments

place	(string) Region or country for which to retrieve data. May be one of the following regional aggregates: "WORLD" (default), "UN_EUR" (Europe), "UN_AFR" (Africa), "UN_ASI" (Asia), "UN_AME" (Americas), or "UN_OCE" (Oceania). Alternatively, supply an ISO 3166-1 alpha-3 country code such as "USA", "CAN", or "BRA".
year	(numeric) Four-digit year for which to retrieve data. Must be 2012 or later. Defaults to the current year.
use_cache	(boolean) Return cached data if available, defaults to TRUE. Use FALSE to fetch updated data.
write_cache	(boolean) Write data to cache, defaults to FALSE. Use TRUE to write data to cache for later use. Can also be set using options(hs_write_cache=TRUE)

Details

get_fires_area invisibly returns a tibble with weekly and year-to-date cumulative wildfire burnt area (in hectares) and fire event counts sourced from the GWIS API. Historical minimum, maximum, and average baselines are included for comparison. Data coverage begins in 2012. Cache filenames include the place and year so that multiple queries can be cached independently.

Value

Invisibly returns a tibble with weekly and cumulative wildfire burnt area and fire count data for the requested place and year. Weekly columns include date, events, events_min, events_max, events_avg, area_ha, area_ha_min, area_ha_avg, and area_ha_max. Cumulative columns are prefixed with cum_.

Author(s)

Hernando Cortina, <hch@alum.mit.edu>

References

Global Wildfire Information System (GWIS), Copernicus Emergency Management Service / European Commission Joint Research Centre. https://gwis.jrc.ec.europa.eu/apps/gwis.statistics/seasonaltrend

Examples

```
# Fetch global data for the current year from cache if available:
fires <- get_fires_area()
#
# Fetch data for Canada in 2023:
fires_can <- get_fires_area(place = 'CAN', year = 2023)
#
# Force cache refresh:
fires <- get_fires_area(use_cache = FALSE)
#
# Review cache contents and last update dates:
hockeystick_cache_details()
```

---

get_fires_emissions *Download and plot essential climate data*

Description

Retrieves global wildfire greenhouse gas emissions data from the Global Wildfire Information System (GWIS), operated by the Copernicus Emergency Management Service. https://gwis.jrc.ec.europa.eu/apps/gwis.statistics/seasonaltrend

Usage

```
get_fires_emissions(
  place = "WORLD",
  year = as.numeric(format(Sys.Date(), "%Y")),
  use_cache = TRUE,
  write_cache = getOption("hs_write_cache")
)
```

Arguments

place	(string) Region or country for which to retrieve data. May be one of the following regional aggregates: "WORLD" (default), "UN_EUR" (Europe), "UN_AFR" (Africa), "UN_ASI" (Asia), "UN_AME" (Americas), or "UN_OCE" (Oceania). Alternatively, supply an ISO 3166-1 alpha-3 country code such as "USA", "CAN", or "BRA".
year	(numeric) Four-digit year for which to retrieve data. Must be 2012 or later. Defaults to the current year.
use_cache	(boolean) Return cached data if available, defaults to TRUE. Use FALSE to fetch updated data.
write_cache	(boolean) Write data to cache, defaults to FALSE. Use TRUE to write data to cache for later use. Can also be set using options(hs_write_cache=TRUE)

Details

`get_fires_emissions` invisibly returns a tibble with weekly and year-to-date cumulative wildfire emissions by pollutant, sourced from the GWIS emissions API. Historical minimum, maximum, and average baselines are included for comparison. Data coverage begins in 2012. Filter on the `plt` column to select a specific pollutant (e.g. `filter(plt == "CO2")`). Cache filenames include the place and year so that multiple queries can be cached independently.

Value

Invisibly returns a tibble with weekly and cumulative wildfire emissions data for the requested place and year. The `plt` column identifies the pollutant (e.g. `"CO2"`, `"CO"`, `"PM2.5"`). Weekly value columns include `curv` (current), `minv`, `maxv`, and `avgv`. Cumulative columns are prefixed with `cum_`.

Author(s)

Hernando Cortina, <hch@alum.mit.edu>

References

Global Wildfire Information System (GWIS), Copernicus Emergency Management Service / European Commission Joint Research Centre. https://gwis.jrc.ec.europa.eu/apps/gwis.statistics/seasonaltrend

Examples

```
# Fetch global emissions data for the current year from cache if available:
emissions <- get_fires_emissions()
#
# Fetch CO2 emissions for the USA in 2023:
emissions_usa <- get_fires_emissions(place = 'USA', year = 2023)
#
# Force cache refresh:
emissions <- get_fires_emissions(use_cache = FALSE)
#
# Review cache contents and last update dates:
hockeystick_cache_details()
```

---

get_hurricanes *Download and plot essential climate data*

Description

Retrieves Atlantic basin hurricane data since 1851 from National Oceanic and Atmospheric Administration HURDAT Atlantic Hurricane Database Re-analysis Project. https://www.aoml.noaa.gov/hrd/hurdat/Data_Storm.html

Usage

```
get_hurricanes(use_cache = TRUE, write_cache = getOption("hs_write_cache"))
```

Arguments

use_cache	(boolean) Return cached data if available, defaults to TRUE. Use FALSE to fetch updated data.
write_cache	(boolean) Write data to cache, defaults to FALSE. Use TRUE to write data to cache for later use. Can also be set using options(hs_write_cache=TRUE)

Details

get_hurricanes invisibly returns a tibble with NOAA's annual North Atlantic revised HURDAT hurricane data since 1851. The returned object includes Year, and number of named storms, hurricanes, major hurricanes (S-S scale 3-5), Accumulated Cyclone Energy (ACE), and U.S. hurricane strikes.

ACE is an index that combines the number of systems, how long they existed and how intense they became. It is calculated by squaring the maximum sustained surface wind in the system every six hours that the cyclone is a Named Storm and summing it up for the season. Please refer to above website for details.

Value

Invisibly returns a tibble with the annual HURDAT hurricane data since 1851

Author(s)

Hernando Cortina, <hch@alum.mit.edu>

References

HURDAT North Atlantic Hurricane Database Re-analysis Project, Hurricane Research Division, NOAA https://www.aoml.noaa.gov/hrd/hurdat/Data_Storm.html.

Data from: https://www.aoml.noaa.gov/hrd/hurdat/comparison_table.html

https://en.wikipedia.org/wiki/Accumulated_cyclone_energy

Examples

```
# Fetch from cache if available:
hurricanes <- get_hurricanes()
#
# Force cache refresh:
hurricanes <- get_hurricanes(use_cache=FALSE)
#
# Review cache contents and last update dates:
hockeystick_cache_details()
#
# Plot output using package's built-in ggplot2 settings
plot_hurricanes(hurricanes)
```

`get_icecurves` *Download and plot essential climate data*

Description

Retrieves Arctic or Antarctic monthly Sea Ice Index time series (in million square km). Source is the National Snow and Ice Data Center, defaults to Arctic (Northern Hemisphere) monthly sea ice extent since 1979. https://nsidc.org/data/explore-data

Usage

```
get_icecurves(
  pole = "N",
  measure = "extent",
  use_cache = TRUE,
  write_cache = getOption("hs_write_cache")
)
```

Arguments

`pole`	'N' for Arctic or 'S' for Antarctic
`measure`	Must be 'extent' or 'area', defaults to 'extent'. Please see terminology link in references for details.
`use_cache`	(boolean) Return cached data if available, defaults to TRUE. Use FALSE to fetch updated data, or to change pole or month in cache.
`write_cache`	(boolean) Write data to cache, defaults to FALSE. Use TRUE to write data to cache for later use. Can also be set using options(hs_write_cache=TRUE)

Value

Invisibly returns a tibble with the series of monthly Sea Ice Index since 1979 (in million square km).

`get_icecurves` invisibly returns a tibble with time series of monthly Sea Ice Index since 1979 (in million square km).

User may select monthly Arctic or Antarctic sea ice extent or area (in millions of square kilometers). Defaults to Arctic sea ice extent. https://nsidc.org/sea-ice-today/about-data#area_extent

Author(s)

Hernando Cortina, <hch@alum.mit.edu>

References

- NSIDC Data Archive: https://nsidc.org/data/explore-data
- All About Sea Ice: https://nsidc.org/learn/parts-cryosphere/sea-ice
- Sea Ice terminology: https://nsidc.org/learn

Examples

```
# Fetch monthly sea ice history from cache if available:
icecurves <- get_icecurves()
#
# Force cache refresh:
icecurves <- get_icecurves(use_cache = FALSE)
# change region
south_icecurves <- get_icecurves(pole='S',  use_cache = FALSE)
#
# Review cache contents and last update dates:
hockeystick_cache_details()
#
# Plot output using package's built-in ggplot2 settings
plot_icecurves(icecurves)
```

`get_methane` *Download and plot essential climate data*

Description

Retrieves globally averaged marine surface methane monthly mean data from National Oceanic and Atmospheric Administration. The Global Monitoring Division of NOAA's Earth System Research Laboratory has measured methane since 1983 at a globally distributed network of air sampling sites. A global average is constructed by first smoothing the data for each site as a function of time, and then smoothed values for each site are plotted as a function of latitude. Global means are calculated from the latitude plot at each time step. https://gml.noaa.gov/ccgg/trends_ch4/ https://gml.noaa.gov/ccgg/about/global_means.html

Usage

```
get_methane(use_cache = TRUE, write_cache = getOption("hs_write_cache"))
```

Arguments

`use_cache`	(boolean) Return cached data if available, defaults to TRUE. Use FALSE to fetch updated data.
`write_cache`	(boolean) Write data to cache, defaults to FALSE. Use TRUE to write data to cache for later use. Can also be set using options(hs_write_cache=TRUE)

Details

get_methane invisibly returns a tibble with NOAA's monthly globally averaged methane measurement. The returned object includes year, month, date, average, average uncertainty, trend, and trend uncertainty columns. Trend is NOAA's published trend. Please refer to above website for details. CH4 expressed as a mole fraction in dry air, nanomol/mol, abbreviated as ppb.

Value

Invisibly returns a tibble with the monthly methane series

Author(s)

Hernando Cortina, <hch@alum.mit.edu>

References

Lan, X., K.W. Thoning, and E.J. Dlugokencky: Trends in globally-averaged CH4, N2O, and SF6 determined from NOAA Global Monitoring Laboratory measurements. Version 2022-11, doi:10.15138/P8XGAA10

Examples

```
# Fetch from cache if available, otherwise download:
ch4 <- get_methane()
#
# Force fetch from source:
ch4 <- get_methane(use_cache=FALSE)
#
# Review cache contents and last update dates:
hockeystick_cache_details()
#
# Plot output using package's built-in ggplot2 settings
plot_methane(ch4)
```

---

get_paleo *Download and plot essential climate data*

Description

Retrieves historical carbon dioxide and temperature records from the Vostok ice core during the past 420,000 years. Source of data is the U.S. Department of Energy's (DOE) Environmental System Science Data Infrastructure for a Virtual Ecosystem (ESS-DIVE). https://ess-dive.lbl.gov/ and https://data.ess-dive.lbl.gov/datasets/doi:10.3334/CDIAC/CLI.006

Usage

```
get_paleo(use_cache = TRUE, write_cache = getOption("hs_write_cache"))
```

Arguments

use_cache	(boolean) Return cached data if available, defaults to TRUE. Use FALSE to fetch updated data.
write_cache	(boolean) Write data to cache, defaults to FALSE. Use TRUE to write data to cache for later use. Can also be set using options(hs_write_cache=TRUE)

Value

Invisibly returns a tibble with the age of the ice (years before C.E.), carbon dioxide (ppm) and temperature (degrees C).

get_paleo invisibly returns a tibble with Vostok ice core data: the age of the ice (years before C.E.), carbon dioxide (ppm) and temperature (degrees C).

Data are from: Barnola J; Raynaud D; Lorius C; Barkov N (2003): Historical Carbon Dioxide Record from the Vostok Ice Core (417,160 - 2,342 years BP) and Petit J R ; Raynaud D ; Lorius C ; Delaygue G ; Jouzel J ; Barkov N I ; Kotlyakov V M (2000): Historical Isotopic Temperature Record from the Vostok Ice Core. CDIAC.

Author(s)

Hernando Cortina, <hch@alum.mit.edu>

References

- Historical Carbon Dioxide Record from the Vostok Ice Core (US Dept of Energy): https://data.ess-dive.lbl.gov/datasets/doi:10.3334/CDIAC/CLI.006
- Petit J R ; Raynaud D ; Lorius C ; Delaygue G ; Jouzel J ; Barkov N I ; Kotlyakov V M (2000): Historical Isotopic Temperature Record from the Vostok Ice Core. CDIAC. doi:10.3334/CDIAC/CLI.006. https://data.ess-dive.lbl.gov/view/doi:10.3334/CDIAC/CLI.006
- Barnola J ; Raynaud D ; Lorius C ; Barkov N (2003): Historical Carbon Dioxide Record from the Vostok Ice Core (417,160 - 2,342 years BP). None. doi:10.3334/CDIAC/ATG.009 https://data.ess-dive.lbl.gov/view/doi:10.3334/CDIAC/ATG.009

Examples

```
# Fetch Vostok paleo carbon and temperature data from cache if available:
vostok <- get_paleo()
#
# Force cache refresh:
vostok <- get_paleo(use_cache=FALSE)
#
# Review cache contents and last update dates:
hockeystick_cache_details()
#
# Plot output using package's built-in ggplot2 settings
plot_paleo(vostok)
```

get_seaice *Download and plot essential climate data*

Description

Retrieves Arctic or Antarctic annual Sea Ice Index (in million square km). Source is the National Snow and Ice Data Center, defaults to Arctic (Northern Hemisphere) July sea ice extent. https://nsidc.org/data/explore-data

Usage

```
get_seaice(
  pole = "N",
  month = 7,
  measure = "extent",
  use_cache = TRUE,
  write_cache = getOption("hs_write_cache")
)
```

Arguments

pole	'N' for Arctic or 'S' for Antarctic
month	2-digit month to retrieve sea ice for, defaults to 7 (July)
measure	Must be 'extent' or 'area', defaults to 'extent'. Please see terminology link in references for details.
use_cache	(boolean) Return cached data if available, defaults to TRUE. Use FALSE to fetch updated data, or to change pole or month in cache.
write_cache	(boolean) Write data to cache, defaults to FALSE. Use TRUE to write data to cache for later use. Can also be set using options(hs_write_cache=TRUE)

Value

Invisibly returns a tibble with the annual series of monthly Sea Ice Index since 1979 (in million square km).

get_seaice invisibly returns a tibble with annual series of monthly Sea Ice Index since 1979 (in million square km).

User may select Arctic or Antarctic sea ice extent or area (in millions of square kilometers) by year for a given month, specified by argument month. Defaults to Arctic July sea ice extent. https://nsidc.org/sea-ice-today/about-data#area_extent

Author(s)

Hernando Cortina, <hch@alum.mit.edu>

References

- NSIDC Data Archive: https://nsidc.org/data/explore-data
- All About Sea Ice: https://nsidc.org/learn/parts-cryosphere/sea-ice
- Sea Ice terminology: https://nsidc.org/learn

Examples

```
# Fetch sea ice from cache if available:
seaice <- get_seaice()
#
# Force cache refresh:
seaice <- get_seaice(use_cache = FALSE)
# change region and month
seaice <- get_seaice(pole='S', month=9, use_cache = FALSE)
#
# Review cache contents and last update dates:
hockeystick_cache_details()
#
# Plot output using package's built-in ggplot2 settings
plot_seaice(seaice)
```

`get_sealevel` *Download and plot essential climate data*

Description

Retrieves global mean sea level (GMSL) data from historic tide gauge and recent satellite altimeter observations (in mm). Source for tide gauge data is Commonwealth Scientific and Industrial Research Organisation (CSIRO) as described in Church and White (2011). https://research.csiro.au/slrwavescoast/sea-level/measurements-and-data/sea-level-data/

Source for satellite data is NOAA Laboratory for Satellite Altimetry: https://www.star.nesdis.noaa.gov/socd/lsa/SeaLevelRise/

Usage

```
get_sealevel(use_cache = TRUE, write_cache = getOption("hs_write_cache"))
```

Arguments

`use_cache`	(boolean) Return cached data if available, defaults to TRUE. Use FALSE to fetch updated data.
`write_cache`	(boolean) Write data to cache, defaults to FALSE. Use TRUE to write data to cache for later use. Can also be set using options(hs_write_cache=TRUE)

Value

Invisibly returns a tibble with the global mean sea level time series (in mm) over 1880-2013 using tide gauges and since 1993 for satellite measurements.

`get_sealevel` invisibly returns a tibble with mean sea level in mm time series from tide gauges and satellite observations.

The satellite observations have been releveled so that their mean level in 1993 matches that of the tide gauges.

The tide gauge data are no longer updated and cover the period from 1880 to 2013, per Church, J. A. and N.J. White (2011) https://research.csiro.au/slrwavescoast/sea-level/measurements-and-data/sea-level-data/

Satellite data are updated monthly or more frequently from the NOAA Laboratory for Satellite Altimetry. TOPEX and Jason-1,-2,-3 satellites dataset, with seasonal signals removed. https://www.star.nesdis.noaa.gov/socd/lsa/SeaLevelRise/

Author(s)

Hernando Cortina, <hch@alum.mit.edu>

References

- CSIRO reconstructed tide gauge GMSL for 1880 to 2009: https://research.csiro.au/slrwavescoast/sea-level/measurements-and-data/sea-level-data/
- Church, J. A. and N.J. White (2011), Sea-level rise from the late 19th to the early 21st Century. *Surveys in Geophysics*, doi:10.1007/s10712-011-9119-1. https://link.springer.com/article/10.1007/s10712-011-9119-1
- NOAA Laboratory for Satellite Altimetry https://www.star.nesdis.noaa.gov/socd/lsa/SeaLevelRise/

Examples

```
# Fetch sea level from cache if available:
gmsl <- get_sealevel()
#
# Force cache refresh:
gmsl <- get_sealevel(use_cache=FALSE)
#
# Review cache contents and last update dates:
hockeystick_cache_details()
#
# Plot output using package's built-in ggplot2 settings
plot_sealevel(gmsl)
```

get_temp *Download and plot essential climate data*

Description

Retrieves the combined global land- and sea-surface temperature anomaly (Land-Ocean Temperature Index, LOTI). Source is NASA/GISS Surface Temperature Analysis (GISTEMP v4), an estimate of global surface temperature change. https://data.giss.nasa.gov/gistemp/

Usage

```
get_temp(use_cache = TRUE, write_cache = getOption("hs_write_cache"))
```

Arguments

use_cache	(boolean) Return cached data if available, defaults to TRUE. Use FALSE to fetch updated data.
write_cache	(boolean) Write data to cache, defaults to FALSE. Use TRUE to write data to cache for later use. Can also be set using options(hs_write_cache=TRUE)

Value

Invisibly returns a tibble with the annual mean and monthly Combined Land-Surface Air and Sea-Surface Water Temperature Anomalies.

get_temp invisibly returns a tibble with the NASA/GISS annual mean and monthly global temperature anomaly. Data are global from 1880 to present, and represent the deviations from the 1951-1980 mean.

The returned object includes monthly and annual average anomalies, as well as seasonal anomalies. GISS Surface Temperature Analysis (GISTEMP v4) is an estimate of global surface temperature change.

Data are updated around the middle of every month using current data files from NOAA GHCN v4 (meteorological stations) and ERSST v5 (ocean areas). Station data are combined as described in Hansen et al. (2010) https://data.giss.nasa.gov/gistemp/references.html and Lenssen et al. (2019) https://www.giss.nasa.gov/pubs/abs/le05800h.html

Author(s)

Hernando Cortina, <hch@alum.mit.edu>

References

- GISS Surface Temperature Analysis (GISTEMP v4): https://data.giss.nasa.gov/gistemp/
- GISTEMP Team, 2020: *GISS Surface Temperature Analysis (GISTEMP), version 4.* NASA Goddard Institute for Space Studies.
- Lenssen, N., G. Schmidt, J. Hansen, M. Menne, A. Persin, R. Ruedy, and D. Zyss, 2019: Improvements in the GISTEMP uncertainty model. *J. Geophys. Res. Atmos.*, 124, no. 12, 6307-6326, doi:10.1029/2018JD029522. https://www.giss.nasa.gov/pubs/abs/le05800h.html

Examples

```
# Fetch temp anomaly from cache if available:
anomaly <- get_temp()
#
# Force cache refresh:
anomaly <- get_temp(use_cache=FALSE)
#
# Review cache contents and last update dates:
hockeystick_cache_details()
#
# Plot output using package's built-in ggplot2 settings
plot_temp(anomaly)
```

`get_temp2k` *Download and plot essential climate data*

Description

Retrieves the Common Era Global Surface Temperature Reconstructions. Source is PAGES2k Consortium and NOAA National Centers for Environmental Information. https://www.ncei.noaa.gov/access/paleo-search/study/26872

Usage

```
get_temp2k(use_cache = TRUE, write_cache = getOption("hs_write_cache"))
```

Arguments

`use_cache`	(boolean) Return cached data if available, defaults to TRUE. Use FALSE to fetch updated data.
`write_cache`	(boolean) Write data to cache, defaults to FALSE. Use TRUE to write data to cache for later use. Can also be set using options(hs_write_cache=TRUE)

Value

Invisibly returns a tibble with filtered and unfiltered temperature reconstructions and Cowtan & Way instrumental temperatures.

`get_temp2k` invisibly returns a tibble with the PAGES2k Consortium temperature reconstruction (years 1-2000 CE) and instrumental record (years 1850-2017 CE). Temperatures represent deviations from the 1961-1990 mean.

The returned object includes annual average temperature anomalies as well as filtered anomalies using a 31-year Butterworth filter. Reconstructions use seven different statistical methods that draw from a global collection of temperature-sensitive palaeoclimate records.

Methodology described in PAGES2k (2019) https://www.nature.com/articles/s41561-019-0400-0

Author(s)

Hernando Cortina, <hch@alum.mit.edu>

References

- PAGES2k Common Era Surface Temperature Reconstructions. https://www.ncei.noaa.gov/access/paleo-search/study/26872
- PAGES 2k Consortium., Neukom, R., Barboza, L.A. et al. Consistent multidecadal variability in global temperature reconstructions and simulations over the Common Era. *Nat. Geosci.* 12, 643–649 (2019). doi:10.1038/s4156101904000

Examples

```
# Fetch temp anomaly from cache if available:
anomaly <- get_temp2k()
#
# Force cache refresh:
anomaly <- get_temp2k(use_cache=FALSE)
#
# Review cache contents and last update dates:
hockeystick_cache_details()
#
# Plot output using package's built-in ggplot2 settings
plot_temp2k(anomaly)
```

---

hockeystick_cache *Manage cached datasets*

Description

Manage cached datasets

Usage

```
hockeystick_cache_list()

hockeystick_cache_delete(files, force = TRUE)

hockeystick_cache_delete_all(force = TRUE)

hockeystick_cache_details(files = NULL)

hockeystick_update_all()
```

Arguments

files	(character) one or more complete file names
force	(logical) Should files be force deleted? Default: TRUE

Details

cache_delete only accepts 1 file name, while cache_delete_all doesn't accept any names, but deletes all files. For deleting many specific files, use cache_delete in a lapply() type call

We cache using tools::R_user_dir(), find your cache folder by executing tools::R_user_dir("hockeystick","cache'

Value

hockeystick_cache_list() returns a character vector of full path filenames in cache.

hockeystick_cache_delete() no return value, called for side effect.

hockeystick_cache_delete_all() no return value, called for side effect.

hockeystick_cache_details() returns list of filenames and sizes of cached files.

hockeystick_update_all() updates all datasets and caches them. No return value, called for side effect.

Functions

- hockeystick_cache_list() returns a character vector of full path file names in cache
- hockeystick_cache_delete() deletes one or more files, returns nothing
- hockeystick_cache_delete_all() delete all files, returns nothing
- hockeystick_cache_details() prints file name and file size of each file, supply with one or more files, or no files (and get details for all available)
- hockeystick_update_all() updates the cache with the latest co2, temperature, sea level, and sea ice data.

References

Caching data sets: ROpenSci guide to persistent config and data for R packages: https://blog.r-hub.io/2020/03/12/user-preferences/

Examples

```
# list files in cache
hockeystick_cache_list()

# List info for single files
hockeystick_cache_details(files = hockeystick_cache_list()[1])
hockeystick_cache_details(files = hockeystick_cache_list()[2])

# List info for all files
hockeystick_cache_details()

# Delete cached files by name
hockeystick_cache_delete(files = hockeystick_cache_list()[1])

# Update all hockeystick data and place in cache
hockeystick_update_all()
```

```
# Delete all cached data
hockeystick_cache_delete_all()
```

`hscache_path` *Return path of data cache directory*

Description

Internal Function

Usage

```
hscache_path()
```

Value

Return path of data cache directory

`merge_carbontemp` *Download and plot essential climate data*

Description

Merge NOAA carbon and NASA temperature datasets on common dates.

Usage

```
merge_carbontemp(carbon = get_carbon(), temp = get_temp())
```

Arguments

`carbon`	Name of the tibble generated by `get_carbon`
`temp`	Name of the tibble generated by `get_temp`

Details

`merge_carbontemp` invisibly returns a tibble with the merged data from from `get_carbon` and `get_temp` functions. Tibble only includes data from dates when both datasets are available, essentially from 1960.

Value

Invisibly returns a tibble with merged datasets from `get_carbon` and `get_temp` functions.

Author(s)

Hernando Cortina, <hch@alum.mit.edu>

Examples

```
# Create merged tibble

mergedcarbontemp <- merge_carbontemp()
```

`plot_carbon` *Download and plot essential climate data*

Description

Plots carbon dioxide data retrieved using `get_carbon()` with ggplot2. The output ggplot2 object may be modified.

Usage

```
plot_carbon(dataset = get_carbon(), print = TRUE, annot = TRUE)
```

Arguments

`dataset`	Name of the tibble generated by `get_carbon`
`print`	(boolean) Display carbon dioxide ggplot2 chart, defaults to TRUE. Use FALSE to not display chart.
`annot`	(boolean) Include chart annotation with latest date and value, defaults to TRUE.

Details

`plot_carbon` invisibly returns a ggplot2 object with a pre-defined carbon dioxide chart using data from `get_carbon`. By default the chart is also displayed. Users may further modify the output ggplot2 chart.

Value

Invisibly returns a ggplot2 object with carbon dioxide chart

Author(s)

Hernando Cortina, <hch@alum.mit.edu>

Examples

```
# Fetch carbon dioxide data:
maunaloa <- get_carbon()
#
# Plot output using package's built-in ggplot2 defaults
plot_carbon(maunaloa)

# Or just call plot_carbon(), which defaults to get_carbon() dataset
plot_carbon()

p <- plot_carbon(maunaloa, print = FALSE)
# Modify plot such as: p + ggplot2::labs(title='The Keeling Curve')
```

---

plot_carbontemp *Download and plot essential climate data*

Description

Plots the global monthly mean temperature anomaly vs atmospheric carbon with ggplot2. The output ggplot2 object may be further modified.

Usage

```
plot_carbontemp(dataset = merge_carbontemp(), print = TRUE)
```

Arguments

dataset	Name of the tibble generated by merge_carbontemp
print	(boolean) Display temperature anomaly ggplot2 chart, defaults to TRUE. Use FALSE to not display chart.

Details

plot_carbontemp invisibly returns a ggplot2 object with a pre-defined temperature anomaly vs carbon chart using data from merge_carbontemp. By default the chart is also displayed. Users may further modify the output ggplot2 chart.

Value

Invisibly returns a ggplot2 object with temperature anomaly vs carbon chart

Author(s)

Hernando Cortina, <hch@alum.mit.edu>

Examples

```
# Fetch temperature anomaly:
mergedtemp <- merge_carbontemp()
#
# Plot output using package's built-in ggplot2 defaults
plot_carbontemp(mergedtemp)

# Or just call plot_carbontemp(), which defaults to merge_carbontemp() dataset
plot_carbontemp()

p <- plot_carbontemp(mergedtemp, print = FALSE)
# Modify plot such as: p + ggplot2::labs(title='The Signature of Climate Change')
```

---

plot_cmip6 *Download and plot essential climate data*

Description

Plots gridded CMIP6 climate projections retrieved using `get_cmip6()` with ggplot2.

Usage

```
plot_cmip6(
  dataset = get_cmip6(),
  palette = "Spectral",
  xlim = NULL,
  ylim = NULL,
  country = NULL,
  state = NULL,
  mask = TRUE,
  print = TRUE
)
```

Arguments

`dataset`	Name of the tibble generated by `get_cmip6`
`palette`	(string) Color palette name passed to RColorBrewer. Defaults to "Spectral" reversed for temperature-type variables.
`xlim`, `ylim`	(numeric) Optional two-element longitude/latitude bounds to zoom the map to a region, e.g. `xlim = c(-130, -60)`, `ylim = c(20, 70)`.
`country`	(string) Optional country name or ISO 3166-1 alpha-3 code to zoom the map to that country. Requires the `geodata` package.
`state`	(string) Optional state/province name (admin-1), used together with `country` to zoom to a subnational region, e.g. `country = "MEX"`, `state = "Sonora"`. Requires the `geodata` package.

mask	(boolean) If country is given, blank cells outside the country border, defaults to TRUE.
print	(boolean) Display map, defaults to TRUE.

Details

plot_cmip6 returns a pre-defined ggplot2 world map of projected values. Regions may be selected by coordinates (xlim/ylim), by country, or by state/province. Users may further modify the output chart.

Value

Invisibly returns a ggplot2 object with the projection map.

Examples

```
proj <- get_cmip6()
plot_cmip6(proj)
#
# Zoom to the contiguous United States. Explicit xlim/ylim override the
# country's bounding box, excluding Alaska and Hawaii from the view:
plot_cmip6(proj, country = 'USA', xlim = c(-125, -66), ylim = c(24, 50))
#
# Or zoom to a single state:
plot_cmip6(proj, country = 'MEX', state = 'Sonora')
```

---

plot_cmip6_anom *Download and plot essential climate data*

Description

Plots expected climate change (projection minus 1970-2000 baseline) computed by get_cmip6_anom() with a diverging color scale centered on zero.

Usage

```
plot_cmip6_anom(
  dataset = get_cmip6_anom(),
  zero_centered = FALSE,
  xlim = NULL,
  ylim = NULL,
  country = NULL,
  state = NULL,
  mask = TRUE,
  print = TRUE
)
```

Arguments

dataset	Name of the tibble generated by get_cmip6_anom
zero_centered	(boolean) Use a symmetric diverging scale around zero, defaults to FALSE (scale runs from zero up).
xlim, ylim	(numeric) Optional two-element longitude/latitude bounds to zoom the map to a region, e.g. xlim = c(-130, -60), ylim = c(20, 70). The color scale is recomputed on the visible region.
country	(string) Optional country name or ISO 3166-1 alpha-3 code to zoom the map to that country. Requires the geodata package.
state	(string) Optional state/province name (admin-1), used together with country to zoom to a subnational region, e.g. country = "MEX", state = "Sonora". Requires the geodata package.
mask	(boolean) If country is given, blank cells outside the country border, defaults to TRUE.
print	(boolean) Display map, defaults to TRUE.

Details

plot_cmip6_anom returns a pre-defined ggplot2 world map of expected change in degrees C (or percent for precipitation variables), with blue for cooling/drying and red for warming/wetting. Precipitation maps with both increases and decreases automatically use the diverging scale even when zero_centered = FALSE. Users may further modify the output chart.

Value

Invisibly returns a ggplot2 object with the change map.

Author(s)

Hernando Cortina, <hch@alum.mit.edu>

References

See get_cmip6.

Examples

```
plot_cmip6_anom()
```

plot_dailytemp *Download and plot essential climate data*

Description

Plots the daily temperatures since 1940 and current anomaly data retrieved using `get_dailytempcop()` with ggplot2. The output ggplot2 object may be further modified.

Usage

```
plot_dailytemp(
  dataset = get_dailytempcop(),
  print = TRUE,
  anomaly = FALSE,
  maxtemp = FALSE,
  current_year = as.numeric(substr(Sys.Date(), 1, 4)),
  title_lab = "Daily Average Air Temperature",
  cop = TRUE
)
```

Arguments

dataset	Name of the tibble generated by `get_dailytempcop` or `get_dailytemp`
print	(boolean) Display daily temperature ggplot2 chart, defaults to TRUE. Use FALSE to not display chart.
anomaly	(boolean) Display current anomaly versus historic mean, defaults to FALSE.
maxtemp	(boolean) Display current deviation versus historic max, defaults to FALSE.
current_year	(numeric) Year to highlight in alternate color, defaults to current year.
title_lab	(string) Title to override default chart title. Default title pulls region name from dataset attributes.
cop	(boolean) Flag for chart caption, TRUE = Copernicus, FALSE =. ClimateReanalyzer.org

Details

`plot_temp` invisibly returns a ggplot2 object with a pre-defined daily temperature anomaly chart using data from `get_dailytemp`. By default the chart is also displayed. Plots one line per year, as well as mean and anomaly (which may be disabled). Users may further modify the output ggplot2 chart.

Value

Invisibly returns a ggplot2 object with daily temperature anomaly chart

Author(s)

Hernando Cortina, <hch@alum.mit.edu>

Examples

```
# Fetch temperature anomaly:
dailydata <- get_dailytempcop()
#
# Plot output using package's built-in ggplot2 defaults
plot_dailytemp(dailydata)

# Don't plot anomaly shading and highight specific year
plot_dailytemp(anomaly = FALSE, current_year = 2012)

# Or just call plot_temp(), which defaults to get_dailytempcop() dataset
plot_dailytemp()

p <- plot_dailytemp(dailydata, print = FALSE)
# Modify plot such as: p + ggplot2::labs(title='Record Temperatures in 2023')
```

---

plot_emissions *Download and plot essential climate data*

Description

Plots carbon dioxide emissions retrieved using `get_emissions()` with ggplot2. The output ggplot2 object may be modified. Alternative columns from the dataset may also be plotted.

Usage

```
plot_emissions(
  dataset = get_emissions(),
  start_year = 1900,
  region = "World",
  field = "co2",
  print = TRUE,
  annot = TRUE,
  title_expression = expression("Fossil Combustion " * CO[2] * " Emissions"),
  yaxis_expression = expression("Gt " * CO[2] * " per year")
)
```

Arguments

`dataset`	Name of the tibble generated by `get_emissions`
`start_year`	Year to start plot at. Defaults to 1900. Data is available since 1750.
`region`	ISO code of region to plot. Defaults to 'OWID_WRL' which signifies entire world.

`field` Field from GCP dataset to be plotted, defaults to 'co2'

`print` (boolean) Display carbon dioxide emissions ggplot2 chart, defaults to TRUE. Use FALSE to not display chart.

`annot` (boolean) Include chart annotation with latest date and value, defaults to TRUE.

`title_expression`
Chart title, defaults to CO2 emissions

`yaxis_expression`
y-axis label, defaults to Gt CO2 emissions

Details

`plot_emissions` invisibly returns a ggplot2 object with a pre-defined carbon dioxide emissions chart using data from `get_emissions`. Use the `field` parameter to select alternative columns from the data set such as co2_per_capita. By default the chart is also displayed. Users may further modify the output ggplot2 chart.

Value

Invisibly returns a ggplot2 object with carbon dioxide emissions chart

Author(s)

Hernando Cortina, <hch@alum.mit.edu>

Examples

```
# Fetch carbon dioxide emissions:
emissions <- get_emissions()

# Plot output using package's built-in ggplot2 defaults
plot_emissions(emissions)

# Or just call plot_emissions(), which defaults to get_emissions() dataset
plot_emissions()

# You can also select a region by name and start year
plot_emissions(region='United States', start_year=1950)
p <- plot_emissions(emissions, print = FALSE)

# Modify plot such as: p + ggplot2::labs(title='Anthropogenic Carbon Emissions')

# Plot a different field from GCP dataset
plot_emissions(field='co2_per_capita', yaxis_expression=expression(CO[2]*' per capita' ))
```

plot_emissions_with_land

Download and plot essential climate data

Description

Plots carbon dioxide emissions (including land use change) retrieved using `get_emissions()` with ggplot2. The output ggplot2 object may be modified. Alternative columns from the dataset may also be plotted.

Usage

```
plot_emissions_with_land(
  dataset = get_emissions(),
  start_year = 1900,
  region = "World",
  print = TRUE,
  annot = TRUE,
 title_expression = expression("Fossil + Land Use Change " * CO[2] * " Emissions"),
  yaxis_expression = expression("Gt " * CO[2] * " per year")
)
```

Arguments

`dataset`	Name of the tibble generated by `get_emissions`
`start_year`	Year to start plot at. Defaults to 1900. Data is available since 1750.
`region`	ISO code of region to plot. Defaults to 'OWID_WRL' which signifies entire world.
`print`	(boolean) Display carbon dioxide emissions ggplot2 chart, defaults to TRUE. Use FALSE to not display chart.
`annot`	(boolean) Include chart annotation with latest date and value, defaults to TRUE.
`title_expression`	Chart title, defaults to CO2 emissions
`yaxis_expression`	y-axis label, defaults to Gt CO2 emissions

Details

`plot_emissions_with_land` invisibly returns a ggplot2 object with a pre-defined carbon dioxide emissions (including land use change) chart using data from `get_emissions`. By default the chart is also displayed. Users may further modify the output ggplot2 chart.

Value

Invisibly returns a ggplot2 object with carbon dioxide emissions chart

Author(s)

Hernando Cortina, <hch@alum.mit.edu>

Examples

```
# Fetch carbon dioxide emissions:
emissions <- get_emissions()

# Plot output (including land use change) using package's built-in ggplot2 defaults
plot_emissions_with_land(emissions)

# Or just call plot_emissions_with_land(), which defaults to get_emissions() dataset
plot_emissions_with_land()

# You can also select a region by name and starting year
plot_emissions_with_land(region='United States', start_year=1950)
p <- plot_emissions_with_land(emissions, print = FALSE)

# Modify plot such as: p + ggplot2::labs(title='Anthropogenic Carbon Emissions')
```

---

plot_fires_area *Download and plot essential climate data*

Description

Plots the wildfire burnt area or fire count data retrieved using `get_fires_area()` with ggplot2. The output ggplot2 object may be further modified.

Usage

```
plot_fires_area(
  dataset = get_fires_area(),
  var = c("area", "count"),
  style = c("cumulative", "weekly"),
  print = TRUE
)
```

Arguments

`dataset`	Name of the tibble generated by `get_fires_area`
`var`	(string) Select which variable to plot. May be `"area"` (burnt area in hectares, the default) or `"count"` (number of fire events).
`style`	(string) Select cumulative or weekly view. May be `"cumulative"` (year-to-date totals, the default) or `"weekly"` (weekly values).
`print`	(boolean) Display the ggplot2 chart, defaults to TRUE. Use FALSE to suppress display.

Details

plot_fires_area invisibly returns a ggplot2 object with a pre-defined wildfire chart using data from get_fires_area. By default the chart is also displayed. The shaded ribbon shows the historical minimum-to-maximum range since 2012, and the blue line shows the historical average. Users may further modify the output ggplot2 chart.

Value

Invisibly returns a ggplot2 object with the wildfire chart

Author(s)

Hernando Cortina, <hch@alum.mit.edu>

Examples

```
# Fetch wildfire data:
fires <- get_fires_area()
#
# Plot output using package's built-in ggplot2 defaults
plot_fires_area(fires)

# Or just call plot_fires_area(), which defaults to get_fires_area() dataset
plot_fires_area()

# Plot weekly fire count instead of cumulative burnt area
plot_fires_area(fires, var = 'count', style = 'weekly')

p <- plot_fires_area(fires, print = FALSE)
# Modify plot such as: p + ggplot2::labs(title='Global Wildfire Burnt Area')
```

---

plot_fires_emissions *Download and plot essential climate data*

Description

Plots the wildfire greenhouse gas emissions data retrieved using get_fires_emissions() with ggplot2. The output ggplot2 object may be further modified.

Usage

```
plot_fires_emissions(
  dataset = get_fires_emissions(),
 pollutant = c("CO2", "BC", "C", "CH4", "CO", "NMHC", "NOX", "OC", "PM2.5", "SO2", "TC"),
  style = c("cumulative", "weekly"),
  print = TRUE
)
```

Arguments

dataset	Name of the tibble generated by get_fires_emissions
pollutant	(string) Select which pollutant to plot. May be one of "CO2" (the default), "BC", "C", "CH4", "CO", "NMHC", "NOX", "OC", "PM2.5", "SO2", or "TC".
style	(string) Select cumulative or weekly view. May be "cumulative" (year-to-date totals, the default) or "weekly" (weekly values).
print	(boolean) Display the ggplot2 chart, defaults to TRUE. Use FALSE to suppress display.

Details

plot_fires_emissions invisibly returns a ggplot2 object with a pre-defined wildfire emissions chart using data from get_fires_emissions. By default the chart is also displayed. The shaded ribbon shows the historical minimum-to-maximum range since 2003, and the blue line shows the historical average. Emissions are expressed in millions of tons. Users may further modify the output ggplot2 chart.

Value

Invisibly returns a ggplot2 object with the wildfire emissions chart

Author(s)

Hernando Cortina, <hch@alum.mit.edu>

Examples

```
# Fetch wildfire emissions data:
emissions <- get_fires_emissions()
#
# Plot output using package's built-in ggplot2 defaults
plot_fires_emissions(emissions)

# Or just call plot_fires_emissions(), which defaults to get_fires_emissions() dataset
plot_fires_emissions()

# Plot weekly PM2.5 emissions instead of cumulative CO2
plot_fires_emissions(emissions, pollutant = 'PM2.5', style = 'weekly')

p <- plot_fires_emissions(emissions, print = FALSE)
# Modify plot such as: p + ggplot2::labs(title='Global CO2 Emissions from Wildfires')
```

plot_hurricanes *Download and plot essential climate data*

Description

Plots the hurricane data retrieved using `get_hurricanes()` with ggplot2. The output ggplot2 object may be further modified.

Usage

```
plot_hurricanes(dataset = get_hurricanes(), cat = "major", print = TRUE)
```

Arguments

`dataset`	Name of the tibble generated by `get_hurricanes`
`cat`	(string) Select which category of hurricane to plot. May be "major", "hurricane", or "storm". Defaults to "major".
`print`	(boolean) Display hurricane ggplot2 chart, defaults to TRUE. Use FALSE to not display chart.

Details

`plot_hurricanes` invisibly returns a ggplot2 object with a pre-defined hurricane data chart using data from `get_hurricanes`. By default the chart is also displayed. Smooths using ggplot2's built-in loess smoother. Users may further modify the output ggplot2 chart. Categories may be "major" (category 3-5), "hurricane" (category 1-5), or "storm" (named storm).

Value

Invisibly returns a ggplot2 object with hurricanes chart

Author(s)

Hernando Cortina, <hch@alum.mit.edu>

Examples

```
# Fetch hurricane data:
hurricanes <- get_hurricanes()
#
# Plot output using package's built-in ggplot2 defaults
plot_hurricanes(hurricanes)

# Or just call plot_hurricanes(), which defaults to get_hurricanes() dataset
plot_hurricanes()

p <- plot_hurricanes(hurricanes, print = FALSE)
# Modify plot such as: p + ggplot2::labs(title='Growing number of North Atlantic named storms')
```

plot_hurricane_nrg *Download and plot essential climate data*

Description

Plots the hurricane energy data (ACE) retrieved using `get_hurricanes()` with ggplot2. The output ggplot2 object may be further modified.

Usage

```
plot_hurricane_nrg(dataset = get_hurricanes(), print = TRUE)
```

Arguments

`dataset`	Name of the tibble generated by `get_hurricanes`
`print`	(boolean) Display hurricane ggplot2 chart, defaults to TRUE. Use FALSE to not display chart.

Details

`plot_hurricane_nrg` invisibly returns a ggplot2 object with a pre-defined hurricane energy data chart using data from `get_hurricanes`. By default the chart is also displayed. Smooths using ggplot2's built-in loess smoother. Users may further modify the output ggplot2 chart.

ACE is an index that combines the number of systems, how long they existed and how intense they became. It is calculated by squaring the maximum sustained surface wind in the system every six hours that the cyclone is a Named Storm and summing it up for the season. Please refer to above website for details.

Value

Invisibly returns a ggplot2 object with hurricane energy chart

Author(s)

Hernando Cortina, <hch@alum.mit.edu>

Examples

```
# Fetch hurricane data:
hurricanes <- get_hurricanes()
#
# Plot output using package's built-in ggplot2 defaults
plot_hurricane_nrg(hurricanes)

# Or just call plot_hurricane_nrg(), which defaults to get_hurricanes() dataset
plot_hurricane_nrg()

p <- plot_hurricane_nrg(hurricanes, print = FALSE)
```

```
# Modify plot such as: p + ggplot2::labs(title='Accumulated Cyclone Energy')
```

`plot_icecurves` *Download and plot essential climate data*

Description

Plots the monthly Sea Ice Index data retrieved using `get_icecurves()` with ggplot2. The output ggplot2 object may be further modified.

Usage

```
plot_icecurves(dataset = get_icecurves(), print = TRUE)
```

Arguments

`dataset`	Name of the tibble generated by `get_icecurves`, defaults to calling `get_icecurves`
`print`	(boolean) Display sea ice ggplot2 chart, defaults to TRUE. Use FALSE to not display chart.

Details

`plot_icecurves` invisibly returns a ggplot2 object with a pre-defined Sea Ice Index chart using data from `get_icecurves`. By default the chart is also displayed. Users may further modify the output ggplot2 chart. Chart consists of one line per year showing monthly sea ice from January through December. Current year is highlighted.

Value

Invisibly returns a ggplot2 object with Sea Ice Index chart

Author(s)

Hernando Cortina, <hch@alum.mit.edu>

Examples

```
# Fetch historic monthly sea ice data since 1979:
icecurves <- get_icecurves()
#
# Plot output using package's built-in ggplot2 defaults
plot_icecurves(icecurves)

# Or just call plot_icecurves(), which defaults to get_icecurves() dataset
plot_icecurves()

p <- plot_icecurves(icecurves, print = FALSE)
# Modify plot such as: p + ggplot2::labs(title='Shrinking Arctic Sea Ice')
```

plot_methane *Download and plot essential climate data*

Description

Plots atmospheric methane data retrieved using `get_methane()` with ggplot2. The output ggplot2 object may be modified.

Usage

```
plot_methane(dataset = get_methane(), print = TRUE, annot = TRUE)
```

Arguments

dataset	Name of the tibble generated by get_methane
print	(boolean) Display methane ggplot2 chart, defaults to TRUE. Use FALSE to not display chart.
annot	(boolean) Include chart annotation with latest date and value, defaults to TRUE.

Details

`plot_methane` invisibly returns a ggplot2 object with a pre-defined methane chart using data from `get_methane`. By default the chart is also displayed. Users may further modify the output ggplot2 chart.

Value

Invisibly returns a ggplot2 object with methane chart

Author(s)

Hernando Cortina, <hch@alum.mit.edu>

Examples

```
# Fetch methane data:
ch4 <- get_methane()
#
# Plot output using package's built-in ggplot2 defaults
plot_methane(ch4)

# Or just call plot_methane(), which defaults to get_methane() dataset
plot_methane()

p <- plot_methane(ch4, print = FALSE)
# Modify plot such as: p + ggplot2::labs(title='Trend in Atmospheric Methane')
```

plot_paleo *Download and plot essential climate data*

Description

Plots the Vostok ice core data retrieved using `get_paleo()` with ggplot2. The output ggplot2 object may be further modified.

Usage

```
plot_paleo(dataset = get_paleo(), print = TRUE)
```

Arguments

`dataset`	Name of the tibble generated by `get_paleo`, defaults to calling `get_paleo`
`print`	(boolean) Display Vostok ice core ggplot2 chart, defaults to TRUE. Use FALSE to not display chart.

Details

`plot_paleo` invisibly returns a ggplot2 object with a pre-defined Vostok ice core chart using data from `get_paleo`. The returned chart stacks carbon dioxide concentration over temperature over 420,000 years. By default the chart is also displayed. Users may further modify the output ggplot2 chart.

Value

Invisibly returns a ggplot2 object with the Vostok chart

Author(s)

Hernando Cortina, <hch@alum.mit.edu>

References

Barnola, J.-M., D. Raynaud, C. Lorius, and N.I. Barkov. 2003. Historical CO2 record from the Vostok ice core. In Trends: A Compendium of Data on Global Change. Carbon Dioxide Information Analysis Center, Oak Ridge National Laboratory, U.S. Department of Energy, Oak Ridge, Tenn., U.S.A

Examples

```
# Fetch Vostok data:
vostok <- get_paleo()
#
# Plot output using package's built-in ggplot2 defaults
plot_paleo(vostok)
```

```
# Or just call plot_paleo(), which defaults to get_paleo() dataset
plot_paleo()

p <- plot_paleo(vostok, print = FALSE)
# Modify plot such as: p + patchwork::plot_annotation(title='A Long History of Carbon')
```

`plot_seaice` *Download and plot essential climate data*

Description

Plots the Sea Ice Index data retrieved using `get_seaice()` with ggplot2. The output ggplot2 object may be further modified.

Usage

```
plot_seaice(dataset = get_seaice(), title = "Arctic Sea Ice", print = TRUE)
```

Arguments

`dataset`	Name of the tibble generated by `get_seaice`, defaults to calling `get_seaice`
`title`	chart title, defaults to Arctic Sea Ice
`print`	(boolean) Display sea ice ggplot2 chart, defaults to TRUE. Use FALSE to not display chart.

Details

`plot_seaice` invisibly returns a ggplot2 object with a pre-defined Sea Ice Index chart using data from `get_seaice`. By default the chart is also displayed. Users may further modify the output ggplot2 chart.

Value

Invisibly returns a ggplot2 object with Sea Ice Index chart

Author(s)

Hernando Cortina, <hch@alum.mit.edu>

Examples

```
# Fetch sea ice data:
seaice <- get_seaice()
#
# Plot output using package's built-in ggplot2 defaults
plot_seaice(seaice)
```

```
# Or just call plot_seaice(), which defaults to get_seaice() dataset
plot_seaice()

p <- plot_seaice(seaice, print = FALSE)
# Modify plot such as: p + ggplot2::labs(title='Shrinking Arctic Sea Ice')
```

`plot_sealevel` *Download and plot essential climate data*

Description

Plots the global mean sea level data retrieved using `get_sealevel()` with ggplot2. The output ggplot2 object may be further modified.

Usage

```
plot_sealevel(dataset = get_sealevel(), print = TRUE)
```

Arguments

`dataset`	Name of the tibble generated by `get_sealevel`, defaults to calling `get_sealevel`
`print`	(boolean) Display sealevel ggplot2 chart, defaults to TRUE. Use FALSE to not display chart.

Details

`plot_sealevel` invisibly returns a ggplot2 object with a pre-defined sealevel change chart using data from `get_sealevel`. By default the chart is also displayed. Users may further modify the output ggplot2 chart.

Value

Invisibly returns a ggplot2 object with sealevel chart

Author(s)

Hernando Cortina, <hch@alum.mit.edu>

Examples

```
# Fetch sealevel data:
gmsl <- get_sealevel()
#
# Plot output using package's built-in ggplot2 defaults
plot_sealevel(gmsl)

# Or just call plot_sealevel(), which defaults to get_sealevel() dataset
plot_sealevel()
```

```
p <- plot_sealevel(gmsl, print = FALSE)
# Modify plot such as: p + ggplot2::labs(title='Rising Waters')
```

---

plot_temp *Download and plot essential climate data*

Description

Plots the global annual mean temperature anomaly retrieved using `get_temp()` with ggplot2. The output ggplot2 object may be further modified.

Usage

```
plot_temp(dataset = get_temp(), print = TRUE)
```

Arguments

`dataset`	Name of the tibble generated by `get_temp`
`print`	(boolean) Display temperature anomaly ggplot2 chart, defaults to TRUE. Use FALSE to not display chart.

Details

`plot_temp` invisibly returns a ggplot2 object with a pre-defined temperature annual mean anomaly chart using data from `get_temp`. By default the chart is also displayed. Smooths using ggplot2's built-in loess smoother. Users may further modify the output ggplot2 chart.

Value

Invisibly returns a ggplot2 object with temperature anomaly chart

Author(s)

Hernando Cortina, <hch@alum.mit.edu>

Examples

```
# Fetch temperature anomaly:
anomaly <- get_temp()
#
# Plot output using package's built-in ggplot2 defaults
plot_temp(anomaly)

# Or just call plot_temp(), which defaults to get_temp() dataset
plot_temp()

p <- plot_temp(anomaly, print = FALSE)
```

```
# Modify plot such as: p + ggplot2::labs(title='The Signature of Climate Change')
```

`plot_temp2k` *Download and plot essential climate data*

Description

Plots the Common Era 2000-year global temperature anomaly retrieved using `get_temp2k()` with ggplot2. The output ggplot2 object may be further modified.

Usage

```
plot_temp2k(
  dataset = get_temp2k(),
  instrumental = TRUE,
  filtered = TRUE,
  print = TRUE
)
```

Arguments

`dataset`	Name of the tibble generated by `get_temp2k`
`instrumental`	(boolean) Include the Cowtan & Way instrumental temperatures through 2017. Defaults to TRUE.
`filtered`	(boolean) Use the filtered temperatures provided by PAGES2k Consortium. Temperatures filtered using a 31-year Butterworth filter. Defaults to TRUE.
`print`	(boolean) Display temperature anomaly ggplot2 chart, defaults to TRUE. Use FALSE to not display chart.

Details

`plot_temp2k` invisibly returns a ggplot2 object with a pre-defined temperature anomaly chart using data from `get_temp2k`. By default the chart is also displayed. Users may further modify the output ggplot2 chart.

Value

Invisibly returns a ggplot2 object with temperature anomaly chart

Author(s)

Hernando Cortina, <hch@alum.mit.edu>

Examples

```
# Fetch temperature anomaly:
anomaly <- get_temp2k()
#
# Plot output using package's built-in ggplot2 defaults
plot_temp2k(anomaly)

# Or just call plot_temp2k(), which defaults to get_temp2k() dataset
plot_temp2k()

p <- plot_temp2k(anomaly, print = FALSE)
# Modify plot such as: p + ggplot2::labs(title='Temperature Increase in the Common Era')
```

---

plot_temp_monthly *Download and plot essential climate data*

Description

Plots the monthly mean temperature anomaly retrieved using `get_temp()` with ggplot2. The output ggplot2 object may be further modified.

Usage

```
plot_temp_monthly(dataset = get_temp(), print = TRUE)
```

Arguments

`dataset`	Name of the tibble generated by `get_temp`
`print`	(boolean) Display temperature anomaly ggplot2 chart, defaults to TRUE. Use FALSE to not display chart.

Details

`plot_temp_monthly` invisibly returns a ggplot2 object with a pre-defined temperature monthly mean anomaly chart using data from `get_temp`. By default the chart is also displayed. Smooths using ggplot2's built-in loess smoother. Users may further modify the output ggplot2 chart.

Value

Invisibly returns a ggplot2 object with temperature anomaly chart

Author(s)

Hernando Cortina, <hch@alum.mit.edu>

Examples

```
# Fetch temperature anomaly:
anomaly <- get_temp()
#
# Plot output using package's built-in ggplot2 defaults
plot_temp_monthly(anomaly)

# Or just call plot_temp_monthly(), which defaults to get_temp() dataset
plot_temp_monthly()

p <- plot_temp_monthly(anomaly, print = FALSE)
# Modify plot such as: p + ggplot2::labs(title='The Signature of Climate Change')
```

---

plot_temp_scatter *Download and plot essential climate data*

Description

Plots scatter of monthly temperature anomaly retrieved using `get_temp()` with ggplot2. The output ggplot2 object may be further modified.

Usage

```
plot_temp_scatter(
  dataset = get_temp(),
  print = TRUE,
  labelmax = FALSE,
  labellatest = TRUE
)
```

Arguments

`dataset`	Name of the tibble generated by `get_temp`
`print`	(boolean) Display temperature anomaly ggplot2 chart, defaults to TRUE. Use FALSE to not display chart.
`labelmax`	(boolean) Display date of max value, default = FALSE
`labellatest`	(boolean) Display date of latest value, default = TRUE

Details

`plot_temp_scatter` invisibly returns a ggplot2 object with a pre-defined temperature monthly mean anomaly chart using data from `get_temp`. By default the chart is also displayed. Smooths using ggplot2's built-in loess smoother. Users may further modify the output ggplot2 chart.

Value

Invisibly returns a ggplot2 object with temperature anomaly chart

Author(s)

Hernando Cortina, <hch@alum.mit.edu>

Examples

```
# Fetch temperature anomaly:
anomaly <- get_temp()
#
# Plot output using package's built-in ggplot2 defaults
plot_temp_scatter(anomaly)

# Or just call plot_temp_scatter(), which defaults to get_temp() dataset
plot_temp_scatter()

p <- plot_temp_scatter(anomaly, print = FALSE)
# Modify plot such as: p + ggplot2::labs(title='The Signature of Climate Change')
```

---

print.hockeystick_cache_info

Display data cache info Shows filenames and cache file sizes

Description

Display data cache info Shows filenames and cache file sizes

Usage

```
## S3 method for class 'hockeystick_cache_info'
print(x, ...)
```

Arguments

x	filenames
...	Additional parameters

Value

Display data cache info. No return value, called for side effect.

`warming_stripes` *Download and plot essential climate data*

Description

Plots global "warming stripes" graph in the style popularized by Ed Hawkins, based on temperature anomaly retrieved using `get_temp()`. Function can output stripes chart with legend or a minimal chart. The output ggplot2 object may be further modified.

Usage

```
warming_stripes(
  dataset = get_temp(),
  stripe_only = FALSE,
  col_strip = RColorBrewer::brewer.pal(11, "RdBu"),
  print = TRUE
)
```

Arguments

`dataset`	Name of the tibble generated by `get_temp`
`stripe_only`	Display legend and axes, defaults to TRUE
`col_strip`	Color palette to use. Defaults to Red-Blue RColorBrewer palette.
`print`	(boolean) Display warming stripe ggplot2 chart, defaults to TRUE. Use FALSE to not display chart.

Details

`warming_stripes` invisibly returns a ggplot2 object with warming stripes chart using data from `get_temp`. By default the chart is also displayed. User may modify color palette or remove axes and legend. Users may further modify the output ggplot2 chart.

Value

Invisibly returns a ggplot2 object with warming stripes

Author(s)

Hernando Cortina, <hch@alum.mit.edu>

References

- Climate Lab. 2018. https://www.climate-lab-book.ac.uk/2018/warming-stripes/
- GISS Surface Temperature Analysis (GISTEMP v4): https://data.giss.nasa.gov/gistemp/
- GISTEMP Team, 2020: *GISS Surface Temperature Analysis (GISTEMP), version 4.* NASA Goddard Institute for Space Studies.
- Dr. Dominic Roye blog post "How to Create Warming Stripes in R": https://dominicroye.github.io/en/2018/how-to-create-warming-stripes-in-r/

Examples

```
# Draw with axes and legend
stripes <- warming_stripes()

# Draw stripes only
stripes <- warming_stripes(stripe_only = TRUE)

# Don't display, store for further modifications
stripes <- warming_stripes(print = FALSE)

# Change color palette
stripes <- warming_stripes(stripe_only = TRUE, col_strip = viridisLite::viridis(11))
```

Index

climate_grid, 3

emissions_map, 4

file_info_, 5

get_carbon, 6
get_cmip6, 7, *9, 10*, *35*
get_cmip6_anom, 9
get_dailytemp, 10
get_dailytempcop, 12
get_emissions, 13
get_fires_area, 14
get_fires_emissions, 16
get_hurricanes, 17
get_icecurves, 19
get_methane, 20
get_paleo, 21
get_seaice, 23
get_sealevel, 24
get_temp, 26
get_temp2k, 27
getsize, 5

hockeystick_cache, 28
hockeystick_cache_delete
        *(*hockeystick_cache*)*, 28
hockeystick_cache_delete_all
        *(*hockeystick_cache*)*, 28
hockeystick_cache_details
        *(*hockeystick_cache*)*, 28
hockeystick_cache_list
        *(*hockeystick_cache*)*, 28
hockeystick_update_all
        *(*hockeystick_cache*)*, 28
hscache_path, 30

lapply(), *29*

merge_carbontemp, 30

plot_carbon, 31
plot_carbontemp, 32
plot_cmip6, 33
plot_cmip6_anom, 34
plot_dailytemp, 36
plot_emissions, 37
plot_emissions_with_land, 39
plot_fires_area, 40
plot_fires_emissions, 41
plot_hurricane_nrg, 44
plot_hurricanes, 43
plot_icecurves, 45
plot_methane, 46
plot_paleo, 47
plot_seaice, 48
plot_sealevel, 49
plot_temp, 50
plot_temp2k, 51
plot_temp_monthly, 52
plot_temp_scatter, 53
print.hockeystick_cache_info, 54

tools::R_user_dir(), *29*

warming_stripes, 55