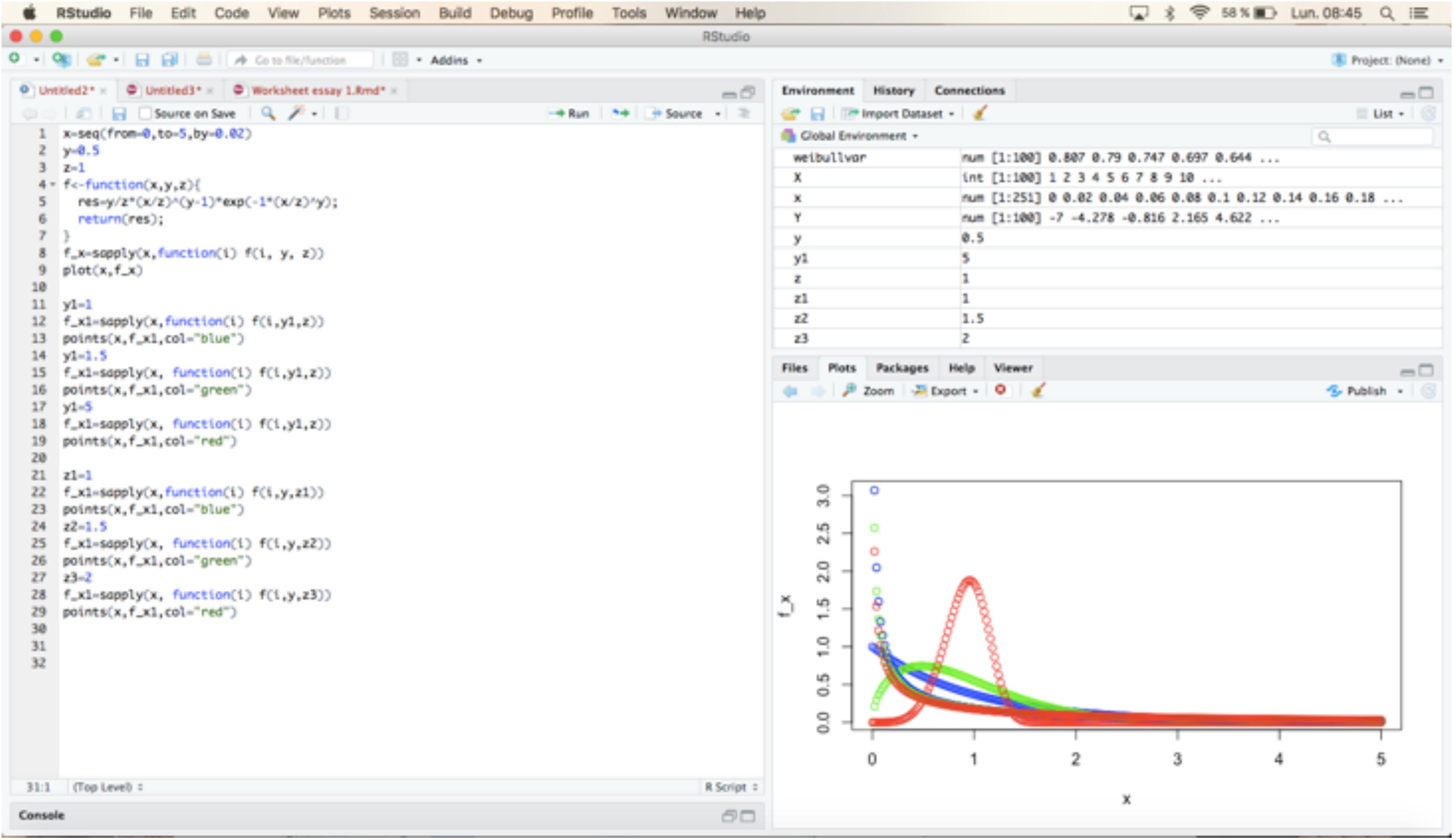
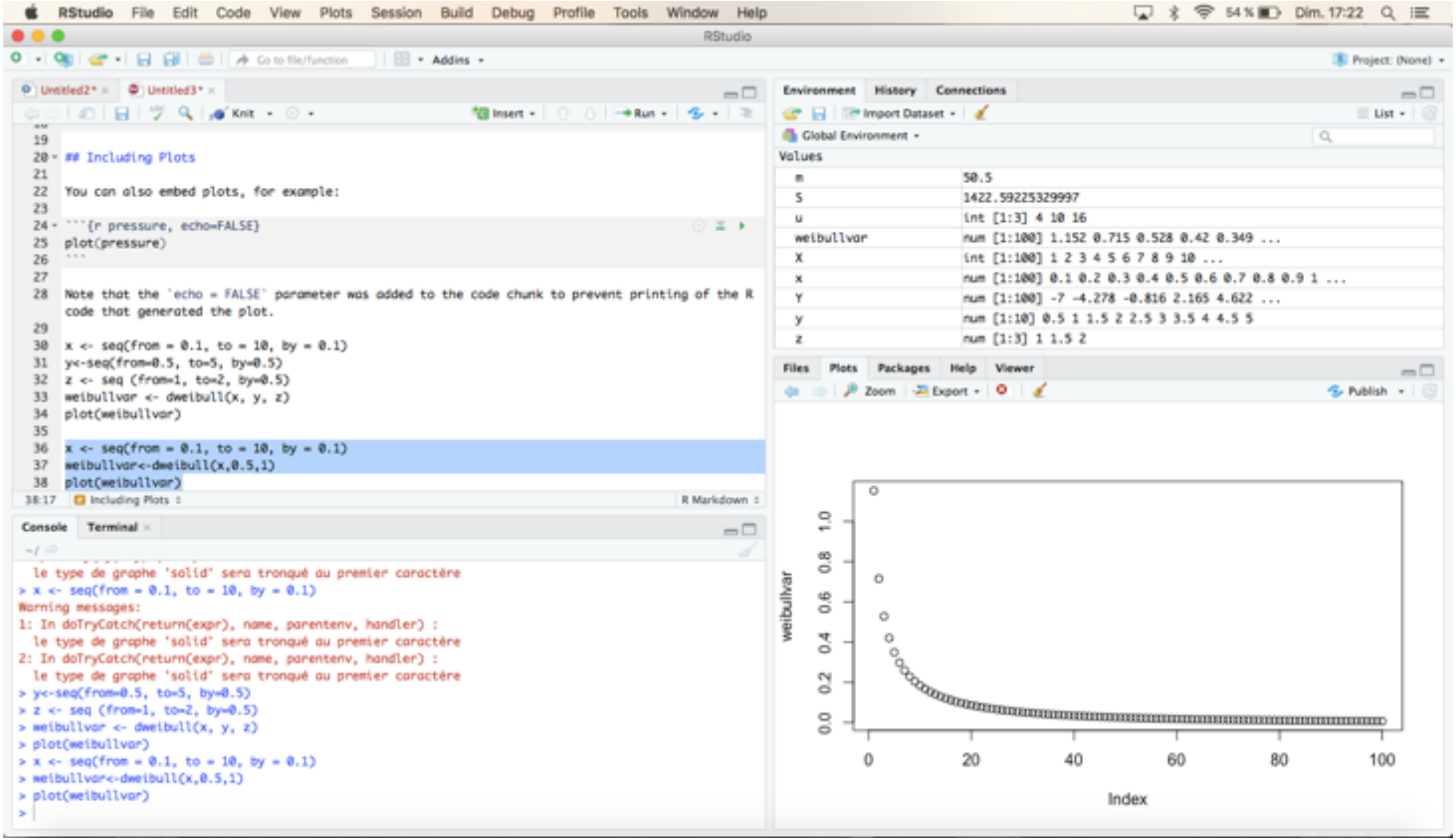


```
x=seq(from=0,to=5,by=0.02)
y=0.5
z=1
f<-function(x,y,z){
  res=y/z*(x/z)^(y-1)*exp(-1*(x/z)^y);
  return(res);
}
f_x=sapply(x,function(i) f(i, y, z))
plot(x,f_x)

y1=1
f_x1=sapply(x,function(i) f(i,y1,z))
points(x,f_x1,col="blue")
y1=1.5
f_x1=sapply(x, function(i) f(i,y1,z))
points(x,f_x1,col="green")
y1=5
f_x1=sapply(x, function(i) f(i,y1,z))
points(x,f_x1,col="red")

z1=1
f_x1=sapply(x,function(i) f(i,y,z1))
points(x,f_x1,col="blue")
z2=1.5
f_x1=sapply(x, function(i) f(i,y,z2))
points(x,f_x1,col="green")
z3=2
f_x1=sapply(x, function(i) f(i,y,z3))
points(x,f_x1,col="red")
```

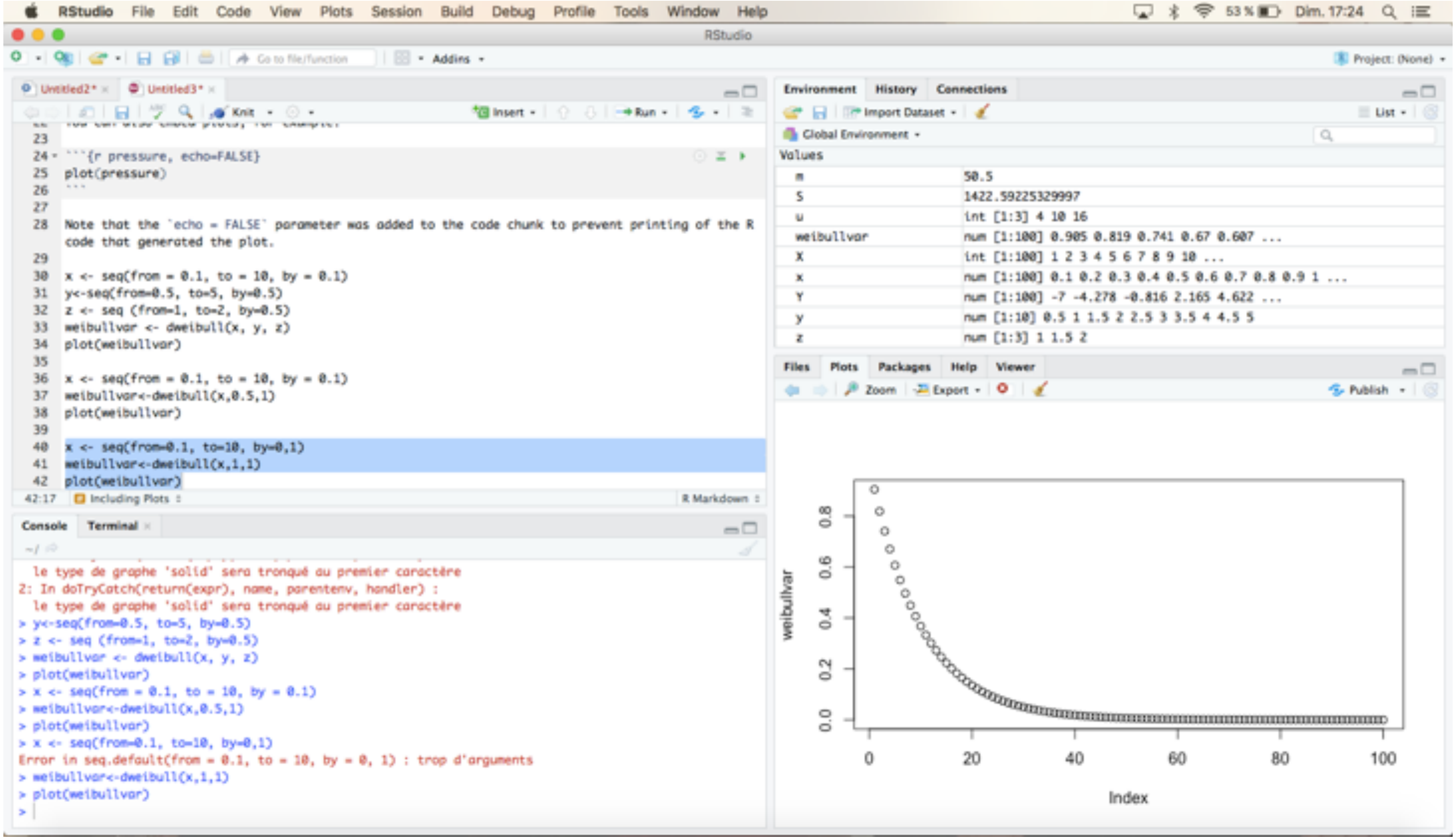
```
weibullvar   num [1:100] 0.807 0.79 0.747 0.697 0.644 ...
X            int [1:100] 1 2 3 4 5 6 7 8 9 10 ...
x            num [1:251] 0 0.02 0.04 0.06 0.08 0.1 0.12 0.14 0.16 0.18 ...
Y            num [1:100] -7 -4.278 -0.816 2.165 4.622 ...
y            0.5
y1           5
z            1
z1           1
z2           1.5
z3           2
```



~~~
## Including Plots

You can also embed plots, for example:

```{r pressure, echo=FALSE}
plot(pressure)
```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

x <- seq(from = 0.1, to = 10, by = 0.1)
y<-seq(from=0.5, to=5, by=0.5)
z <- seq (from=1, to=2, by=0.5)
weibullvar <- dweibull(x, y, z)
plot(weibullvar)

x <- seq(from = 0.1, to = 10, by = 0.1)
weibullvar<-dweibull(x,0.5,1)
plot(weibullvar)
~~~

```
Values
m            50.5
S            1422.59225329997
u            int [1:3] 4 10 16
weibullvar   num [1:100] 1.152 0.715 0.528 0.42 0.349 ...
X            int [1:100] 1 2 3 4 5 6 7 8 9 10 ...
x            num [1:100] 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1 ...
Y            num [1:100] -7 -4.278 -0.816 2.165 4.622 ...
y            num [1:10] 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5
z            num [1:3] 1 1.5 2
```

```
  le type de graphe 'solid' sera tronqué au premier caractère
> x <- seq(from = 0.1, to = 10, by = 0.1)
Warning messages:
1: In doTryCatch(return(expr), name, parentenv, handler) :
  le type de graphe 'solid' sera tronqué au premier caractère
2: In doTryCatch(return(expr), name, parentenv, handler) :
  le type de graphe 'solid' sera tronqué au premier caractère
> y<-seq(from=0.5, to=5, by=0.5)
> z <- seq (from=1, to=2, by=0.5)
> weibullvar <- dweibull(x, y, z)
> plot(weibullvar)
> x <- seq(from = 0.1, to = 10, by = 0.1)
> weibullvar<-dweibull(x,0.5,1)
> plot(weibullvar)
> 
```

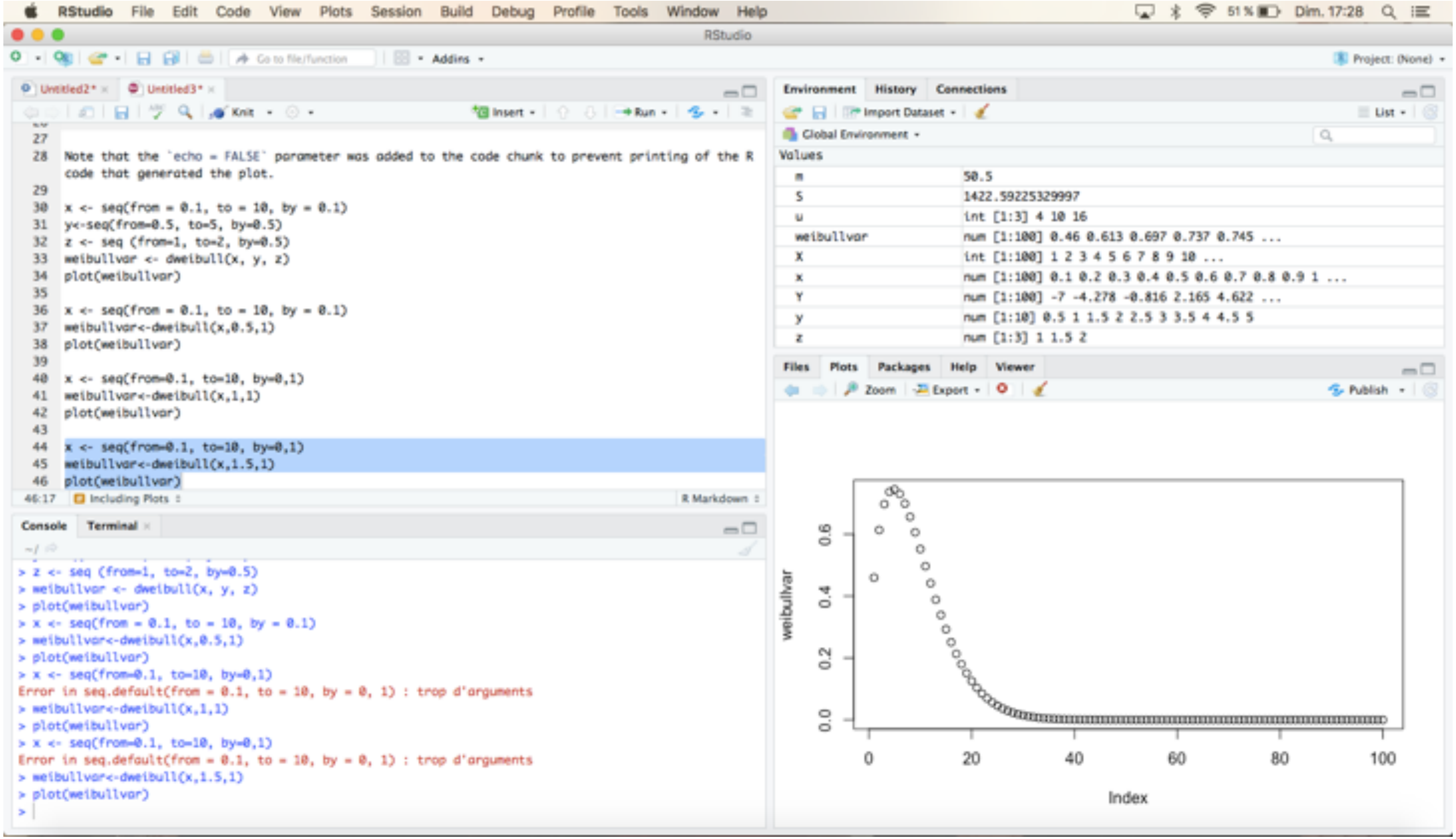


~~~
```{r pressure, echo=FALSE}
plot(pressure)
```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R
code that generated the plot.

x <- seq(from = 0.1, to = 10, by = 0.1)
y<-seq(from=0.5, to=5, by=0.5)
z <- seq (from=1, to=2, by=0.5)
weibullvar <- dweibull(x, y, z)
plot(weibullvar)

x <- seq(from = 0.1, to = 10, by = 0.1)
weibullvar<-dweibull(x,0.5,1)
plot(weibullvar)

x <- seq(from=0.1, to=10, by=0,1)
weibullvar<-dweibull(x,1,1)
plot(weibullvar)
~~~

Values

```
m             50.5
S             1422.59225329997
u             int [1:3] 4 10 16
weibullvar    num [1:100] 0.905 0.819 0.741 0.67 0.607 ...
X             int [1:100] 1 2 3 4 5 6 7 8 9 10 ...
x             num [1:100] 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1 ...
Y             num [1:100] -7 -4.278 -0.816 2.165 4.622 ...
y             num [1:10] 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5
z             num [1:3] 1 1.5 2
```

```
  le type de graphe 'solid' sera tronqué au premier caractère
2: In doTryCatch(return(expr), name, parentenv, handler) :
  le type de graphe 'solid' sera tronqué au premier caractère
> y<-seq(from=0.5, to=5, by=0.5)
> z <- seq (from=1, to=2, by=0.5)
> weibullvar <- dweibull(x, y, z)
> plot(weibullvar)
> x <- seq(from = 0.1, to = 10, by = 0.1)
> weibullvar<-dweibull(x,0.5,1)
> plot(weibullvar)
> x <- seq(from=0.1, to=10, by=0,1)
Error in seq.default(from = 0.1, to = 10, by = 0, 1) : trop d'arguments
> weibullvar<-dweibull(x,1,1)
> plot(weibullvar)
> 
```

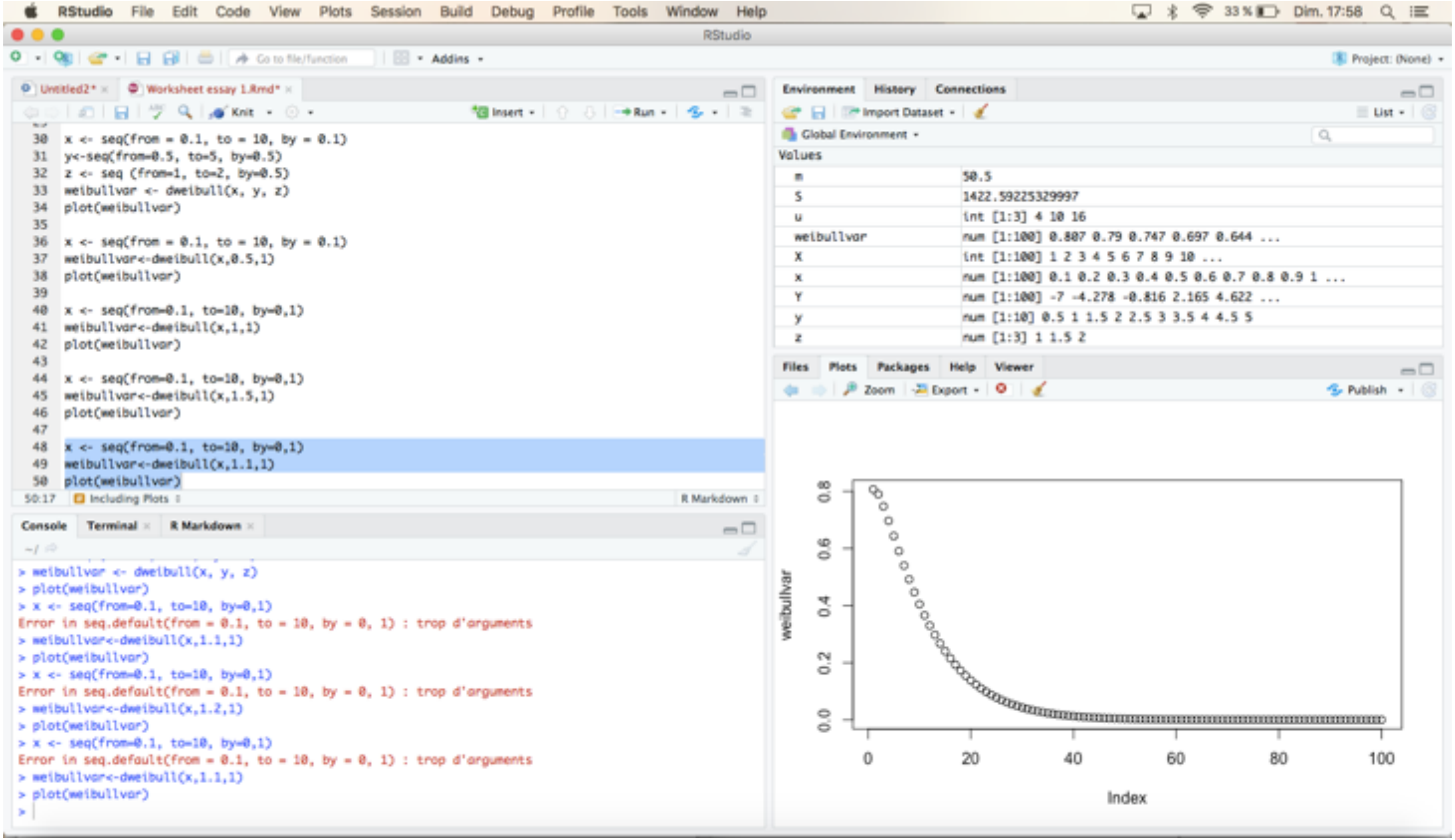


```
Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R
code that generated the plot.

x <- seq(from = 0.1, to = 10, by = 0.1)
y<-seq(from=0.5, to=5, by=0.5)
z <- seq (from=1, to=2, by=0.5)
weibullvar <- dweibull(x, y, z)
plot(weibullvar)

x <- seq(from = 0.1, to = 10, by = 0.1)
weibullvar<-dweibull(x,0.5,1)
plot(weibullvar)

x <- seq(from=0.1, to=10, by=0,1)
weibullvar<-dweibull(x,1,1)
plot(weibullvar)

x <- seq(from=0.1, to=10, by=0,1)
weibullvar<-dweibull(x,1.5,1)
plot(weibullvar)
```

Values

m	50.5
S	1422.59225329997
u	int [1:3] 4 10 16
weibullvar	num [1:100] 0.46 0.613 0.697 0.737 0.745 ...
X	int [1:100] 1 2 3 4 5 6 7 8 9 10 ...
x	num [1:100] 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1 ...
Y	num [1:100] -7 -4.278 -0.816 2.165 4.622 ...
y	num [1:10] 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5
z	num [1:3] 1 1.5 2

```
> z <- seq (from=1, to=2, by=0.5)
> weibullvar <- dweibull(x, y, z)
> plot(weibullvar)
> x <- seq(from = 0.1, to = 10, by = 0.1)
> weibullvar<-dweibull(x,0.5,1)
> plot(weibullvar)
> x <- seq(from=0.1, to=10, by=0,1)
Error in seq.default(from = 0.1, to = 10, by = 0, 1) : trop d'arguments
> weibullvar<-dweibull(x,1,1)
> plot(weibullvar)
> x <- seq(from=0.1, to=10, by=0,1)
Error in seq.default(from = 0.1, to = 10, by = 0, 1) : trop d'arguments
> weibullvar<-dweibull(x,1.5,1)
> plot(weibullvar)
>
```



```
x <- seq(from = 0.1, to = 10, by = 0.1)
y<-seq(from=0.5, to=5, by=0.5)
z <- seq (from=1, to=2, by=0.5)
weibullvar <- dweibull(x, y, z)
plot(weibullvar)

x <- seq(from = 0.1, to = 10, by = 0.1)
weibullvar<-dweibull(x,0.5,1)
plot(weibullvar)

x <- seq(from=0.1, to=10, by=0,1)
weibullvar<-dweibull(x,1,1)
plot(weibullvar)

x <- seq(from=0.1, to=10, by=0,1)
weibullvar<-dweibull(x,1.5,1)
plot(weibullvar)

x <- seq(from=0.1, to=10, by=0,1)
weibullvar<-dweibull(x,1.1,1)
plot(weibullvar)
```

Values

m	50.5
S	1422.59225329997
u	int [1:3] 4 10 16
weibullvar	num [1:100] 0.807 0.79 0.747 0.697 0.644 ...
X	int [1:100] 1 2 3 4 5 6 7 8 9 10 ...
x	num [1:100] 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1 ...
Y	num [1:100] -7 -4.278 -0.816 2.165 4.622 ...
y	num [1:10] 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5
z	num [1:3] 1 1.5 2

```
> weibullvar <- dweibull(x, y, z)
> plot(weibullvar)
> x <- seq(from=0.1, to=10, by=0,1)
Error in seq.default(from = 0.1, to = 10, by = 0, 1) : trop d'arguments
> weibullvar<-dweibull(x,1.1,1)
> plot(weibullvar)
> x <- seq(from=0.1, to=10, by=0,1)
Error in seq.default(from = 0.1, to = 10, by = 0, 1) : trop d'arguments
> weibullvar<-dweibull(x,1.2,1)
> plot(weibullvar)
> x <- seq(from=0.1, to=10, by=0,1)
Error in seq.default(from = 0.1, to = 10, by = 0, 1) : trop d'arguments
> weibullvar<-dweibull(x,1.1,1)
> plot(weibullvar)
>
```

DETEVE Louise

Worksheet 1 :

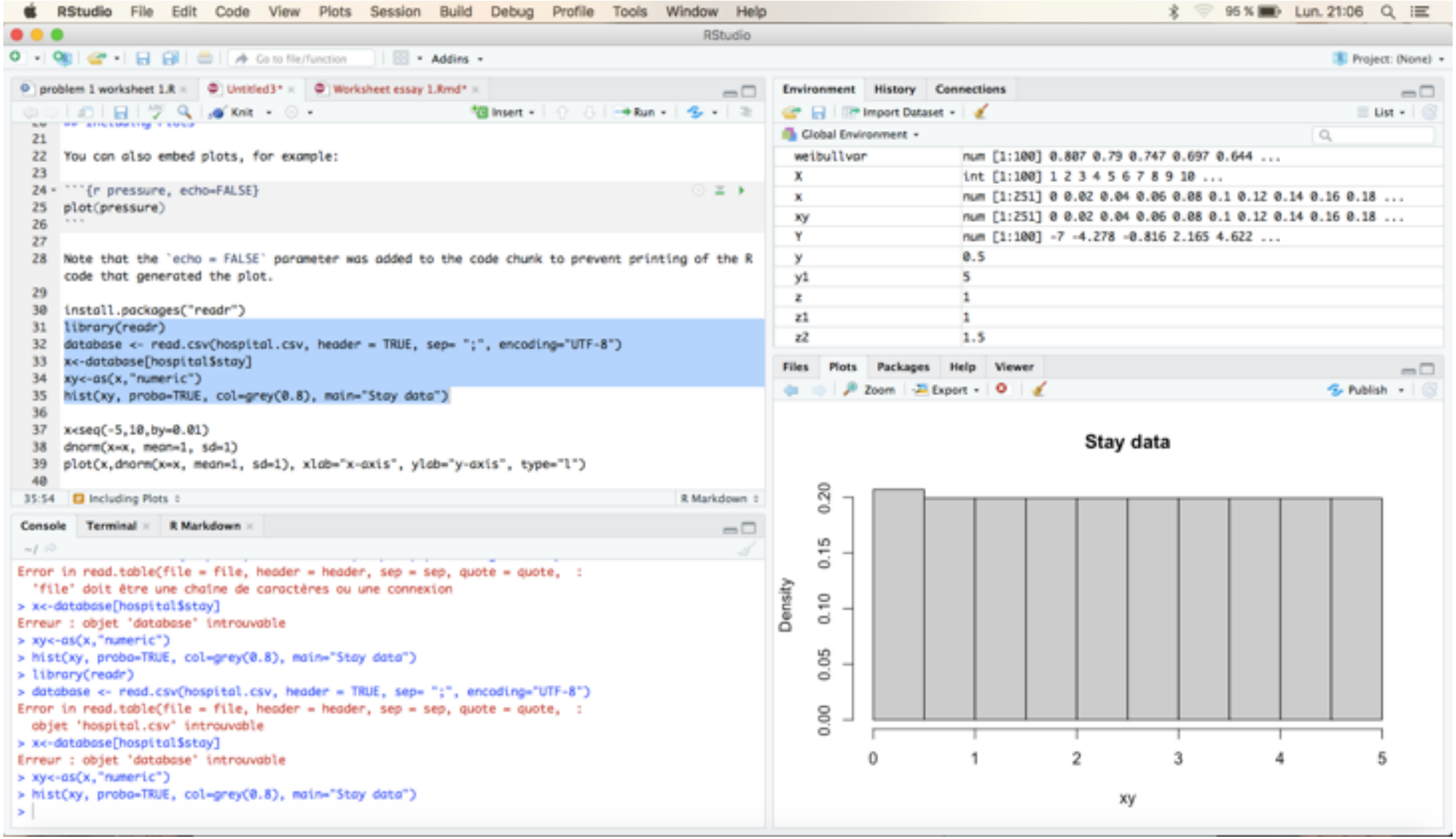
Problem 1 :

a) I have plot the density of the Weibull distribution for several values, using the first program (see on first page). To have all the values we can also use a boucle for like this :

```
y1=c(1,1.5,5)
z1=c(1,1.5,2)
for(j in y1){
        for(k in z1){
                f_x1=sapply(x, function(i) f(i,j,z))
                points(x, f_x1)
}
```

b) I choose to use this values for the different cases :

- for 0<K<1, I choose to plot the density for K=0,5. We can see the graph on the second page of this PDF. We can see that the curve quickly falls on the X axis, the density is not big.
- for K=1, we can see on the third page of this sheet that the curve decrease more slowly, so the density is more present, is bigger.
- for K>1, I choose to plot the density for K=1,5 first, so we can see that the curve show an increase and then a decrease, it is more relevant and more representative of a density. We can work with such a graph. Then, I create the graph for the value K=1,1, and we can conclude that the density is clearly represent for every value when K>1. (see on page 5).



~~~
You can also embed plots, for example:

```{r pressure, echo=FALSE}
plot(pressure)
```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R
code that generated the plot.

install.packages("readr")
library(readr)
database <- read.csv(hospital.csv, header = TRUE, sep= ";", encoding="UTF-8")
x<-database[hospital$stay]
xy<-as(x,"numeric")
hist(xy, proba=TRUE, col=grey(0.8), main="Stay data")

x<seq(-5,10,by=0.01)
dnorm(x=x, mean=1, sd=1)
plot(x,dnorm(x=x, mean=1, sd=1), xlab="x-axis", ylab="y-axis", type="l")
~~~

```
weibullvar   num [1:100] 0.807 0.79 0.747 0.697 0.644 ...
X            int [1:100] 1 2 3 4 5 6 7 8 9 10 ...
x            num [1:251] 0 0.02 0.04 0.06 0.08 0.1 0.12 0.14 0.16 0.18 ...
xy           num [1:251] 0 0.02 0.04 0.06 0.08 0.1 0.12 0.14 0.16 0.18 ...
Y            num [1:100] -7 -4.278 -0.816 2.165 4.622 ...
y            0.5
y1           5
z            1
z1           1
z2           1.5
```

```
Error in read.table(file = file, header = header, sep = sep, quote = quote,  :
  'file' doit être une chaîne de caractères ou une connexion
> x<-database[hospital$stay]
Erreur : objet 'database' introuvable
> xy<-as(x,"numeric")
> hist(xy, proba=TRUE, col=grey(0.8), main="Stay data")
> library(readr)
> database <- read.csv(hospital.csv, header = TRUE, sep= ";", encoding="UTF-8")
Error in read.table(file = file, header = header, sep = sep, quote = quote,  :
  objet 'hospital.csv' introuvable
> x<-database[hospital$stay]
Erreur : objet 'database' introuvable
> xy<-as(x,"numeric")
> hist(xy, proba=TRUE, col=grey(0.8), main="Stay data")
>
```

Problem 2 :

You can see the code on page 7 but I don't succeed to make it work.

a) I think that a Poisson random variable could fit well the given counts because the Poisson random variable described the behavior of the number of events occurring in an interval of fixed time, if these events occur with an average frequency or a known hope and independently of the elapsed time since the previous event. It is exactly the kind of data we have here, with « stay » in « hospital ».

b) We use this command : any(x>0) and the program respond that it is TRUE. So we know that all the values are positive. Here it can be a problem because the Poisson variable is used with rare events. And for our values, we don't have any rare value. Moreover, we don't have any value equal to 0, because the minimum of stay is 1. Actually, the probability p should tends to 0,1, but here it will not. So, to model our data, we will have an illisible frame of all the values which will not be used correctly. We have too much values to use a Poisson random variable.

c) Since the ZTP is a truncated distribution with the truncation stipulated as K>0, we can derive the probability mass function g(K;λ) from a standard Poisson distribution f(K,λ) as :

$g(K,\lambda) = P(X=K|X>0)= f(K;\lambda)/(1-f(0;\lambda))=(\lambda^{\wedge}Ke^{\wedge}(-\lambda))/(K!(1-e^{\wedge}(-\lambda))=(\lambda^{\wedge}K)/((e^{\wedge}\lambda-1)K!)$