Package 'HEXCleanR'

January 22, 2026

Title Werkzeuge zur Datenbereinigung im HEX-Projekt

Version 0.5.13

Description Das Paket stellt Hilfsfunktionen zur qualitätsgesicherten Aufbereitung und Bereinigung der im HEX anfallenden Daten bereit. Es unterstützt u.a. insbesondere bei der Prüfung und Säuberung von Organisationsangaben, dem Erkennen auffälliger Veränderungen in kategorialen Merkmalen über Semester hinweg sowie der Vereinheitlichung und Plausibilisierung von Rohdaten aus unterschiedlichen Quellen.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

URL http://srv-data01:30080/hex/hexcleanr

RoxygenNote 7.3.3

Depends R (>= 4.1.0)

Imports dplyr,
pointblank (>= 0.12.2.9000),
rlang,
stringr,
reticulate

Contents

check_db *Sicherheitscheck für die* db_data.rds *einer Universität*

Description

check_db prüft mithilfe des Paketes pointblank verschiedene Struktur-, Typ- und Plausibilitätsprüfungen für db_data.rds. Der Check bildet den Abschluss des Cleaningprozesses einer Universität.

Usage

```
check_db(test_data)
```

Arguments

test_data	Ein Data Frame mit den zu prüfenden Daten.

Details

Folgende Prüfungen werden durchgeführt:

1. **Vollständigkeits-Check:** Alle erwarteten Variablen (z. B. hochschule, jahr, semester, Future-Skills-Variablen etc.) sind im Datensatz enthalten.
2. **Datentypen-Check Strings:** Alle inhaltlichen Text-Variablen (außer den Future-Skills-Variablen und der Hilfsspalte kursbeschreibung_len) müssen vom Typ character sein.
3. **Datentypen-Check Numeric:** Die Future-Skills-Variablen data_analytics_ki, softwareentwicklung, nutzerzentriertes_design, it_architektur, hardware_robotikentwicklung und quantencomputing müssen numerisch sein.
4. **Mindestlänge Kursbeschreibung:** Die Länge der Kursbeschreibung (kursbeschreibung_len) muss mindestens 20 Zeichen betragen. Kürzere Beschreibungen gelten als fehlerhaft und sollten NA gesetzt werden.
5. **Hochschulnamen:** Die Werte in hochschule müssen in der geladenen Referenzliste zulässiger Hochschulnamen enthalten sein (inst/extdata/hochschulen_namen_kuerzel.sql).
6. **Hochschulkürzel:** Die Werte in hochschule_kurz müssen in der entsprechenden Referenzliste zulässiger Kürzel enthalten sein (inst/extdata/hochschulen_namen_kuerzel.sql).
7. **Pflichtfelder Jahr/Semester:** Die Felder jahr und semester dürfen keine fehlenden Werte (NA) enthalten.
8. **Sprach-Codierung:** Die Werte in sprache_recoded müssen zu der im Wiki definierten, erlaubten Menge gehören, z. B. "Deutsch", "Englisch", "Deutsch/Englisch", weitere Sprachen, "Sonstiges" oder NA.
9. **Kursformat-Codierung:** Die Werte in kursformat_recoded müssen zu der im Wiki definierten, festen Menge gehören, z. B. "Vorlesung", "Seminar", "Übung", "Austausch", "Erfahrung", "Sprachkurs", "Sonstiges" oder NA.
10. **Semester-Format:** Die Werte in semester müssen dem Muster "YYYYs" oder "YYYYw" entsprechen (vierstellige Jahreszahl, gefolgt von s für Sommer- bzw. w für Wintersemester).

Value

Ein einzelnes ptblank_agent-Objekt mit den Prüfergebnissen Zusätzlich wird ein HTML-Report im Viewer angezeigt.

Examples

```
## Not run:
  res <- check_db(test_data)
  res

## End(Not run)
```

`check_distinct_level_change`
Checkt auf auffällige Änderungen der Anzahl einzigartiger Werte einer Variablen zwischen Gruppen

---

Description

Diese Funktion dient der Qualitäts- und Plausibilitätskontrolle einer Variable, indem sie prüft, ob sich die Anzahl einzigartiger Werte (z. B. Organisationen, Sprachen, Studiengänge) zwischen Gruppen (typischerweise Semestern oder Jahre) auffällig verändert. Ziel ist das frühzeitige Erkennen potenziell fehlerhafter Daten (z. B. unvollständiges Scraping, fehlerhafte Joins).

Usage

```
check_distinct_level_change(data, value_col, group_col, threshold = 0.75)
```

Arguments

`data`	Ein `data.frame` oder `tibble` mit den zu prüfenden Daten.
`value_col`	Ungequoteter Spaltenname: Spalte, deren Anzahl einzigartiger Werte geprüft werden soll (z. B. Organisation, Semestern).
`group_col`	Ungequoteter Spaltenname: Gruppierungsvariable (z. B. Semester, Jahr).
`threshold`	Numerischer Schwellenwert für den relativen Vergleich. Ein Wert von `0.75` entspricht einer erlaubten Abweichung von bis zu 25%.

Details

Je nach Anzahl der Gruppen wird automatisch zwischen zwei Prüfmodi unterschieden:

- mehr als 3 Gruppen: Plausibilitätscheck auf harte Sprünge mittels relativem Vergleich zum Maximum.
- weniger/gleich 3 Gruppen: Stabilitätscheck mittels Vergleich zum Durchschnitt.

Die Funktion zählt pro Gruppe die Anzahl einzigartiger Werte in `value_col`. Bei nur zwei Gruppen wird geprüft, ob eine Gruppe deutlich vom Maximum abweicht (harter Sprung). Bei drei oder mehr Gruppen wird geprüft, ob einzelne Gruppen deutlich unter dem durchschnittlichen Niveau liegen.

Die Funktion ist explizit als Plausibilitäts- und Fehlerdetektor konzipiert und nicht als inferenzstatistisches Verfahren.

Value

Ein tibble mit folgenden Spalten:

- grp_val: Gruppierungswert (Werte aus group_col).
- n_distinct: Anzahl einzigartiger Werte in value_col je Gruppe.
- rel_change: Relativer Vergleichswert (zur Referenz oder zum Durchschnitt).
- flagged: Logisch, ob eine auffällige Abweichung vorliegt.

Examples

```
## Not run:
  check_distinct_level_change(
    db_data_universitaet_jena,
    organisation,
    semester,
    threshold = 0.75
  )

## End(Not run)
```

---

check_distinct_level_change_df

Wendet die Plausibilitätsprüfung auf alle Spalten eines Datensatzes an.

Description

Diese Hilfsfunktion wendet check_distinct_level_change() für alle Variablen eines Datensatzes an, außer für die angegebene Gruppierungsvariable. So kann in einem Schritt für alle Variablen eines Datensatzes geprüft werden, ob sich die Anzahl einzigartiger Werte zwischen Gruppen (z. B. Semestern) auffällig verändert.

Usage

```
check_distinct_level_change_df(data, group_col, threshold = 0.75)
```

Arguments

data	Ein data.frame oder tibble mit den zu prüfenden Daten.
group_col	Ungequoteter Spaltenname: Gruppierungsvariable, die für alle Checks als Referenz verwendet wird (z. B. semester).
threshold	Numerischer Schwellenwert für den relativen Vergleich, wird direkt an check_distinct_level_cha weitergegeben.

Details

Alle Spalten in data, die nicht group_col entsprechen, werden als value_col nacheinander an check_distinct_level_change() übergeben. Die Ergebnisse werden mit purrr::map_dfr() zu einem gemeinsamen Tibble zusammengeführt und erhalten zusätzlich eine Spalte variable, die den Namen der jeweils geprüften Spalte enthält.

Value

Ein tibble, das die gebündelten Ergebnisse von check_distinct_level_change() für alle geprüften Variablen enthält. Zurückgegeben werden nur die Kombinationen, bei denen flagged == TRUE ist (also auffällige Abweichungen). Das Ergebnis enthält mindestens die Spalten variable, grp_val, n_distinct, rel_change und flagged. Falls keine auffälligen Abweichungen gefunden werden, wird ein leeres tibble zurückgegeben und eine entsprechende Erfolgsmeldung ausgegeben.

See Also

check_distinct_level_change

Examples

```
## Not run:
  check_distinct_level_change_df(db_data_universitaet_jena,
                                 semester)

## End(Not run)
```

---

check_nas *Visualisiert NA-Konzentration nach Semester*

Description

Erzeugt einen Plot mit der NA-Konzentration pro Variable und sortiert die Semester-Faktoren nach der darin enthaltenen vierstelligen Jahreszahl. Falls keine vierstellige Jahreszahl in der angegebenen Spalte gefunden wird, bricht die Funktion mit einer aussagekräftigen Fehlermeldung ab.

Usage

```
check_nas(data, semester)
```

Arguments

data	Ein data.frame oder tibble mit den zu analysierenden Daten.
semester	Unquoted Spaltenname mit Semester-Informationen (z.B. WS 2019/20 oder 2019).

Value

Ein ggplot-Objekt mit der Visualisierung der NA-Konzentration.

Examples

```
## Not run:
check_nas(df, semester = semester_col)

## End(Not run)
```

`check_organisation` *Prüft Organisations-Variable auf definierte Qualitätsregeln*

Description

Diese Funktion verwendet das Paket `pointblank`, um die Werte in der Organisations-Variable systematisch zu validieren. Optional wird ein HTML-Report mit Badges ausgegeben.

Usage

```
check_organisation(
  data,
  organisation_col = "organisation",
  semester_col = "semester",
  stop_at = 1,
  show_report = TRUE
)
```

Arguments

`data`	Ein `data.frame`, das die zu prüfende Organisations-Variable enthält.
`organisation_col`	Name der Organisations-Variable (Standard: `organisation`).
`semester_col`	Name der Semester-Spalte (Standard: `semester`). Wird für den Check auf Schwankungen zwischen Semestern benötigt.
`stop_at`	Schwellenwert für Fehler-Toleranz. Kann relativ (Anteil 0 bis 1) oder absolut (z. B. 1 für "Null Toleranz") angegeben werden. Standard: 1 (= Null Toleranz).
`show_report`	Logisch; wenn `TRUE`, wird ein pointblank-Report mit Badges im Viewer ausgegeben. Standard: `TRUE`.

Details

Die folgenden Prüfungen werden durchgeführt:

1. Nur korrektes Semikolon " ; " oder kein Semikolon.
2. Kein "|" enthalten.
3. Kein ">" enthalten.
4. Länge der Zeichenkette ist kleiner oder gleich 1000.
5. Keine überflüssigen Leerzeichen (entspricht `stringr::str_squish()`).
6. Warnung bei Abschluss-Begriffen (z. B. Bachelor, Master, Diplom).

Value

Ein `pointblank`-Agent-Objekt (invisible). Über `pointblank::get_agent_report()` kann der Report auch separat erzeugt werden.

Examples

```
## Not run:
  agent <- check_organisation(test_data, organisation_col = "organisation")
  pointblank::get_agent_report(agent)

## End(Not run)
```

`classify_fs` *Klassifiziert Kursdaten mit SetFit-Modell und pflegt neue Labels ein*

Description

Diese Funktion identifiziert Future-Skills-Schlagwörter in Kursdaten mithilfe eines vortrainierten SetFit-Modells (siehe http://srv-data01:30080/hex/future_skill_classification), das über das reticulate-Paket aus Python heraus aufgerufen wird. Dabei werden zunächst noch nicht klassifizierte Kurse ermittelt, diese im Modell bewertet und die resultierenden Future-Skills-Labels anschließend mit bestehenden Klassifizierungsinformationen aus einem optionalen RDS-Datensatz zusammengeführt.

Usage

```
classify_fs(
  raw_data,
  db_data_path = NULL,
  model_path = "Chernoffface/fs-setfit-multilable-model",
  key_vars = c("titel", "nummer")
)
```

Arguments

`raw_data`	Data Frame mit Rohkursdaten; sollte mindestens die Spalten für die Join-Schlüssel (`key_vars`) sowie, idealerweise, `kursbeschreibung` und `lernziele` enthalten.
`db_data_path`	(Optional) Pfad zu einer bestehenden RDS-Datei mit bereits klassifizierten Kursen. Wenn `NULL` oder nicht vorhanden, werden alle Zeilen in `raw_data` neu klassifiziert.
`model_path`	HuggingFace-Modellpfad des SetFit-Modells, das zur Klassifizierung verwendet werden soll.
`key_vars`	Zeichenvektor mit den Spaltennamen, die als Join-Keys verwendet werden (Standard: `c("titel", "nummer")`). Nur vorhandene und nicht vollständig fehlende Variablen werden verwendet.

Details

Die Funktion setzt ein funktionierendes Python-/Conda-Environment mit dem entsprechenden SetFit-Modell voraus und erwartet in den Rohdaten mindestens eine sinnvolle Schlüsselvariable (z.B. "titel" oder "nummer") sowie Spalten mit Kursbeschreibung und Lernzielen.

Value

Data Frame mit allen Zeilen von `raw_data` und den ergänzten bzw. aktualisierten Future-Skills-Klassifikationsspalten.

Examples

```
## Not run:
  raw_data_fs <- classify_fs(
    raw_data   = raw_data_uni_musterstadt,
   db_data_path = "C:/SV/HEX/Scraping/data/single_universities/Friedrich-Schiller-Universitaet_Jena/db_data_
    model_path = "Chernoffface/fs-setfit-multilable-model",
    key_vars   = c("titel", "nummer")
  )

## End(Not run)
```

`remove_semantic_na_values` *Setzt Werte einer String-Variable NA, wenn diese weniger als 20 Zeichen enthalten*

Description

Diese Funktion ersetzt Einträge in einem Vektor von Texten durch `NA`, wenn die Anzahl der Buchstaben (ohne Satzzeichen) kleiner als ein definierter Schwellenwert ist.

Usage

```
remove_semantic_na_values(texts, min_num_letters = 20)
```

Arguments

`texts`	Ein Vektor von Zeichenketten (Character-Vektor), der bereinigt werden soll.
`min_num_letters`	Minimale Anzahl an Buchstaben (Standard: 20), die ein Text enthalten muss, damit er nicht als semantisches NA betrachtet wird.

Value

Ein Character-Vektor, in dem zu kurze Texte durch `NA` ersetzt wurden.

Examples

```
## Not run:
library(dplyr)
db_data_universitaet_jena  |>
mutate(
kursbeschreibung_cleaned = remove_semantic_na_values(kursbeschreibung, min_num_letters = 30)
)

## End(Not run)
```

use_cleaning_template *Erzeugt ein Cleaning-Template im aktuellen Projekt*

Description

Diese Funktion kopiert die im Paket mitgelieferte Datei `cleaning_template.R` in das Verzeichnis `R` des aktuellen Projekts und benennt sie in `data_preparation_<university_name>.R` um. Existiert unter diesem Namen bereits eine Datei, wird eine Warnung ausgegeben und die vorhandene Datei nicht überschrieben.

Usage

```
use_cleaning_template(university_name)
```

Arguments

`university_name`
: Zeichenkette mit dem Namen der Universität, für die ein Cleaning-Template erstellt werden soll.

Value

Unsichtbar der Pfad zur erstellten (oder bereits vorhandenen) Datei.

Examples

```
## Not run:
  # Legt die Datei R/data_preparation_Universitaet_Musterstadt.R an,
  # sofern sie noch nicht existiert
  use_cleaning_template("Universitaet_Musterstadt")

## End(Not run)
```

Index