FINISHED

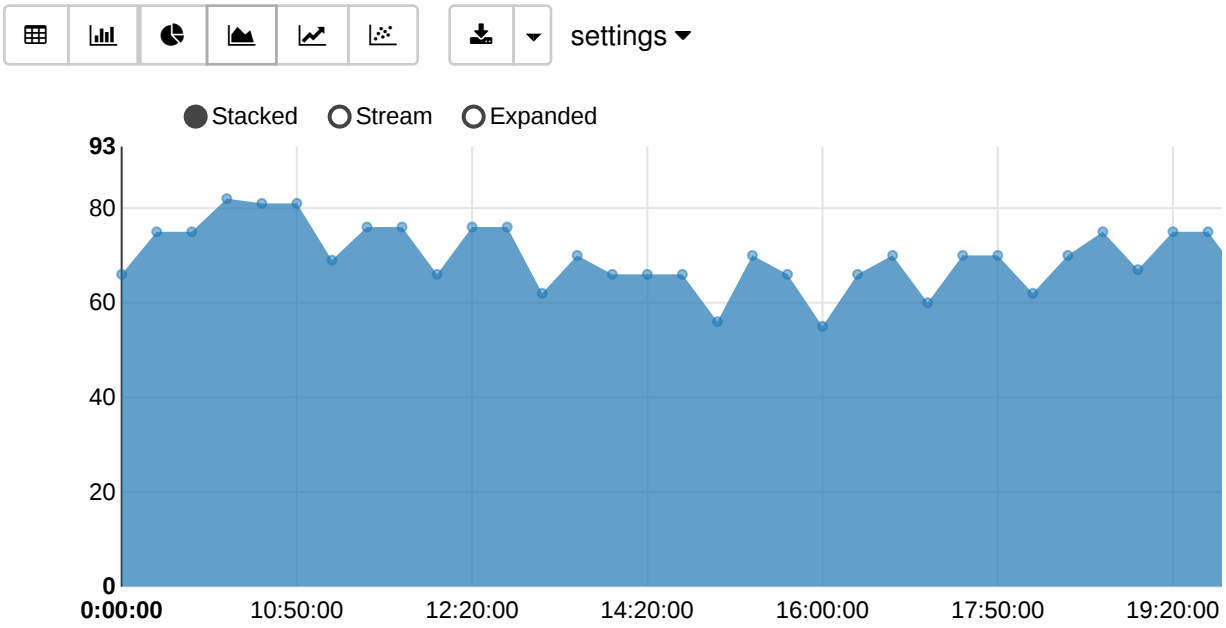
```
val humData = sqlContext.read.format("com.databricks.spark.csv").load("/home/scarface/desktop/sem-3
humData.toDF().registerTempTable("humidity")
```

```
humData: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [_c0: string, _c1: string ... 1 mor
e field]
warning: there was one deprecation warning; re-run with -deprecation for details
```

Took 15 sec. Last updated by anonymous at April 18 2017, 7:21:58 PM.

FINISHED

```
%sql
select _c0, _c1 as time,SUM(_c2) as humidity_per_day from humidity where _c0 = '13-02-2014' group b
```



Took 5 sec. Last updated by anonymous at April 18 2017, 7:24:11 PM. (outdated)

READY

FINISHED

```
import org.apache.spark.sql.functions._
import org.joda.time.format.DateTimeFormat
import org.apache.commons.io.IOUtils
```

Zeppelin Notebook

Zeppelin Visualiza...

anonymous

```
import org.apache.spark.sql.functions._
import org.joda.time.format.DateTimeFormat
import org.apache.commons.io.IOUtils
import java.net.URL
import java.nio.charset.Charset
```

Took 28 sec. Last updated by anonymous at April 18 2017, 6:12:27 PM.

default

FINISHED

```
%pyspark
from pandas import Series, DataFrame
import pandas as pd
import numpy as np
```

Took 28 sec. Last updated by anonymous at April 18 2017, 6:12:31 PM.

FINISHED

```
%pyspark
inputPath =  "/home/scarface/Desktop/sem-3/capstone/sravan/raw_weather_data_aarhus"

dewpoint = pd.read_csv(inputPath+"/dewptm.csv")

humidity = pd.read_csv(inputPath+"/hum.csv")

pressure = pd.read_csv(inputPath+"/pressurem.csv")


temp = pd.read_csv(inputPath+"/tempm.csv")

wind = pd.read_csv(inputPath+"/wdird.csv")
```

Took 0 sec. Last updated by anonymous at April 18 2017, 6:17:19 PM.

FINISHED

```
%pyspark
weather = pd.concat([dewpoint,humidity.ix[:,1],pressure.ix[:,1],temp.ix[:,1],winddirection.ix[:,1]]
weather.head()
```

```
   0  2014-02-13 00:00:00   -0  2014-02-13 00:00:00  2014-02-13 00:00:00  \
0  1  2014-02-13 00:20:00  0.0  2014-02-13 00:20:00  2014-02-13 00:20:00
1  2  2014-02-13 00:50:00  0.0  2014-02-13 00:50:00  2014-02-13 00:50:00
2  3  2014-02-13 01:00:00 -0.0  2014-02-13 01:00:00  2014-02-13 01:00:00
3  4  2014-02-13 01:20:00  0.0  2014-02-13 01:20:00  2014-02-13 01:20:00
4  5  2014-02-13 01:50:00  0.0  2014-02-13 01:50:00  2014-02-13 01:50:00
   2014-02-13 00:00:00  2014-02-13 00:00:00
0  2014-02-13 00:20:00  2014-02-13 00:20:00
1  2014-02-13 00:50:00  2014-02-13 00:50:00
2  2014-02-13 01:00:00  2014-02-13 01:00:00
3  2014-02-13 01:20:00  2014-02-13 01:20:00
4  2014-02-13 01:50:00  2014-02-13 01:50:00
```

Took 0 sec. Last updated by anonymous at April 18 2017, 6:16:38 PM.

Zeppelin Notebook

Zeppelin Visualiza...

anonymous

```
%pyspark
head(weather.ix[:,1:])
```

ERROR

default

```
%pyspark
weather.ix[:,1:].sum(axis=1)
```

FINISHED

```
%pyspark
weather.ix[:,1:].mean(axis=1,skipna=False)
```

FINISHED

```
%pyspark
weather.isnull().any()
```

```
0                     False
2014-02-13 00:00:00    False
-0                     True
2014-02-13 00:00:00    False
2014-02-13 00:00:00    False
2014-02-13 00:00:00    False
2014-02-13 00:00:00    False
dtype: bool
```

Took 0 sec. Last updated by anonymous at April 18 2017, 6:14:42 PM.

FINISHED

```
%pyspark
#In this step, we will try to update null values.
#filling null values could be complicated.As we seen in previous data exploration steps
#that 116 was the maximum null values and total datasize is 12563. Since, maximum percent of null v
#So, null values will be replaced by mean of the particular parameter.
def updatenullvalues(dataset):
    for col in dataset.ix[:,1:]:
        if dataset[col].isnull().any:
            mean = dataset[col].mean()
            dataset[col].fillna(mean,inplace=True)
    return dataset
```

Took 0 sec. Last updated by anonymous at April 18 2017, 6:14:48 PM.

ERROR

```
%pyspark

#Let's update null values in our dataset.
weather = updatenullvalues(weather_dataset)
#verify is there still any null value left in the dataset
weather.isnull().sum()
```

Zeppelin Notebook

Zeppelin Visualiza...

anonymous

```
%pyspark
import matplotlib.pyplot as plt

#Now that we in good state as our null values are vanished
#In this code step, we will check distribution of our data.

data = [weather.ix[:,1], weather.ix[:,2], weather.ix[:,3], weather.ix[:,4], weather.ix[:,5]]

parameter_names = ['Dewpoint', 'Humidity', 'Pressure', 'Temperature', 'WindDirection']

fig, axis = plt.subplots()
axis.set_title("Distribution of weather parameters")
axis.set_xlabel('Weather Parameters')
axis.set_ylabel('Values')
day_plot = plt.boxplot(data, sym='o', vert=1, whis=1.5)
plt.setp(day_plot['boxes'], color = 'black')
plt.setp(day_plot['whiskers'], color = 'black')
plt.setp(day_plot['fliers'], color = 'black', marker = 'o')
axis.set_xticklabels(parameter_names)
plt.show()
```

default

READY

```
%pyspark
# column-wise and Multiple Function Application
grouped_pressure = weather.groupby(['Pressure'])
```

READY

```
%pyspark
grouped_pressure.apply(lambda weather: weather['Humidity'].corr(weather['Temperature']))
```

READY

```
%pyspark
import statsmodels.api as sm
def regression(data, yvar, xvars):
  Y = data[yvar]
  X = data[xvars]
  X['intercept'] = 1.
  result = sm.OLS(Y,X).fit()
  return result.params
```

READY

Zeppelin Notebook

Zeppelin Visualiza...

anonymous

default

```
#regression(weather_dataset,'Humidity','Temperature')
grouped_pressure.apply(regression,'Humidity',['Temperature','Winddirection'])
# [['Dewpoint', 'Humidity', 'Pressure', 'Temperature', 'WindDirection']
```

FINISHED

```
%spark.r
dewpoint = read.csv("/home/scarface/Desktop/sem-3/capstone/sravan/raw_weather_data_aarhus/dewptm.csv
humidity = read.csv("/home/scarface/Desktop/sem-3/capstone/sravan/raw_weather_data_aarhus/hum.csv", h
pressure = read.csv("/home/scarface/Desktop/sem-3/capstone/sravan/raw_weather_data_aarhus/pressurem.
temp = read.csv("/home/scarface/Desktop/sem-3/capstone/sravan/raw_weather_data_aarhus/tempm.csv", hea
wind = read.csv("/home/scarface/Desktop/sem-3/capstone/sravan/raw_weather_data_aarhus/wdird.csv", hea
```

Took 0 sec. Last updated by anonymous at April 18 2017, 6:35:55 PM.

FINISHED

```
%spark.r
dataSet <- cbind(dewpoint,humidity$V3,pressure$V3,temp$V3,wind$V3)
```

Took 1 sec. Last updated by anonymous at April 18 2017, 7:51:18 PM.

FINISHED

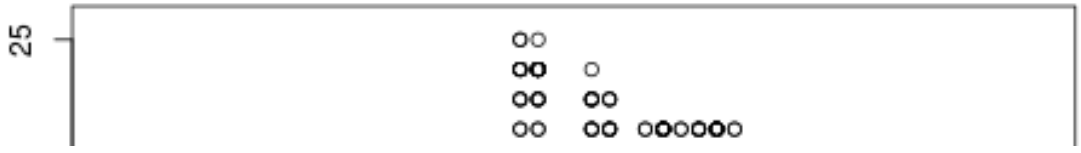
```
%spark.r
colnames(dataSet)<- c("id", "date", "dew", "hum", "pre", "tem", "win")
head(dataSet)
dataSet<-na.omit(dataSet)
```

```
id                date dew hum pre tem win
1  0 2014-02-13 00:00:00   0  66 995   4 160
2  1 2014-02-13 00:20:00   0  75 994   4 160
3  2 2014-02-13 00:50:00   0  75 993   4 160
4  3 2014-02-13 01:00:00   0  65 994   4 160
5  4 2014-02-13 01:20:00   0  75 993   4 170
6  5 2014-02-13 01:50:00   0  75 993   4 170
```

Took 0 sec. Last updated by anonymous at April 18 2017, 7:55:17 PM.

FINISHED

```
%spark.r
plot(dataSet[5:6])
```

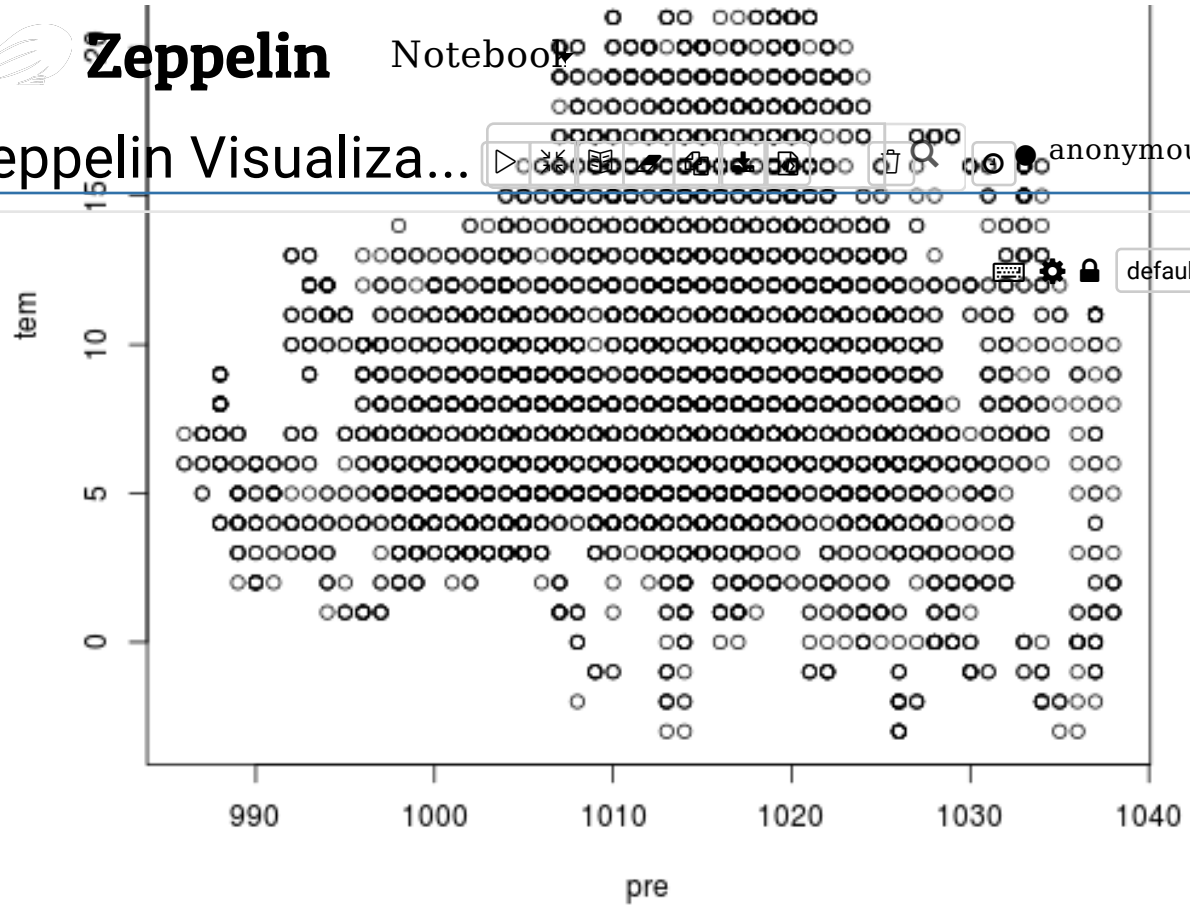


Zeppelin Notebook

Zeppelin Visualiza...

anonymous

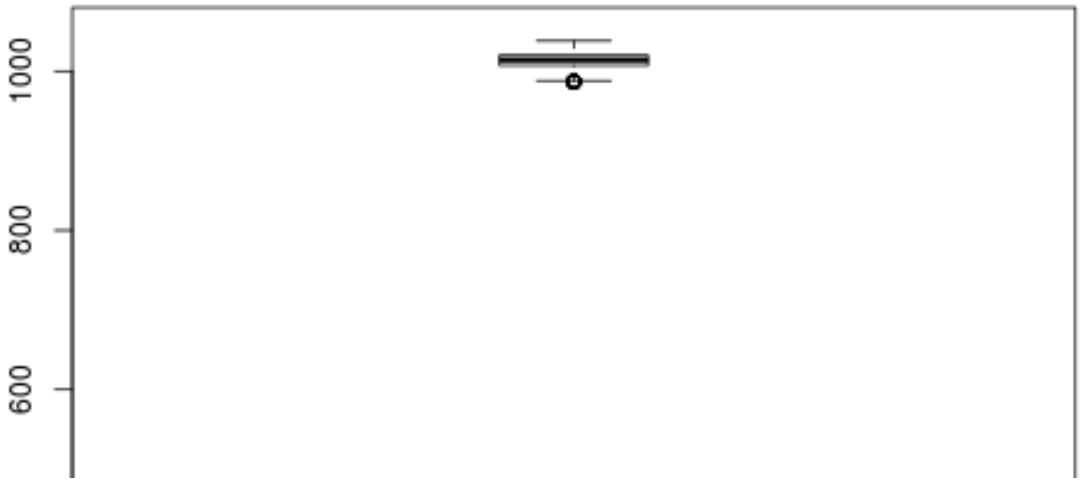
default



Took 0 sec. Last updated by anonymous at April 18 2017, 7:51:25 PM.

FINISHED

```
%spark.r
boxplot(dataSet[3:7])
```

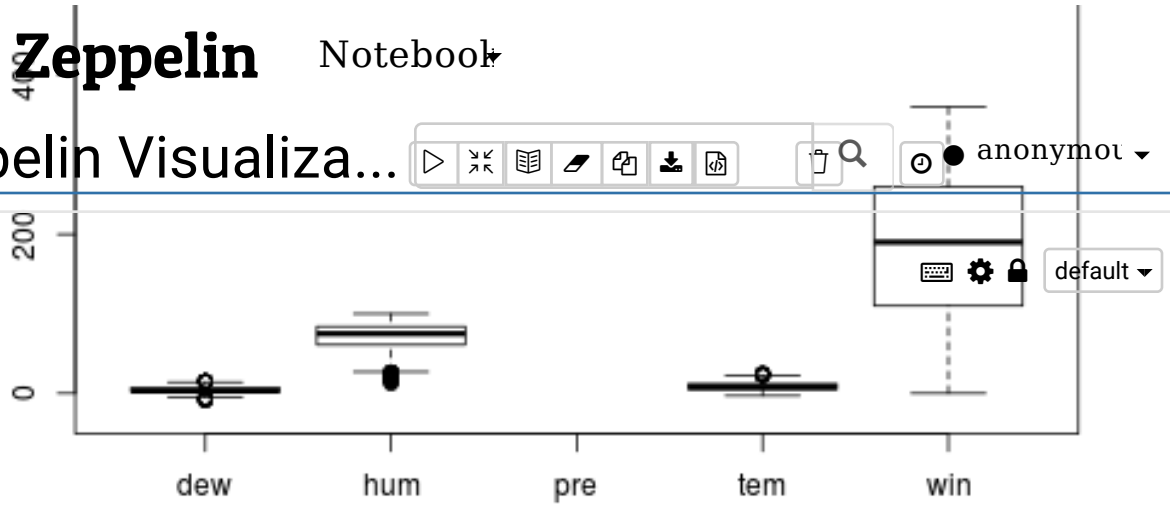


Zeppelin Notebook

Zeppelin Visualiza...

anonymou

default



Took 0 sec. Last updated by anonymous at April 18 2017, 7:34:45 PM.

FINISHED

```
%spark.r
ds=max(na.omit(dataSet[,3]))
hs=max(na.omit(dataSet[,4]))
ps=max(na.omit(dataSet[,5]))
ts=max(na.omit(dataSet[,6]))
ws=max(na.omit(dataSet[,7]))

dataSet<-na.omit(dataSet)
```

Took 0 sec. Last updated by anonymous at April 18 2017, 7:36:19 PM.

FINISHED

```
%spark.r
for (i in 1:length(dataSet$id)){
    dataSet[i,3]<-dataSet[i,3]/ds
    dataSet[i,4]<-dataSet[i,4]/hs
    dataSet[i,5]<-dataSet[i,5]/ps
    dataSet[i,6]<-dataSet[i,6]/ts
    dataSet[i,7]<-dataSet[i,7]/ws
}
#dataSet
```

Took 4 sec. Last updated by anonymous at April 18 2017, 7:39:38 PM.

FINISHED

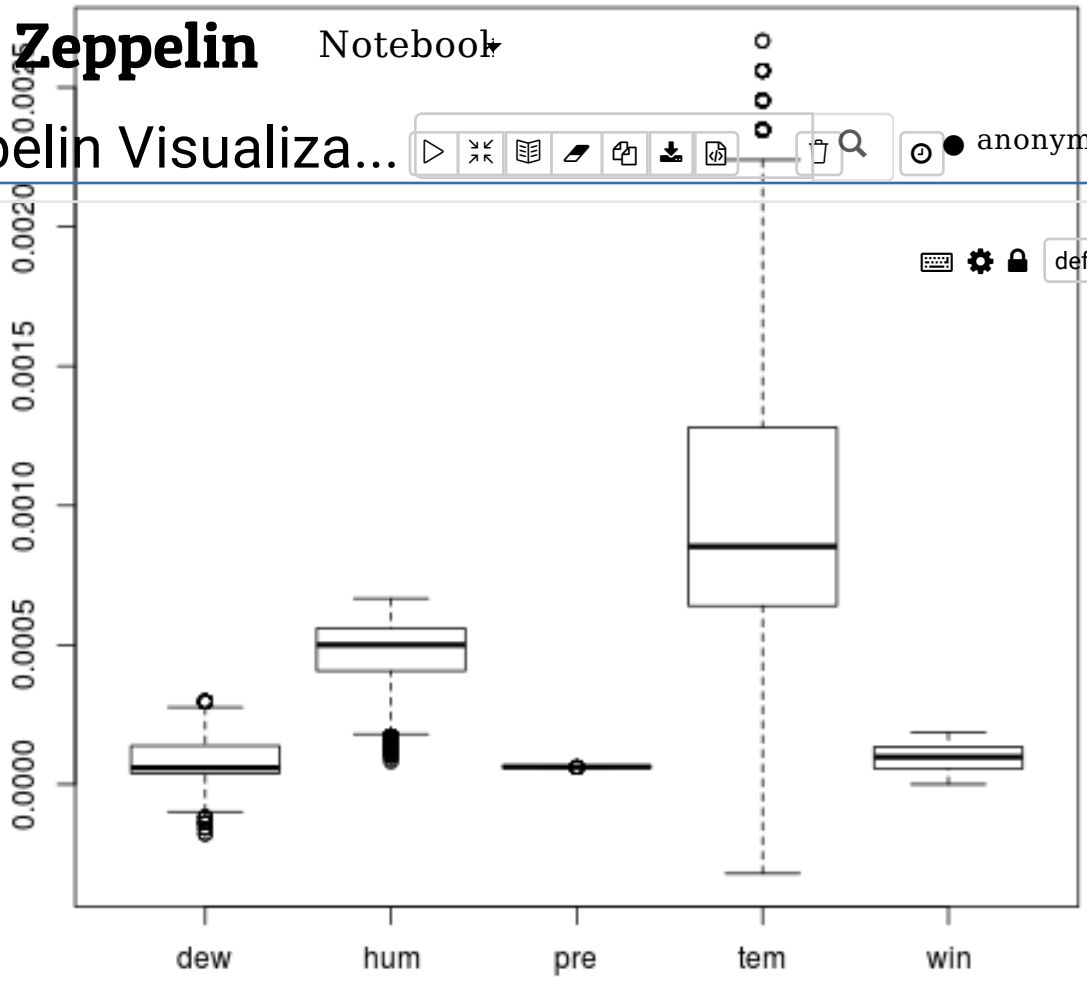
```
%spark.r
boxplot(dataSet[3:7])
```

Zeppelin Notebook

Zeppelin Visualiza...

anonymous

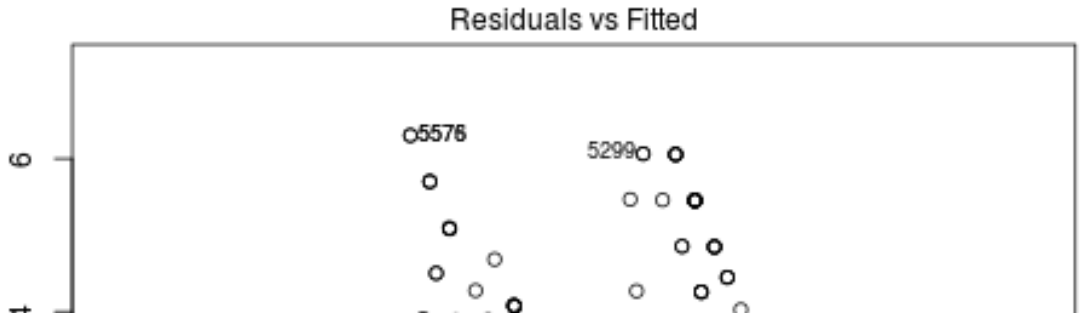
default



Took 0 sec. Last updated by anonymous at April 18 2017, 7:40:07 PM.

FINISHED

```
%spark.r
lmF <- lm(dataSet$tem ~ dataSet$dew+dataSet$hum+dataSet$pre)
plot(lmF)
```

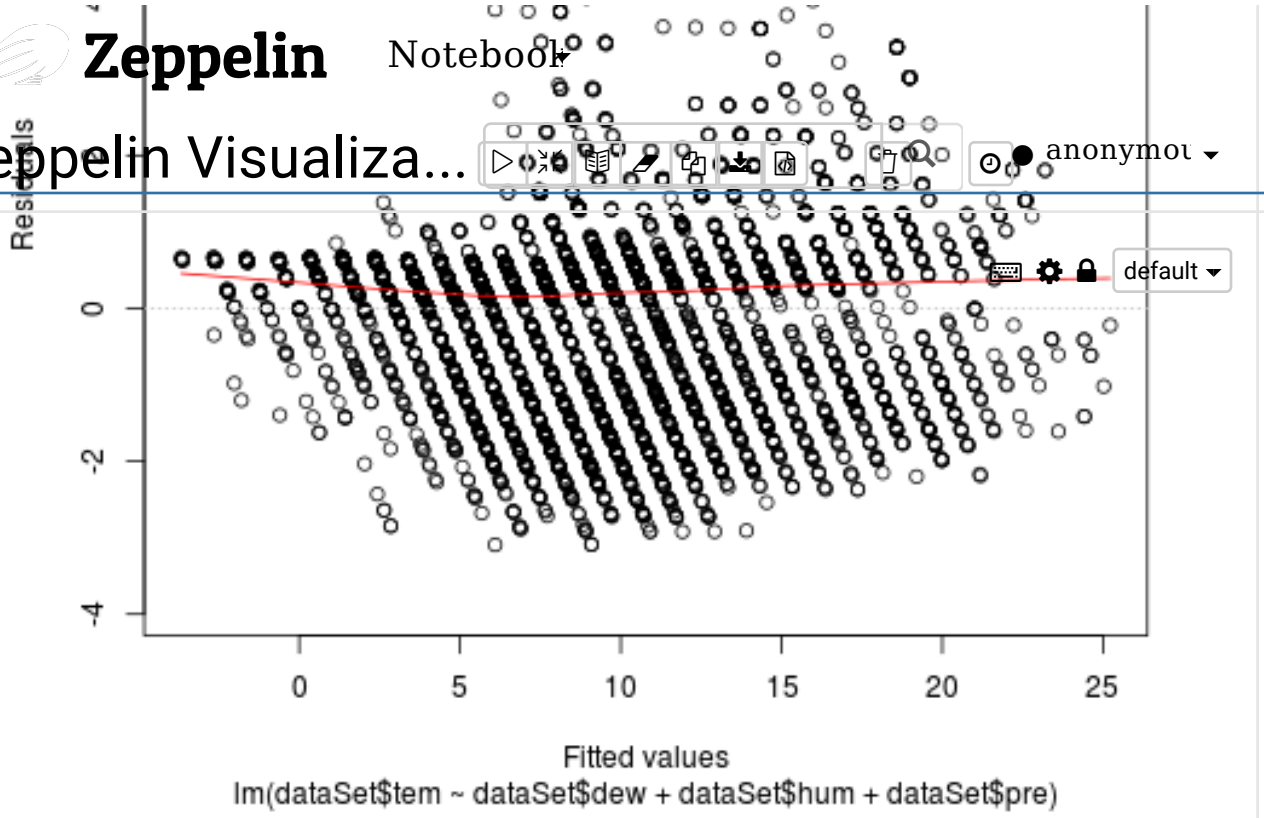


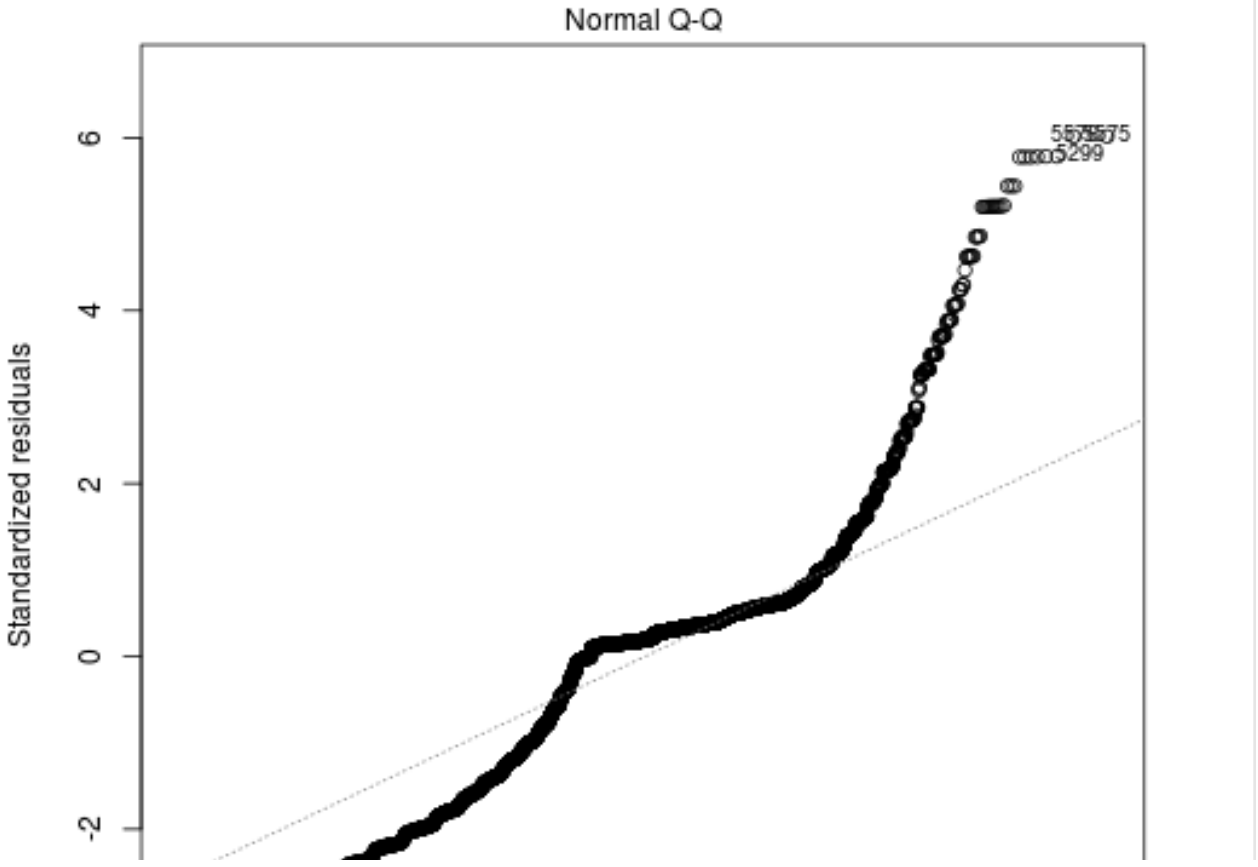
Zeppelin Notebook

Zeppelin Visualiza...

anonymou

default



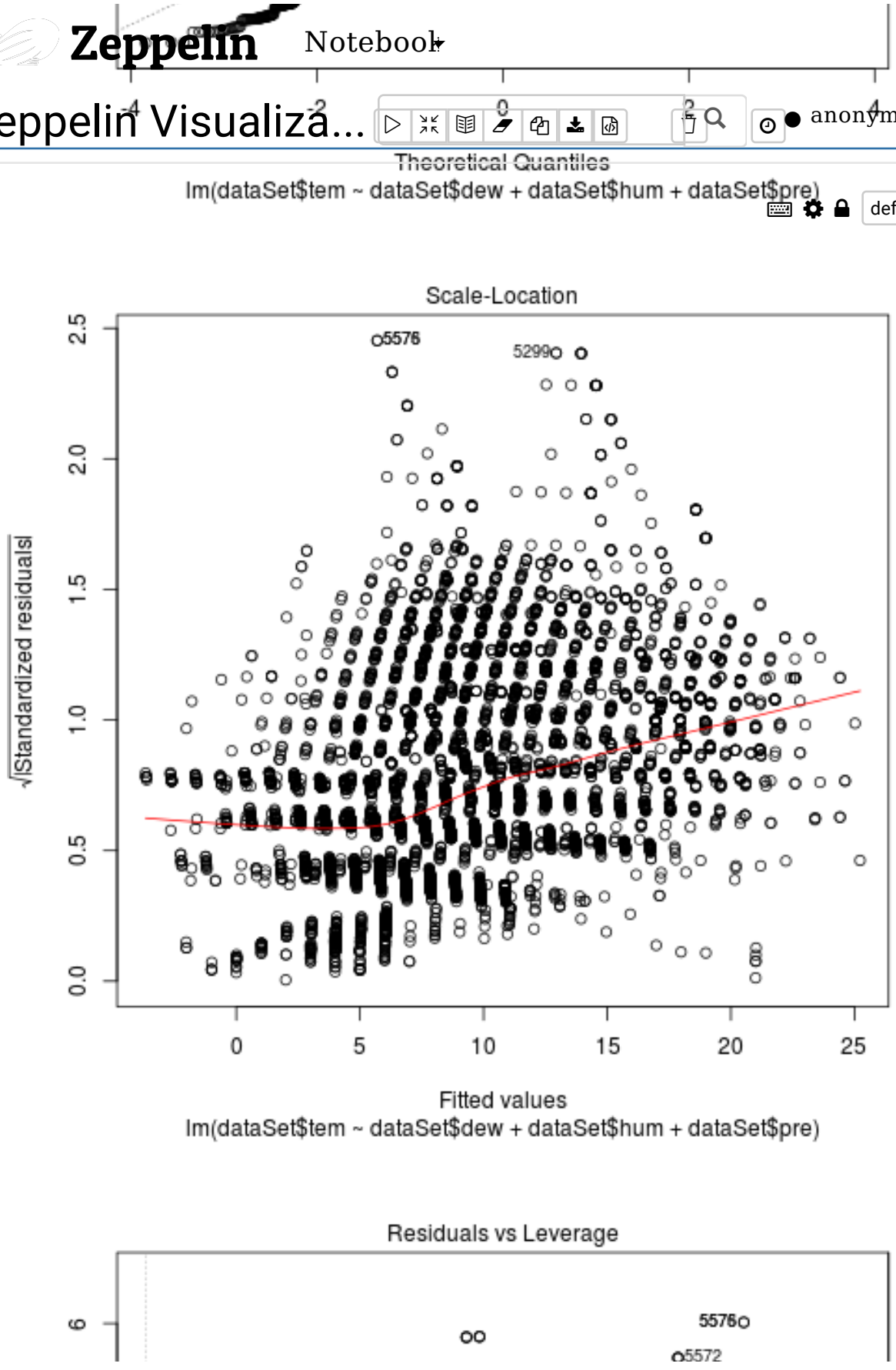


Zeppelin Notebook

Zeppelin Visualiza...

anonymou

default

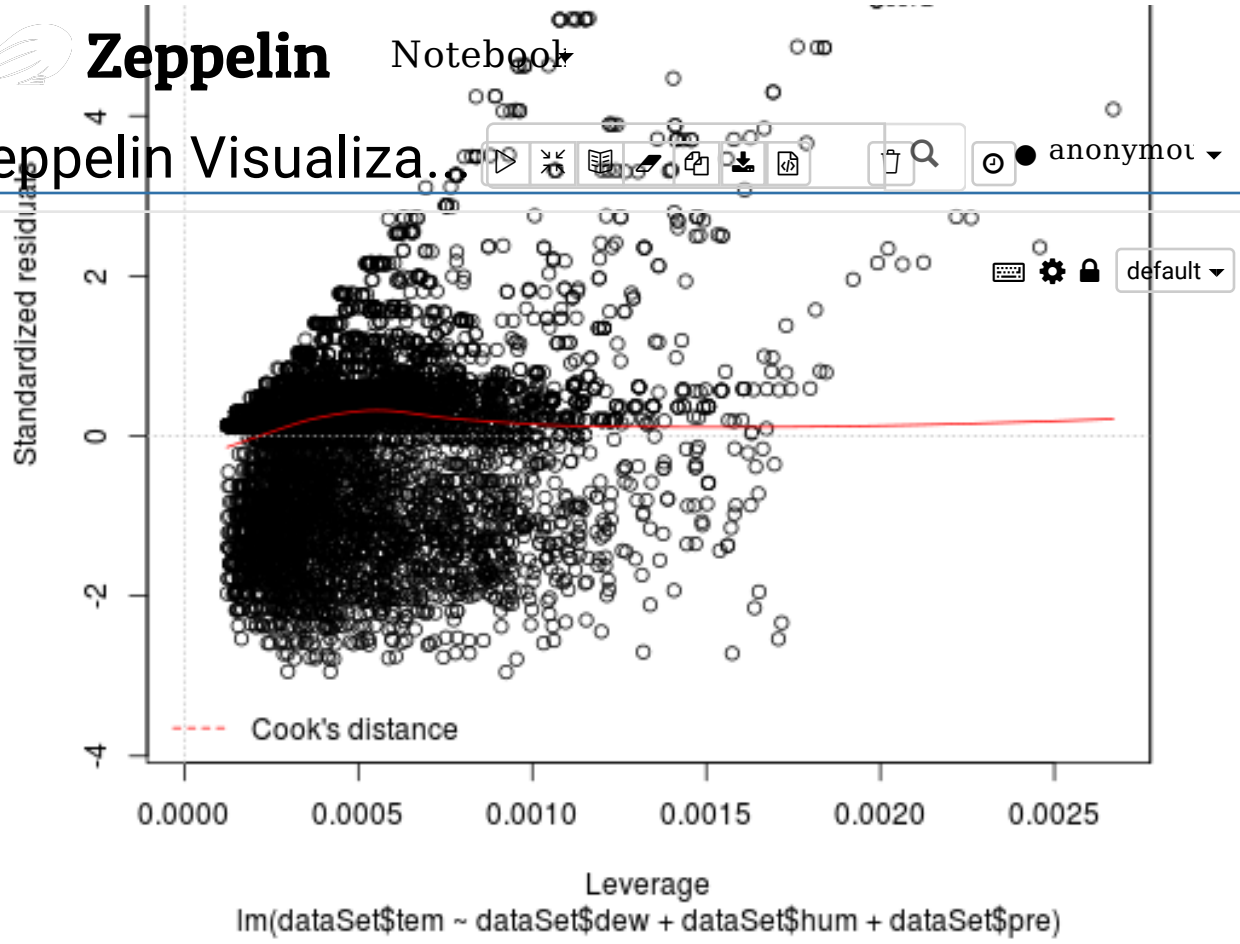


Zeppelin Notebook

Zeppelin Visualiza...

anonymous

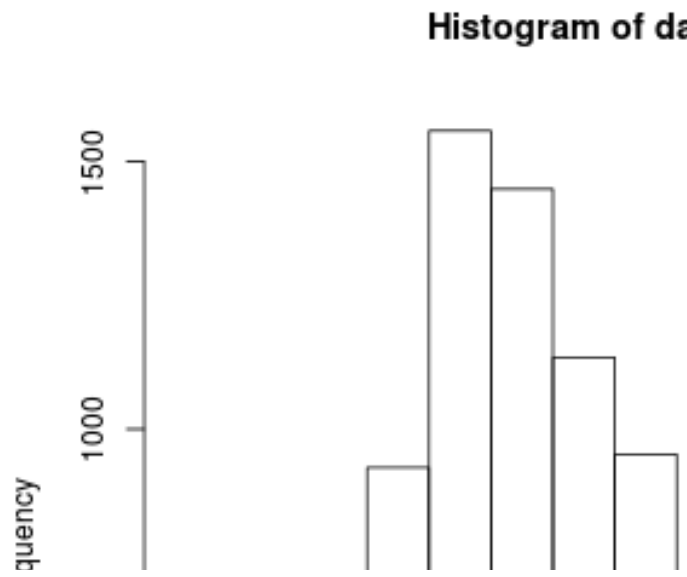
default



Took 1 sec. Last updated by anonymous at April 18 2017, 7:17:10 PM.

FINISHED

```
%spark.r
hist(dataSet$tem)
```

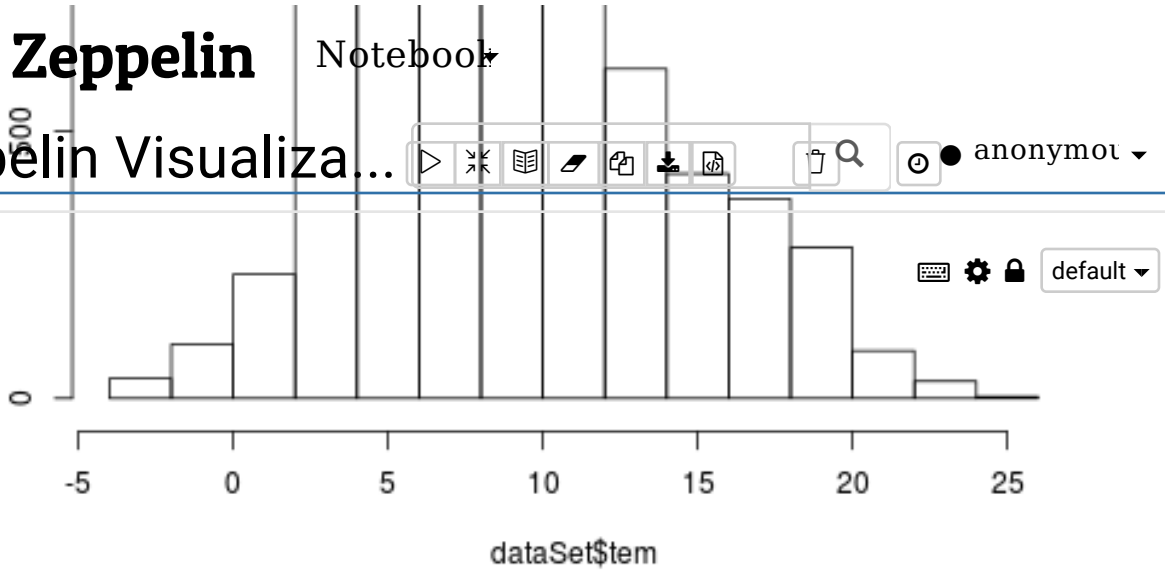


Zeppelin Notebook

Zeppelin Visualiza...

anonymou

default



Took 0 sec. Last updated by anonymous at April 18 2017, 7:28:06 PM.

FINISHED

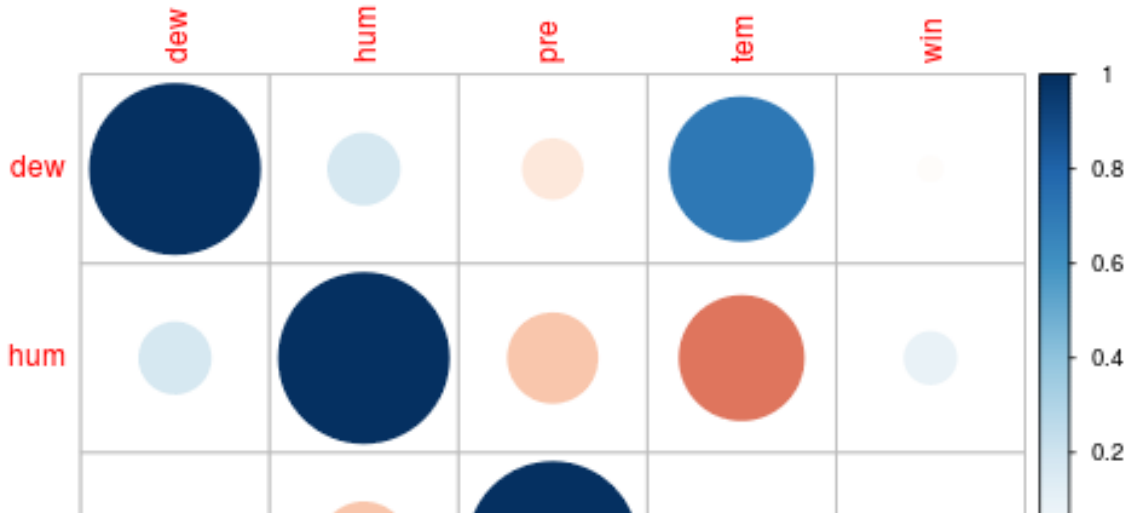
```
%spark.r
install.packages("corrplot", repos = "http://cran.us.r-project.org")
```

```
The downloaded source packages are in
    '/tmp/RtmpWXXF7Y/downloaded_packages'
```

Took 8 sec. Last updated by anonymous at April 18 2017, 7:44:34 PM.

FINISHED

```
%spark.r
library(corrplot)
M <- cor(dataSet[3:7])
corrplot(M, method="circle")
```

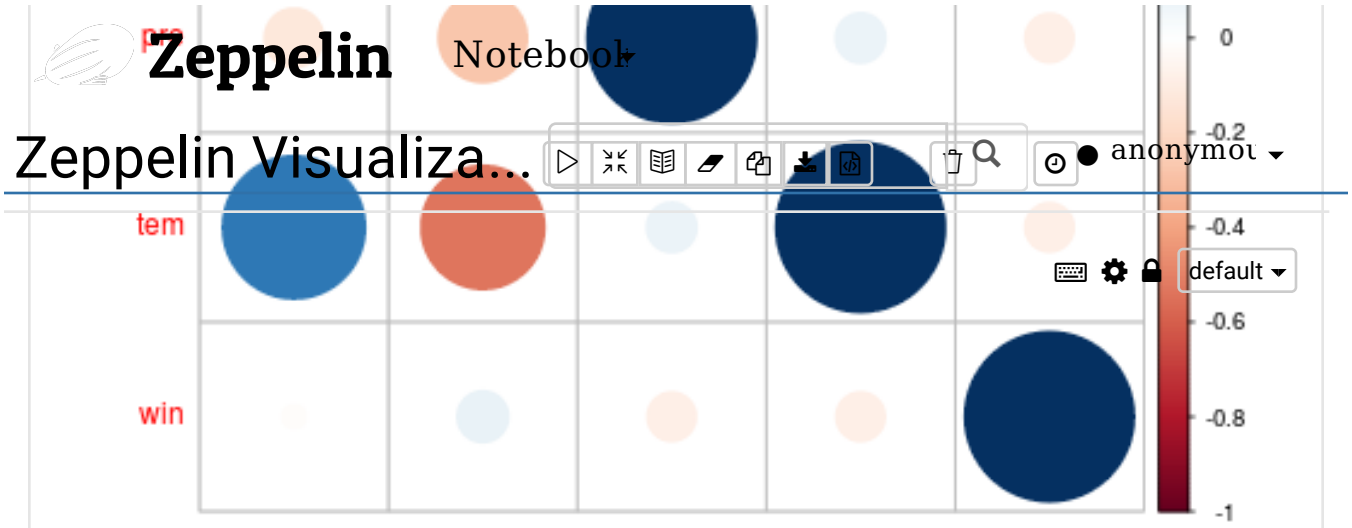


Zeppelin Notebook

Zeppelin Visualiza...

anonymous

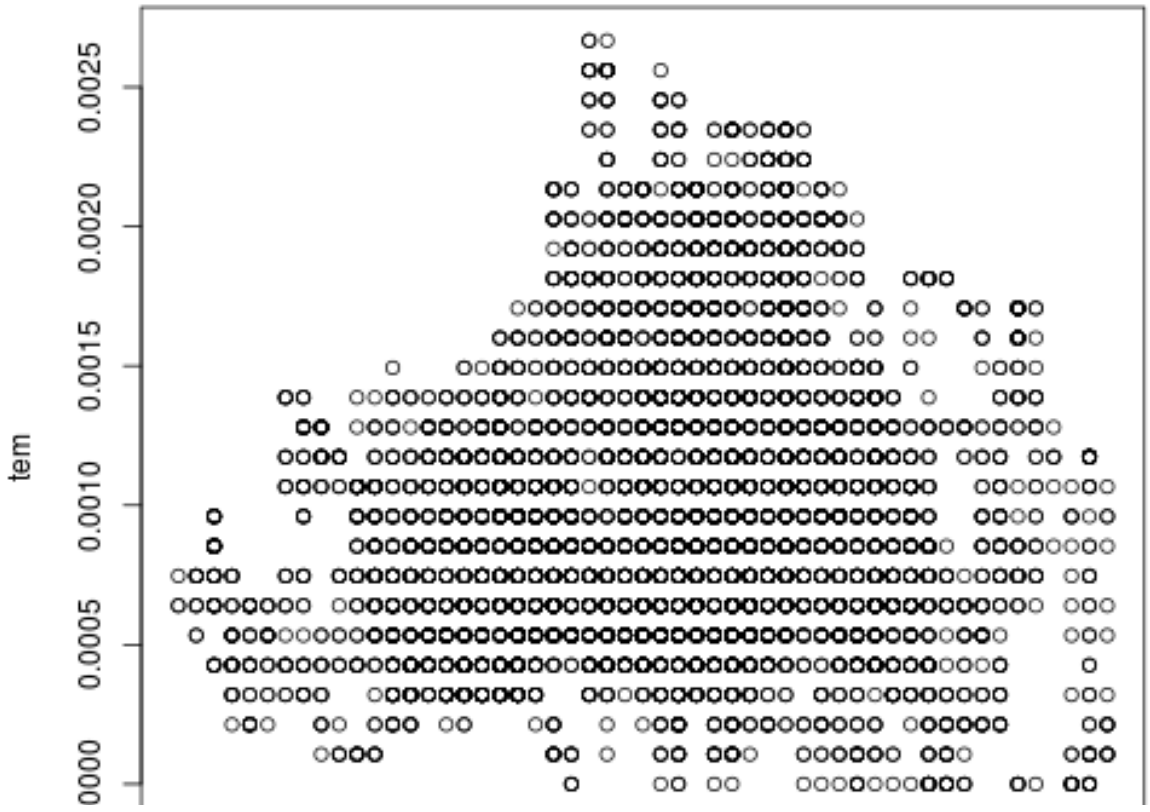
default

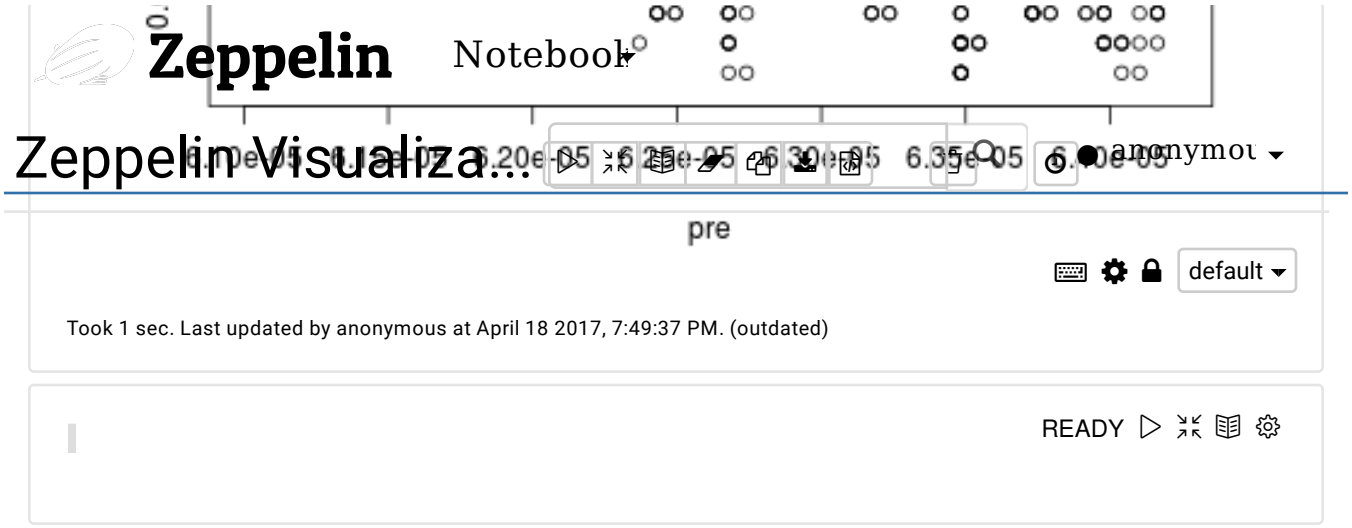


Took 0 sec. Last updated by anonymous at April 18 2017, 7:45:25 PM.

```
%spark.r
```

FINISHED




Zeppelin
Notebook
Zeppelin Visualiza...
anonymous
pre
default
Took 1 sec. Last updated by anonymous at April 18 2017, 7:49:37 PM. (outdated)
READY