

Package: ukhousing (via r-universe)

June 3, 2026

Title Access UK Housing Data from Land Registry, EPC, and Planning

Version 0.1.0

Description Fetch UK housing data from official sources. Access the UK House Price Index and Price Paid Data from 'HM Land Registry' <<https://landregistry.data.gov.uk/>>, domestic and non-domestic Energy Performance Certificates from the 'MHCLG' Open Data service <<https://epc.opendatacommunities.org/>>, and planning application, brownfield land, and local plan data from 'planning.data.gov.uk' <<https://www.planning.data.gov.uk/>>. Data covers all 441 UK local authorities from 1995 to the present. Functions accept flexible filters (postcode, local authority, property type, date range) and return tidy data frames. Downloaded data is cached locally for subsequent calls.

Depends R (>= 4.1.0)

License MIT + file LICENSE

Encoding UTF-8

Language en-US

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Imports cli (>= 3.6.0), httr2 (>= 1.0.0), jsonlite, tools, utils

Suggests sf, testthat (>= 3.0.0)

Config/testthat/edition 3

URL <https://charlescoverdale.github.io/ukhousing/>,
<https://github.com/charlescoverdale/ukhousing>

BugReports <https://github.com/charlescoverdale/ukhousing/issues>

Config/pak/sysreqs libssl-dev

Repository <https://charlescoverdale.r-universe.dev>

Date/Publication 2026-05-04 19:11:44 UTC

RemoteUrl <https://github.com/charlescoverdale/ukhousing>

RemoteRef HEAD

RemoteSha 4086d210cf93f8df77b2e4f23af15ab312f68f9c

Contents

ukh_cache_info	2
ukh_clear_cache	3
ukh_epc_bulk	3
ukh_epc_certificate	4
ukh_epc_recommendations_summary	5
ukh_epc_search	6
ukh_epc_set_key	8
ukh_epc_summary	9
ukh_hpi	10
ukh_hpi_compare	11
ukh_pipr	12
ukh_planning	13
ukh_planning_datasets	14
ukh_ppd	14
ukh_ppd_address	16
ukh_ppd_bulk	16
ukh_ppd_summary	17
ukh_ppd_transaction	18
ukh_ppd_years	19
ukh_regions	20
ukh_sparql	21
ukh_transactions	22
Index	23

ukh_cache_info	<i>Inspect the local ukhousing cache</i>
----------------	--

Description

Returns information about the local cache: where it lives, how many files it contains, and how much disk space they take. Useful when debugging stale results or deciding whether to call [ukh_clear_cache\(\)](#).

Usage

```
ukh_cache_info()
```

Value

A list with elements `dir`, `n_files`, `size_bytes`, `size_human`, and `files` (a data frame).

See Also

Other configuration: [ukh_clear_cache\(\)](#), [ukh_epc_set_key\(\)](#)

Examples

```
op <- options(ukhousing.cache_dir = tempdir())
ukh_cache_info()
options(op)
```

ukh_clear_cache	<i>Clear the ukhousing cache</i>
-----------------	----------------------------------

Description

Deletes all locally cached UK housing data files. The next call to any data function will re-download fresh data.

Usage

```
ukh_clear_cache()
```

Value

Invisibly returns NULL.

See Also

Other configuration: [ukh_cache_info\(\)](#), [ukh_epc_set_key\(\)](#)

Examples

```
op <- options(ukhousing.cache_dir = tempdir())
ukh_clear_cache()
options(op)
```

ukh_epc_bulk	<i>Download a bulk EPC ZIP for a local authority</i>
--------------	--

Description

Downloads and extracts the MHCLG bulk ZIP containing all domestic certificates and recommendations for one local authority. The ZIP is typically 5 to 50 MB and contains `certificates.csv`, `recommendations.csv`, and a licence file.

Usage

```
ukh_epc_bulk(
  la,
  refresh = FALSE,
  type = c("domestic", "non-domestic", "display")
)
```

Arguments

la	Character. Local authority GSS code (e.g. "E09000033").
refresh	Logical. Re-download even if cached? Default FALSE.
type	Character. Register: "domestic" (default), "non-domestic", or "display".

Value

A list with elements certificates and recommendations, each giving the path to the extracted CSV.

See Also

Other energy performance certificates: [ukh_epc_certificate\(\)](#), [ukh_epc_recommendations_summary\(\)](#), [ukh_epc_search\(\)](#), [ukh_epc_summary\(\)](#)

Examples

```
## Not run:
paths <- ukh_epc_bulk("E09000033")
certs <- read.csv(paths$certificates)

## End(Not run)
```

ukh_epc_certificate *Fetch a single EPC by LMK key*

Description

Returns all available fields and any improvement recommendations for one Energy Performance Certificate identified by its LMK key.

Usage

```
ukh_epc_certificate(lmk_key, type = c("domestic", "non-domestic", "display"))
```

Arguments

lmk_key	Character. The certificate's LMK key (found in the lmk_key column of ukh_epc_search() results).
type	Character. Register: "domestic" (default), "non-domestic", or "display".

Value

A list with two elements:

certificate A one-row data frame with all certificate fields.

recommendations A data frame of improvement recommendations (may be empty).

See Also

Other energy performance certificates: [ukh_epc_bulk\(\)](#), [ukh_epc_recommendations_summary\(\)](#), [ukh_epc_search\(\)](#), [ukh_epc_summary\(\)](#)

Examples

```
## Not run:
cert <- ukh_epc_certificate("0000-0000-0000-0000-0000")
cert$certificate
cert$recommendations

## End(Not run)
```

ukh_epc_recommendations_summary

Summarise EPC improvement recommendations for a local authority

Description

Aggregates the improvement recommendations across certificates in a local authority, returning the frequency of each recommendation and the mean estimated cost and savings where available.

Usage

```
ukh_epc_recommendations_summary(  
  la,  
  type = c("domestic", "non-domestic", "display"),  
  refresh = FALSE  
)
```

Arguments

la	Character. Local authority GSS code.
type	Character. "domestic" (default), "non-domestic", or "display".
refresh	Logical. Re-download bulk ZIP? Default FALSE.

Details

This uses the bulk per-LA ZIP download rather than the paginated API, which is much faster for this aggregation. Requires EPC API credentials.

Value

A data frame with columns `improvement_id`, `improvement_summary`, `count`, `mean_indicative_cost` (where numeric costs are reported), ordered by count descending.

See Also

Other energy performance certificates: [ukh_epc_bulk\(\)](#), [ukh_epc_certificate\(\)](#), [ukh_epc_search\(\)](#), [ukh_epc_summary\(\)](#)

Examples

```
## Not run:
recs <- ukh_epc_recommendations_summary("E09000033")
head(recs)

## End(Not run)
```

ukh_epc_search	<i>Search domestic Energy Performance Certificates</i>
----------------	--

Description

Queries the MHCLG EPC Open Data service for domestic certificates matching the given filters. Results are paginated automatically using search-after tokens (not the `from` parameter, which caps at 10,000 records).

Usage

```
ukh_epc_search(
  postcode = NULL,
  la = NULL,
  property_type = NULL,
  rating = NULL,
  built_form = NULL,
  from = NULL,
  to = NULL,
  size = 1000L,
  max_records = 10000L,
  type = c("domestic", "non-domestic", "display")
)
```

Arguments

<code>postcode</code>	Optional character. Postcode or partial postcode.
<code>la</code>	Optional character. Local authority GSS code (e.g. "E09000033" for Westminster).
<code>property_type</code>	Optional character. "House", "Flat", "Bungalow", "Maisonette", "Park home".

rating	Optional character. Current energy rating ("A" to "G").
built_form	Optional character. "Detached", "Semi-Detached", "End-Terrace", "Mid-Terrace", "Enclosed End-Terrace", "Enclosed Mid-Terrace".
from, to	Optional. Lodgement date range (YYYY-MM-DD).
size	Integer. Results per page, max 5000. Default 1000.
max_records	Integer. Maximum total records to fetch across pages. Default 10000. Set higher for bulk analysis.
type	Character. Register to query: "domestic" (default, housing), "non-domestic" (commercial buildings), or "display" (DECs for public buildings).

Details

Registration at <https://epc.opendatacommunities.org/> is required and free.

Value

A data frame of certificates. Columns include lmk_key, address, postcode, uprn, local_authority, constituency, property_type, built_form, inspection_date, lodgement_date, current_energy_rating, current_energy_efficiency, potential_energy_rating, potential_energy_efficiency, total_floor_area, co2_emissions_current, and more.

See Also

Other energy performance certificates: [ukh_epc_bulk\(\)](#), [ukh_epc_certificate\(\)](#), [ukh_epc_recommendations_summary](#), [ukh_epc_summary\(\)](#)

Examples

```
## Not run:
# Requires EPC API credentials
certs <- ukh_epc_search(postcode = "SW1A 1AA")
head(certs)

# All E-rated flats in Westminster lodged since 2020
wm <- ukh_epc_search(
  la = "E09000033", property_type = "Flat",
  rating = "E", from = "2020-01-01"
)

## End(Not run)
```

ukh_epc_set_key	<i>Set EPC API credentials</i>
-----------------	--------------------------------

Description

Stores the email and API key used to authenticate requests to the MHCLG Energy Performance Certificate Open Data service. Credentials persist for the current R session. Alternatively, set the EPC_EMAIL and EPC_API_KEY environment variables in your .Renviron file.

Usage

```
ukh_epc_set_key(email, key)
```

Arguments

email	Character. The email you registered with.
key	Character. The API key.

Details

Register for a free API key at <https://epc.opendatacommunities.org/>.

Value

Invisible NULL.

See Also

Other configuration: [ukh_cache_info\(\)](#), [ukh_clear_cache\(\)](#)

Examples

```
## Not run:  
ukh_epc_set_key("you@example.com", "your_api_key_here")  
  
## End(Not run)
```

ukh_epc_summary	<i>EPC rating distribution for a local authority</i>
-----------------	--

Description

Summarises the distribution of current energy ratings (A-G) for domestic certificates in a local authority. Useful for area-level comparisons of housing stock efficiency.

Usage

```
ukh_epc_summary(  
  la,  
  from = NULL,  
  to = NULL,  
  type = c("domestic", "non-domestic", "display")  
)
```

Arguments

la	Character. Local authority GSS code.
from, to	Optional. Lodgement date range.
type	Character. Register: "domestic" (default), "non-domestic", or "display".

Value

A data frame with one row per rating (A-G) and columns rating, count, percentage, mean_floor_area, mean_co2_emissions.

See Also

Other energy performance certificates: [ukh_epc_bulk\(\)](#), [ukh_epc_certificate\(\)](#), [ukh_epc_recommendations_summary](#), [ukh_epc_search\(\)](#)

Examples

```
## Not run:  
ukh_epc_summary(la = "E09000033")  
  
## End(Not run)
```

ukh_hpi	<i>UK House Price Index</i>
---------	-----------------------------

Description

Fetches monthly UK House Price Index data for a region from the HM Land Registry linked data service. Coverage: 441+ areas (countries, regions, counties, local authorities) from January 1995 (England and Wales), January 2004 (Scotland), January 2005 (Northern Ireland).

Usage

```
ukh_hpi(region, from = NULL, to = NULL, refresh = FALSE)
```

Arguments

region	Character. Region slug, GSS code, or common name (e.g. "westminster", "Westminster", "E09000033"). See ukh_regions() for common options.
from, to	Optional character or Date. Start and end dates in ISO format (YYYY-MM-DD). Default returns the full available history.
refresh	Logical. Re-download even if cached? Default FALSE.

Details

The returned data frame includes the headline index, average price, monthly and annual percentage change, sales volume, and breakdowns by property type (detached, semi-detached, terraced, flat) and by buyer type (cash, mortgage, first-time buyer, former owner occupier, new build, existing property).

Sales volumes lag the headline index by approximately five months because the Land Registry needs the full transaction set to settle.

Value

A data frame with one row per month. Columns include date, region, hpi, avg_price, pct_change_monthly, pct_change_annual, sales_volume, plus property-type and buyer-type average prices.

See Also

Other house price index: [ukh_hpi_compare\(\)](#), [ukh_transactions\(\)](#)

Examples

```
op <- options(ukhousing.cache_dir = tempdir())

# All UK average house prices
uk <- ukh_hpi("united-kingdom")
head(uk)
```

```
# London, last 10 years
london <- ukh_hpi("london", from = "2016-01-01")
tail(london)

options(op)
```

ukh_hpi_compare	<i>Compare UK HPI across multiple regions</i>
-----------------	---

Description

Fetches the UK House Price Index for several regions and returns a wide data frame with one measure (e.g. average price or annual % change) as a column per region. Useful for regional comparison charts.

Usage

```
ukh_hpi_compare(  
  regions,  
  measure = "avg_price",  
  from = NULL,  
  to = NULL,  
  refresh = FALSE  
)
```

Arguments

regions	Character vector of region slugs, GSS codes, or names.
measure	Character. Which measure to return. One of "avg_price", "hpi", "pct_change_annual", "pct_change_monthly", or "sales_volume". Default "avg_price".
from, to	Optional character or Date. Date range.
refresh	Logical. Re-download even if cached? Default FALSE.

Value

A wide data frame with a date column and one column per region.

See Also

Other house price index: [ukh_hpi\(\)](#), [ukh_transactions\(\)](#)

Examples

```
op <- options(ukhousing.cache_dir = tempdir())

prices <- ukh_hpi_compare(
  c("london", "manchester", "newcastle-upon-tyne"),
  measure = "avg_price",
  from = "2015-01-01"
)
head(prices)

options(op)
```

ukh_pipr	<i>ONS Index of Private Housing Rental Prices / Price Index of Private Rents</i>
----------	--

Description

Fetches the UK's official private-rental price index from the ONS Beta API. The dataset replaced the Index of Private Housing Rental Prices (IPHRP) with the Price Index of Private Rents (PIPR) in January 2024, but the same dataset slug covers both.

Usage

```
ukh_pipr(refresh = FALSE)
```

Arguments

refresh Logical. Re-download even if cached? Default FALSE.

Details

Coverage: UK aggregate plus countries (England, Scotland, Wales, Northern Ireland) and English regions, monthly from January 2015.

Value

A data frame with columns date, region, and index.

Examples

```
op <- options(ukhousing.cache_dir = tempdir())
rents <- ukh_pipr()
head(rents)
options(op)
```

ukh_planning	<i>Query planning.data.gov.uk</i>
--------------	-----------------------------------

Description

Fetches records from the Digital Land planning data platform, which hosts over 100 datasets on planning applications, brownfield land, local plans, conservation areas, listed buildings, flood risk zones, and more.

Usage

```
ukh_planning(  
  dataset,  
  la = NULL,  
  limit = 1000L,  
  format = c("data.frame", "raw", "sf")  
)
```

Arguments

dataset	Character. Dataset slug. Common values: "planning-application", "brownfield-land", "local-plan", "conservation-area", "listed-building", "article-4-direction-area", "tree-preservation-zone", "flood-risk-zone". See ukh_planning_datasets() for the full list.
la	Optional character. Local authority name (matched case-insensitively against the organisation field).
limit	Integer. Maximum records to return. Default 1000.
format	Character. Response format. "data.frame" (default) returns tidy data. "raw" returns the parsed JSON list.

Value

A data frame (or list if format = "raw").

See Also

Other planning: [ukh_planning_datasets\(\)](#)

Examples

```
op <- options(ukhousing.cache_dir = tempdir())  
bf <- ukh_planning("brownfield-land", limit = 100)  
head(bf)  
options(op)
```

ukh_planning_datasets *List available planning datasets*

Description

Returns a data frame of datasets hosted by planning.data.gov.uk, with their slugs, names, and descriptions.

Usage

```
ukh_planning_datasets()
```

Value

A data frame with slug, name, typology, and record_count.

See Also

Other planning: [ukh_planning\(\)](#)

Examples

```
op <- options(ukhousing.cache_dir = tempdir())
ds <- ukh_planning_datasets()
head(ds)
options(op)
```

ukh_ppd

Price Paid Data

Description

Fetches individual property transaction records from the HM Land Registry Price Paid Data, filtered by local authority, postcode, property type, and other criteria.

Usage

```
ukh_ppd(
  year = as.integer(format(Sys.Date(), "%Y")),
  la = NULL,
  postcode = NULL,
  property_type = NULL,
  new_build = NULL,
  tenure = NULL,
  from = NULL,
```

```

    to = NULL,
    refresh = FALSE
  )

```

Arguments

year	Integer. Year of transactions to fetch. Defaults to the current calendar year.
la	Optional character. Local authority name (matched case-insensitively against the district field).
postcode	Optional character. Postcode or postcode prefix.
property_type	Optional character. One of "detached", "semi", "terraced", "flat", or "other". Also accepts the single-letter codes "D", "S", "T", "F", "O".
new_build	Optional logical. If TRUE, only newly built properties. If FALSE, only established properties.
tenure	Optional character. "freehold" or "leasehold".
from, to	Optional character or Date. Date range within the year (YYYY-MM-DD).
refresh	Logical. Re-download the yearly file even if cached? Default FALSE.

Details

The full yearly CSV is ~150 MB (about 900,000 transactions). This function downloads the yearly file, caches it, and then filters in memory. Memory footprint during the call is roughly 1 GB because R data frames are considerably larger than the source CSV; on memory-constrained machines, prefer [ukh_ppd_address\(\)](#) for postcode lookups or [ukh_ppd_summary\(\)](#) for aggregated stats. Subsequent queries against the same year use the cache. For multi-year queries, call the function once per year or use [ukh_ppd_years\(\)](#).

Value

A data frame of individual transactions with columns `transaction_id`, `price`, `date`, `postcode`, `property_type`, `new_build`, `tenure`, `paon`, `saon`, `street`, `locality`, `town`, `district`, `county`, `category`, `record_status`.

See Also

Other price paid data: [ukh_ppd_address\(\)](#), [ukh_ppd_bulk\(\)](#), [ukh_ppd_summary\(\)](#), [ukh_ppd_transaction\(\)](#), [ukh_ppd_years\(\)](#)

Examples

```

op <- options(ukhousing.cache_dir = tempdir())

# Westminster flats sold in 2024
wm <- ukh_ppd(2024, la = "Westminster", property_type = "flat")
head(wm)

options(op)

```

ukh_ppd_address *Look up Price Paid transactions by postcode address*

Description

Uses the Land Registry linked data address lookup to find all transactions at a given postcode. Faster than downloading the yearly bulk file when you only want a single postcode.

Usage

```
ukh_ppd_address(postcode)
```

Arguments

postcode Character. Full UK postcode (e.g. "SW1A 1AA").

Value

A data frame of transactions at addresses matching the postcode.

See Also

Other price paid data: [ukh_ppd\(\)](#), [ukh_ppd_bulk\(\)](#), [ukh_ppd_summary\(\)](#), [ukh_ppd_transaction\(\)](#), [ukh_ppd_years\(\)](#)

Examples

```
op <- options(ukhousing.cache_dir = tempdir())
tx <- ukh_ppd_address("SW1A 1AA")
head(tx)
options(op)
```

ukh_ppd_bulk *Download a Price Paid Data bulk file*

Description

Downloads and caches a yearly (or the complete) Price Paid Data CSV from HM Land Registry. Returns the path to the cached file. Yearly files are typically 100-200 MB; the complete file (1995-present) is approximately 5.3 GB.

Usage

```
ukh_ppd_bulk(
  year = as.integer(format(Sys.Date(), "%Y")),
  full = FALSE,
  refresh = FALSE
)
```

Arguments

year	Integer. Year to download. Ignored when full = TRUE. Defaults to the current year.
full	Logical. If TRUE, download the complete file (pp-complete.csv, ~5.3 GB) instead of a single year. Default FALSE.
refresh	Logical. Re-download even if cached? Default FALSE.

Value

Character. The path to the cached CSV file.

See Also

Other price paid data: [ukh_ppd\(\)](#), [ukh_ppd_address\(\)](#), [ukh_ppd_summary\(\)](#), [ukh_ppd_transaction\(\)](#), [ukh_ppd_years\(\)](#)

Examples

```
op <- options(ukhousing.cache_dir = tempdir())
path <- ukh_ppd_bulk(2025)
options(op)
```

ukh_ppd_summary	<i>Price Paid Data summary</i>
-----------------	--------------------------------

Description

Returns summary statistics from Price Paid Data for a given year, aggregated by month, property type, or local authority. Useful when the user wants counts and median prices without loading 150 MB of individual transactions.

Usage

```
ukh_ppd_summary(
  year = as.integer(format(Sys.Date(), "%Y")),
  by = c("month", "property_type", "la"),
  la = NULL,
  refresh = FALSE
)
```

Arguments

year	Integer. Year to summarise.
by	Character. Aggregation dimension: "month", "property_type", or "la". Default "month".
la	Optional character. Restrict to one local authority.
refresh	Logical. Re-download even if cached? Default FALSE.

Value

A data frame with one row per group and columns n, median_price, mean_price, total_value.

See Also

Other price paid data: [ukh_ppd\(\)](#), [ukh_ppd_address\(\)](#), [ukh_ppd_bulk\(\)](#), [ukh_ppd_transaction\(\)](#), [ukh_ppd_years\(\)](#)

Examples

```
op <- options(ukhousing.cache_dir = tempdir())
# Monthly transaction counts and medians, nationally
s <- ukh_ppd_summary(2025, by = "month")
head(s)
options(op)
```

`ukh_ppd_transaction` *Look up a single Price Paid transaction by ID*

Description

Fetches the full metadata for one transaction from the Land Registry linked data service by its transaction unique identifier.

Usage

```
ukh_ppd_transaction(id)
```

Arguments

id	Character. A transaction unique identifier (GUID, with or without curly braces).
----	--

Value

A one-row data frame with the transaction fields, or an empty data frame if the transaction is not found.

See Also

Other price paid data: [ukh_ppd\(\)](#), [ukh_ppd_address\(\)](#), [ukh_ppd_bulk\(\)](#), [ukh_ppd_summary\(\)](#), [ukh_ppd_years\(\)](#)

Examples

```
op <- options(ukhousing.cache_dir = tempdir())
tx <- ukh_ppd_transaction("{A4C5B0C6-4D5D-47E2-E053-6C04A8C07E7C}")
tx
options(op)
```

ukh_ppd_years

Price Paid Data across multiple years

Description

Convenience wrapper that calls [ukh_ppd\(\)](#) for each year in a vector and row-binds the results. Caches each year independently.

Usage

```
ukh_ppd_years(years, ...)
```

Arguments

`years` Integer vector. Years to fetch.
`...` Additional arguments passed to [ukh_ppd\(\)](#) (e.g. `la`, `postcode`, `property_type`).

Value

A single data frame combining transactions from all requested years.

See Also

Other price paid data: [ukh_ppd\(\)](#), [ukh_ppd_address\(\)](#), [ukh_ppd_bulk\(\)](#), [ukh_ppd_summary\(\)](#), [ukh_ppd_transaction\(\)](#)

Examples

```
op <- options(ukhousing.cache_dir = tempdir())
five_year <- ukh_ppd_years(2020:2024, la = "Westminster",
                          property_type = "flat")
nrow(five_year)
options(op)
```

ukh_regions	<i>List valid UK HPI region slugs</i>
-------------	---------------------------------------

Description

Returns a data frame of common UK HPI region slugs with their names, GSS codes, and tier (country, region, county, or local authority). Useful for looking up the slug to pass to [ukh_hpi\(\)](#).

Usage

```
ukh_regions(tier = c("all", "country", "region", "county", "la"))
```

Arguments

tier Character. Filter by tier: "all" (default), "country", "region", "county", or "la".

Details

This is a selection of the most commonly used regions, not an exhaustive list. The UK HPI covers 441+ areas total; any valid slug can be passed to [ukh_hpi\(\)](#) directly.

Value

A data frame with columns slug, name, gss_code, and tier.

See Also

Other helpers: [ukh_sparql\(\)](#)

Examples

```
# All regions
head(ukh_regions())

# Just the nine English regions
ukh_regions(tier = "region")

# Country-level series
ukh_regions(tier = "country")
```

ukh_sparql	<i>Run a SPARQL query</i>
------------	---------------------------

Description

Runs a SPARQL query against one of the supported endpoints and returns the result as a data frame. Useful for queries that aren't covered by the dedicated helpers, including custom HPI aggregations and Price Paid Data lookups by postcode.

Usage

```
ukh_sparql(
  query,
  endpoint = c("land-registry", "opendatacommunities"),
  timeout = 60L
)
```

Arguments

query	Character. A SPARQL query string.
endpoint	Character. One of "land-registry" (default), "opendatacommunities", or a full URL to any SPARQL endpoint.
timeout	Integer. Request timeout in seconds. Default 60.

Details

Both endpoints are free and require no authentication. The Land Registry endpoint covers HPI and Price Paid Data; the Open Data Communities endpoint covers 300+ housing-market datasets published by MHCLG.

Value

A data frame of bindings. Column names match the SELECT variables in the query.

See Also

Other helpers: [ukh_regions\(\)](#)

Examples

```
op <- options(ukhousing.cache_dir = tempdir())

# All HPI observations for Westminster in January 2020
q <- '
PREFIX ukhpi: <http://landregistry.data.gov.uk/def/ukhpi/>
SELECT ?hpi ?avgPrice WHERE {
  <http://landregistry.data.gov.uk/data/ukhpi/region/city-of-westminster/month/2020-01>
    ukhpi:housePriceIndex ?hpi ;
```

```

      ukhpi:averagePrice    ?avgPrice .
    }'
    ukh_sparql(q)

    options(op)

```

ukh_transactions *Transaction volumes for a region*

Description

Returns monthly residential transaction counts for a UK region. This is a thin wrapper over [ukh_hpi\(\)](#) that extracts the `sales_volume` column with its date. Transaction volumes lag the headline index by approximately five months because the Land Registry needs the full transaction set to settle.

Usage

```
ukh_transactions(region, from = NULL, to = NULL, refresh = FALSE)
```

Arguments

<code>region</code>	Character. Region slug, GSS code, or common name.
<code>from, to</code>	Optional. Date range (YYYY-MM-DD).
<code>refresh</code>	Logical. Re-download? Default FALSE.

Value

A data frame with `date`, `region`, and `sales_volume`.

See Also

Other house price index: [ukh_hpi\(\)](#), [ukh_hpi_compare\(\)](#)

Examples

```

op <- options(ukhousing.cache_dir = tempdir())
tx <- ukh_transactions("england", from = "2020-01-01")
head(tx)
options(op)

```

Index

- * **configuration**
 - ukh_cache_info, [2](#)
 - ukh_clear_cache, [3](#)
 - ukh_epc_set_key, [8](#)
 - * **energy performance certificates**
 - ukh_epc_bulk, [3](#)
 - ukh_epc_certificate, [4](#)
 - ukh_epc_recommendations_summary, [5](#)
 - ukh_epc_search, [6](#)
 - ukh_epc_summary, [9](#)
 - * **helpers**
 - ukh_regions, [20](#)
 - ukh_sparql, [21](#)
 - * **house price index**
 - ukh_hpi, [10](#)
 - ukh_hpi_compare, [11](#)
 - ukh_transactions, [22](#)
 - * **ons**
 - ukh_pipr, [12](#)
 - * **planning**
 - ukh_planning, [13](#)
 - ukh_planning_datasets, [14](#)
 - * **price paid data**
 - ukh_ppd, [14](#)
 - ukh_ppd_address, [16](#)
 - ukh_ppd_bulk, [16](#)
 - ukh_ppd_summary, [17](#)
 - ukh_ppd_transaction, [18](#)
 - ukh_ppd_years, [19](#)
-
- ukh_cache_info, [2](#), [3](#), [8](#)
 - ukh_clear_cache, [2](#), [3](#), [8](#)
 - ukh_clear_cache(), [2](#)
 - ukh_epc_bulk, [3](#), [5–7](#), [9](#)
 - ukh_epc_certificate, [4](#), [4](#), [6](#), [7](#), [9](#)
 - ukh_epc_recommendations_summary, [4](#), [5](#), [5](#), [7](#), [9](#)
 - ukh_epc_search, [4](#), [5](#), [6](#), [6](#), [9](#)
 - ukh_epc_search(), [4](#)
 - ukh_epc_set_key, [2](#), [3](#), [8](#)
 - ukh_epc_summary, [4–7](#), [9](#)
 - ukh_hpi, [10](#), [11](#), [22](#)
 - ukh_hpi(), [20](#), [22](#)
 - ukh_hpi_compare, [10](#), [11](#), [22](#)
 - ukh_pipr, [12](#)
 - ukh_planning, [13](#), [14](#)
 - ukh_planning_datasets, [13](#), [14](#)
 - ukh_planning_datasets(), [13](#)
 - ukh_ppd, [14](#), [16–19](#)
 - ukh_ppd(), [19](#)
 - ukh_ppd_address, [15](#), [16](#), [17–19](#)
 - ukh_ppd_address(), [15](#)
 - ukh_ppd_bulk, [15](#), [16](#), [16](#), [18](#), [19](#)
 - ukh_ppd_summary, [15](#), [16](#), [17](#), [17](#), [19](#)
 - ukh_ppd_summary(), [15](#)
 - ukh_ppd_transaction, [15–17](#), [18](#), [18](#), [19](#)
 - ukh_ppd_years, [15–18](#), [19](#), [19](#)
 - ukh_ppd_years(), [15](#)
 - ukh_regions, [20](#), [21](#)
 - ukh_regions(), [10](#)
 - ukh_sparql, [20](#), [21](#)
 - ukh_transactions, [10](#), [11](#), [22](#)