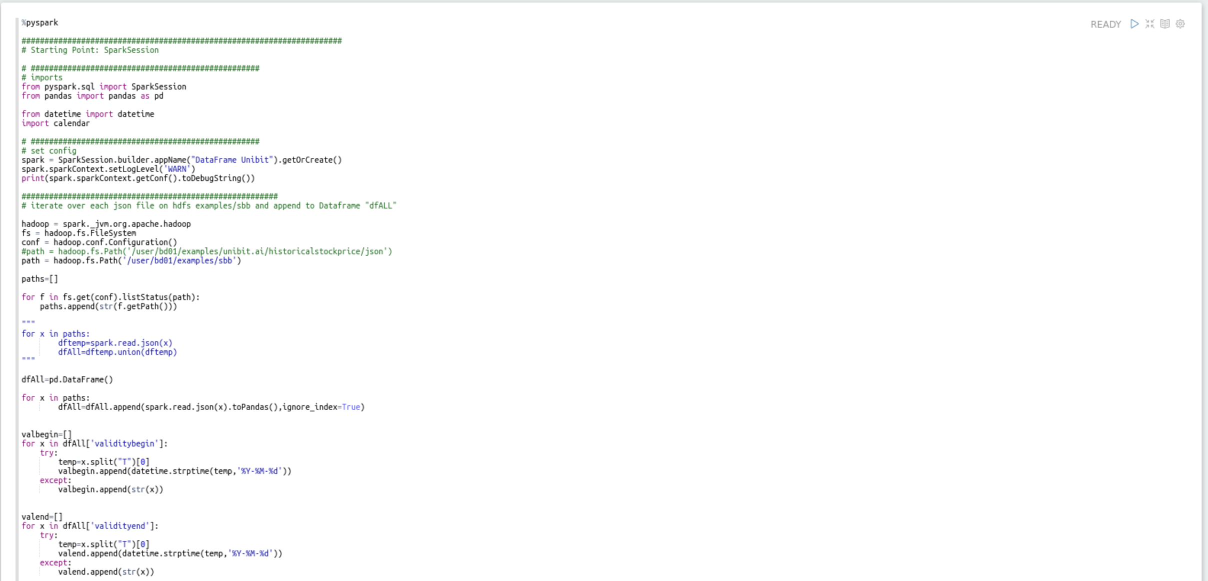
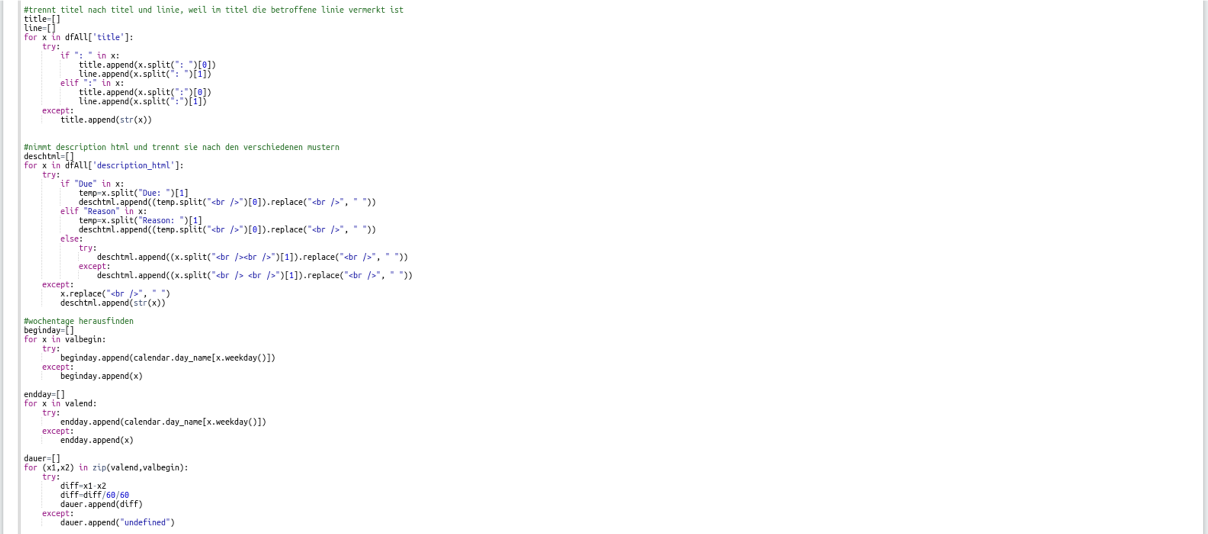
Zeppelin-Screenshots

1.



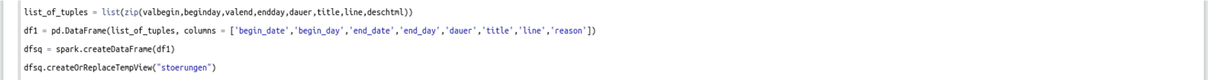
```
%pyspark

######################################################################
# Starting Point: SparkSession

# ##################################################
# imports
from pyspark.sql import SparkSession
from pandas import pandas as pd

from datetime import datetime
import calendar

# ##################################################
# set config
spark = SparkSession.builder.appName("DataFrame Unibit").getOrCreate()
spark.sparkContext.setLogLevel('WARN')
print(spark.sparkContext.getConf().toDebugString())

########################################################
# iterate over each json file on hdfs examples/sbb and append to Dataframe "dfALL"

hadoop = spark._jvm.org.apache.hadoop
fs = hadoop.fs.FileSystem
conf = hadoop.conf.Configuration()
#path = hadoop.fs.Path('/user/bd01/examples/unibit.ai/historicalstockprice/json')
path = hadoop.fs.Path('/user/bd01/examples/sbb')

paths=[]

for f in fs.get(conf).listStatus(path):
    paths.append(str(f.getPath()))

"""
for x in paths:
        dftemp=spark.read.json(x)
        dfAll=dftemp.union(dftemp)
"""

dfAll=pd.DataFrame()

for x in paths:
        dfAll=dfAll.append(spark.read.json(x).toPandas(),ignore_index=True)


valbegin=[]
for x in dfAll['validitybegin']:
    try:
        temp=x.split("T")[0]
        valbegin.append(datetime.strptime(temp,'%Y-%M-%d'))
    except:
        valbegin.append(str(x))


valend=[]
for x in dfAll['validityend']:
    try:
        temp=x.split("T")[0]
        valend.append(datetime.strptime(temp,'%Y-%M-%d'))
    except:
        valend.append(str(x))
```

2.



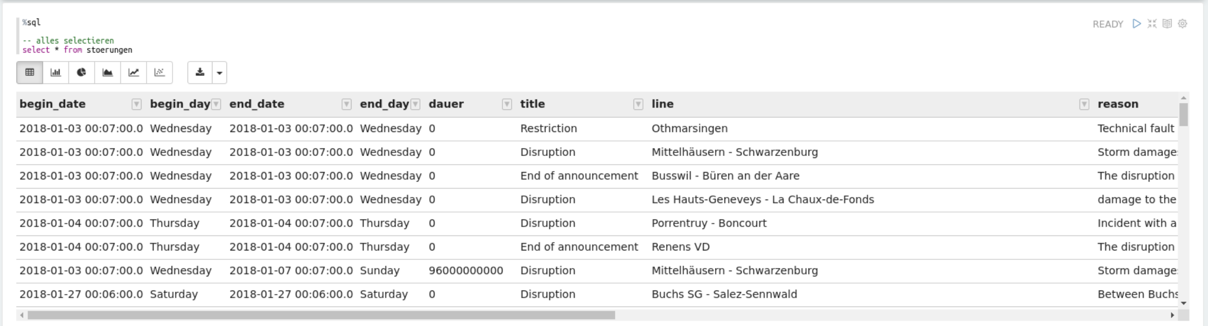
```
#trennt titel nach titel und linie, weil im titel die betroffene linie vermerkt ist
title=[]
line=[]
for x in dfAll['title']:
    try:
        if ": " in x:
            title.append(x.split(": ")[0])
            line.append(x.split(": ")[1])
        elif ":" in x:
            title.append(x.split(":")[0])
            line.append(x.split(":")[1])
    except:
        title.append(str(x))


#nimmt description html und trennt sie nach den verschiedenen mustern
deschtml=[]
for x in dfAll['description_html']:
    try:
        if "Due" in x:
            temp=x.split("Due: ")[1]
            deschtml.append((temp.split("<br />")[0]).replace("<br />", " "))
        elif "Reason" in x:
            temp=x.split("Reason: ")[1]
            deschtml.append((temp.split("<br />")[0]).replace("<br />", " "))
        else:
            try:
                deschtml.append((x.split("<br /><br />")[1]).replace("<br />", " "))
            except:
                deschtml.append((x.split("<br /> <br />")[1]).replace("<br />", " "))
    except:
        x.replace("<br />", " ")
        deschtml.append(str(x))

#wochentage herausfinden
beginday=[]
for x in valbegin:
    try:
        beginday.append(calendar.day_name[x.weekday()])
    except:
        beginday.append(x)

endday=[]
for x in valend:
    try:
        endday.append(calendar.day_name[x.weekday()])
    except:
        endday.append(x)

dauer=[]
for (x1,x2) in zip(valend,valbegin):
    try:
        diff=x1-x2
        diff=diff/60/60
        dauer.append(diff)
    except:
        dauer.append("undefined")
```

3.



```
list_of_tuples = list(zip(valbegin,beginday,valend,endday,dauer,title,line,deschtml))

df1 = pd.DataFrame(list_of_tuples, columns = ['begin_date','begin_day','end_date','end_day','dauer','title','line','reason'])

dfsq = spark.createDataFrame(df1)

dfsq.createOrReplaceTempView("stoerungen")
```

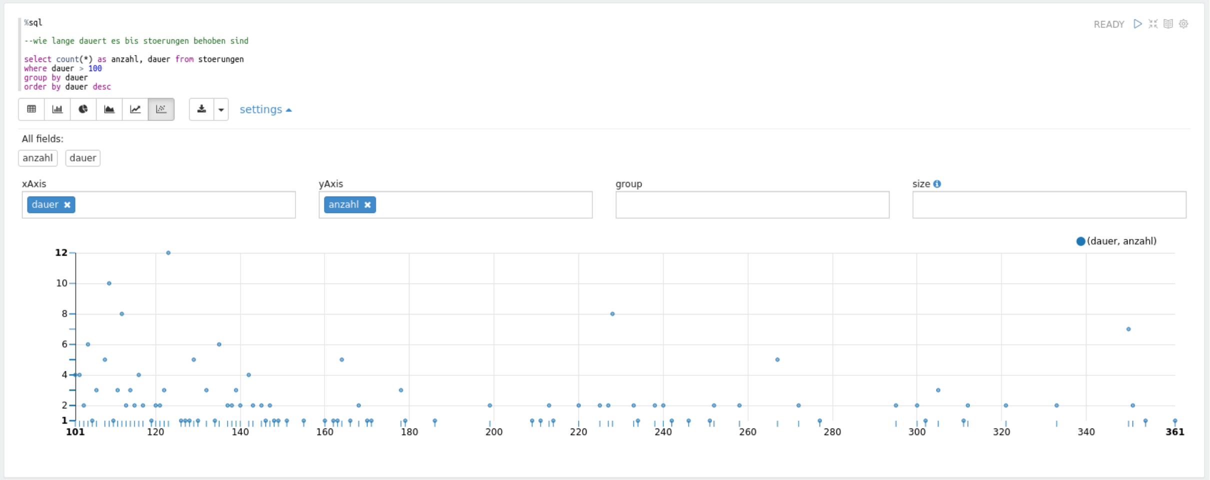
4.



```
%sql

-- alles selectieren
select * from stoerungen
```

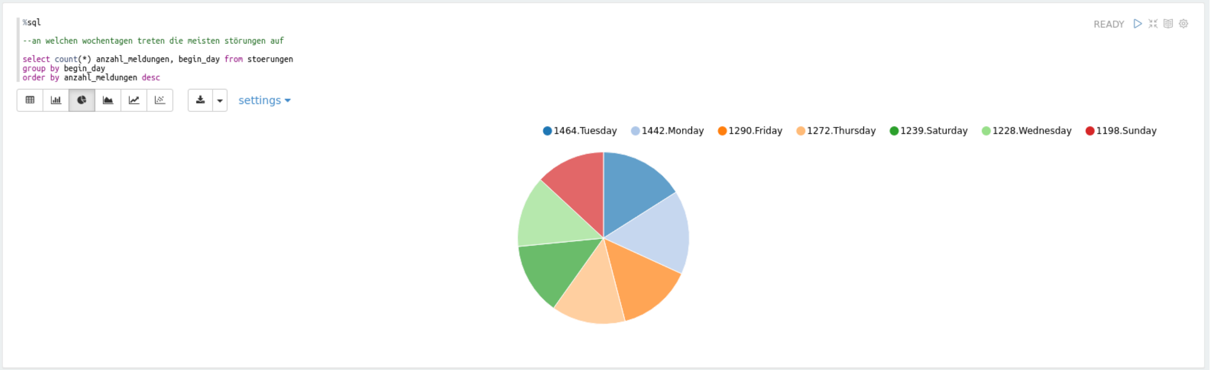
begin_date	begin_day	end_date	end_day	dauer	title	line	reason
2018-01-03 00:07:00.0	Wednesday	2018-01-03 00:07:00.0	Wednesday	0	Restriction	Othmarsingen	Technical fault
2018-01-03 00:07:00.0	Wednesday	2018-01-03 00:07:00.0	Wednesday	0	Disruption	Mittelhäusern - Schwarzenburg	Storm damage
2018-01-03 00:07:00.0	Wednesday	2018-01-03 00:07:00.0	Wednesday	0	End of announcement	Busswil - Büren an der Aare	The disruption
2018-01-03 00:07:00.0	Wednesday	2018-01-03 00:07:00.0	Wednesday	0	Disruption	Les Hauts-Geneveys - La Chaux-de-Fonds	damage to the
2018-01-04 00:07:00.0	Thursday	2018-01-04 00:07:00.0	Thursday	0	Disruption	Porrentruy - Boncourt	Incident with a
2018-01-04 00:07:00.0	Thursday	2018-01-04 00:07:00.0	Thursday	0	End of announcement	Renens VD	The disruption
2018-01-03 00:07:00.0	Wednesday	2018-01-07 00:07:00.0	Sunday	96000000000	Disruption	Mittelhäusern - Schwarzenburg	Storm damage
2018-01-27 00:06:00.0	Saturday	2018-01-27 00:06:00.0	Saturday	0	Disruption	Buchs SG - Salez-Sennwald	Between Buchs

5.



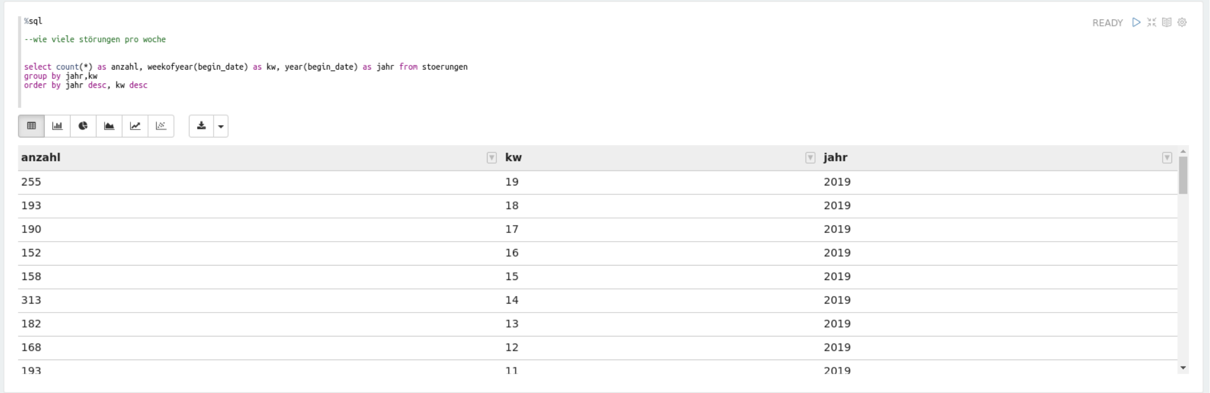
```
%sql

--wie lange dauert es bis stoerungen behoben sind

select count(*) as anzahl, dauer from stoerungen
where dauer > 100
group by dauer
order by dauer desc
```

6.



```
%sql

--an welchen wochentagen treten die meisten störungen auf

select count(*) anzahl_meldungen, begin_day from stoerungen
group by begin_day
order by anzahl_meldungen desc
```

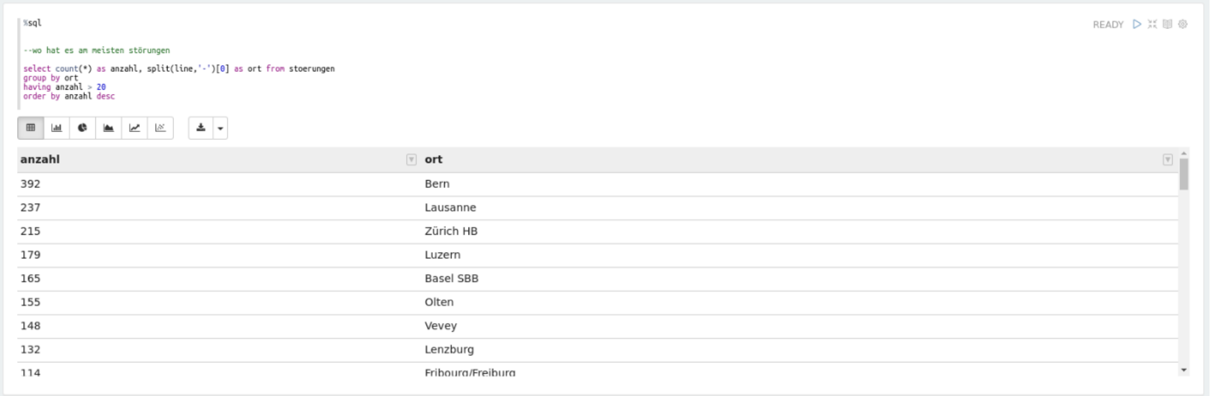
7.



```
%sql

--wie viele störungen pro woche


select count(*) as anzahl, weekofyear(begin_date) as kw, year(begin_date) as jahr from stoerungen
group by jahr,kw
order by jahr desc, kw desc
```

anzahl	kw	jahr
255	19	2019
193	18	2019
190	17	2019
152	16	2019
158	15	2019
313	14	2019
182	13	2019
168	12	2019
193	11	2019

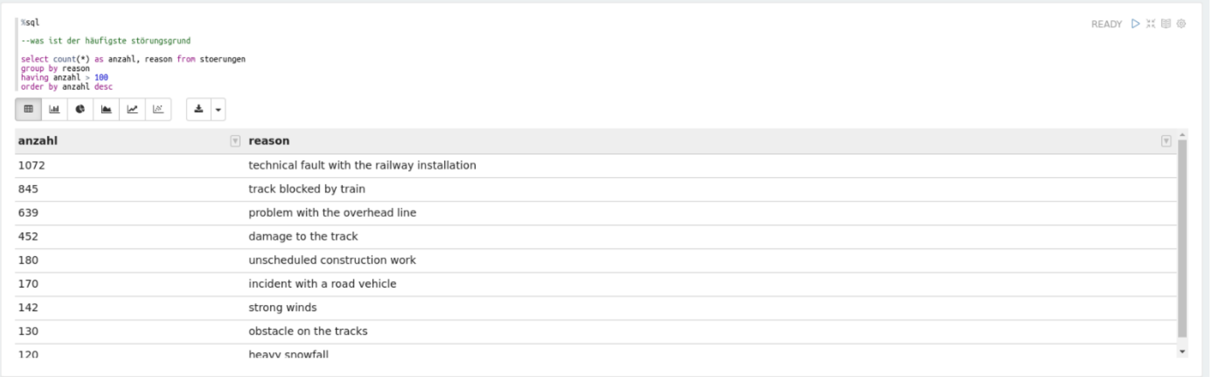
8.



```
%sql


--wo hat es am meisten störungen

select count(*) as anzahl, split(line,'-')[0] as ort from stoerungen
group by ort
having anzahl > 20
order by anzahl desc
```

anzahl	ort
392	Bern
237	Lausanne
215	Zürich HB
179	Luzern
165	Basel SBB
155	Olten
148	Vevey
132	Lenzburg
114	Fribourg/Freiburg

9.



```
%sql

--was ist der häufigste störungsgrund

select count(*) as anzahl, reason from stoerungen
group by reason
having anzahl > 100
order by anzahl desc
```

anzahl	reason
1072	technical fault with the railway installation
845	track blocked by train
639	problem with the overhead line
452	damage to the track
180	unscheduled construction work
170	incident with a road vehicle
142	strong winds
130	obstacle on the tracks
120	heavy snowfall