Racing Insights

Datenbericht

Letzte Änderung: 25.05.2025

# Rohdaten

## Übersichtstabelle der Rohdatensätze

|  |  |  |
| --- | --- | --- |
| **Datensatz Name** | **Quelle / Beschreibung** | **Speicherort** |
| Calendar | FastF1 Python-Bibliothek: Kalenderdaten der Saison aus offiziellen F1-Quellen. | <https://docs.fastf1.dev> - Daten werden automatisch von offiziellen F1-Endpunkten über FastF1 geladen und lokal zwischengespeichert |
| Session Data | FastF1 Python-Bibliothek: Telemetrie-, Wetter-, Positions-, Rundenzeiten- und Punkte-Daten aus offiziellen F1-Quellen. | <https://docs.fastf1.dev> - Daten werden automatisch von offiziellen F1-Endpunkten über FastF1 geladen und lokal zwischengespeichert |

## Details zu den Datensätze Calendar- und Session-Data

|  |  |  |
| --- | --- | --- |
|  | **Calendar Data** | **Session Data** |
| Beschreibung, welche Informationen die Daten enthalten | Calendar-Daten enthält Informationen wann, wo und welches Rennen für eine bestimmtes Saison stattgefunden hat, | Session-Daten enthält Informationen wie Rundenzeiten, Telemetrie (Geschwindigkeit, Gaspedal, Bremsen, G-Kräfte), Wetterdaten, Positionsdaten und Punkte aller Fahrer. |
| Details zur Datenquelle | <https://docs.fastf1.dev/index.html>  Offizielle Formel-1-Daten, bereitgestellt über API-Zugriff und Webscraping durch FastF1 | |
| Informationen zur Datenbeschaffung | Zuerst wird das Python Paket fastf1 importiert und mit fastf1.get\_event\_schedule(jahr) wird der Kalender für die entsprechende Saison geladen. | Daten werden über Python mit fastf1.get\_session(jahr, gp, session) geladen. Nutzung von Caching möglich |
| Rechtliche Aspekte zur Nutzung der Daten, Lizenzen etc | FastF1 steht unter MIT-Lizenz. Datennutzung gemäss F1-Richtlinien, nur für nicht-kommerzielle Zwecke empfohlen. | |
| Data Governance-Aspekte | Daten sind öffentlich und für alle zugänglich. Sie enthalten keine personenbezogenen Daten | |
| Wie kann auf die Daten zugegriffen werden | Via Python: fastf1, Nutzung von Pandas für Analyse, Plotly/Matplotlib für Visualisierungen. | |

## Datenkatalog: Datensatz FastF1 - Calendar Data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Spaltenname** | **Datentyp** | **Werte (Wertebereich)** | **Kurze Beschreibung** | **Primary/Foreign Key** |
| RoundNumber | int | 1 - 24 | Nummer der Runde | Ja |
| Country | chr | "Bahrain", "Saudi Arabian", "Australian", "Japanese", "Chinese", "Miami", "Emilia Romagna", "Monaco", "Canadian", "Spanish", "Austrian", "British", "Hungarian", "Belgian", "Dutch", "Italian", "Azerbaijan", "Singapore", "United States", "Mexico City", "São Paulo", "Las Vegas", "Qatar", "Abu Dhabi" | Land, wo es stattfindet | Nein |
| Location | chr | "Sakhir", "Jeddah", "Melbourne", "Suzuka", "Shanghai", "Miami", "Imola", "Monaco", "Montréal", "Barcelona", "Spielberg", "Silverstone", "Budapest", "Spa-Francorchamps", "Zandvoort", "Monza", "Baku", "Marina Bay", "Austin", "Mexico City", "São Paulo", "Las Vegas", "Lusail", "Yas Island" | Ort, wo es stattfindet | Nein |
| OfficialEventName | chr | FORMULA 1 \*SPONSOR\* \*COUNTRY \* GRAND PRIX \*YEAR\* | Der Offizielle Name vom Event | Nein |
| EventDate | date | yyyy-mm-dd | Datum, wann das Event stattfinded | Nein |
| EventName | chr | \*Contry\* Grand Prix | Name vom Event | Nein |
| EventFormat | chr | "conventional", "sprint\_qualifying" | Format vom Event | Nein |
| Session1 | chr | "Practice 1" | Name von Session 1 | Nein |
| Session1Date | date-time | yyyy-mm-dd hh:mm:ss+timezone | Datum & Uhrzeit (Lokal), wann Session 1 stattfindet | Nein |
| Session1DateUtc | date-time | yyyy-mm-dd hh:mm:ss | Datum & Uhrzeit (UTC), wann Session 1 stattfindet | Nein |
| Session2 | chr | "Practice 2" "Sprint Qualifying" | Name von Session 2 | Nein |
| Session2Date | date-time | yyyy-mm-dd hh:mm:ss+timezone | Datum & Uhrzeit (Lokal), wann Session 2 stattfindet | Nein |
| Session2DateUtc | date-time | yyyy-mm-dd hh:mm:ss | Datum & Uhrzeit (UTC), wann Session 2 stattfindet | Nein |
| Session3 | chr | "Practice 3" "Sprint" | Name von Session | Nein |
| Session3Date | date-time | yyyy-mm-dd hh:mm:ss+timezone | Datum & Uhrzeit (Lokal), wann Session 3 stattfindet | Nein |
| Session3DateUtc | date-time | yyyy-mm-dd hh:mm:ss | Datum & Uhrzeit (UTC), wann Session 3 stattfindet | Nein |
| Session4 | chr | Qualifying" | Name von Session 4 | Nein |
| Session4Date | date-time | yyyy-mm-dd hh:mm:ss+timezone | Datum & Uhrzeit (Lokal), wann Session 4 stattfindet | Nein |
| Session4DateUtc | date-time | yyyy-mm-dd hh:mm:ss | Datum & Uhrzeit (UTC), wann Session 4 stattfindet | Nein |
| Session5 | chr | "Race" | Name von Session 5 | Nein |
| Session5Date | date-time | yyyy-mm-dd hh:mm:ss+timezone | Datum & Uhrzeit (Lokal), wann Session 5 stattfindet | Nein |
| Session5DateUtc | date-time | yyyy-mm-dd hh:mm:ss | Datum & Uhrzeit (UTC), wann Session 5 stattfindet | Nein |
| F1ApiSupport | bool | "True", "False" | Ob F1 API support vorhanden ist | Nein |

## Datenkatalog: Datensatz FastF1 - Session Data

Die ersten zwölf Spalten gehören zum session.laps DataFrame und enthalten Runden-, Sektor- und Boxendaten pro Fahrer, während die letzten sechs Spalten aus dem lap.get\_car\_data() DataFrame stammen und detaillierte Telemetriedaten wie Geschwindigkeit, Gasstellung, Bremsen und Gangwahl enthalten.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Spaltenname** | **Datentyp** | **Werte (Wertebereich)** | **Kurze Beschreibung** | **Primary/Foreign Key** |
| Time | timedelta64[ns] | ≥ 0 | Zeitstempel relativ zum Session-Start | Nein |
| Driver | str | 3-Buchstaben-Code | Fahrerkennung (z. B. 'VER', 'HAM') | Ja (Fahrer-ID) |
| LapTime | timedelta64[ns] | ≥ 0 | Dauer der Runde | Nein |
| LapNumber | int | ≥ 1 | Rundenanzahl | Nein |
| Sector1Time | timedelta64[ns] | ≥ 0 | Zeit für Sektor 1 | Nein |
| Sector2Time | timedelta64[ns] | ≥ 0 | Zeit für Sektor 2 | Nein |
| Sector3Time | timedelta64[ns] | ≥ 0 | Zeit für Sektor 3 | Nein |
| Compound | str | 'SOFT', 'MEDIUM', 'HARD' | Reifenmischung | Nein |
| TyreLife | int | ≥ 0 | Rundenanzahl auf Reifensatz | Nein |
| PitOutTime | timedelta64[ns] | ≥ 0 oder NaT | Zeit beim Boxen-Ausgang | Nein |
| PitInTime | timedelta64[ns] | ≥ 0 oder NaT | Zeit beim Boxen-Eingang | Nein |
| IsAccurate | bool | True / False | Zuverlässigkeit der Rundendaten | Nein |
| Time | timedelta64[ns] | ≥ 0 | Zeitstempel relativ zur Session-Zeit | Nein |
| Speed | float | ≥ 0 (km/h) | Momentangeschwindigkeit | Nein |
| Throttle | float | 0.0 – 100.0 (%) | Gasstellung | Nein |
| Brake | bool | True / False | Ob Bremse betätigt wird | Nein |
| RPM | int | 0 – 15000 | Motordrehzahl | Nein |
| nGear | int | 0 – 8 | Aktueller Gang | Nein |

## Datenqualität: Datensatz FastF1 - Calendar Data

In Python wurden die Datensätze für das Jahr 2024 aufgerufen und anschliessend als eine .csv Datei abgespeichert. Diese Datei wurde dann in eine Streamlit-Applikation (<https://dsg-app.streamlit.app/EDA>) hochgeladen, welche die Explorative Datenanalyse automatisiert durchführt.

### Calender-Datensatz:

|  |  |
| --- | --- |
| **Metrik** | **Wert** |
| Anzahl Spalten | 23 |
| Anzahl Zeilen | 24 |
| Anzahl leerer Zellen | 0 |
| Anteil (%) leerer Zellen | 0.00% |
| Anzahl duplizierter Zeilen | 0 |
| Anteil (%) duplizierter Zeilen | 0.00% |

|  |  |
| --- | --- |
| **Datentyp** | **Anzahl Spalten** |
| Kategorisch | 21 |
| Numerisch | 2 |
| Datum | 0 |

Die Datenstruktur ist vollständig und fehlerfrei. Alle 23 Spalten und 24 Zeilen sind vollständig befüllt, ohne Duplikate oder Nullwerte. Dies spricht für eine sehr hohe Datenqualität. Der Datentyp-Mix ist überwiegend kategorisch, was zur Beschreibung von Events, Sessions und Veranstaltungsorten passend ist.

### Session-Datensatz:

### Track\_status-Tabelle:

|  |  |
| --- | --- |
| **Metrik** | **Wert** |
| Anzahl Spalten | 3 |
| Anzahl Zeilen | 7 |
| Anzahl leerer Zellen | 0 |
| Anteil (%) leerer Zellen | 0.00% |
| Anzahl duplizierter Zeilen | 0 |
| Anteil (%) duplizierter Zeilen | 0.00% |

|  |  |
| --- | --- |
| **Datentyp** | **Anzahl Spalten** |
| Kategorisch | 2 |
| Numerisch | 1 |
| Datum | 0 |

Trotz der kleinen Datenmenge (7 Zeilen) zeigt sich auch hier eine fehlerfreie Datenbasis mit nur drei Spalten. Die Kombination aus zwei kategorialen und einer numerischen Spalte ermöglicht eine einfache Analyse von Streckenstatuswechseln (z. B. Green, Yellow, SC) und deren zeitliche Einordnung.

### Race\_control- Tabelle:

|  |  |
| --- | --- |
| **Metrik** | **Wert** |
| Anzahl Spalten | 9 |
| Anzahl Zeilen | 69 |
| Anzahl leerer Zellen | 233 |
| Anteil (%) leerer Zellen | 37.52% |
| Anzahl duplizierter Zeilen | 0 |
| Anteil (%) duplizierter Zeilen | 0.00% |

|  |  |
| --- | --- |
| **Datentyp** | **Anzahl Spalten** |
| Kategorisch | 6 |
| Numerisch | 3 |
| Datum | 0 |

Der einzige Datensatz mit auffälliger Datenlücke: Mit 37.5 % fehlenden Werten (233 leere Zellen) ist die Vollständigkeit eingeschränkt. Trotzdem liefern die vorhandenen Einträge wertvolle Informationen zu offiziellen Mitteilungen der Rennleitung. Da keine Duplikate vorliegen, sind die gültigen Einträge einzigartig und relevant.

### Weather- Tabelle:

|  |  |
| --- | --- |
| **Metrik** | **Wert** |
| Anzahl Spalten | 8 |
| Anzahl Zeilen | 157 |
| Anzahl leerer Zellen | 0 |
| Anteil (%) leerer Zellen | 0.00% |
| Anzahl duplizierter Zeilen | 0 |
| Anteil (%) duplizierter Zeilen | 0.00% |

|  |  |
| --- | --- |
| **Datentyp** | **Anzahl Spalten** |
| Kategorisch | 7 |
| Numerisch | 1 |
| Datum | 0 |

Dieser Datensatz ist vollständig und präzise. Über 157 Einträge mit Wetterparametern wurden erfasst, die ideal für zeitliche Verlaufsauswertungen geeignet sind. Die hohe Anzahl an kategorialen Spalten unterstützt qualitative Bewertungen der Bedingungen, etwa Windrichtung, Regen oder Wolkenbedeckung.

Durch die stichprobenartige Überprüfung der CSV-Dateien unseres Datensatzes für das Jahr 2024 konnten wir eine hohe Datenvollständigkeit feststellen. Auf Basis dieser Qualitätsprüfung gehen wir davon aus, dass auch die Datensätze aus den übrigen Jahren eine vergleichbar hohe Datenqualität aufweisen. Daher halten wir es für vertretbar, diese ebenfalls ohne Einschränkungen in unseren Datenkatalog aufzunehmen und für Analysen zu verwenden.“

# Prozessierte Daten

## Übersichtstabelle der Prozessierten Daten

|  |  |  |
| --- | --- | --- |
| Name | Input-Datensätze | Speicherort |
| Laps | Session Timing Data, Driver Info | session.laps (Pandas DataFrame) |
| SessionResults | Final Classification Data | session.results  (Pandas DataFrame) |
| CarData | Raw Telemetry Streams | lap.get\_car\_data()  (Pandas DataFrame) |
| PositionData | GPS Data Stream | lap.get\_pos\_data()  (Pandas DataFrame) |
| WeatherData | Session Weather Feed | session.weather\_data  (Pandas DataFrame) |
| TrackStatus | Race Control Feed | session.track\_status  (Pandas DataFrame) |
| CircuitInfo | Static Track Metadata | session.get\_circuit\_info() (dict) |
| RaceControlMessages | Race Control Announcements | session.race\_control\_messages (Pandas DataFrame) |

## Details der Prozessierten Daten

Die Daten wurden nur mit dem Datensatz Session-Data prozessiert.

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Beschreibung** | **Prozessierung** | **Zugriff** |
| Laps | Informationen zu jeder gefahrenen Runde: Zeiten, Sektoren, Reifentyp, Boxenstopps. | Verknüpfung von Timing-Daten mit Fahrer- und Reifeninformationen. Aufbereitung in DataFrame. | session.laps – direkt nach session.load(). Zugriff über Pandas-Methoden möglich. |
| SessionResults | Endergebnisse einer Session: Platzierung, Punkte, Status (DNF, DNS). | Zusammenstellung aus offiziellen Session-Klassifikationen. Standardisiert. | session.results – Zugriff nach session.load() als DataFrame. |
| CarData | Telemetrie pro Runde: Geschwindigkeit, Gaspedal, Bremse, Gang, RPM. | Extraktion und Normierung der Roh-Telemetrie. Interpolation bei Lücken. | lap.get\_car\_data() – Rückgabe als DataFrame, nutzbar mit Matplotlib oder Plotly. |
| PositionData | GPS-Positionen des Fahrzeugs auf der Strecke über Zeit. | Konvertierung und Interpolation der GPS-Telemetrie. Zeitlich synchronisiert. | lap.get\_pos\_data() – als DataFrame mit X/Y-Koordinaten und Zeitstempeln. |
| WeatherData | Wetterbedingungen: Temperatur, Luftfeuchtigkeit, Wind, Regen. | Erfassung in festen Zeitintervallen, Transformation in tabellarische Form. | session.weather\_data – DataFrame mit zeitlichem Verlauf. |
| TrackStatus | Status der Strecke: Green, Yellow, Red, Safety Car, VSC. | Mapping der Rennleitungscodes zu Textstatus. Zeitlich sortiert. | session.track\_status – DataFrame mit Zeitfenstern und Statuscodes. |
| CircuitInfo | Streckeninformationen: Kurvenanzahl, Länge, DRS-Zonen. | Bereitstellung statischer Metadaten über die Strecke. | session.get\_circuit\_info() – Dictionary mit Streckendetails. |
| RaceControlMessages | Offizielle Mitteilungen der Rennleitung: Strafen, Hinweise, SC. | Parsing des Race-Control-Feeds. Kategorisierung nach Typ. | session.race\_control\_messages – DataFrame mit Zeitstempeln und Meldungstyp. |

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

KI-generierte Inhalte können fehlerhaft sein.