Package ‘metScanR’

February 2, 2019

Title Find, Map, and Gather Environmental Data and Metadata
Version 1.2.2
Date 2019-02-01
Maintainer Josh Roberti <jaroberti87@gmail.com>
Description A tool for locating, mapping, and gathering environmental data and metadata, worldwide. Users can search for and filter metadata from > 157,000 environmental monitoring stations among 219 countries/territories and >20 networks/organizations via elevation, location, active dates, elements measured (e.g., temperature, precipitation), country, network, and/or known identifier. Future updates to the package will allow the user to obtain datasets from stations within the database.

Depends R (>= 3.3.0)
Imports geosphere, matlab, leaflet, grDevices, plyr, RCurl, utils
License GPL-3
Encoding UTF-8
LazyData no
RoxygenNote 6.1.1
URL https://github.com/jaroberti/metScanR
Suggests testthat
NeedsCompilation no
Author Josh Roberti [aut, cre],
   Cody Flagg [aut],
   Lee Stanish [aut],
   Robert Lee [aut]
Repository CRAN
Date/Publication 2019-02-02 08:20:03 UTC

R topics documented:

  dbLog .................................................. 2
  getCountry ............................................ 2
**Description**

A simple log file for the version and date of database revisions. More information on specific updates to the database can be found at [https://jaroberti.github.io/metScanR/](https://jaroberti.github.io/metScanR/).

**Usage**

dbLog

**Format**

(data frame) Version number and publication dates.

**Description**

Filter environmental monitoring stations by country

Return metadata of environmental monitoring stations from specific country(ies)/territory(ies) within the metScanR database.

**Usage**

getchountry(crosty, ...)}
getDates

Arguments

  country (character) Country(ies)/territory(ies) to filter environmental stations.
  ...

Value

A list comprising metadata of environmental monitoring stations from country(ies)/territory(ies)
specified in country

Author(s)

Josh Roberti <jaroberti87@gmail.com>

See Also

  siteFinder mapSiteFinder metScanR_DB

Examples

## Not run:
getCountry(country="United States")
getCountry(country=c("Italy","Portugal"))
## End(Not run)

getDates

Filter environmental monitoring stations by active date(s)

Description

Return metadata of environmental monitoring stations that were/are active during specified dates.

Usage

getDates(startDate, endDate, includeUnk = FALSE, ...)

Arguments

  startDate (character) "YYYY-MM-DD" used to filter start dates of environmental stations
  within the metScanR database. Optional if endDate is initialized. Required if
  endDate is missing.
  
  endDate (character) "YYYY-MM-DD" used to filter end dates of environmental stations
  within the metScanR database. Optional if startDate is initialized. Required if
  startDate is missing.
includeUnk: (logical) Defaults to FALSE and excludes sites with unknown start dates. Setting to TRUE will include sites with unknown start dates. Sites with unknown start dates account for ~71 percent of the metScanR database. This is a result of undocumented, government (or network/governing body) metadata. Nearly all stations within the database have a known end date, however. Initializing endDate (while leaving startDate uninitialized) and setting includeUnk=TRUE will more than likely return results than if startDate is also initialized.

auto-populates when called from siteFinder() wrapper

Value
A list comprising metadata of environmental monitoring sites that were/are active between the startDate and/or endDate

Author(s)
Josh Roberti <jaroberti87@gmail.com>
Lee Stanish

References
see reference links above

See Also
siteFinder mapSiteFinder metScanR_DB

Examples
```r
## Not run:
#return metadata of sites that were active from at least 1940-01-01 through 1970-04-18
getDates(startDate="1940-01-01",endDate = "1970-04-18")
#return metadata of sites that were active up through at least 1950-07-08
getDates(endDate = "1950-07-08")
#return metadata of sites that were active up through at least 1950-07-08
#and have an unknown start date:
getDates(endDate = "1950-07-08", includeUnk=TRUE)
## End(Not run)
```

getElevation

Filter environmental monitoring stations by elevation

Description
Return metadata of environmental monitoring stations that have a specific elevation.
getId

Usage

getElevation(elevMin, elevMax, ...)

Arguments

elevMin (numeric) defines the minimum elevation to filter metadata. Units are in meters (m) Above Sea Level.
elevMax (numeric) defines the maximum elevation to filter metadata. Units are in meters (m) Above Sea Level.
...
  auto-populates when called from siteFinder() wrapper

Value

A list comprising metadata of environmental monitoring stations that have elevations conforming to the criteria specified in elevThresh

Author(s)

Josh Roberti <jaroberti87@gmail.com>

See Also

siteFinder mapSiteFinder metScanR_DB

Examples

```r
## Not run:
# return metadata of sites that have elevations above 1500 (m) Above Sea Level
getElevation(elevMin=1500)
# return metadata of sites that have elevations less than 35 (m) Above Sea Level
getElevation(elevMax=35)
## End(Not run)
```

---

getId

Filter environmental monitoring stations by identifier type

Description

Return metadata of environmental monitoring stations that have specific identifier types.

Usage

getId(id, ...)
Arguments

**id**

(character) identifier(s) of interest. Metadata are available for stations with any of the identifiers listed below. It should be noted that a station may have multiple, associated identifiers. For instance, a single station may have COOP, GHCND, and FAA identifiers. See reference links for further information.

AIRMMon: Atmospheric Integrated Research Monitoring Network

http://nadp.slh.wisc.edu/AIRMOn/

Ameriflux

http://ameriflux.lbl.gov/sites/site-search/#filter-type=all

AMNet: Atmospheric Mercury Network

http://nadp.slh.wisc.edu/amn/

AMoN: Ammonia Monitoring Network

http://nadp.slh.wisc.edu/AMoN/

BOR: Bureau of Reclamation


COOP: Cooperative Observer Network


FAA: Federal Aviation Administration

https://www.faa.gov/

GHCND: Global Historical Climatology Network - Daily


GHCNMLT: Global Historical Climatology Network - Monthly Land Temperature v4

https://www.ncdc.noaa.gov/ghcnm/

ICAO: International Civil Aviation Organization

http://www.icao.int/Pages/default.aspx

MDN: Mercury Deposition Network

https://nadp.slh.wisc.edu/MDN/

MPRC: Manual Precipitation Network

MSNT: Non-Telemetered Snow Telemetry Network

NCDCSTNID: National Climatic Data Center

https://www.ncdc.noaa.gov/homr/
Value

A list comprising metadata of environmental monitoring stations having identifier types specified in id

NEON: National Ecological Observatory Network  
http://www.neonscience.org/science-design/field-sites

NWSLI: National Weather Service Location Identifier  
https://www.weather.gov/arh/stationlist

NTN: National Trends Network  
http://nadp.slh.wisc.edu/NTN/

OTHER: Telemetered Natural Resource Conservation Service (NRCS) Stations that do not meet criteria for SNOTEL, SNOLITE, SCAN, or NRCS Experimental hydromet

SCAN: Soil Climate Analysis Network  
https://www.wcc.nrcs.usda.gov/about/mon_scan.html

SNOW: Snow Course and Aerial Marker Network  
https://www.wcc.nrcs.usda.gov/about/mon_manual.html

SNTL: Snow Telemetry Network  
https://www.wcc.nrcs.usda.gov/about/mon_automate.html

SNTLT: Snow Telemetry Network, Limited Sensors  
https://www.wcc.nrcs.usda.gov/about/mon_automate.html

TRANS: *miscellaneous IDs that do not fall into National Centers for Environmental Information (NCEI) support  
https://www.ncei.noaa.gov/

USGS: Streamflow Network (United States Geological Survey)  
https://water.usgs.gov/nsip/

WBAN: Weather Bureau Army Navy  

WMO: World Meteorological Organization  
https://www.wmo.int/pages/index_en.html

... auto-populates when called from siteFinder() wrapper
Author(s)

Josh Roberti <jaroberti87@gmail.com>

References

see reference links above

See Also

siteFinder mapSiteFinder metScanR_DB

Examples

```
## Not run:
#return a list of sites that have an WBAN identifier
getId(id="WBAN")
#return a list of sites that have either an FAA, ICAO, or NWSLI identifier
getId(id=c("FAA","ICAO","NWSLI"))
## End(Not run)
```

---

**getNearby** Filter environmental monitoring stations by POI

Description

Return metadata of environmental monitoring stations nearby a specific environmental station (see siteID) or near a Latitude/Longitude pair (see lat,lon).

Usage

```
getNearby(siteID, lat, lon, radius, ...)
```

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>siteID</td>
<td>(character) in the form of: [NETWORK]:[ID]. Environmental monitoring network to use as your Point of Interest (POI). Required if lat &amp; lon are missing.</td>
</tr>
<tr>
<td>lat</td>
<td>(numeric) Latitude of (POI). lat and lon are required if siteID is missing.</td>
</tr>
<tr>
<td>lon</td>
<td>(numeric) Longitude of POI. lat and lon are required if siteID is missing.</td>
</tr>
<tr>
<td>radius</td>
<td>(numeric) Search radius outward from POI for finding environmental monitoring stations. Defined in kilometers (km); Required</td>
</tr>
</tbody>
</table>

... auto-populates when called from siteFinder() wrapper

Value

A list comprising metadata of environmental monitoring stations located within radius from the user-entered siteID or Lat/Lon POI.
getNetwork

Author(s)

Josh Roberti <jaroberti87@gmail.com>

See Also

siteFinder mapSiteFinder metScanR_DB

Examples

## Not run:
# returns metadata from all sites within 50 km of NEON site 'CPER'
getNearby(siteID="NEON:CPER", radius=50)
# return metadata of sites within 10 km of Lat=41.7821 & Lon = -71.4204 (Cranston, RI, USA)
getNearby(lat=41.7821, lon = -71.4204, radius=40)
## End(Not run)

description

Return metadata of environmental monitoring stations from networks/platforms within the metScanR database.

Usage

getNetwork(network, ...)

Arguments

<table>
<thead>
<tr>
<th>network (character) Network(s)/platform(s) to filter environmental monitoring stations. Metadata are available for stations in the networks below. See reference links for further information.</th>
</tr>
</thead>
<tbody>
<tr>
<td>AL USCRN: United States Regional Climate Reference Network - Alabama</td>
</tr>
<tr>
<td><a href="https://catalog.data.gov/dataset/al-uscrn-station-information">https://catalog.data.gov/dataset/al-uscrn-station-information</a></td>
</tr>
<tr>
<td>Ameriflux</td>
</tr>
<tr>
<td><a href="http://ameriflux.lbl.gov/sites/site-search/#filter-type=all">http://ameriflux.lbl.gov/sites/site-search/#filter-type=all</a></td>
</tr>
<tr>
<td>ASOS: Automated Surface Observing System</td>
</tr>
<tr>
<td>AWOS: Automated Weather Observing System</td>
</tr>
</tbody>
</table>
getNetwork

automated-weather-observing-system-awos

BOR: Bureau of Reclamation
https://www.wcc.nrcs.usda.gov/wsf/wsf-reservoir.html

COCORAHS: Community Collaborative Rain, Hail & Snow Network
https://www.cocorahs.org/

COOP: Cooperative Observer Network

NEON: National Ecological Observatory Network
http://www.neonscience.org/

NADP: National Atmospheric Deposition Program
http://nadp.slh.wisc.edu/airmon/

NRCS: Natural Resources Conservation Service
https://www.wcc.nrcs.usda.gov/web_service/NRCS_Station_Networks.pdf

UKN: *unknown* (unidentified network)

UPPERAIR: Upper Air network
https://www.weather.gov/upperair/nws_upper

USCRN: United States Climate Reference Network
https://www.ncdc.noaa.gov/crn/

USGS: Streamflow Network (United States Geological Survey)
https://water.usgs.gov/nsip/

USRCRN: United States Regional Climate Reference Network
https://www.ncdc.noaa.gov/crn/

... auto-populates when called from siteFinder() wrapper

Value

A list comprising metadata of environmental monitoring sites from network(s)/platform(s) specified in network

Author(s)

Josh Roberti <jaroberti87@gmail.com>
Derek Smith
getStation

References

see reference links above

See Also

siteFinder mapSiteFinder metScanR_DB

Examples

```r
## Not run:
# returns metadata from all NRCS sites within the database
getNetwork(network="NRCS")
# returns metadata from ASOS, USCRN, and NEON sites within the database
getNetwork(network=c("ASOS","USCRN","NEON"))
## End(Not run)
```

getStation  

return metadata for selected environmental station(s)

Description

Return metadata of user specified environmental monitoring stations. This is a standalone function.

Usage

getStation(siteID, ...)

Arguments

- **siteID** (character) in the form of: [idType]:[ID]. Required.
- **...** auto-populates when called from siteFinder() wrapper

Value

A list comprising metadata for the entered environmental monitoring site(s)

Author(s)

Josh Roberti <jaroberti87@gmail.com>

See Also

siteFinder mapSiteFinder metScanR_DB
getTerritory

Filter environmental monitoring stations by state/territory

Description

Return metadata of environmental monitoring stations from a state/territory within the metScanR database.

Usage

getTerritory(territory, ...)

Arguments

territory (character) state/territory abbreviation (e.g., 'RI' = Rhode Island, United States;'YT' = Yukon Territory, Canada) to filter environmental monitoring stations.

... auto-populates when called from siteFinder() wrapper

Value

A list comprising metadata of environmental monitoring sites from state/territory specified in territory

Author(s)

Josh Roberti <jaroberti87@gmail.com>

References

see reference links above

See Also

siteFinder mapSiteFinder metScanR_DB
getVars

Examples

## Not run:
returns metadata from all stations within Rhode Island (RI)
getTerritory(territory = "RI")
returns metadata from stations within Colorado and Utah
getTerritory(territory=c("CO","UT"))
## End(Not run)

getVars  Filter environmental monitoring stations by reported elements

Description

Return metadata of environmental monitoring stations that collect specific element-level (environmental variables, e.g., air temperature) metadata via 'fuzzy search'.

Usage

getVars(vars, startVarsDate, endVarsDate, ...)

Arguments

vars (character) Elements(s)/variables(s) of interest. The user can search for general, environmental terms, such as 'temperature,' or 'wind,' and the function will return environmental stations that collect the specified elements ('fuzzy search'). Keep in mind that the database contains ~107,000 stations, worldwide. Searching for a general term such as 'temperature' will return many stations. The user is advised to search for more granular terms, e.g., using sub terms such as 'air temperature,' or 'soil temperature,' if they wish to narrow their results.

startVarsDate (character) start date in the form of "YYYY-MM-DD" for filtering environmental variables by active measurement dates. Optional

date

endVarsDate (character) end date in the form of "YYYY-MM-DD" for filtering environmental variables by active measurement dates. Optional

... auto-populates when called from siteFinder() wrapper

Value

A list comprising environmental monitoring sites that observe or collect the element(s)/variable(s) specified in vars

Author(s)

Josh Roberti <jaroberti87@gmail.com>
See Also

siteFinder mapSiteFinder metScanR_DB

Examples

## Not run:
#return a list of sites that collect humidity data
getVars(vars="humidity")
#return a list of sites that collect soil temperature and/or wind data
getVars(vars=c("soil temperature","wind"))
#return a list of sites that collected snow depth data during 1970-01-01 thru 1985-05-10
getVars(vars = "snow depth",startVarsDate = "1970-01-01",endVarsDate = "1985-05-10")
## End(Not run)

mapSiteFinder( )

Map environmental monitoring stations

Description

A plotting tool to map environmental monitoring stations from the metScanR database. **NOTE:** This function requires internet connection!

Usage

mapSiteFinder(x, limit = 5000)

Arguments

x (list) Metadata of environmental monitoring stations.

limit (numeric) maximum number of stations to plot. Defaults to 5000. Setting this >5000 may result in wait times of up to minute if internet connection is slow.

Value

A map of environmental monitoring stations

Author(s)

Josh Roberti <jaroberti87@gmail.com>
Lee Stanish
Cody Flagg
Sam Weintraub
Derek Smith

See Also

getNearby getElevation getDates getNetwork getVars getCountry getId siteFinder metScanR_DB
Examples

```r
## Not run:
#map environmental monitoring stations located in Italy
mapSiteFinder(getCountry(country="Italy"))
#map environmental monitoring stations within 50 km of Boulder, CO, USA
mapSiteFinder(getNearby(lat=40.0149,lon=-105.2705,radius=50))
## End(Not run)
```

Description

Metadata from ~107,000 environmental monitoring stations among 219 countries/territories and 18 environmental networks. Metadata are gathered from a growing number of sources and the database is continually updated to reflect the increase of information.

Usage

metScanR_DB

Format

(list) Metadata from ~107,000 environmental monitoring stations, worldwide. The metadata for each site include:

name (character) Name of environmental monitoring site as defined by governing network

identifiers (data.frame) Station identifiers of environmental monitoring site as defined by associated networks

platform (character) Station platform (type). A single station may be a specific platform but it might be associated with many networks / identifiers.

elements (data.frame) Elements (environmental phenomena) measured, along with associated start and end dates

location (data.frame) Spatial location information (latitude, longitude, country, elevation) of environmental monitoring site
metScanR_terms

Environmental metadata terms used within metScanR_DB

Description

Environmental metadata terms (e.g., air temperature, wind speed, etc.) and data product identifiers used by the environmental monitoring networks within the metScanR_DB. Terms are gathered from a growing number of sources and the terms database is continually updated to reflect the increase of information.

Usage

metScanR_terms

Format

(data frame) Environmental terms and data product identifiers from environmental monitoring networks, worldwide

siteFinder

Filter environmental monitoring stations (wrapper)

Description

A wrapper function comprising all filtering functions within the metScanR package. The metScanR database contains metadata from roughly 107,000 stations among ~200 countries/territories and ~18 networks/platforms, worldwide. This function returns metadata for a subset of those stations, based on the criteria specified by a user.

Usage

siteFinder(country, siteID, lat, lon, radius, network, vars, startVarsDate, endVarsDate, id, startDate, endDate, includeUnk, elevMin, elevMax, territory, ...)

Arguments

country (character) Country(ies)/territory(ies) to filter environmental stations. See ’?getCountry’ for more information.
siteID (character) in the form of: *idType:id*. Environmental monitoring network to use as your Point of Interest (POI). See ’?getNearby’ for more help.
lat (numeric) Latitude of (POI). See ’?getNearby’ for more help.
lon (numeric) Longitude of (POI). See ’?getNearby’ for more help.
radius (numeric) Search radius outward from POI for finding environmental monitoring stations. Defined in kilometers (km). See ’?getNearby’ for more help.
**network** (character) Network(s)/platform(s) to filter environmental monitoring stations. Metadata are available for stations in the networks below. See '?getNetwork' for more information.

**vars** (character) Elements(s)/variables(s) of interest. The user can search for general, environmental terms, such as 'temperature,' or 'wind,' and the function will return environmental stations that collect the specified elements ('fuzzy search'). Keep in mind that the database contains ~107,000 stations, worldwide. Searching for a general term such as 'temperature' will return many stations. The user is advised to search for more granular terms, e.g., using sub terms such as 'air temperature,' or 'soil temperature,' if they wish to narrow their results. See '?getVars' for more help.

**startVarsDate** (character) start date in the form of "YYYY-MM-DD" for filtering environmental variables by active measurement dates. Optional

**endVarsDate** (character) end date in the form of "YYYY-MM-DD" for filtering environmental variables by active measurement dates. Optional

**id** (character) identifier(s) of interest. Metadata are available for stations with any of the identifiers listed below. It should be noted that a single station may have multiple, associated identifiers. See '?getId' for more information.

**startDate** (character) "YYYY-MM-DD" used to filter start dates of environmental stations within the metScanR database. Optional if endDate is initialized. Required if endDate is missing. See '?getDates' for more information.

**endDate** (character) "YYYY-MM-DD" used to filter end dates of environmental stations within the metScanR database. Optional if startDate is initialized. Required if startDate is missing. See '?getDates' for more information.

**includeUnk** (logical) Defaults to FALSE and excludes sites with unknown start dates. Setting to TRUE will include sites with unknown start dates. Sites with unknown start dates account for ~71 percent of the metScanR database. This is a result of undocumented, government (or network/governing body) metadata. Nearly all stations within the database have a known end date, however. Initializing endDate (while leaving startDate uninitialized) and setting includeUnk=TRUE will more than likely return results than if startDate is also initialized. See '?getDates' for more information.

**elevMin** (numeric) defines minimum elevation (m) to filter metadata.

**elevMax** (numeric) defines maximum elevation (m) to filter metadata.

**territory** (character) state/territory abbreviation (e.g., 'RI'= Rhode Island, United States;'YT'= Yukon Territory, Canada) to filter environmental monitoring stations.

... Depracated terms from previous version of function.

**Value**

A list comprising metadata of environmental monitoring stations from country(ies)/territory(ies) specified in country
updateDatabase

**Description**

Updates the metScanR database to the latest version hosted on GitHub at: https://github.com/jaroberti/metScanR. When installed, the metScanR package contains only a small (subset) database comprising ~5300 environmental monitoring stations. This function will update the local version of the database to the most up-to-date version.

**Usage**

```r
updateDatabase()
```

**Author(s)**

Robert Lee <rhlee@colorado.edu>

**See Also**

metScanR_DB

---

**Examples**

```r
## Not run:
# Return metadata of sites within 50 km of NEON's HARV active from 1965-10-20 to 1986-09-02
siteFinder(siteID="NEON:HARV",startDate="1965-10-20", endDate="1986-09-02",radius=50)

# Return metadata of SCAN, SNTL, and ASOS sites active from at least 2000-01-05 onward
siteFinder(network=c("SCAN","SNTL","ASOS"),startDate="2000-01-05")

# Return metadata of sites in Brazil with elevations of 1500 +/- 250 (m) Above Sea Level
siteFinder(minElev=1000,maxElev=1800,country="Brazil")
## End(Not run)
```
Examples

```r
## Not run:
updateDatabase()

## End(Not run)
```
Index

*Topic AL
  getNetwork, 9
  getTerritory, 12
*Topic ASOS,
  getNetwork, 9
  getTerritory, 12
*Topic AWOS,
  getNetwork, 9
  getTerritory, 12
*Topic COCORAHs,
  getNetwork, 9
  getTerritory, 12
*Topic COOP,
  getId, 5
  getNetwork, 9
  getTerritory, 12
*Topic Earth,
  getCountry, 2
  getDates, 3
  updateDatabase, 18
*Topic FAA,
  getId, 5
*Topic GHCND,
  getId, 5
*Topic GHCNMLT,
  getId, 5
*Topic ICAO,
  getId, 5
*Topic MPRC,
  getNetwork, 9
  getTerritory, 12
*Topic MSNT,
  getNetwork, 9
  getTerritory, 12
*Topic NCD CSTNID,
  getId, 5
*Topic NEON,
  getId, 5
  getNetwork, 9
  getTerritory, 12
*Topic NRCS,
  getId, 5
*Topic NWSLI,
  getId, 5
*Topic SCAN,
  getNetwork, 9
  getTerritory, 12
*Topic SNOW,
  getNetwork, 9
  getTerritory, 12
*Topic SNTL,
  getNetwork, 9
  getTerritory, 12
*Topic SNTLT,
  getNetwork, 9
  getTerritory, 12
*Topic TRANS,
  getId, 5
*Topic UPPERAIT,
  getNetwork, 9
  getTerritory, 12
*Topic USCRN,
  getNetwork, 9
  getTerritory, 12
*Topic USGS,
  getNetwork, 9
  getTerritory, 12
*Topic USRCRN
  getNetwork, 9
  getTerritory, 12
*Topic WBAN,
  getId, 5
*Topic WMO,
  getId, 5
*Topic air,
getCountry, 2
getDates, 3
gElevation, 4
gId, 5
gNearby, 8
gNetwork, 9
gStation, 11
g Territory, 12
g Vars, 13
updateDatabase, 18

*Topic **air**
getCountry, 2
getDates, 3
gElevation, 4
gId, 5
gNearby, 8
gNetwork, 9
gStation, 11
g Territory, 12
g Vars, 13
updateDatabase, 18

*Topic **atmospheric**
getCountry, 2
getDates, 3
gElevation, 4
gId, 5
gNearby, 8
gNetwork, 9
gStation, 11
g Territory, 12
g Vars, 13
mapSiteFinder, 14
siteFinder, 16
updateDatabase, 18

*Topic **atmosphere,**
getCountry, 2
getDates, 3
gElevation, 4
gId, 5
gNearby, 8
gNetwork, 9
gStation, 11
g Territory, 12
g Vars, 13
mapSiteFinder, 14
siteFinder, 16
updateDatabase, 18

*Topic **atmospheric**
getCountry, 2
getDates, 3
gElevation, 4
gId, 5
gNearby, 8
gNetwork, 9
gStation, 11
g Territory, 12
g Vars, 13
dbLog, 2
metScanR_DB, 15
metScanR_terms, 16

*Topic **canopy,**
getCountry, 2
getDates, 3
gElevation, 4
gId, 5
gNearby, 8
gNetwork, 9
gStation, 11
g Territory, 12
g Vars, 13
updateDatabase, 18

*Topic **chemical,**
getCountry, 2
getDates, 3
gElevation, 4
gId, 5
gNearby, 8
gNetwork, 9
gStation, 11
g Territory, 12
g Vars, 13
updateDatabase, 18

*Topic **climate,**
getCountry, 2
getDates, 3
gElevation, 4
gId, 5
gNearby, 8
gNetwork, 9
gStation, 11
g Territory, 12
g Vars, 13
mapSiteFinder, 14
siteFinder, 16
updateDatabase, 18

*Topic **cloud,**
getCountry, 2
getDates, 3
gElevation, 4
gId, 5
gNearby, 8
gNetwork, 9
gStation, 11
g Territory, 12
g Vars, 13
updateDatabase, 18
*Topic conductivity,
  getCountry, 2
  getDates, 3
  getElevation, 4
  getId, 5
  getNearby, 8
  getNetwork, 9
  getStation, 11
  getTerritory, 12
  getVars, 13
  updateDatabase, 18
*Topic country,
  getCountry, 2
  getDates, 3
  updateDatabase, 18
*Topic data,
  dbLog, 2
  getCountry, 2
  getDates, 3
  getElevation, 4
  getId, 5
  getNearby, 8
  getNetwork, 9
  getStation, 11
  getTerritory, 12
  getVars, 13
  mapSiteFinder, 14
  siteFinder, 16
  updateDatabase, 18
*Topic datasets,
  dbLog, 2
  getCountry, 2
  getDates, 3
  getElevation, 4
  updateDatabase, 18
*Topic dates,
  getDates, 3
*Topic elevation,
  getElevation, 4
*Topic environment,
  getCountry, 2
  getDates, 3
  getElevation, 4
  getId, 5
  getNearby, 8
  getNetwork, 9
  getStation, 11
  getTerritory, 12
  getVars, 13
  mapSiteFinder, 14
  siteFinder, 16
  updateDatabase, 18
*Topic flux,
  getCountry, 2
  getDates, 3
  getElevation, 4
  getId, 5
  getNearby, 8
  getNetwork, 9
  getStation, 11
  getTerritory, 12
  getVars, 13
  updateDatabase, 18
*Topic groundwater,
  getCountry, 2
  getDates, 3
  getElevation, 4
  getId, 5
  getNearby, 8
  getNetwork, 9
  getStation, 11
  getTerritory, 12
  getVars, 13
  updateDatabase, 18
*Topic humidity,
  getElevation, 4
  getId, 5
  getNearby, 8
  getNetwork, 9
getStation, 11
getTerritory, 12
getVars, 13
*Topic **in-situ**,
getCountry, 2
getDates, 3
getElevation, 4
getId, 5
getNearby, 8
getNetwork, 9
getStation, 11
getTerritory, 12
getVars, 13
mapSiteFinder, 14
siteFinder, 16
updateDatabase, 18
*Topic **lake**,
getCountry, 2
gIDates, 3
getElevation, 4
getId, 5
getNearby, 8
getNetwork, 9
getStation, 11
getTerritory, 12
getVars, 13
updateDatabase, 18
*Topic **latitude**,
getNearby, 8
*Topic **longitude**,
getNearby, 8
*Topic **mapping**
mapSiteFinder, 14
*Topic **maps**,
mapSiteFinder, 14
*Topic **metadata**,
dbLog, 2
getElevation, 4
getId, 5
getNetwork, 9
getTerritory, 12
metScanR_DB, 15
metScanR_terms, 16
*Topic **meters**
getElevation, 4
*Topic **n-grams**
dbLog, 2
metScanR_terms, 16
*Topic **phenology**,
getCountry, 2
gIDates, 3
getElevation, 4
getId, 5
getNearby, 8
getNetwork, 9
getStation, 11
getTerritory, 12
getVars, 13
updateDatabase, 18
*Topic **pollution**,
getCountry, 2
gIDates, 3
getElevation, 4
getId, 5
getNearby, 8
getNetwork, 9
getStation, 11
getTerritory, 12
getVars, 13
updateDatabase, 18
*Topic **precipitation**,
getCountry, 2
gIDates, 3
getElevation, 4
getId, 5
getNearby, 8
getNetwork, 9
getStation, 11
getTerritory, 12
getVars, 13
updateDatabase, 18
*Topic **meteorological**,
mapSiteFinder, 14
*Topic **meteorological**
siteFinder, 16
*Topic **meteorology**,
dbLog, 2
mapSiteFinder, 14
metScanR_DB, 15
metScanR_terms, 16
siteFinder, 16
*Topic **meters**
getElevation, 4
*Topic **n-grams**
dbLog, 2
metScanR_terms, 16
*Topic **phenology**,
getCountry, 2
gIDates, 3
getElevation, 4
getId, 5
getNearby, 8
getNetwork, 9
getStation, 11
getTerritory, 12
getVars, 13
updateDatabase, 18
*Topic **pollution**,
getCountry, 2
gIDates, 3
getElevation, 4
getId, 5
getNearby, 8
getNetwork, 9
getStation, 11
getTerritory, 12
getVars, 13
updateDatabase, 18
*Topic **precipitation**,
getCountry, 2
gIDates, 3
getElevation, 4
getId, 5
getNearby, 8
getNetwork, 9
getStation, 11
getTerritory, 12
getVars, 13
updateDatabase, 18
*Topic radiation,
getCountry, 2
getDates, 3
getElevation, 4
getId, 5
getNearby, 8
getNetwork, 9
getStation, 11
getTerritory, 12
getVars, 13
updateDatabase, 18
*Topic radius,
getNearby, 8
*Topic reservoir,
getCountry, 2
getDates, 3
getElevation, 4
getId, 5
getNearby, 8
getNetwork, 9
getStation, 11
getTerritory, 12
getVars, 13
updateDatabase, 18
*Topic river,
getCountry, 2
getDates, 3
getElevation, 4
getId, 5
getNearby, 8
getNetwork, 9
getStation, 11
getTerritory, 12
getVars, 13
updateDatabase, 18
*Topic salinity,
getCountry, 2
getDates, 3
getElevation, 4
getId, 5
getNearby, 8
getNetwork, 9
getStation, 11
getTerritory, 12
getVars, 13
updateDatabase, 18
*Topic snow,
getCountry, 2
getDates, 3
getElevation, 4
getId, 5
getNearby, 8
getNetwork, 9
getStation, 11
getTerritory, 12
getVars, 13
updateDatabase, 18
*Topic soil,
getCountry, 2
getDates, 3
getElevation, 4
getId, 5
getNearby, 8
getNetwork, 9
getStation, 11
getTerritory, 12
getVars, 13
updateDatabase, 18
*Topic soils,
getCountry, 2
getDates, 3
getElevation, 4
getId, 5
getNearby, 8
getNetwork, 9
getStation, 11
getTerritory, 12
getVars, 13
updateDatabase, 18
*Topic stream,
getCountry, 2
getDates, 3
getElevation, 4
getId, 5
getNearby, 8
getNetwork, 9
getStation, 11
getTerritory, 12
getVars, 13
updateDatabase, 18
*Topic temperature,
getCountry, 2
getDates, 3
getElevation, 4
INDEX

getid, 5
getnearby, 8
getnetwork, 9
getstation, 11
getterritory, 12
getvars, 13
updatedatabase, 18

*Topic **times**,
gdates, 3

*Topic **vapor**,
getcountry, 2
gdates, 3
gelevation, 4
gid, 5
gnearby, 8
gnetwork, 9
gstation, 11
gterritory, 12
gvars, 13
updatedatabase, 18

*Topic **water**,
getcountry, 2
gdates, 3
gelevation, 4
gid, 5
gnearby, 8
gnetwork, 9
gstation, 11
gterritory, 12
gvars, 13
updatedatabase, 18

*Topic **weather**,
dbLog, 2
gcountry, 2, 14, 18
gdates, 3, 14, 18
gelevation, 4, 14, 18
gid, 5, 14, 18
gnearby, 8, 14, 18
gnetwork, 9, 14, 18
gstation, 11
gterritory, 12
gvars, 13, 14, 18
mapsitefinder, 3–5, 8, 9, 11, 12, 14, 15, 18
metScanR_DB, 3–5, 8, 9, 11, 12, 14, 15, 18
metScanR_terms, 16
sitefinder, 3–5, 8, 9, 11, 12, 14, 16
updatedatabase, 18

*Topic **wind**,
gcountry, 2
gdates, 3
gelevation, 4
gid, 5
gnearby, 8
gnetwork, 9
gstation, 11
gterritory, 12
gvars, 13
updatedatabase, 18

dbLog, 2
gcountry, 2, 14, 18
gdates, 3, 14, 18
gelevation, 4, 14, 18
gid, 5, 14, 18
gnearby, 8, 14, 18
gnetwork, 9, 14, 18
gstation, 11
gterritory, 12
gvars, 13, 14, 18
mapsitefinder, 3–5, 8, 9, 11, 12, 14, 15, 18
metScanR_DB, 3–5, 8, 9, 11, 12, 14, 15, 18
metScanR_terms, 16
sitefinder, 3–5, 8, 9, 11, 12, 14, 16
updatedatabase, 18