Stt864 Lab2

Nan Cao

February 23, 2016

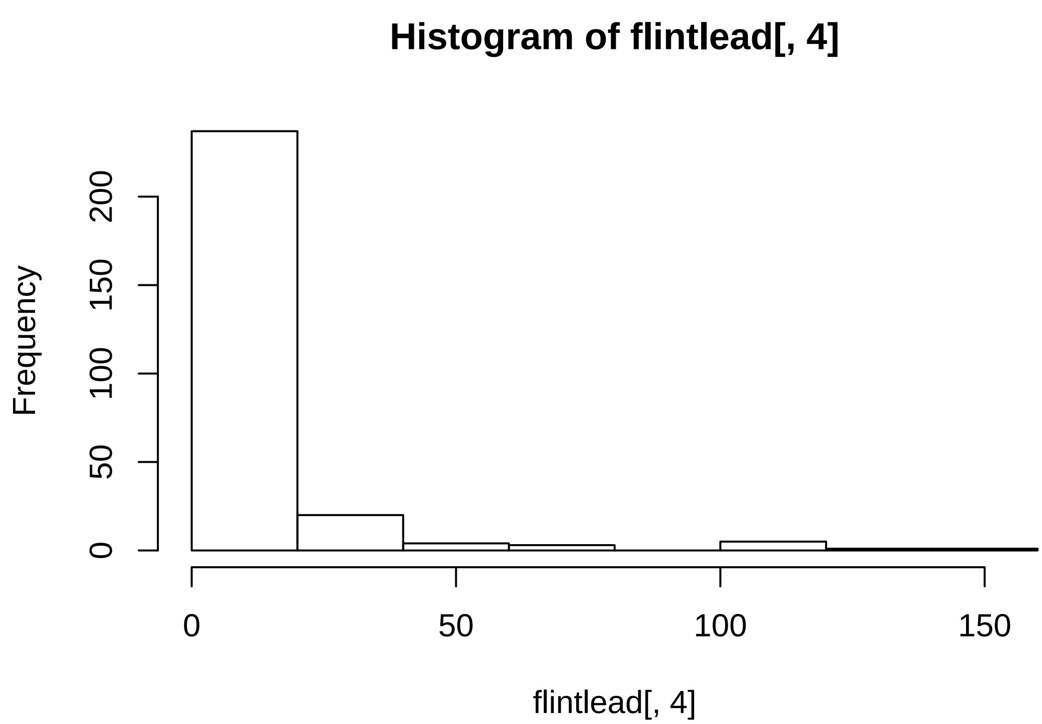
Q1(a)

```
set.seed(52871775)
setwd("C://Users//nan66//Google Drive//stt864//LAB2")
flintlead<-read.csv(file="Flint-water-lead-dataset.csv",header=FALSE)
colnames(flintlead)=c("SampleID","ZipCode","Ward", "0sec", "45sec", "120sec")
table(flintlead[2])
```

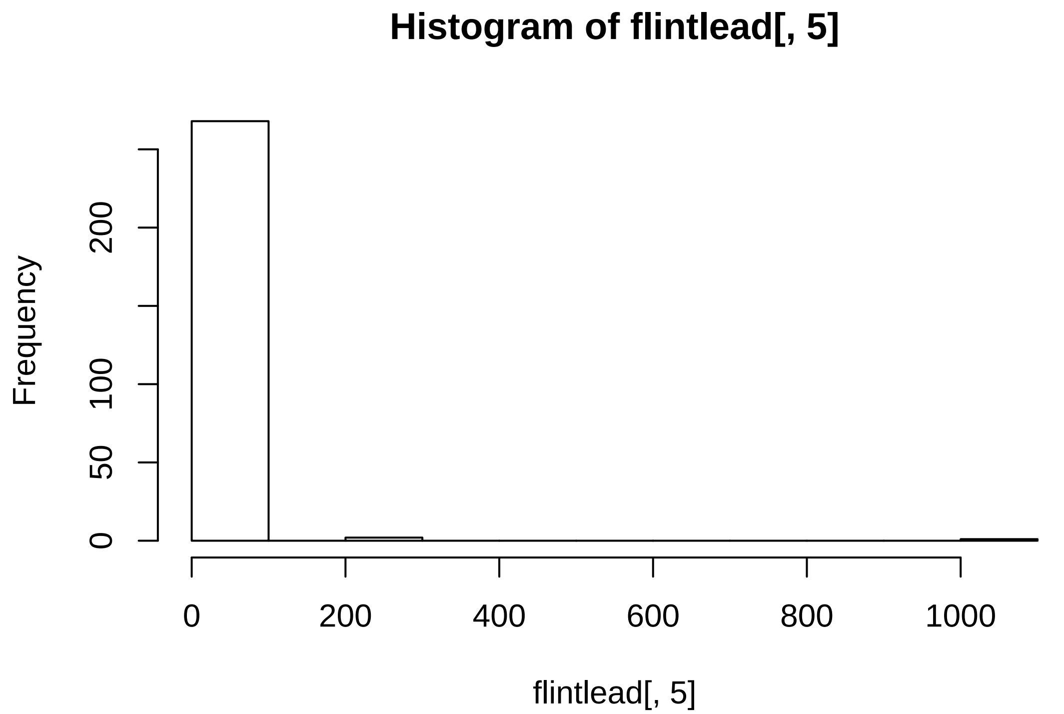
```
##
## 48502 48503 48504 48505 48506 48507 48529 48532
##     1    69    55    48    44    51     1     2
```

They are not evenly sampled.
Q1(b)

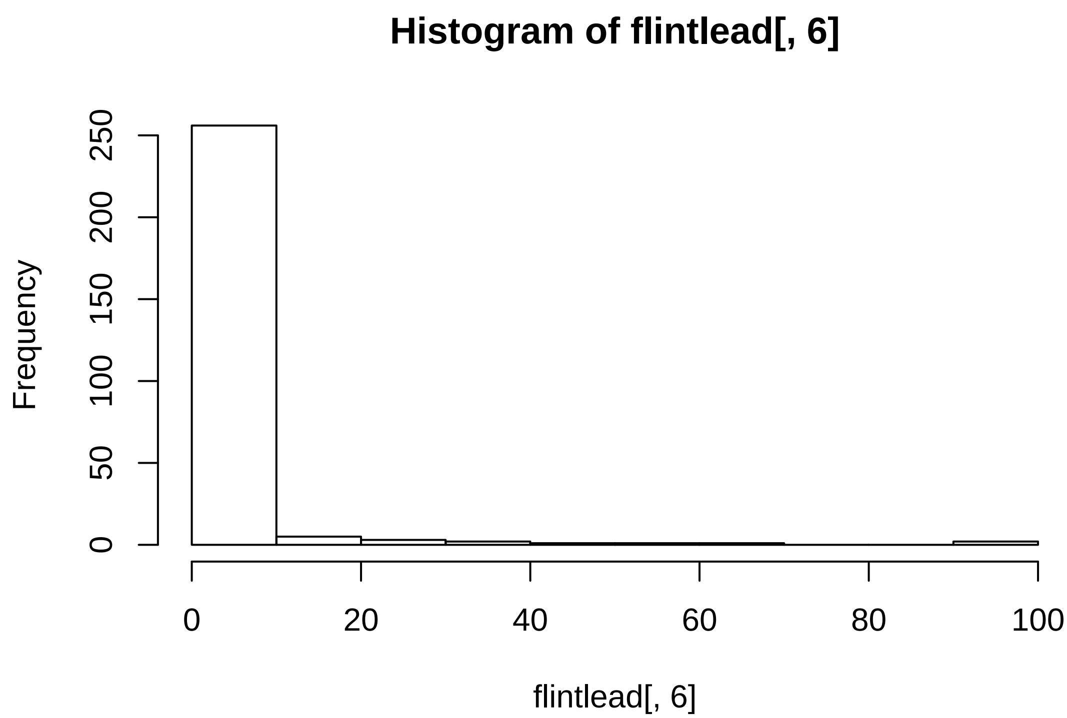
```
hist(flintlead[,4])
```




```
hist(flintlead[,5])
```



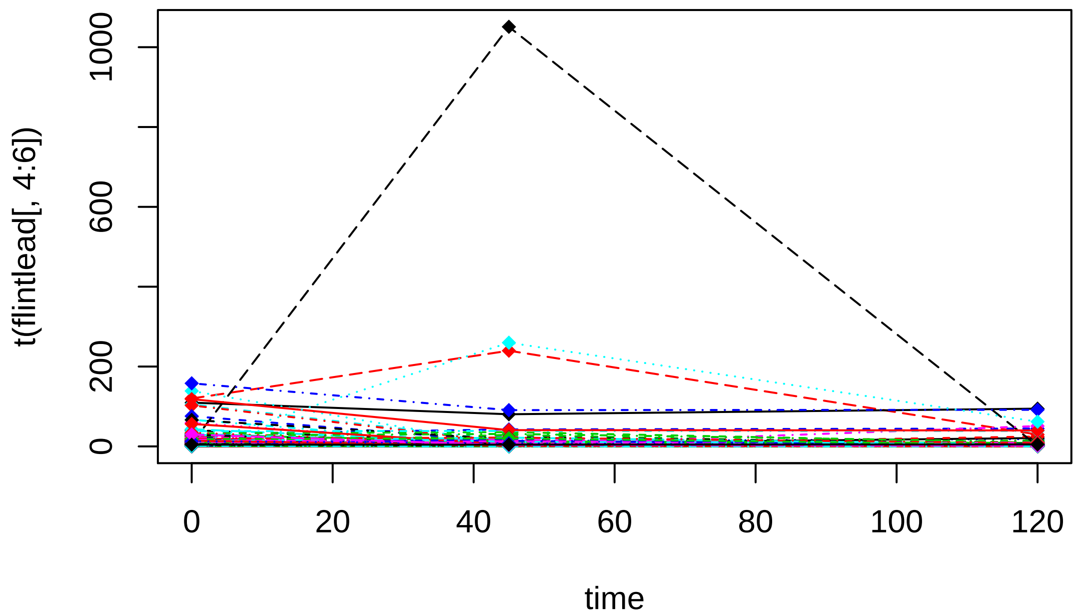

```r
hist(flintlead[,6])
```



All

three histograms have the shape of "L".
Q1(c)

```
time<-c(0, 45, 120)
matplot(time, t(flintlead[,4:6]),pch=18,type="o")
```

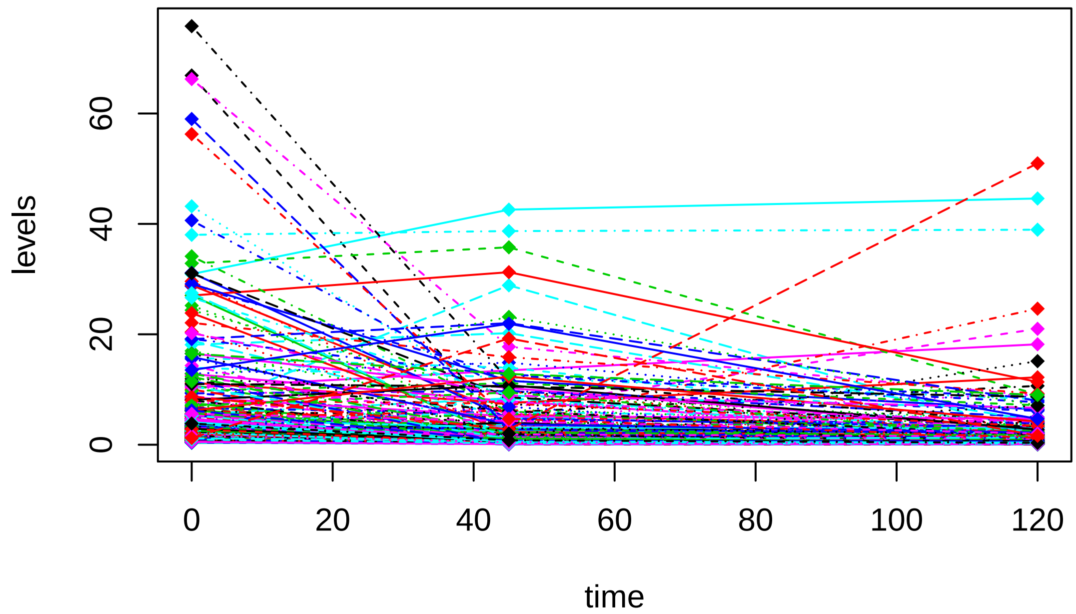


There's an extreme value larger than 1000.

```
flintlead2<-flintlead[
  (flintlead[,4]<100)&(flintlead[,5]<100)&(flintlead[,6]<100),]
```

Q1(d)

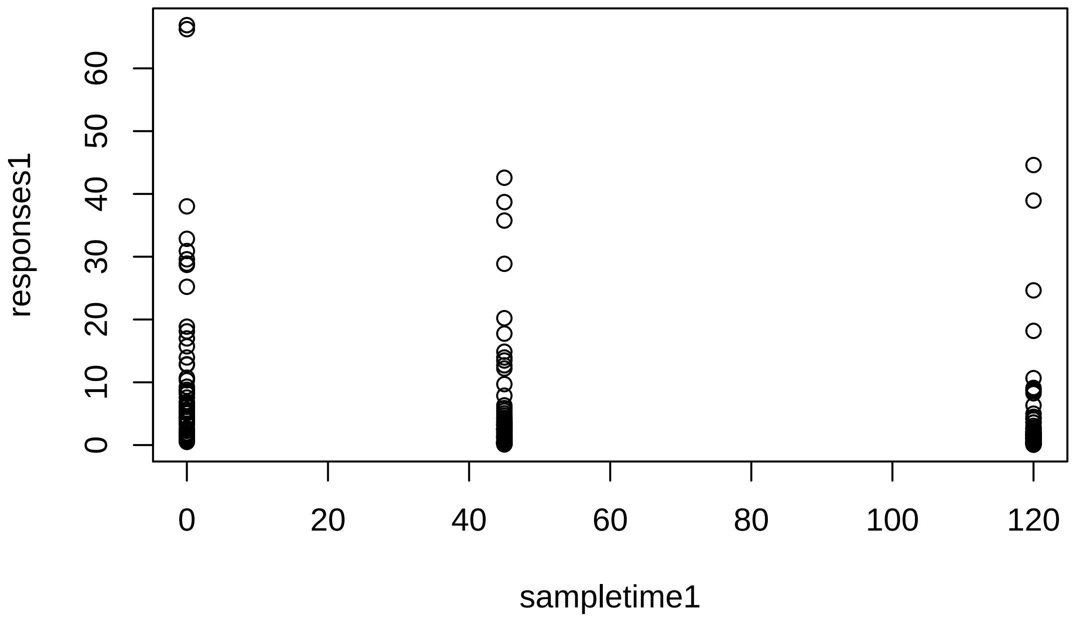
```
for(i in 48503:48507){
    zipcode<-which(flintlead2[,2]==i)
    matplot(time,t(flintlead2[zipcode,4:6]),type="o",ylab="levels",pch=18,
              add= (i!=48503),ylim=c(0,ceiling(max(flintlead2[,4:6]))))
}
```



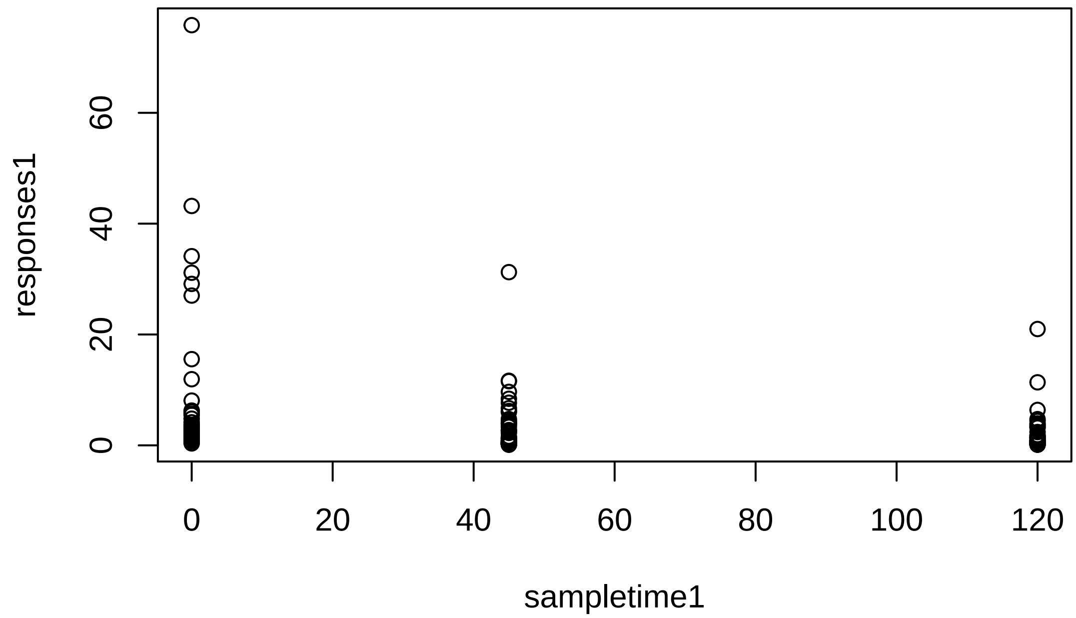
There are some nonlinearity patterns

Q2

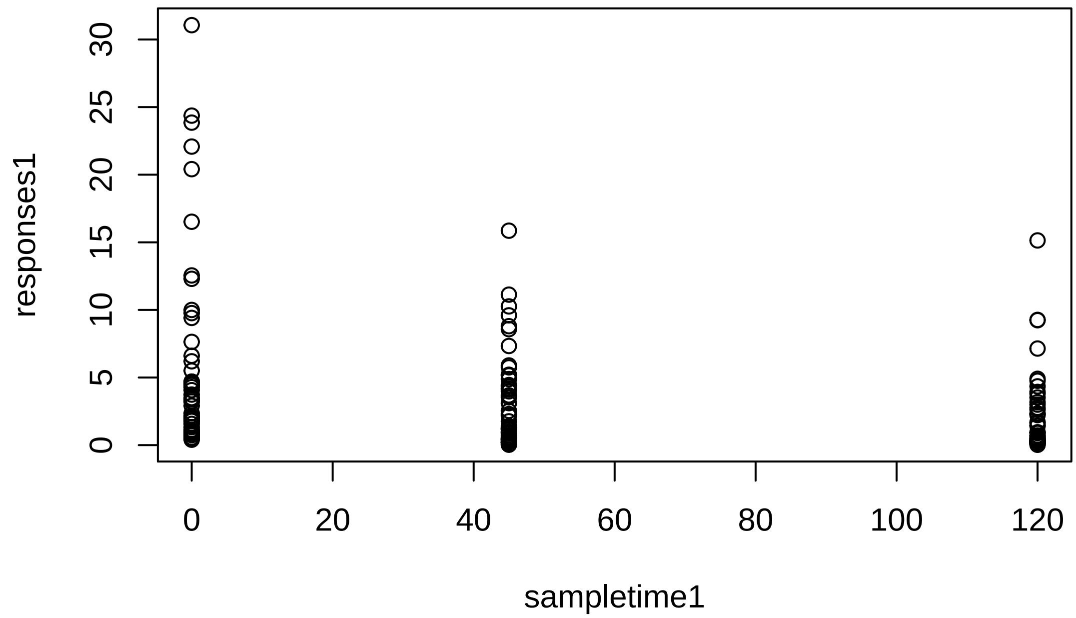
```
Coef_Matr1<-matrix(0,5,2)
Coef_Matr2<-matrix(0,5,3)
Sigma_Matr<-matrix(0,5,2)
for(i in 48503:48507){
zipcode<-which(flintlead2[,2]==i)
subsetflintlead<-flintlead2[zipcode,]
responses1<-unlist(subsetflintlead[,4:6])
sampletime1<-rep(time,each=dim(subsetflintlead)[1])
plot(sampletime1, responses1)
nlsreg1<-nls(responses1~theta1*exp(-sampletime1*theta2),
             start=list(theta1=5,theta2=0.02))
nlsreg2<-nls(responses1~theta1/(1+theta2*(exp(sampletime1*theta3))),
             start=list(theta1=2,theta2=-0.7,theta3=-0.025))
Coef_Matr1[i-48502,]<-coef(nlsreg1)
Coef_Matr2[i-48502,]<-coef(nlsreg2)
Sigma_Matr[i-48502,]<-c(summary(nlsreg1)$sigma,summary(nlsreg2)$sigma)
print(summary(nlsreg1))
print(summary(nlsreg2))
}
```




```
##
## Formula: responses1 ~ theta1 * exp(-sampletime1 * theta2)
##
## Parameters:
##        Estimate Std. Error t value Pr(>|t|)
## theta1 9.272776   1.193044   7.772 3.84e-13 ***
## theta2 0.008745   0.003117   2.805  0.00552 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.4 on 202 degrees of freedom
##
## Number of iterations to convergence: 4
## Achieved convergence tolerance: 9.347e-06
##
##
## Formula: responses1 ~ theta1/(1 + theta2 * (exp(sampletime1 * theta3)))
##
## Parameters:
##         Estimate Std. Error t value Pr(>|t|)
## theta1  2.462919   5.532201   0.445    0.657
## theta2 -0.742616   0.575919  -1.289    0.199
## theta3 -0.006411   0.025204  -0.254    0.799
##
## Residual standard error: 10.41 on 201 degrees of freedom
##
## Number of iterations to convergence: 5
## Achieved convergence tolerance: 1.476e-07
```

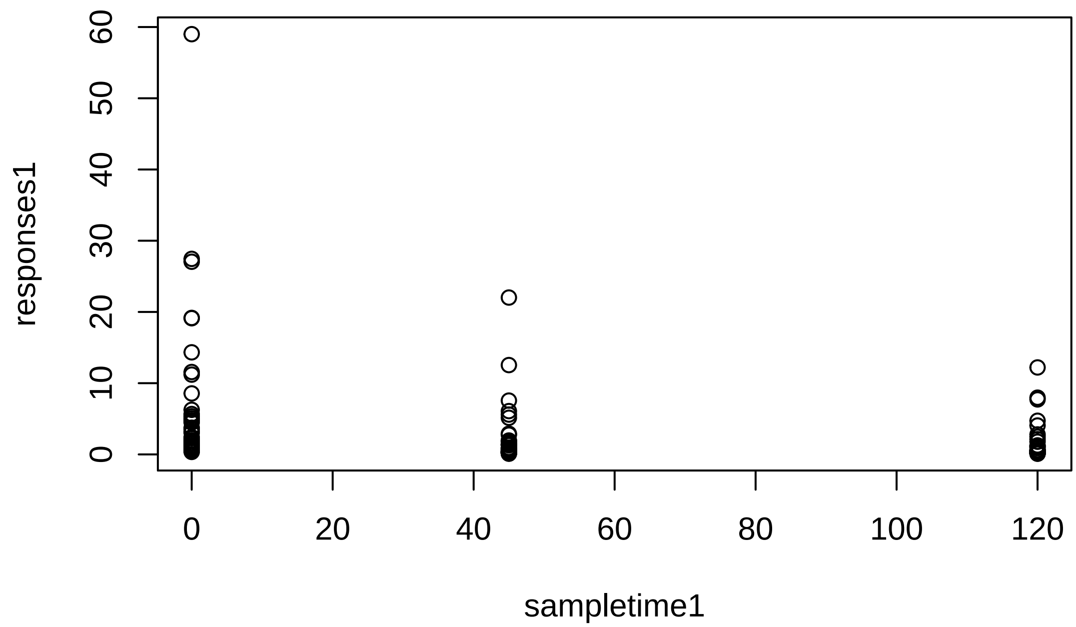



```
##
## Formula: responses1 ~ theta1 * exp(-sampletime1 * theta2)
##
## Parameters:
##        Estimate Std. Error t value Pr(>|t|)
## theta1  7.47117    1.20428   6.204 5.24e-09 ***
## theta2  0.01601    0.00644   2.486    0.014 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.668 on 148 degrees of freedom
##
## Number of iterations to convergence: 6
## Achieved convergence tolerance: 5.917e-06
##
##
## Formula: responses1 ~ theta1/(1 + theta2 * (exp(sampletime1 * theta3)))
##
## Parameters:
##         Estimate Std. Error t value Pr(>|t|)
## theta1  1.167818   3.973293   0.294    0.769
## theta2 -0.847368   0.519010  -1.633    0.105
## theta3 -0.007078   0.036573  -0.194    0.847
##
## Residual standard error: 8.68 on 147 degrees of freedom
##
## Number of iterations to convergence: 4
## Achieved convergence tolerance: 9.57e-08
```



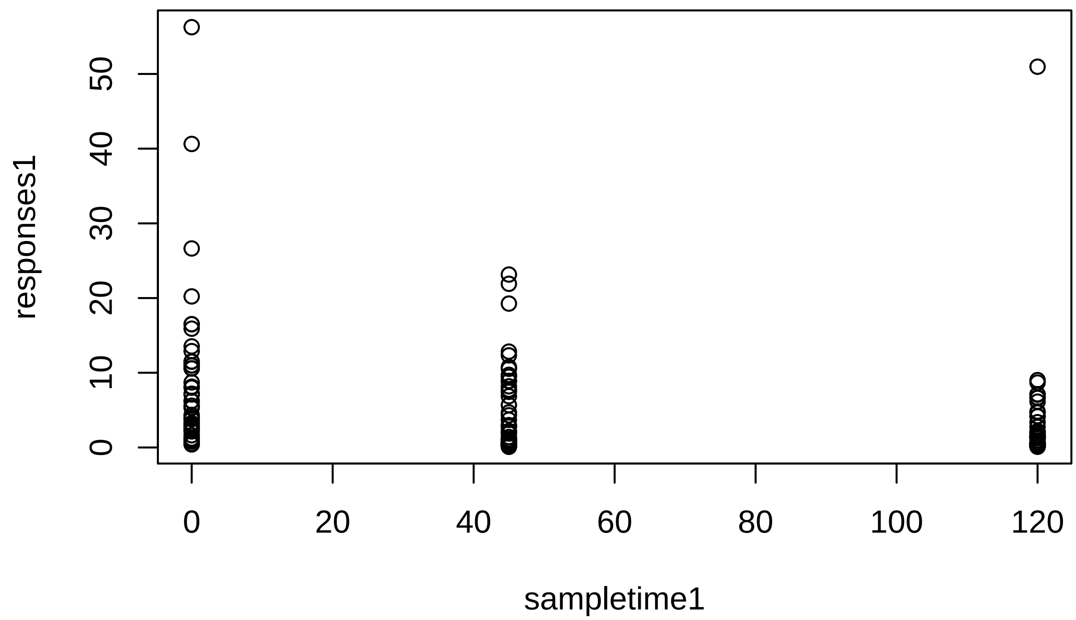

```
##
## Formula: responses1 ~ theta1 * exp(-sampletime1 * theta2)
##
## Parameters:
##        Estimate Std. Error t value Pr(>|t|)
## theta1 5.891855   0.689998   8.539 1.89e-14 ***
## theta2 0.010384   0.003191   3.254  0.00142 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.996 on 142 degrees of freedom
##
## Number of iterations to convergence: 6
## Achieved convergence tolerance: 8.061e-06
##
##
## Formula: responses1 ~ theta1/(1 + theta2 * (exp(sampletime1 * theta3)))
##
## Parameters:
##         Estimate Std. Error t value Pr(>|t|)
## theta1  1.445314   2.643304   0.547   0.5854
## theta2 -0.762648   0.433083  -1.761   0.0804 .
## theta3 -0.007398   0.023681  -0.312   0.7552
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.997 on 141 degrees of freedom
##
```

```
## Number of iterations to convergence: 5
## Achieved convergence tolerance: 1.254e-10
```




```
##
## Formula: responses1 ~ theta1 * exp(-sampletime1 * theta2)
##
## Parameters:
##        Estimate Std. Error t value Pr(>|t|)
## theta1 6.501135   1.031689   6.301 4.69e-09 ***
## theta2 0.020066   0.008106   2.476   0.0147 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.742 on 124 degrees of freedom
##
## Number of iterations to convergence: 2
## Achieved convergence tolerance: 8.089e-06
##
##
## Formula: responses1 ~ theta1/(1 + theta2 * (exp(sampletime1 * theta3)))
##
## Parameters:
##        Estimate Std. Error t value Pr(>|t|)
## theta1  1.23301    2.08397   0.592    0.555
## theta2 -0.81399    0.31537  -2.581    0.011 *
## theta3 -0.01462    0.05044  -0.290    0.772
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.744 on 123 degrees of freedom
##
## Number of iterations to convergence: 3
## Achieved convergence tolerance: 5.566e-08
```




```
##
## Formula: responses1 ~ theta1 * exp(-sampletime1 * theta2)
##
## Parameters:
##        Estimate Std. Error t value Pr(>|t|)
## theta1 7.895872   1.072104   7.365 1.15e-11 ***
## theta2 0.009423   0.003454   2.728  0.00715 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.973 on 148 degrees of freedom
##
## Number of iterations to convergence: 4
## Achieved convergence tolerance: 2.032e-06
##
##
## Formula: responses1 ~ theta1/(1 + theta2 * (exp(sampletime1 * theta3)))
##
## Parameters:
##         Estimate Std. Error t value Pr(>|t|)
```

```
## theta1  2.121807   4.294901   0.494    0.622
## theta2 -0.740089   0.524532  -1.411    0.160
## theta3 -0.007429   0.027337  -0.272    0.786
##
## Residual standard error: 7.984 on 147 degrees of freedom
##
## Number of iterations to convergence: 4
## Achieved convergence tolerance: 5.755e-08
```

Q3
The leaf levels decrease over the flushing time.

$$H_0 : \mu_{120} = \mu_0; H_a : \mu_{120} \neq \mu_0.$$

```r
var(flintlead2[,4])
```

```
## [1] 131.2726
```

```r
var(flintlead2[,6])
```

```
## [1] 31.56412
```

```r
var.test(flintlead2[,4],flintlead2[,6])
```

```
##
##  F test to compare two variances
##
## data:  flintlead2[, 4] and flintlead2[, 6]
## F = 4.1589, num df = 261, denom df = 261, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  3.261173 5.303799
## sample estimates:
## ratio of variances
##           4.158919
```

```r
t.test(flintlead2[,4],flintlead2[,6],var.equal=FALSE,paired=F)
```

```
##
##  Welch Two Sample t-test
##
## data:  flintlead2[, 4] and flintlead2[, 6]
## t = 6.5749, df = 379.65, p-value = 1.613e-10
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  3.633291 6.733495
## sample estimates:
## mean of x mean of y
##  7.719229  2.535836
```

Conclude $H_a : \mu_{120} \neq \mu_0$;It's an efficient way.
Q4

```
Ftest_Mat2<-matrix(0,nrow=10,ncol=5)
i<-1
for(i1 in 48503:48506){
    for (i2 in (i1+1):48507){
        zipcode<-which(flintlead2[,2]==i1|flintlead2[,2]==i2)
        subsetflintlead<-flintlead2[zipcode,]
        subsetflintlead[,7]<-(subsetflintlead[,2]==i1)+0
        ini1<-rep(subsetflintlead[,7],each=3)
        responses1<-unlist(subsetflintlead[,4:6])
        Log_res<-log(responses1)
        sampletime1<-rep(time,each=dim(subsetflintlead)[1])*ini1
        sampletime2<-rep(time,each=dim(subsetflintlead)[1])*(1-ini1)
        reg2<-lm(Log_res~ini1+sampletime1+sampletime2)
        f_stat<-summary(reg2)$fstatistic[1]
        df1<-summary(reg2)$fstatistic[2]
        df2<-summary(reg2)$fstatistic[3]
        Ftest_Mat2[i,1]<-i1
        Ftest_Mat2[i,2]<-i2
        Ftest_Mat2[i,3]<-f_stat
        Ftest_Mat2[i,4]<-qf(0.95,df1,df2)
        Ftest_Mat2[i,5]<-(f_stat<=qf(0.95,df1,df2))
        i<-i+1
    }
}
Ftest_Mat2
```

```
##        [,1]  [,2]     [,3]     [,4] [,5]
##  [1,] 48503 48504 18.41119 2.630420    0
##  [2,] 48503 48505 16.50556 2.630867    0
##  [3,] 48503 48506 17.86897 2.632310    0
##  [4,] 48503 48507 19.97337 2.630420    0
##  [5,] 48504 48505 16.93688 2.635735    0
##  [6,] 48504 48506 20.10540 2.637791    0
##  [7,] 48504 48507 19.24986 2.635106    0
##  [8,] 48505 48506 20.74773 2.638538    0
##  [9,] 48505 48507 16.58577 2.635735    0
## [10,] 48506 48507 18.83960 2.637791    0
```

There's siginificant differences among areas different zip codes.
Q5

```
for(i in 48503:48507){
    time0<-0
    theta1hat<-Coef_Matr1[i-48502,1]
    theta2hat<-Coef_Matr1[i-48502,2]
    meany<-theta1hat*exp(-time0*theta2hat)
    sigmahat<-Sigma_Matr[i-48502,1]
    y90quantile<-qnorm(0.9, meany, sigmahat)
    print(i)
    print(y90quantile)
}
```

```
## [1] 48503
## [1] 22.59692
## [1] 48504
## [1] 18.5802
## [1] 48505
## [1] 12.29387
## [1] 48506
## [1] 15.14081
## [1] 48507
## [1] 18.11319
```

Q6

```r
library(MASS)
f<-function(theta1,theta2,x){
    return(theta1*exp(-theta2*x))
}
Db_hat<-function(theta1,theta2,x){
  D<-cbind(exp(-theta2*x),-x*theta1*exp(-theta2*x))
  return(D)
}
G_hat<-function(theta1,theta2,x){
  G<-matrix(c(exp(-theta2*x),-x*theta1*exp(-theta2*x)),nrow=1,ncol=2)
  return(G)
}
CI<-function(para1,para2,sig,X,x0,alpha){
  D<-Db_hat(para1,para2,X)
  G<-G_hat(para1,para2,x0)
  delta<-qnorm(1-0.5*alpha)*sig*sqrt(G%*%(ginv(t(D)%*%D))%*%t(G))
  up<-f(para1,para2,x0)+delta
  low<-f(para1,para2,x0)-delta
  Interval<-c(low,up)
  return(Interval)
}
for (i in 48503:48507){
    theta1hat<-Coef_Matr1[i-48502,1]
    theta2hat<-Coef_Matr1[i-48502,2]
    sigmahat<-Sigma_Matr[i-48502,1]
    zipcode<-which(flintlead2[,2]==i)
    subsetflintlead<-flintlead2[zipcode,]
    sampletime1<-rep(time,each=dim(subsetflintlead)[1])
    CI_0<-CI(theta1hat,theta2hat,sigmahat,sampletime1,0,0.05)
    CI_45<-CI(theta1hat,theta2hat,sigmahat,sampletime1,45,0.05)
    CI_120<-CI(theta1hat,theta2hat,sigmahat,sampletime1,120,0.05)
    print(i)
    print(CI_0)
    print(CI_45)
    print(CI_120)

}
```

```
## [1] 48503
## [1]  6.934453 11.611099
```

```
## [1] 4.670611 7.841846
## [1] 1.190047 5.303978
## [1] 48504
## [1] 5.110824 9.831508
## [1] 1.740264 5.529863
## [1] -0.4501431  2.6383149
## [1] 48505
## [1] 4.539483 7.244226
## [1] 2.743593 4.641413
## [1] 0.5700071 2.8194663
## [1] 48506
## [1] 4.479061 8.523208
## [1] 0.8759067 4.3947681
## [1] -0.4776228  1.6478710
## [1] 48507
## [1] 5.794586 9.997158
## [1] 3.722932 6.611094
## [1] 0.7413708 4.3558711
```

Q7

```
flintlead3<-flintlead2[flintlead2[,2]!=48502,]
flintlead3<-flintlead3[flintlead3[,2]!=48529,]
flintlead3<-flintlead3[flintlead3[,2]!=48532,]
resp_ALL<-unlist(flintlead3[,4:6])
x48503<-rep(((flintlead3[,2]==48503)+0),3)
x48504<-rep(((flintlead3[,2]==48504)+0),3)
x48505<-rep(((flintlead3[,2]==48505)+0),3)
x48506<-rep(((flintlead3[,2]==48506)+0),3)
x48507<-rep(((flintlead3[,2]==48507)+0),3)
stime<-rep(time,each=dim(flintlead3)[1])
regzip<-nls(resp_ALL~b0*exp(-(
    b1*x48503*stime+
    b2*x48504*stime+
    b3*x48505*stime+
    b4*x48506*stime+
    b5*x48507*stime)),
    start=list(b0=4,b1=0.002,b2=0.001,b3=0.002,b4=-0.0002,b5=0))
regzip
```

```
## Nonlinear regression model
##   model: resp_ALL ~ b0 * exp(-(b1 * x48503 * stime + b2 * x48504 * stime +     b3 * x48505 * stime +
##    data: parent.frame()
##       b0       b1       b2       b3       b4       b5
## 7.630974 0.006204 0.016439 0.014863 0.024148 0.008934
##  residual sum-of-squares: 51952
##
## Number of iterations to convergence: 8
## Achieved convergence tolerance: 4.383e-06
```

Q8

```r
for(i in 48503:48507){
    BootS_Times<-100
    N<-nrow(flintlead2)
    Num<-150
    meany_Accumul<-0
    sigma_Accumul<-0
    for(step in 1:BootS_Times){
        Sample_Num<-sample(1:N,Num,replace = FALSE)
        Boot_Sample<-flintlead2[Sample_Num,]
        time0<-0
        zipcode_BS<-which(Boot_Sample[,2]==i)
        subsetflintlead_BS<-Boot_Sample[zipcode_BS,]
        responses_BS<-unlist(subsetflintlead_BS[,4:6])
        sampletime_BS<-rep(time,each=dim(subsetflintlead_BS)[1])
        nlsreg1_BS<-nls(responses_BS~theta1*exp(-sampletime_BS*theta2),
                start=list(theta1=5,theta2=0.02))
        theta1hat_BS<-coef(nlsreg1_BS)[1]
        theta2hat_BS<-coef(nlsreg1_BS)[2]
        meany_BS<-theta1hat_BS*exp(-time0*theta2hat_BS)
        sigmahat_BS<-summary(nlsreg1_BS)$sigma
        meany_Accumul<-meany_Accumul+meany_BS
        sigma_Accumul<-sigma_Accumul+sigmahat_BS^2
    }
    y90quantile_BS<-qnorm(0.9, (meany_Accumul/BootS_Times), sqrt(sigma_Accumul/BootS_Times))
    print(i)
    print(y90quantile_BS)
}
```

```
## [1] 48503
## [1] 22.93506
## [1] 48504
## [1] 18.58756
## [1] 48505
## [1] 12.53812
## [1] 48506
## [1] 15.23621
## [1] 48507
## [1] 18.2254
```