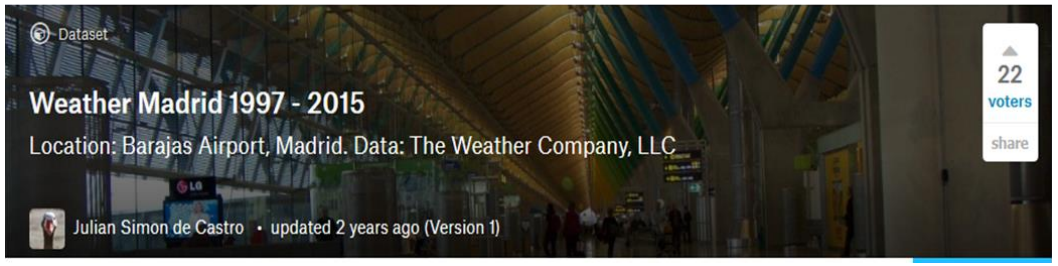
# WETTER UND LUFTQUALITÄT VON MADRID

**Inhalt**

- Beschreibung der Daten
- Spark Befehlsabfolge
- Vergleich CSV und Parquet
- Verwendung von RDD
- Lessons Learned - Fazit


Andreas Fischer, Bernd Novotny, Tobias Schieferdecker 13.01.2019

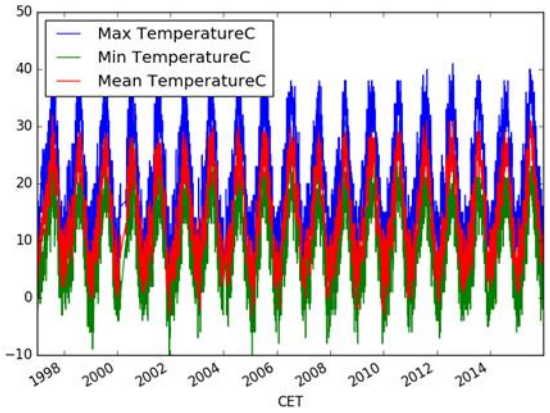
# Beschreibung der Daten «Wetter»

**Weather Madrid 1997 – 2015**

- https://www.kaggle.com/juliansimon/weather_madrid_lemd_1997_2015.csv

- Files:
  weather_madrid_LEMD_1997_2015.csv → 0.5 MB

- 6812 Zeilen
- pro Tag eine Angabe über
  Temperatur, Feuchtigkeit, Luftdruck, Visibility, Windgeschwindigkeit und -richtung, Niederschlag, Bedeckung, Vorkommnisse, Wolkendecke



# Beschreibung der Daten «Luftqualität»

**Air Quality in Madrid (2001 - 2018)**

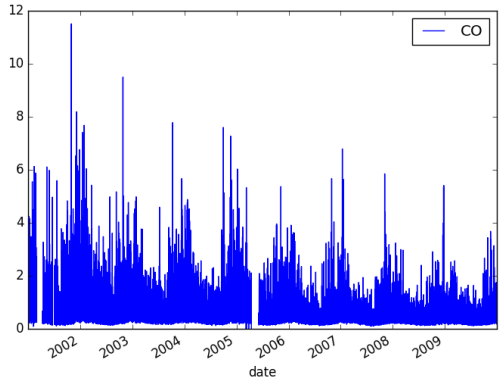
- https://www.kaggle.com/decide-soluciones/air-quality-madrid

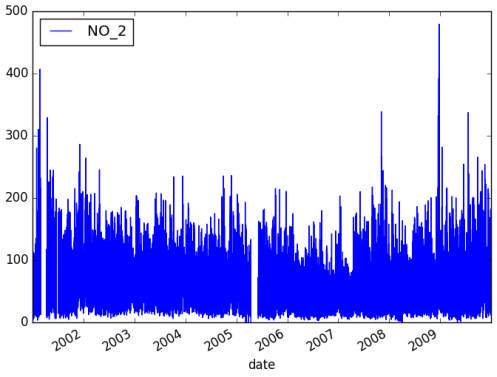
- Files:
  stations.csv – 24 Messstationen
  madrid_2001.csv ... madrid_2018.csv → 500 MB

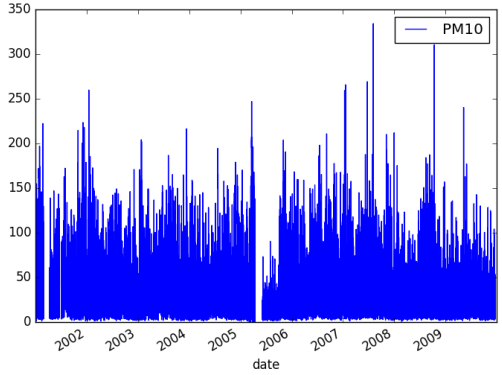
- 3.8 Millionen Tuples insgesamt
- pro Messstation jede Stunde eine Messreihe über
  Kohlenstoffmonoxid CO, Feinstaub PM10, Stickstoffdioxid NO2, Ozon O3, Nicht-Methan Kohlenwasserstoff NMHC, Benzene BEN und einige weitere

# Luftqualität Kohlenstoffmonoxid / Stickstoffdioxid / Feinstaub zwei Stationen

Station:

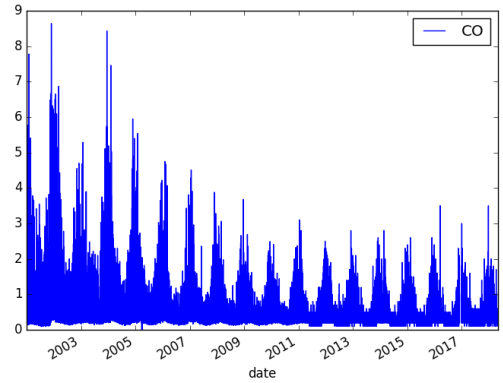


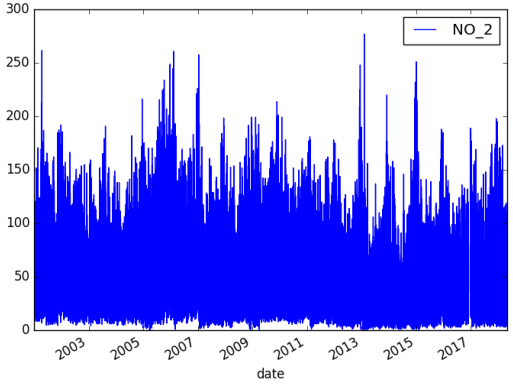



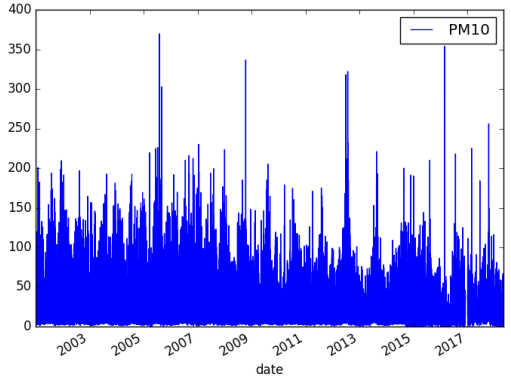




28079007

← nur bis 2010 !



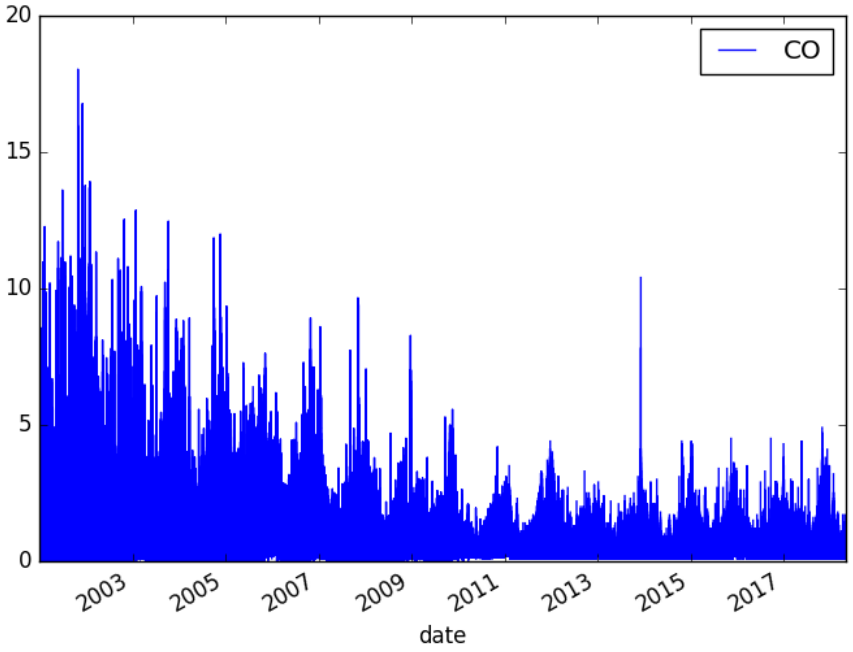



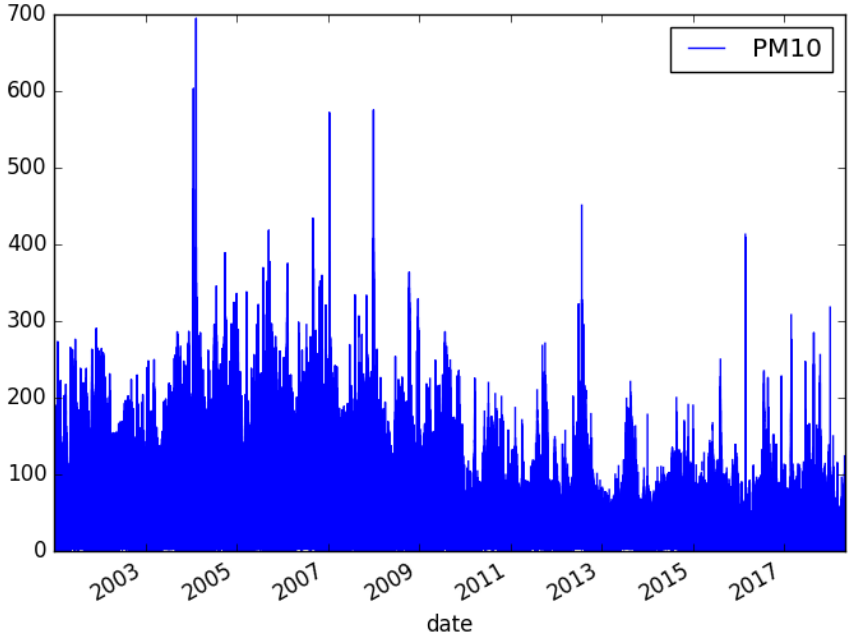

28079036

# Vorgehen Spark - databricks

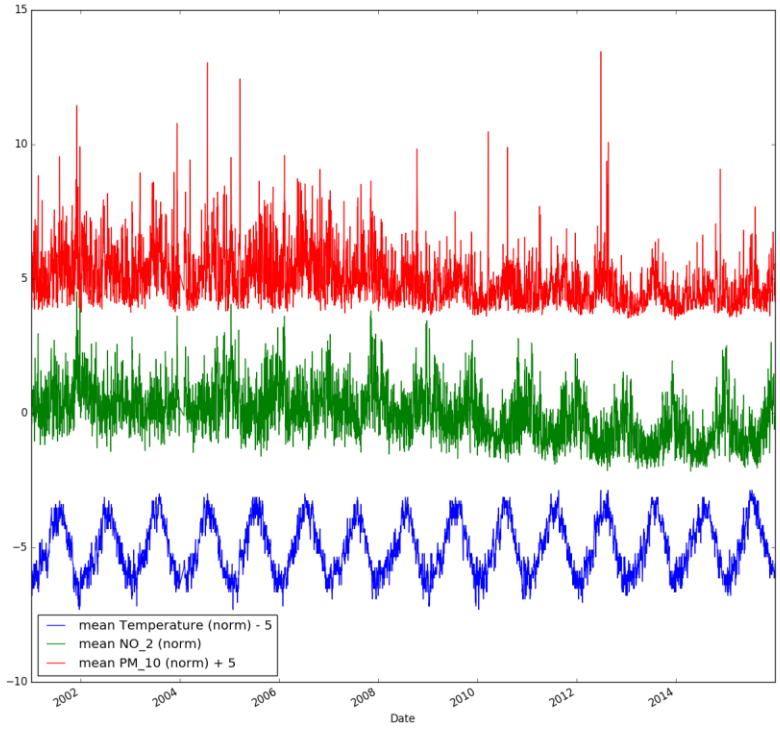
| | |
|---|---|
| • Einlesen der CSV Daten: | `sqlContext.read…` |
| • SQL-Abfrage: | `sqlContext.sql('SELECT … ')` |
| • Daten ansehen: | `show()/count()/printSchema()` |
| • Grafisch Darstellung | `toPandas()/plot(…)/display()` |
| • Daten zusammenführen: | `unionByName() / unionAll()` |
| • Reduktion auf gemeins. Attribute: | `drop()` |
| • Abspeichern in einem CSV-File: | `df.write.csv(…)` |
| • Abspeichern in Parquet-File: | `df.write.parquet(..)` |
| • Format Zeitstempel anpassen: | `AVG()` |
| • Grafiken mit gemittelten Daten: | `plot(…)` |
| • Join mit Wetterdaten: | `join()` |

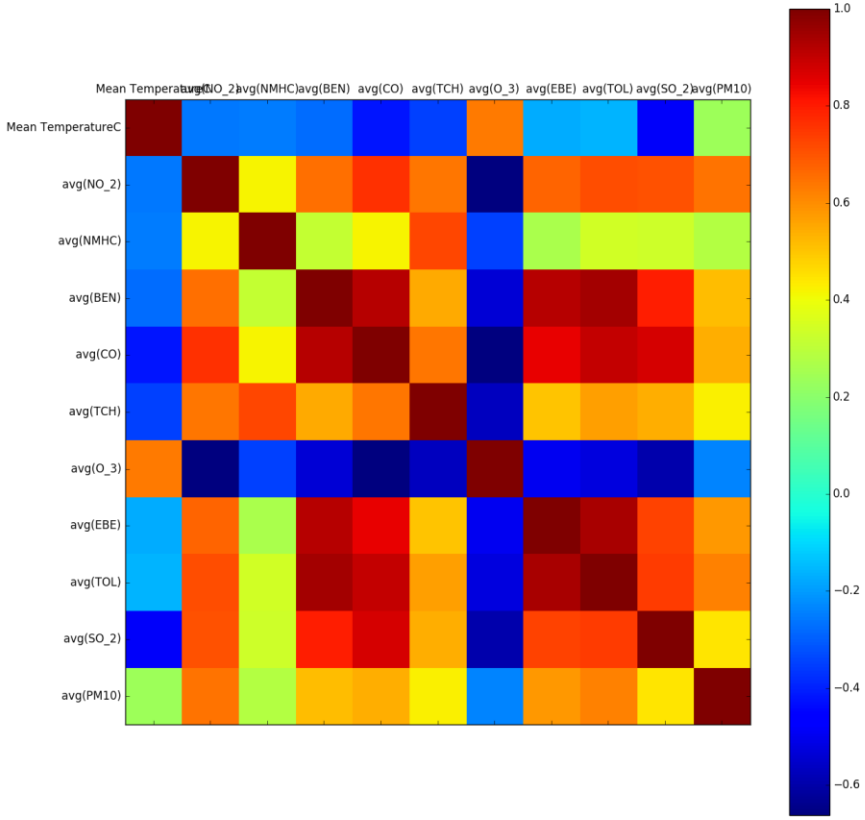
# Kohlenstoffmonoxid und Feinstaub aller Stationen



# Abhängigkeiten Schadstoffe – Temperatur?



# Einsatz von RDDs

- Für die Reduktion der Datumsangaben in den Air Pollution Daten (3.8 Mio Datensätze) von Stunden auf Tage wurde der Dataframe auf ein RDD transformiert und über ein Mapping die Datumsangaben ersetzt.

```
# helper method to allow normalization of dates by "deleting" hours and minutes, i.e. reducing to day information
import datetime
from pyspark import sql

def replace_date (zeile):
  datum = zeile["date"]
  z = zeile.asDict()
  z['date'] = datum.replace(hour=0, minute=0)
  return sql.types.Row(**z)

```

```
airRDD = air_df.rdd
# normalize all timestamps to day only, i.e. overwrite hour and minute by 0
airRDD_new = airRDD.map(lambda x: replace_date(x))
# convert back to dataframe
air_df_new = spark.createDataFrame(airRDD_new)
air_df_new.createOrReplaceTempView("airRDD_new")
```

# Vergleich CSV und Parquet

Join der Wetter Daten mit den Luftqualitätsdaten über das Datum.

Weather (csv)

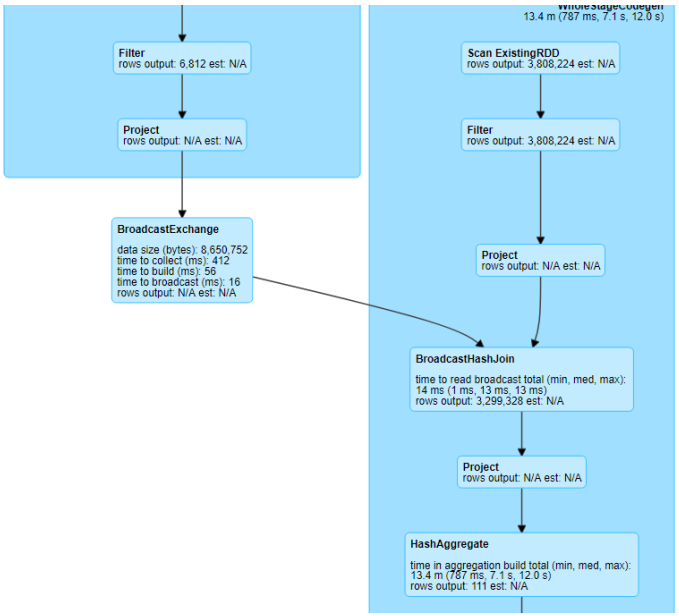
| | | | |
|---|---|---|---|
| … | | | |
| datetime(2001, 8, 1, 0, 0) | | | |
| …. | | | |

24

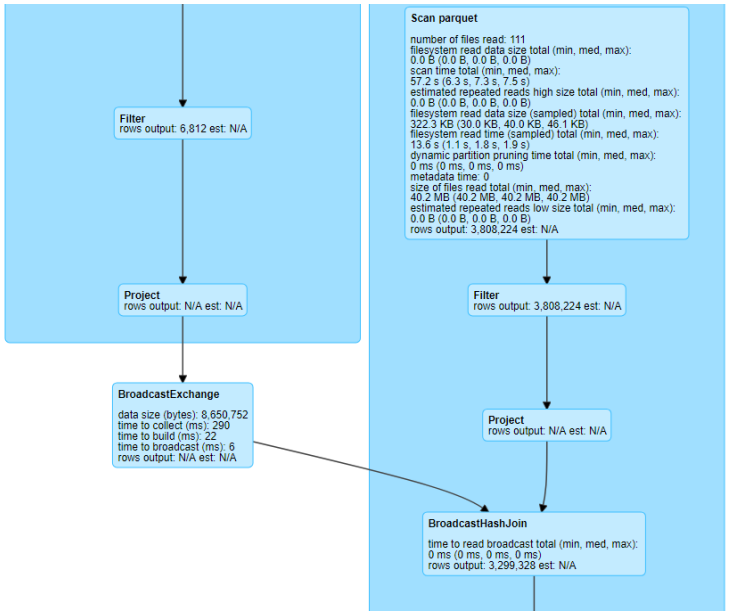
Air

| | | | |
|---|---|---|---|
| … | | | |
| datetime(2001, 8, 1, 0, 0) | | | |
| datetime(2001, 8, 1, 0, 0) | | | |
| … | | | |
| datetime(2001, 8, 1, 0, 0) | | | |
| … | | | |

Air als CSV




Air als Parquet



# Lessons Learned - Fazit

- Einlesen der CSV Dateien (eine pro Jahr) wird sehr schön parallelisiert: 18 Dateien -> 36 Jobs
- Parquet mag keine Leerzeichen in Attributnamen. Umbenennen mit `df.withColumnRenamed(<old-name>, <new-name>).collect()`
- Darauf achten, wo man den Cache Befehl platziert, damit er zum richtigen Zeitpunkt Wirkung erzielt (Lazy Evaluation).
- Bei Caching andere Execution Pläne als ohne Caching
- Auch mit Caching (in memory) join mit Parquet immer noch rund Faktor 5 schneller (ohne Caching Faktor 15)
- Transformation von Spark Dataframes in Pandas Dataframes kann seeehr lange dauern. Folgender Trick kann helfen:
  `spark.conf.set("spark.sql.execution.arrow.enabled", "true")`