



Zeppelin

pro1



default ▾

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

// Load data - adjust the path to the location of your data
val inputPath = "/Users/neha/Documents/Capstone/pollution*"
val data1 = sqlContext.read
    .format("com.databricks.spark.csv")
    .option("header", "true") // Use first line of all files as header
    .option("delimiter", ",")
    .option("inferSchema", "true") // Automatically infer data types
    .load(inputPath)
data1.toDF().registerTempTable("data2")
```

FINISHED ▷ ⌵ 📖 ⚙️

```
%pyspark
DF = sqlContext.sql("select avg(ozone) as avg_ozone, avg(particulate_matter) as particulate,
, avg(nitrogen_dioxide) as nitrogen from data2 group by longitude,latitude")
data = DF.toPandas()
data
```

FINISHED ▷ ⌵ 📖 ⚙️

```
%pyspark
import matplotlib
import matplotlib.pyplot as plt
import seaborn as sns

import StringIO
def show(p):
    img = StringIO.StringIO()
    p.savefig(img, format='svg')
    img.seek(0)
    print "%html " + img.buf
```

FINISHED ▷ ⌵ 📖 ⚙️

Took 0 sec. Last updated by anonymous at March 27 2017, 2:54:31 AM.

```
%pyspark
df1 = sqlContext.sql("SELECT latitude, longitude, count(*) as records, avg(ozone) as avg_ozone")
data3 = df1.toPandas()

value = "avg_ozone"
x = "latitude"
```

FINISHED ▷ ⌵ 📖 ⚙️

pro1

```
grouping = ["longitude"]
```

Zeppelin

```
heatmap_data= data3.pivot_table(values=value, index=x, columns=grouping)
```

```
heatmap_data = heatmap_data[0:100]
```

```
row1 a4_dims = (len(heatmap_data.columns),50)
```

```
fig, ax = plt.subplots(figsize=a4_dims)
```

```
ax.set_title("Avg ozone level")
```

```
sns.heatmap(heatmap_data, ax=ax, annot=True, fmt=".02f")
```

```
show(n1+)
```





```
%sql
```

```
select ozone, particullate_matter from data2
```

FINISHED ▶ ⌵ ⌶ ⚙

```
%pyspark
```

```
data.hist('avg_ozone')
```

ERROR    

```
%pyspark
```

```
from pandas.tools.plotting import scatter_matrix
```

```
stuff = scatter_matrix(data, alpha=0.5, figsize=(10,10), diagonal='kde')
```

```
show(plt)
```

FINISHED ▶ ↻ 📖 ⚙️

```
%pyspark
```

```
import matplotlib.pyplot as plt1
```

```
plt.clf()
```

```
num_bins = 30
```

```
#data.hist('avg_ozone',num_bins)
```

```
n, bins, patches = plt.hist('avg_ozone', num_bins, normed=1, facecolor='green', alpha=0.5, data=)
```

```
show(plt)
```

FINISHED ▶ 🔍 📖 ⚙️

%r

```
library(readr)
```

```
library(dplyr)
```

```
setwd("/Users/neha/Documents/Capstone/pollution")
```

```
temp = list.files(pattern="*.csv")
```

```
tbl = lapply(temp, read_csv) %>% bind_rows()
```

```
head(tbl)
```

FINISHED ▶ 🔍 📖 ⚙️

|pro1

[illegible]

FINISHED ▶ 🔍 📖 ⚙️

[illegible]

```
%n=9<-kmeans(x1,9)
```

Zeppelin

pro 1

```
res10<-kmeans(x1,10)
res10$centers
```

FINISHED default

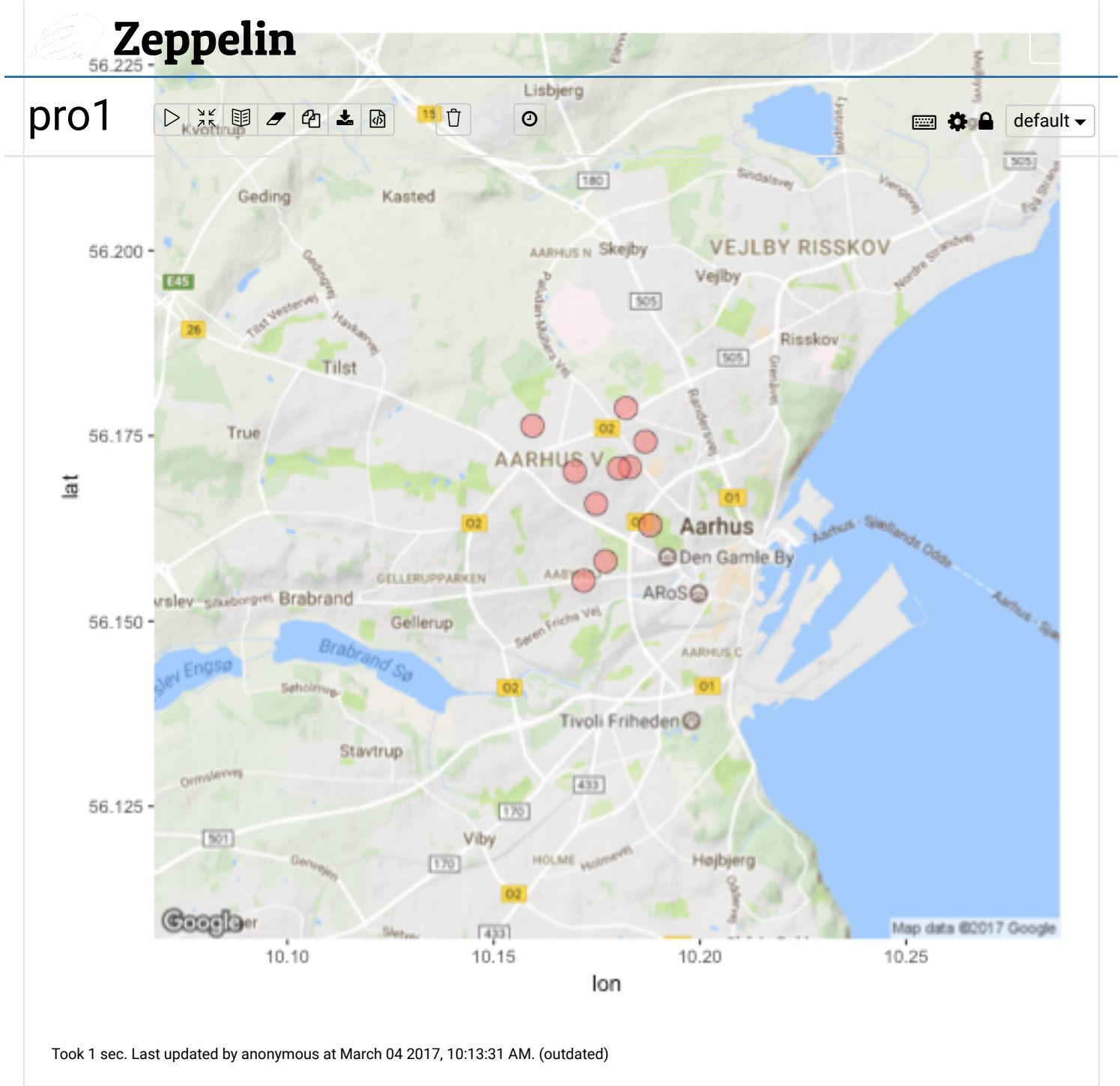
```
%r
plot(x1[c("avg_ozone","particulate")], col = res10$cluster )
```

FINISHED

```
%r
df = res10$centers
df<-as.data.frame(df)
df
```

FINISHED    

```
%r
lon <- as.vector(df['longitude'])
lat <- as.vector(df['latitude'])
df1 <- as.data.frame(cbind(lon,lat))
mapgilbert <- get_map(location = c(lon = mean(df1$lon), lat = mean(df1$lat)), zoom = 12,
maptype = "terrain", scale = 1)
# plotting the map with some points on it
ggmap(mapgilbert) + geom_point(data = df1, aes(x = lon, y = lat, fill = "red", alpha = 0
.1), size = 5, shape = 21) + guides(fill=FALSE, alpha=FALSE, size=FALSE)
```



pro1

```
%r
lon <- as.vector(df['longitude'])
lat <- as.vector(df['latitude'])
df1 <- as.data.frame(cbind(lon,lat))
ggplot() + geom_point(data=df, aes(x=lon, y=lat), color="red")
```

pro1

```
%r
lon <- as.vector(df['longitude'])
```



```

lon <- as.vector(df['latitude'])
lat <- as.vector(df['longitude'])
mapgilbert <- get_map(location = c(lon = mean(df1$lon), lat = mean(df1$lat)), zoom = 12, maptype = "hybrid", scale = 1)
# plotting the map with some points on it
ggmap(mapgilbert) + geom_point(data = df1, aes(x = 10.18032, y = 56.17061, fill = "red", alpha = 0.1), size = 5, shape = 21) + guides(fill=FALSE, alpha=FALSE, size=FALSE)

```



Took 2 sec. Last updated by anonymous at March 04 2017, 10:27:34 AM.

FINISHED ► 🔍 📄 ⚙️