



**Statistisches Amt  
Kanton Zürich**



# VALIDATE PACKAGE

Überprüfe und validiere Daten in R.

Loris Kaufmann

- Datenqualität/-integrität
- Automatisierung Datenprüfung
- Transparenz
- Standardisierung

WARUM VALIDATE?

# ANWENDUNGSBEISPIELE



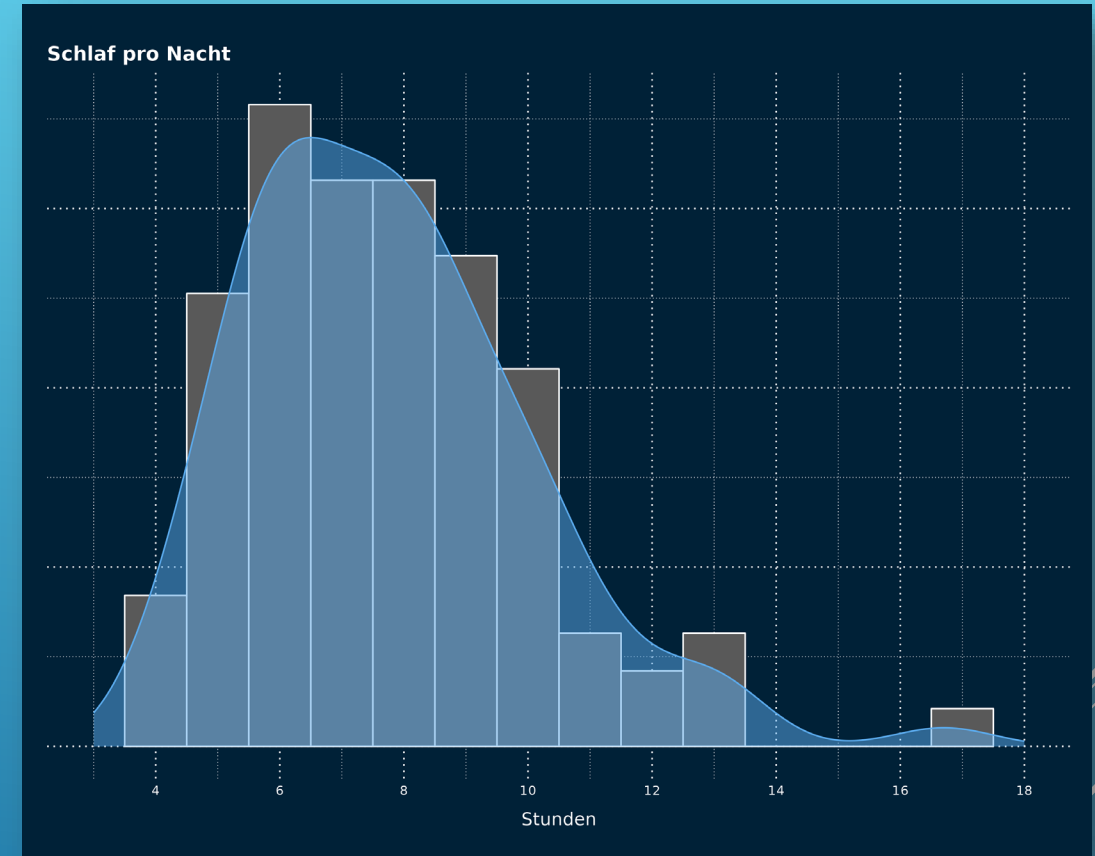
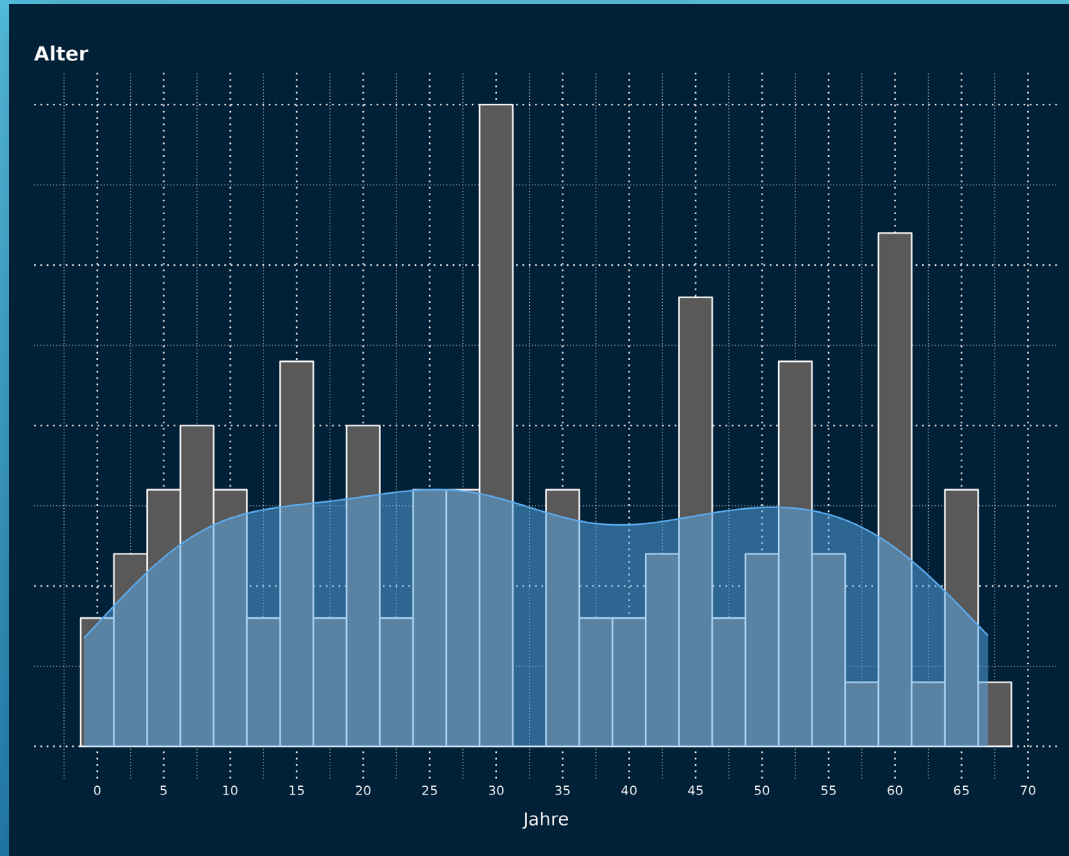
```
# Erstelle Daten
set.seed(123)

data <- data.frame(
  alter = round(runif(100, -1, 67), 0), # Alter
  schlaf = round(rtruncnorm(100, a = 4, mean = 7, sd = 3), 1) # Schlafstunden pro Nacht
)

# Kategorie basierend auf dem Alter
data$alter_kat <- ifelse(data$alter < 18, "Kind",
                        ifelse(data$alter ≤ 65, "Erwachsener", "Rentner"))

# Füge einige NA-Werte ein
na_indices <- sample(1:nrow(data), 5)
data$schlaf[na_indices] <- NA
```

# FIKTIVE DATEN SIMULIEREN



VISUALISIERUNG

# VALIDIERUNGSREGELN ERSTELLEN



```
# Jeder Ausdruck, der einen logisches Ergebnis (TRUE/FALSE) liefert, wird als
Validierungsregel akzeptiert
regeln ← validator(

  # selbst erstellte Regeln
  geboren = alter > 0,
  ausreisser_oben = schlaf < mean(schlaf, na.rm = TRUE) + 3 * sd(schlaf, na.rm = TRUE),

  # praktische Funktion, um beide Grenzen einer Variablen zu überprüfen
  grenzen = in_range(schlaf, min = 4, max = 12),

  # Überprüfung der Alterskategorie-Strings
  valid_alter = alter_kat %in% c("Kind", "Erwachsener")

)
```



```
# Wende Validierungsregeln auf die Daten an  
resultate ← confront(data, regeln)
```

```
# Zusammenfassungen
```

```
summary(resultate)
```

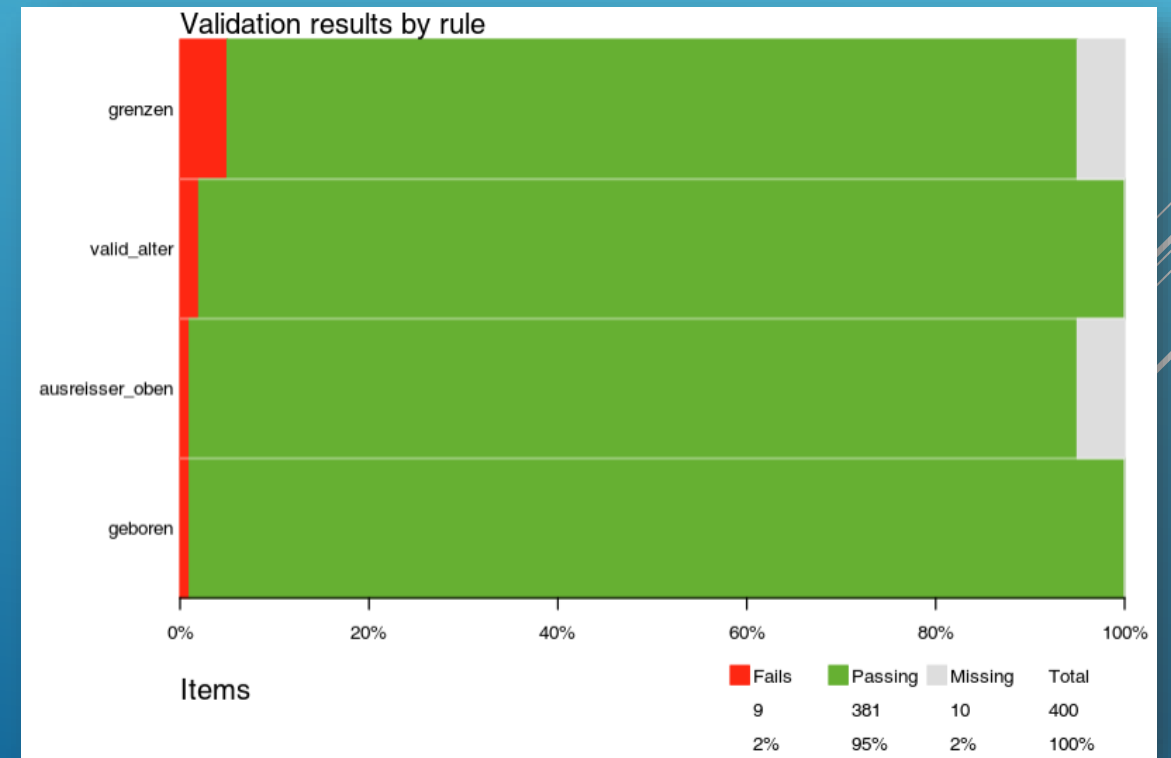
```
plot(resultate)
```

Description: df [4 × 8]

name <chr>	items <int>	passes <int>	fails <int>	nNA <int>	error <lgl>	warning <lgl>	expression <chr>
geboren	100	99	1	0	FALSE	FALSE	alter > 0
ausreisser_oben	100	94	1	5	FALSE	FALSE	schlaf < mean(schlaf, na.rm = TRUE) + 3 * sd(schlaf, na.rm = TRUE)
grenzen	100	90	5	5	FALSE	FALSE	in_range(schlaf, min = 4, max = 12)
valid_alter	100	98	2	0	FALSE	FALSE	alter_kat %vin% c("Kind", "Erwachsener")

4 rows

# REGELN ANWENDEN





```
# Datenpunkte welche die Regeln nicht bestanden haben  
check_failed ← violating(data, resultate)  
print(check_failed)
```

Description: df [8 × 3]

	alter <dbl>	schlaf <dbl>	alter_kat <chr>
17	16	13.2	Kind
24	67	7.5	Rentner
66	29	12.5	Erwachsener
74	-1	7.1	Kind
78	41	12.7	Erwachsener
83	27	13.3	Erwachsener
87	66	6.6	Rentner
94	44	16.7	Erwachsener

8 rows

## FEHLERHAFT DATENPUNKTE

Dataframe mit Datenpunkten  
welche die Regeln nicht  
bestanden haben



# ANWENDUNG OGD

Zwei Beispiele zur Anwendung für OGD -Datensätze

## Bruttolastgang elektrische Energie im Versorgungsgebiet der Elektrizitätswerke des Kantons Zürich

Der Bruttolastgang entspricht der im Netzgebiet der EKZ (Elektrizitätswerke des Kantons Zürich) abgegebenen elektrischen Energie in einer Auflösung von 15 Minuten. Ausgewiesen wird nur die an (direkt angeschlossene) Endverbraucher:innen abgegebene Energie. EKZ beliefert den grössten Teil des Kantons Zürich mit Strom. Das Netzgebiet ist in die Regionen Limmattal, Oberland, Sihl, Weinland unterteilt und geht mit EKZ Einsiedeln über das Kantonsgebiet hinaus. Die genaue Ausdehnung des Versorgungsgebiets ist unten verlinkt. Die Daten werden täglich aus den am Vortag gemessenen Energiewerten berechnet und aggregiert. Es können einzelne Messwerte fehlen oder falsch gemessen sein; sie werden nachträglich manuell angepasst. Mögliche Korrekturen werden bis zu sechs Monate nach der Messung vorgenommen. Bei der Interpretation der Werte ist eine gewisse Vorsicht geboten, da Faktoren wie die Witterung (z.B. Heizung oder Sonnenscheindauer), Home-Office oder Veränderung der Anzahl Kunden einen bedeutenden Einfluss haben auf den Stromverbrauch.

Schlagwörter


Energie Ogd Stromverbrauch

### Ressourcen

#### Detaillierter Bruttolastgang EKZ und EKZ Einsiedeln

Grösse 0 Bytes Format CSV Aktualisiert 05.12.2023

[Details anzeigen](#)

[Herunterladen](#) 

```
EKZ_data <- read_csv("EKZ.csv")
```

```
head(EKZ_data)
```

A tibble: 6 × 2

Zeitstempel <S3: POSIXct>	Bruttolastgang_kWh <dbl>
2021-12-31 23:00:00	110479.18
2021-12-31 23:15:00	111505.70
2021-12-31 23:30:00	107645.08
2021-12-31 23:45:00	106935.39
2022-01-01 00:00:00	104033.95
2022-01-01 00:15:00	99816.01

# EKZ LAST STROMNETZ MESSUNGEN

# PRAKTISCHE FUNKTIONEN



```
# Prüfung auf fehlende Daten (NA's)
```

```
fehlende_daten ← is_complete(EKZ_data$Bruttolastgang_kWh)
```

```
# gibt logischen Wert für jede einzelne Beobachtung zurück
```

```
alle_Daten_vollständig ← all_complete(EKZ_data)
```

```
# gibt einzelnen logischen Wert zurück für alle Beobachtungen
```

```
# Prüfung Sequenz
```

```
sequenz ← is_linear_sequence(EKZ_data$Zeitstempel) # gibt einzelnen logischen Wert zurück
```

```
# Regeln erstellen
regeln_EKZ ← validator(

  kWh_range = in_range(Bruttolastgang_kWh, 50000, 155000),
  # Kilowattstunden innerhalb Bandbreite

  Datum_range = in_range(Zeitstempel, min=as.Date("2021-12-31"), max=as.Date("2024-05-02"))
  # Datum innerhalb Zeitraums

)

# Resultate Überprüfung
resultate_EKZ ← confront(EKZ_data, regeln_EKZ)

summary(resultate_EKZ)
```

# DATEN ÜBERPRÜFEN

# YAML



- „Yet Another Markup Language“
- „YAML Ain't Markup Language“
- Daten-Serialisierungssprache
- Textbasiertes Datenformat zur Konfiguration und Datenstrukturierung
- Klare, hierarchische Struktur

```
rules:
- expr: Bruttolastgang_kWh ≥ 50000 & Bruttolastgang_kWh ≤ 155000
  name: 'kWh_range'
  label: 'kWh Range Validation'
  description: |
    Bruttolastgang_kWh sollte innerhalb der Bandbreite von 50000 bis 155000 kWh sein.
  created: 2023-04-30 10:00:00

- expr: Zeitstempel ≥ as.Date("2021-12-31") & Zeitstempel ≤ as.Date("2024-05-02")
  name: 'Datum_range'
  label: 'Date Range Validation'
  description: |
    Der Zeitstempel soll innerhalb des Zeitraums von 2021-12-31 bis 2024-05-02 liegen.
  created: 2023-04-30 10:00:00

- expr: is_complete(Bruttolastgang_kWh)
  name: 'fehlende_werte'
  label: 'Completeness Check for kWh'
  description: |
    Überprüfung auf Vollständigkeit.
  created: 2023-04-30 10:00:00
```

- Einfachheit
- Wiederverwendbarkeit und Skalierung
- Transparenz
- Kollaboration und Versionskontrolle
- Dynamisches laden
- Integration unterschiedliche Sprachen

## WARUM YAML?

# Messdaten langjähriger Abgasmessungen im realen Fahrbetrieb mittels Remote Sensing (RSD)

Der "Remote Sensing Detector" (RSD) ermöglicht berührungsfreie Messungen von Abgasen vorbeifahrender Fahrzeuge unter realen Verkehrsbedingungen. Diese Messungen können hauptsächlich dazu verwendet werden, Erkenntnisse über die Emissionen von Stickoxiden bestimmter Fahrzeugkategorien zu erlangen. Die Messungen liefern zudem Informationen über die Emissionen von Kohlenstoffmonoxid und Kohlenwasserstoffen. Die Emissionen der Fahrzeuge werden somit in realen Verkehrssituationen bestimmt und erlauben beispielsweise Aussagen über die Emissionen der Fahrzeugflotte, den Anteil hochemittierender Fahrzeuge am gesamten Fahrzeugbestand, dem Alterungsverhalten von Abgasreinigungssystemen sowie der Abhängigkeiten der Schadstoffwerte von einzelnen Abgasstufen (EURO-Normen). Darüber hinaus ermöglichen die Daten eine Untersuchung des Einflusses verkehrsbezogener Faktoren, wie etwa der Fahrdynamik, auf den Abgasausstoss. Der Datensatz enthält hunderttausende Messdaten aus Messkampagnen seit dem Jahr 2002 (jährlich, beziehungsweise seit dem Jahr 2022 mit Unterbrüchen), die mit ausreichendem Fachwissen eine aussagekräftige Auswertung zulassen. Weitere Informationen sind der abgelegten Datensatz-Beschreibung zu entnehmen. Für eine Methodendokumentation, Beispielauswertungen und deren fachliche Einordnung sei auf den verlinkten Fachbericht verwiesen.

Schlagwörter


Abgase Emissionen Fahrzeuge Luft Luftqualitaet Luftschadstoffe Ogd Stickoxide Strassenverkehr

## Ressourcen

### Datensatzbeschreibung (englisch)

Grösse 10 KB Format TXT Aktualisiert 08.01.2024


[Details anzeigen](#)


Herunterladen 

### Daten RSD-Messkampagne 2002

Grösse 27 MB Format CSV Aktualisiert 08.01.2024

[Details anzeigen](#)


Herunterladen 


Konvertiere zu Excel 

### Daten RSD-Messkampagne 2003

Grösse 36 MB Format CSV Aktualisiert 08.01.2024

[Details anzeigen](#)

Herunterladen 

Konvertiere zu Excel 

### Daten RSD-Messkampagne 2004

# AWEL ABGASMESSUNGEN



**Offich, 08.01.2023**  
**Auftrag für Waste, Water, Energy and Air (AWEL) / Canton Zürich / Switzerland**  
 Stämpfenbachstrasse 12, 8090 Zürich  
 www.zh.ch  
 Email: luft@bz.ch

This dataset is the result of long-standing measurement campaigns with vehicle "Remote Emission Sensing" (RES) employing a so-called "Remote Sensing Detector" (RSD).

The RSD is a continuously operating optical open-path device with the light beam passing over one road lane at about the height of the tailpipe. The instrument (in the RES approach, the RSD captures the increase in pollutant concentration due to an emission plume of a passing vehicle. The increase in pollutant concentration is thus used as a measure for the pollutant concentration in the exhaust emissions (see also <https://doi.org/10.1021/ar50649v>). Additionally, vehicle velocity and acceleration at the moment of measurement are recorded. Some general vehicle properties (e.g.: fuel type, European emission norm). These data do not allow in any way to identify individual vehicles, their owner or driver.

Single plume measurements with RSD are imprecise. However, measuring thousands of plumes, the emissions of representative vehicle categories can be statistically estimated.

Since the year 2002, the Canton of Zürich / "Amt für Abfall, Energie, Wasser und Luft AWEL" conducts RSD measurement campaigns during the summer months. The goal of RSD measurements are typically conducted at the same one or two specific locations in the Canton of Zürich.

These measurements are part of the program to monitor the effect of air quality regulation and e.g. help to improve the quantification of real emissions, estimate the impact of traffic measures on air quality.

In the past, focus of data analysis has been laid on NOx emissions. However, RSD also captures (partly only in some years, depending on the instrument model) carbon results for all of these parameters (if available) are included in the dataset, which begins in the year 2002 with yearly measurement campaigns. Since the year 2017, CO<sub>2</sub> emissions are also measured.

The AWEL has provided an extended report about the results of the measurement campaigns between the years 2002 and 2021, see:

Sinternann, J., Alt, G.W., Götsch, M., Baum, F., Delb, V. (2021): Langjährige Abgasmessungen in realen Fahrbetrieb mittels Remote Sensing (No. vl.4); Kantonsrat des Kantons Zürich. [https://www.zh.ch/content/dam/zhibw/bilder-dokumente/themen/umwelt-tiere/luft-strahlung/luftschadstoffqualitaet/verkehr/abgasmessungen-rsd\\_rsd\\_bericht\\_2021.pdf](https://www.zh.ch/content/dam/zhibw/bilder-dokumente/themen/umwelt-tiere/luft-strahlung/luftschadstoffqualitaet/verkehr/abgasmessungen-rsd_rsd_bericht_2021.pdf)

A summary including results from recent measurement campaigns can also be found here (in German):

<https://www.zh.ch/de/umwelt-tiere/luft-strahlung/luftschadstoffqualitaet/emissionen-verkehr/abgasmessungen-rsd.html>

To understand more about the method, the results and their context, the reader is advised to study the appropriate literature (see e.g. references below).

The present dataset contains all basic parameters required to start RES analysis. The calculated undiluted tailpipe concentration of measured pollutants are therefore directly available. Even though we have undertaken all measures required to appropriately conduct RSD measurements, we cannot guarantee correctness of the data. The user is strongly recommended to verify the data by other means.

The provided data files are structured according to the year of measurement and they will be updated periodically with additional results from future measurement campaigns. The datasets are published on <https://opendata.swiss> with a "free" license, meaning:

- the data can be used for non-commercial purposes,
- the data can be used for commercial purposes,
- referencing the data source is not required, but recommended:  
 - German: "Amt für Abfall, Energie, Wasser und Luft (AWEL) / kanton Zürich / Schweiz, Langjährige Abgasmessungen in realen Fahrbetrieb mittels Remote Sensing"  
 - English: "Office for Waste, Water, Energy and Air (AWEL) / Canton Zürich / Switzerland, Monitoring real vehicle emissions using remote emission sensing"

The file headers are comma-separated ".csv" files formatted in UTF-8 typesetting; missing data is marked "NA". They contain the following parameters:

```
- "id": general unique identifying number per measurement record [integer]
- "date": measurement date of individual plume measurement in format YYYY-MM-DD [string]
- "site": generalized measurement site identifier in format "A.X.Y.Z" [string]
- "site_roadname": measurement site's road name [double]
- "site_air_temperature": air temperature at the time of measurement, in degree Celsius, derived from nearby meteorological stations [double]
- "res_id": name of the used res model / instrument-number [string]
- "vehicle_type": general vehicle category 'passenger car' or 'light duty vehicle' [string]
- "vehicle_fuel_type": vehicle fuel type either 'gasoline' or 'diesel' [string]
- "vehicle_euronorm": various emission norms 'EuroII...' etc [string]
- "vehicle_make": various vehicle makes [string]
- "vehicle_model": various vehicle models [string]
- "vehicle_model_year": year of vehicle model, derived the official certificate of conformity data code of homologation (from https://lvs.opendata.ch/opendata/v1/en/homologation-weight) [integer]
- "vehicle_engine_displacement": vehicle engine displacement in ccn [integer]
- "vehicle_unloaded_weight": unloaded weight in kg [integer]
- "vehicle_initial_registration": date of first vehicle registration (if available); in format YYYY-MM-DD [string]
- "parameter": measurement parameter such as vehicle's instant 'velocity' or 'acceleration' or the calculated undiluted (except for CO2) tailpipe concentration [string]
- "value": record of measured parameter [double]
- "unit": accounting unit of the record [string]
```

Vehicle registration data ("vehicle\_\*") in the dataset originate from the cantonal Strassenverkehrsamt (In years prior to 2018) and are provided courtesy of the Swiss Federal Office of Transport (BFS). In case a specific vehicle model has been recorded less than 10 times per measurement campaign, i.e. per year, "vehicle\_model" has been set NA to make it impossible to identify individual vehicles.

Further references (not complete, in alphabetical order):

- Bernard, Y., Tietje, U., Gorman, J., Müncrief, R., 2018. Determination of real-world emissions from passenger vehicles using remote sensing data. ICCT, Berlin, p. 44.
- Bishop, G.A., Steadman, D.J., 2008. A Decade of On-road Emissions Measurements. Environmental Science & Technology 42, 1651–1656. <https://doi.org/10.1021/es07049a>.
- Bishop, G.A., Stedman, D.J., 1996. Measuring the Emissions of Passing Cars. Acc. Chem. Res. 29, 489–495. <https://doi.org/10.1021/ar50649v>.
- Borken-Kleefeld, J., Bernard, Y., Carlswald, D., 5jodina, A., 2018. Contribution of vehicle remote sensing to in-service/realt driving emissions monitoring and assessment. Atmospheric Environment 180, 107–117. <https://doi.org/10.1016/j.atmosenv.2018.01.011>.
- Borken-Kleefeld, J., Chen, Y., 2015. New emission deterioration rates for gasoline cars – Results from long-term measurements. Atmospheric Environment 115, 107–117. <https://doi.org/10.1016/j.atmosenv.2015.01.011>.
- Borken-Kleefeld, J., Chen, Y., 2014. Real-driving emissions from diesel passenger cars – Results from long-term measurements. Atmospheric Environment 115, 107–117. <https://doi.org/10.1016/j.atmosenv.2015.01.011>.
- Chen, Y., Borken-Kleefeld, J., 2014. Real-driving emissions from Diesel Passenger Cars – Results from long-term measurements. Atmospheric Environment 115, 107–117. <https://doi.org/10.1016/j.atmosenv.2015.01.011>.
- Chen, Y., Borken-Kleefeld, J., 2014. Real-driving emissions from cars and light commercial vehicles – Results from 13 years remote sensing at Zurich/CH. Atmospheric Environment 115, 107–117. <https://doi.org/10.1016/j.atmosenv.2015.01.011>.
- Chen, Y., Sun, R., Borken-Kleefeld, J., 2020. On-Road NOx and Smoke Emissions of Diesel Light Commercial Vehicles-Combining Remote Sensing Measurements and Exhaust Gas Analysis. Atmospheric Environment 230, 117777. <https://doi.org/10.1016/j.atmosenv.2020.117777>.
- REMOVES, 2014. Überwachung der Emissionen von Strassenfahrzeugen in der Schweiz. <https://www.mobilityplatform.ch/de/research-data-shop/smobilityvehicles/>

```
rules:
- expr: is.numeric(id) & id == floor(id)
  name: 'id_is_integer'
  description: 'The "id" field must be numeric and an integer.'

- expr: grepl("[0-9]{4}-[0-9]{2}-[0-9]{2}$", date_measured)
  name: 'date_measured_format'
  description: 'The "date_measured" field must be in YYYY-MM-DD format.'

- expr: grepl("[A-Z]$", site)
  name: 'site_format'
  description: 'The "site" field must be a single uppercase letter from A to Z.'

- expr: is.numeric(site_roadgrade)
  name: 'site_roadgrade_is_numeric'
  description: 'The "site_roadgrade" must be a numeric value (double).'

- expr: is.numeric(site_air_temperature)
  name: 'site_air_temperature_is_numeric'
  description: 'The "site_air_temperature" must be a numeric value (double).'

- expr: is.character(rsd_model)
  name: 'rsd_model_is_string'
  description: 'The "rsd_model" field must be a string.'

- expr: vehicle_type %in% c('passenger car', 'light duty vehicle')
  name: 'vehicle_type_valid'
  description: 'The "vehicle_type" must be either "passenger car" or "light duty vehicle".'

- expr: vehicle_fuel_type %in% c('gasoline', 'diesel')
  name: 'vehicle_fuel_type_valid'
  description: 'The "vehicle_fuel_type" must be either "gasoline" or "diesel".'

- expr: grepl("^Euro[0-9]+$", vehicle_euronorm)
  name: 'vehicle_euronorm_format'
  description: 'The "vehicle_euronorm" must follow the format "Euro1", "Euro2", etc.'
```



# INTEGRATION YAML R

```
# Ergebnisse der Datenvalidierung
resultat_EKZ ← confront(EKZ_data, validator(.file = "rules.yaml"))

# Zusammenfassung Ergebnisse
resultat_EK_df ← summary(resultat_EKZ)

# if else Statement
if(any(resultat_EK_df$fails > 0)) {
  warning("!Einige Datenpunkte haben die Validierungsregeln NICHT erfüllt!")
  print(resultat_EK_df)
} else {
  message("Alle Datenpunkte haben die Validierungsregeln erfüllt")
}
...
```

Alle Datenpunkte haben die Validierungsregeln erfüllt

# Daten validieren, Fehler minimieren!



FRAGEN?

IDEEN ANWENDUNG?

Several thin, white, parallel diagonal lines are positioned in the bottom right corner of the slide, extending from the right edge towards the center.