

Hands on Brewer temperature dependence: EUBREWNET Using R/Python

Workshop objective:

- work with data from EUBREWNET DB
- work with the python module brewer

Programming language:

- Python
- R

Python

- General-purpose programming language.
- Download Python: <https://www.python.org/>
- Add c:\python27\;c:\python27\scripts; to path system variable
- Download pip: <https://pip.pypa.io/en/latest/installing/>
- Python module installation: (from command line)
 - pip install numpy
 - pip install requests
 - pip install scikits.statsmodels
- Download brewer modules:
 - brewer
 - brewerjson

R

- Programming language focuses in statistics and data analysis
- Download R: <https://www.cran.r-project.org/>
- Download IDE: <https://www.rstudio.com/>
- Sites about R:
 - <http://www.statmethods.net/>
- Manage R Modules:
 - Packages needed: httr, jsonlite
 - Repositories: Cambridge (UK)
 - Inside R (operating system independent): `install.packages("httr")`
'Jsonlite', 'mime', 'curl', 'openssl', 'R6'
 - Linux repositories (r-cran-httr, r-cran-jsonlite)
- R specificities:
 - Help: `?write.table`
 - Data types:

- + Variables: `a<-23`
- + Vectors: `a <- c(1,2)`; indexes start by 1.
- + Dataframes: `df <- data.frame(site=c('izana','huelva'),o3=c(330,331)); df$o3[1]; df[1,1]`
- + Write data to files: `write.table(df,'file.dat',quote = T, sep = "\t", row.names=F)`
- + Read data from files: `d<-read.table('file.dat',sep='\t')`

Download data from EBREWNET

First, we need to log in the website: (azores/azowork. If you dont have a user, ask for it to Bantor)

<http://rbcce.aemet.es/eubrewnet>

And now the data will be accesible from:

<http://rbcce.aemet.es/eubrewnet/data/get/SL?brewerid=157&date=2015-01-01&enddate=2015-01-02&format=csv>

<http://rbcce.aemet.es/dokuwiki/doku.php?id=codes:dbaccess>

R example

```
#!/usr/bin/Rscript

library(httr)

r<-GET("http://rbcce.aemet.es/eubrewnet/data/get/SL?brewerid=185&date=2015-07-01&enddate=2015-07-31&format=csv",authenticate("azores", "azowork"))
d<-read.table(text=content(r, "text"),header=T,sep=',')

d$gmt<-as.POSIXct(d$gmt,format="%Y%m%dT%H%M%SZ",tz="UTC")

write.table(d,'sl185.dat',quote = T, sep = "\t", row.names=F)

head(d,2)

plot(d$gmt,d$raw_counts_w2)

pdf("rawcs2.pdf")
plot(d$gmt,d$raw_counts_w2,xlab="Date",ylab="Raw Counts Slit 2")
void<-dev.off()

r<-GET("http://rbcce.aemet.es/eubrewnet/data/get/ConfigbyDate?brewerid=185&date=2015-03-01&enddate=2015-06-01&format=csv",authenticate("azores", "azowork"))
```

```

c<-read.table(text=content(r, "text"),header=T,sep=',')

deadtime<-c$deadtime[1]

#  $2(F-F_0)/(cy*it)$ 
d[, "counts_w0"]<-2*(d[, "raw_counts_w0"]-d$dark_count)/(d$cycles*0.1147)

#  $F_0<-F*exp(F_0*deadtime)$ 
F0<-d[,eval(b)]
for (i in seq(9)) {
  F0<-d[,eval(b)]*exp(F0*deadtime)
}

plot(d$temp,d$counts_w0)
fit1<-lm(d$counts_w0 ~ d$temp)
abline(fit1)
summary(fit1)

tc1=-summary(fit1)$coefficients[2]/summary(fit1)$coefficients[1]

plot(d$temp,10000*log10(d$counts_w0))
fit2<-lm(10000*log10(d$counts_w0) ~ d$temp)
abline(fit2)
summary(fit2)

tc2=-summary(fit2)$coefficients[2]

db<-c("counts_w0","counts_w1","counts_w2","counts_w3","counts_w4","counts_w5")
for (b in db){
  #  $2(F-F_0)/(cy*it)$ 
  d[,eval(b)]<-2*(d[,paste("raw_",b,sep='')]-d$dark_count)/(d$cycles*0.1147)
  #d[,paste(b,"_0",sep='')]<-d[,eval(b)]
  #  $F_0<-F*exp(F_0*deadtime)$ 
  F0<-d[,eval(b)]
  for (i in seq(9)) {
    F0<-d[,eval(b)]*exp(F0*deadtime)
  }
  d[,eval(b)]<-10000*log10(F0)
}

tc1<-c()
for (b in db){
  plot(d$temp,d[,eval(b)])
  fit<-lm(d[,eval(b)] ~ d$temp)
  abline(fit)
  tc1<-c(tc1,-summary(fit)$coefficients[2])
}
tcr1<-tc1-tc1[2]

tcr2<-c()
for (b in db){
  plot(d$temp,d[,eval(b)]-d$counts_w1)

```

```
fit<-lm(d[,eval(b)]~d$counts_w1 ~ d$temp)
abline(fit)
tcr2<-c(tcr2,-summary(fit)$coefficients[2])
}

tcr1
tcr2
```

Python example

```
import datetime
import brewer
import brewerjson

brewerid=185
user='azores'
password='azowork'
date1=datetime.date(2015,7,1)
date2=datetime.date(2015,7,31)
sl=brewerjson.getDataList(brewerid,user,password,"SL",date1,date2)
config=brewerjson.getDataList(brewerid,user,password,"ConfigbyDate",date1,date2)

ta=brewer.tempanalysis(sl,config,[],0)
ta.keys()
tc=ta['new']['tcoef']
tcr=ta['new']['tcoef']-ta['new']['tcoef'][1]

import matplotlib.pyplot as plt

fig, ax = plt.subplots()
wl=[303.2,306.3,310.1,313.5,316.8,320.1]
ax.set_ylim(-1,2)
ax.plot(wl,list(tcr),marker='o')
plt.show()
```
