Package ‘hR’

November 10, 2019

Type Package
Title Toolkit for Data Analytics in Human Resources
Version 0.2.1
Author Dale Kube [aut, cre]
Maintainer Dale Kube <dkube@uwalumni.com>
Description
Transform and analyze workforce data in meaningful ways for human resources (HR) analytics.

BugReports https://github.com/dalekube/hR
Encoding UTF-8
License GPL
LazyData true
RoxygenNote 6.1.1
Imports data.tree, data.table, shiny, rhandsontable, knitr
Depends R(>= 2.10)
Suggests rmarkdown
VignetteBuilder knitr
NeedsCompilation no
Repository CRAN
Date/Publication 2019-11-10 05:40:02 UTC

R topics documented:
hierarchyLong .......................................................... 2
hierarchyWide .......................................................... 2
workforceHistory ....................................................... 3
workforcePlan .......................................................... 4

Index 5
The hierarchyLong function transforms a standard set of unique employee and supervisor identifiers (employee IDs, email addresses, etc.) into an elongated format that can be used to aggregate employee data by a particular line of leadership (i.e. include everyone who rolls up to Susan). The function returns a long data.table consisting of one row per employee for every supervisor above them, up to the top of the tree. The levels represent the number of supervisors from the employee (starting with "1" for an employee's direct supervisor).

Usage

hierarchyLong(ee, supv)

Arguments

ee An array containing unique identifiers for employees.
supv An array containing unique identifiers for supervisors. These values should be of the same type as the employee values.

Value
data table

Examples

ee = c("Dale@hR.com","Bob@hR.com","Julie@hR.com","Andrea@hR.com")
supv = c("Julie@hR.com","Julie@hR.com","Andrea@hR.com","Susan@hR.com")
hierarchyLong(ee,supv)

The hierarchyWide function transforms a standard set of unique employee and supervisor identifiers (employee IDs, email addresses, etc.) into a wide format that can be used to aggregate employee data by a particular line of leadership (i.e. include everyone who rolls up to Susan). The function returns a wide data.table with a column for every level in the hierarchy, starting from the top of the tree (i.e. "Supv1" is likely the CEO in your organization).

Usage

hierarchyWide(ee, supv)
Arguments

- **ee**: An array containing unique identifiers for employees.
- **supv**: An array containing unique identifiers for supervisors. These values should be of the same type as the employee values.

Value
data table

Examples

```r
ee = c("Dale@hR.com", "Bob@hR.com", "Julie@hR.com", "Andrea@hR.com")
supv = c("Julie@hR.com", "Julie@hR.com", "Andrea@hR.com", "Susan@hR.com")
hierarchyWide(ee, supv)
```

---

**workforceHistory**

Workforce history data for a sample team of employees and contractors.

Description

Artificial data that reflects the workforce history data structure often used to manage employment records in a human capital management system (HCM). Modern enterprises store data in this format at the core of their HCM. This data is the root source of all data analysis and reporting related to headcount, hiring, turnover, etc.

Usage

data(workforceHistory)

Format

A data table with 45 rows and 10 variables:

- **DATE**: Effective date of the record
- **SEQ**: Effective sequence of the record (used to manage multiple records for the same effective date)
- **ACTION**: Action
- **EMPLID**: Employee ID
- **SUPVID**: Supervisor ID
- **TYPE**: Employee type (employee or contractor)
- **REGTEMP**: Regular, temporary, or contract employment
- **TITLE**: Job title
- **STATUS**: Employment status
- **NAME**: Employee name ...
**Description**

Launch a simple, interactive workforce planning worksheet that helps managers and team leaders to execute basic workforce planning tasks and plan ahead for hiring, turnover, and other factors that influence a team’s talent structure. Data analysts can use this alongside team leaders to convey change and proactively think about recruitment, etc.

**Usage**

```r
workforcePlan(launch.browser = T)
```

**Arguments**

- `launch.browser` Logical; whether the app should launch in the user's default browser
Index

* Topic **datasets**
  - workforceHistory, 3

hierarchyLong, 2
hierarchyWide, 2

workforceHistory, 3
workforcePlan, 4