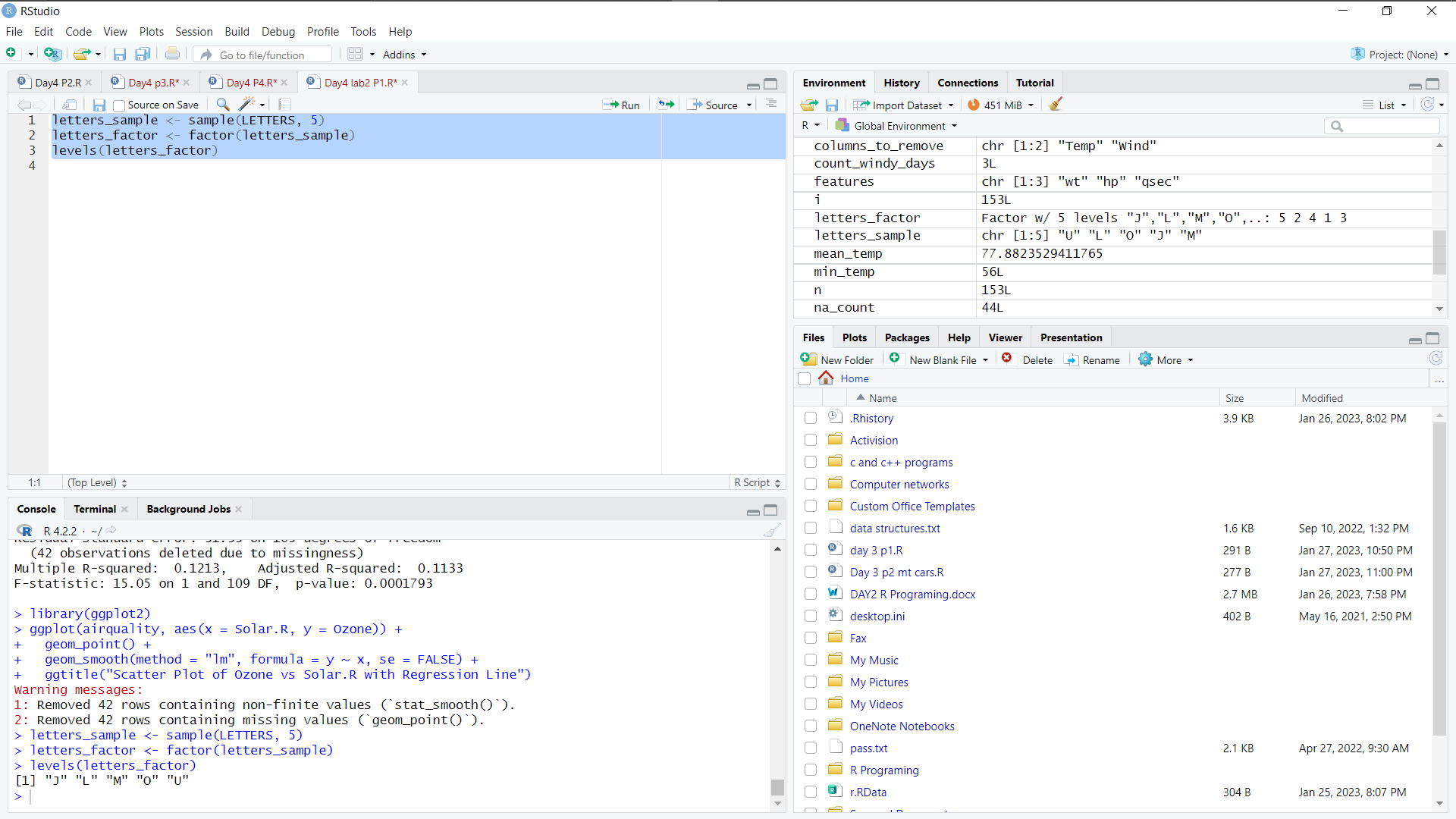
1.(i) Write a R program to extract the five of the levels of factor created from a random sample from the LETTERS (Part of the base R distribution.)

CODE:

letters\_sample <- sample(LETTERS, 5)

letters\_factor <- factor(letters\_sample)

levels(letters\_factor)  
OUTPUT:  


(ii)Write R function to find the range of given vector. Range=Max-Min

Sample input, C<-(9,8,7,6,5,4,3,2,1),

output=8

CODE:

find\_range <- function(x){

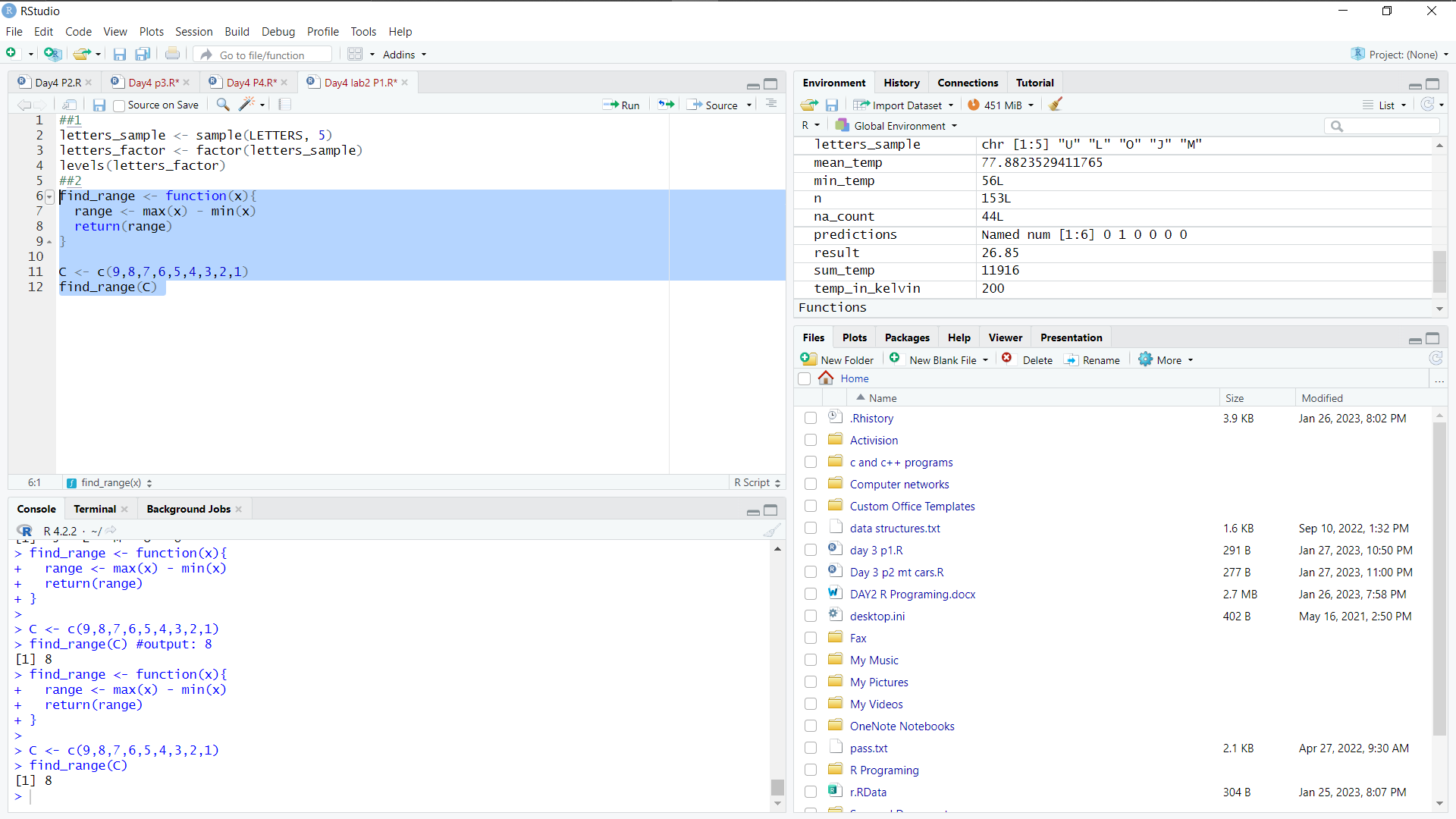
range <- max(x) - min(x)

return(range)

}

C <- c(9,8,7,6,5,4,3,2,1)

find\_range(C)  
OUTPUT:



(iii)Wirte the R function to find the number of vowels in given string Sample input c<- "matrix", output <-2

CODE:  
find\_vowels <- function(str){

vowels <- c("a", "e", "i", "o", "u", "A", "E", "I", "O", "U")

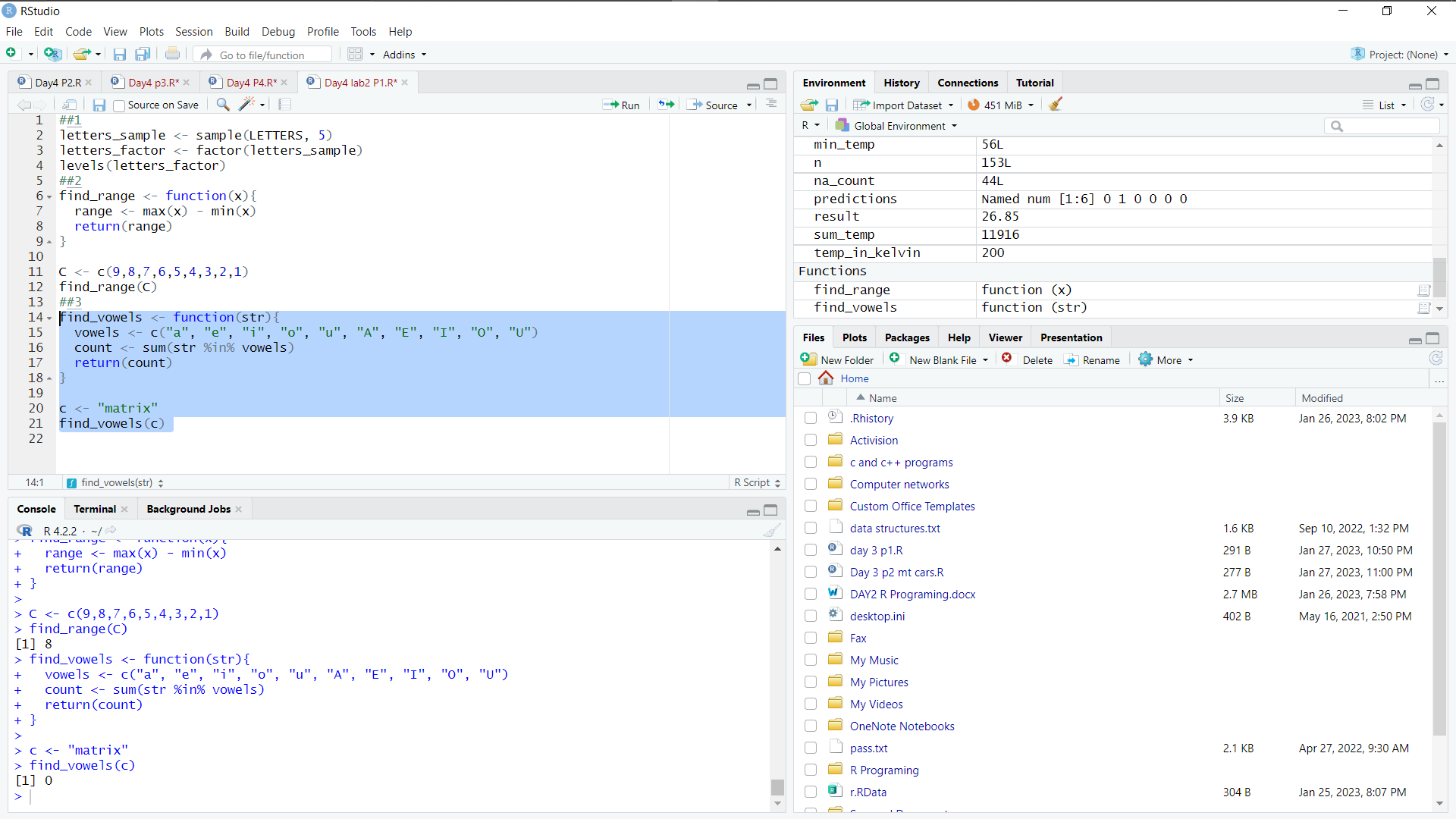
count <- sum(str %in% vowels)

return(count)

}

c <- "matrix"

find\_vowels(c)

OUTPUT:  


2.Load inbuild dataset"ChickWeight" in R

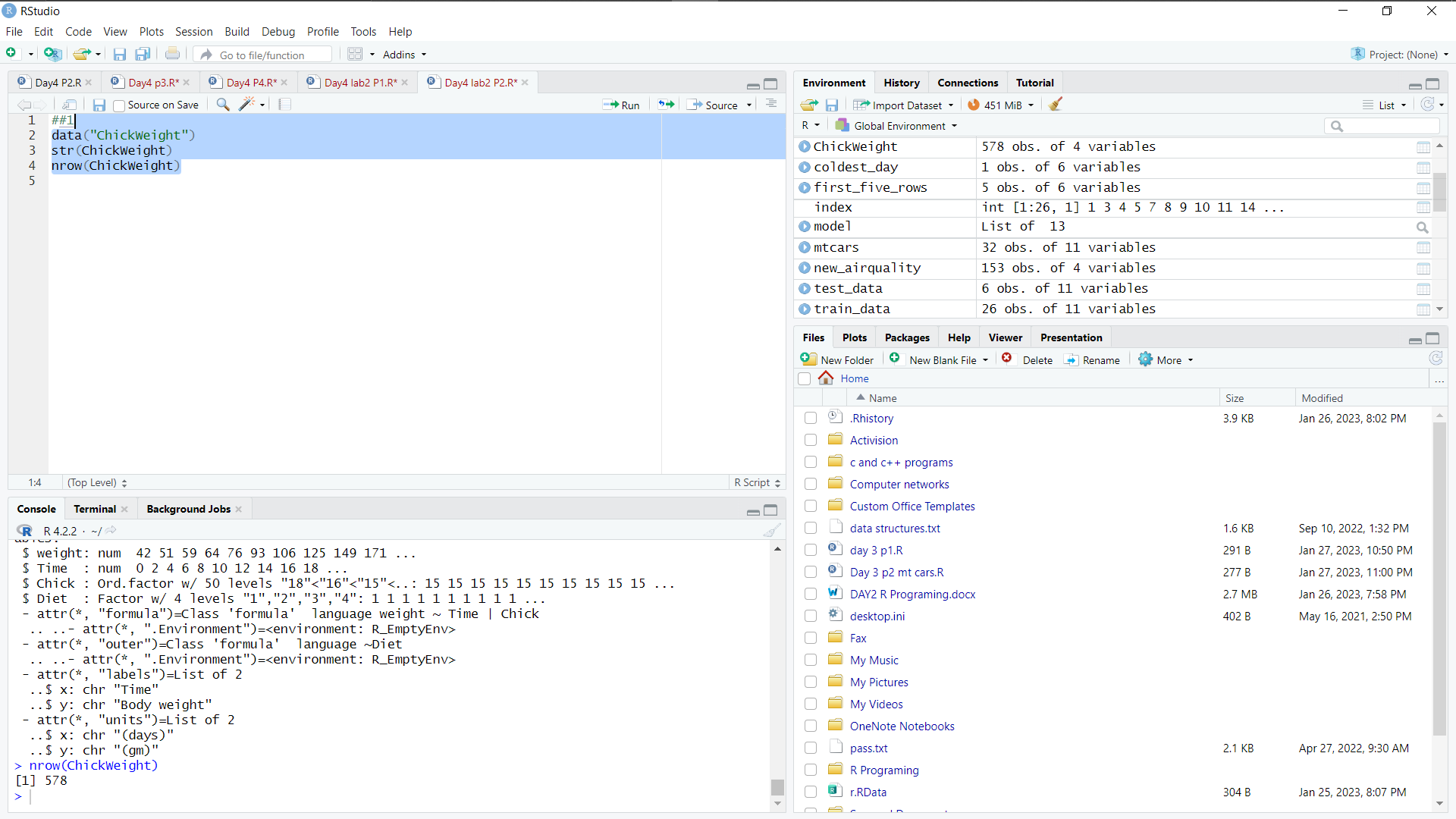
(i) Explore the summary of Data set, like number of Features and its type. Fins the number of records for each features

CODE:

data("ChickWeight")

str(ChickWeight)

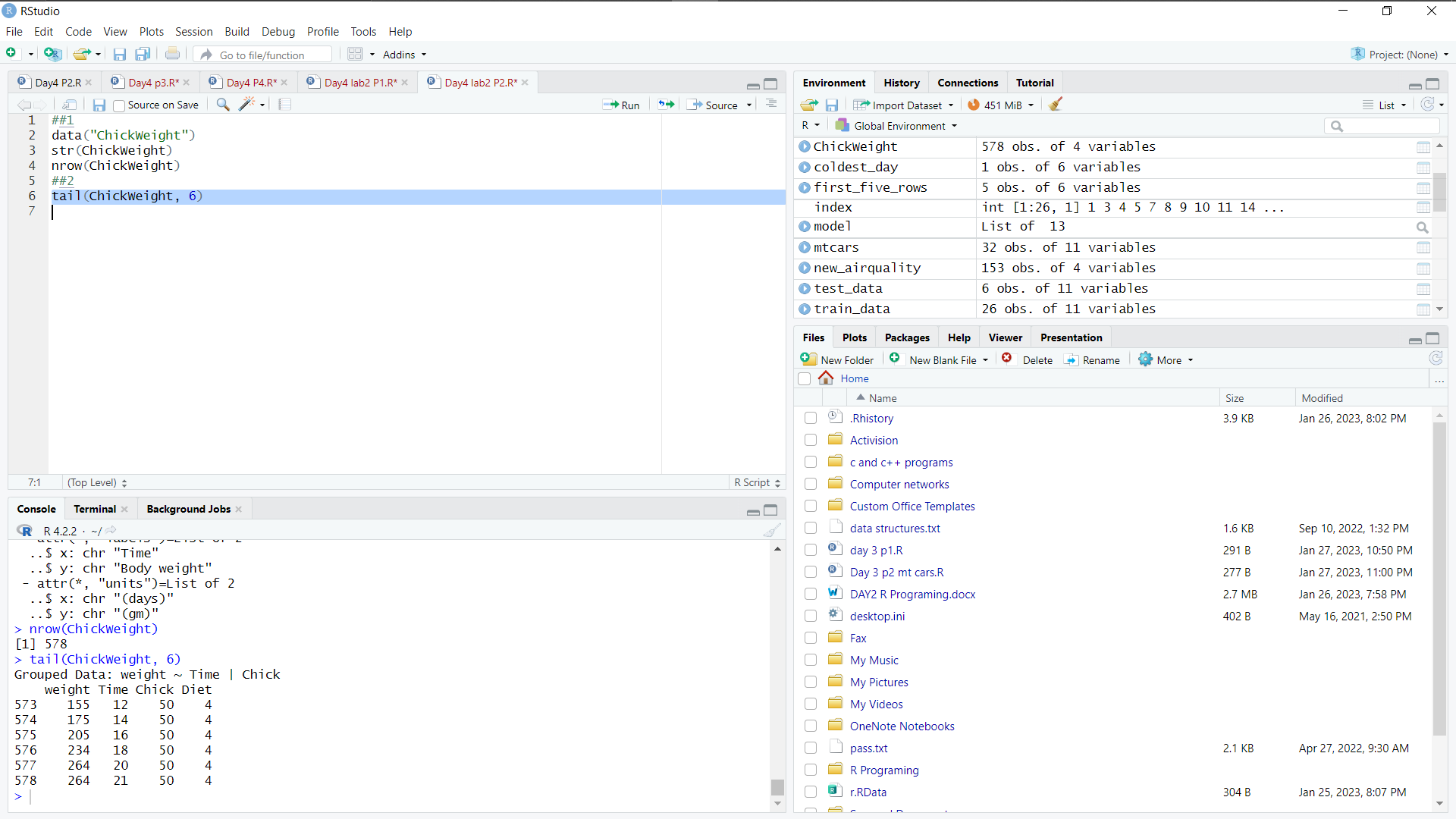
nrow(ChickWeight)   
OUTPUT:



(ii)Extract last 6 records of dataset

CODE:

tail(ChickWeight, 6)  
OUTPUT:



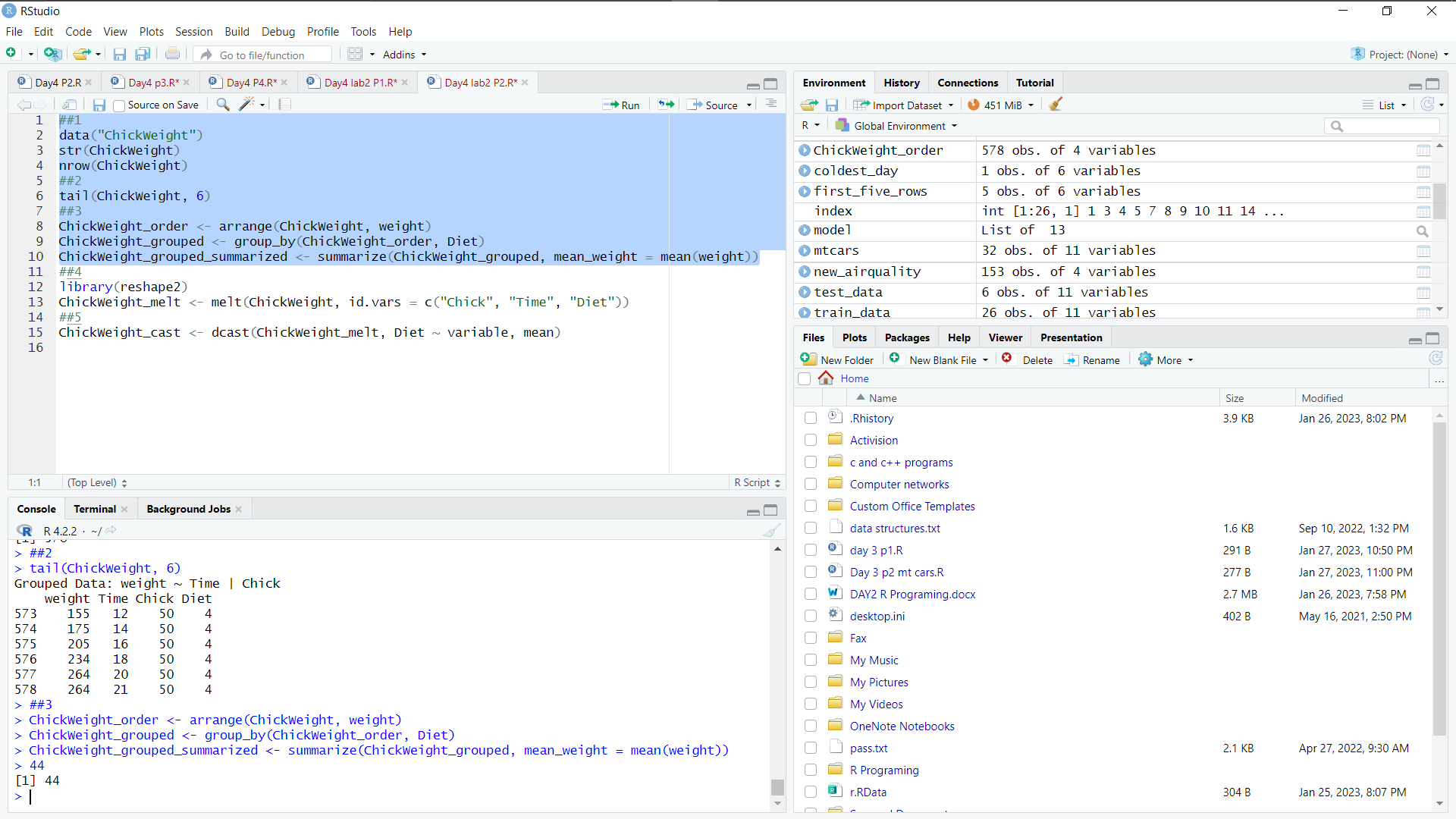
(iii) order the data frame, in ascending order by feature name "weight" grouped by feature "diet"

CODE:

ChickWeight\_order <- arrange(ChickWeight, weight)

ChickWeight\_grouped <- group\_by(ChickWeight\_order, Diet)

ChickWeight\_grouped\_summarized <- summarize(ChickWeight\_grouped, mean\_weight = mean(weight))  
OUTPUT:

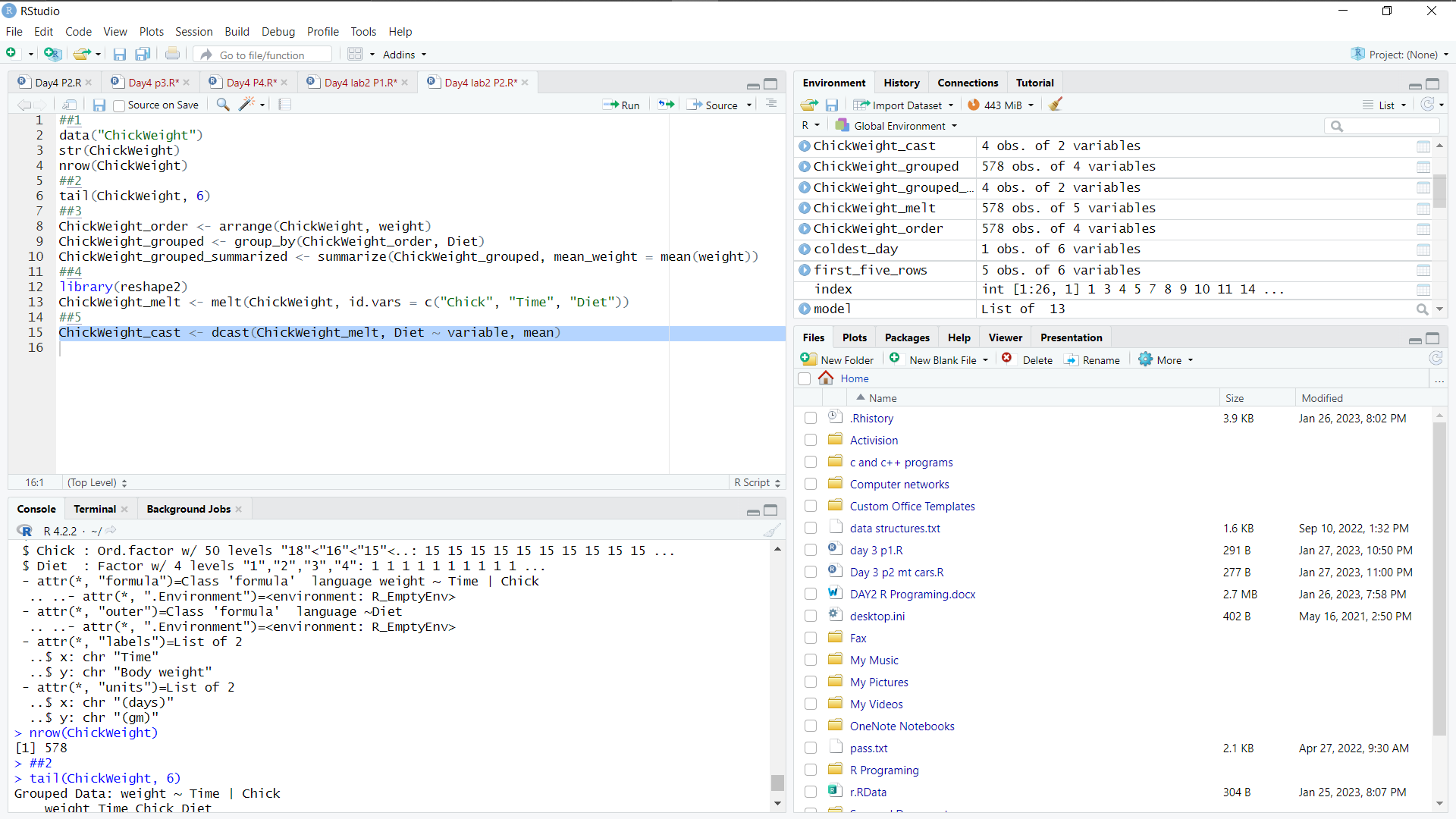


(iv)Perform melting function based on "Chick", "Time","Diet" features as ID variables

CODE:

library(reshape2)

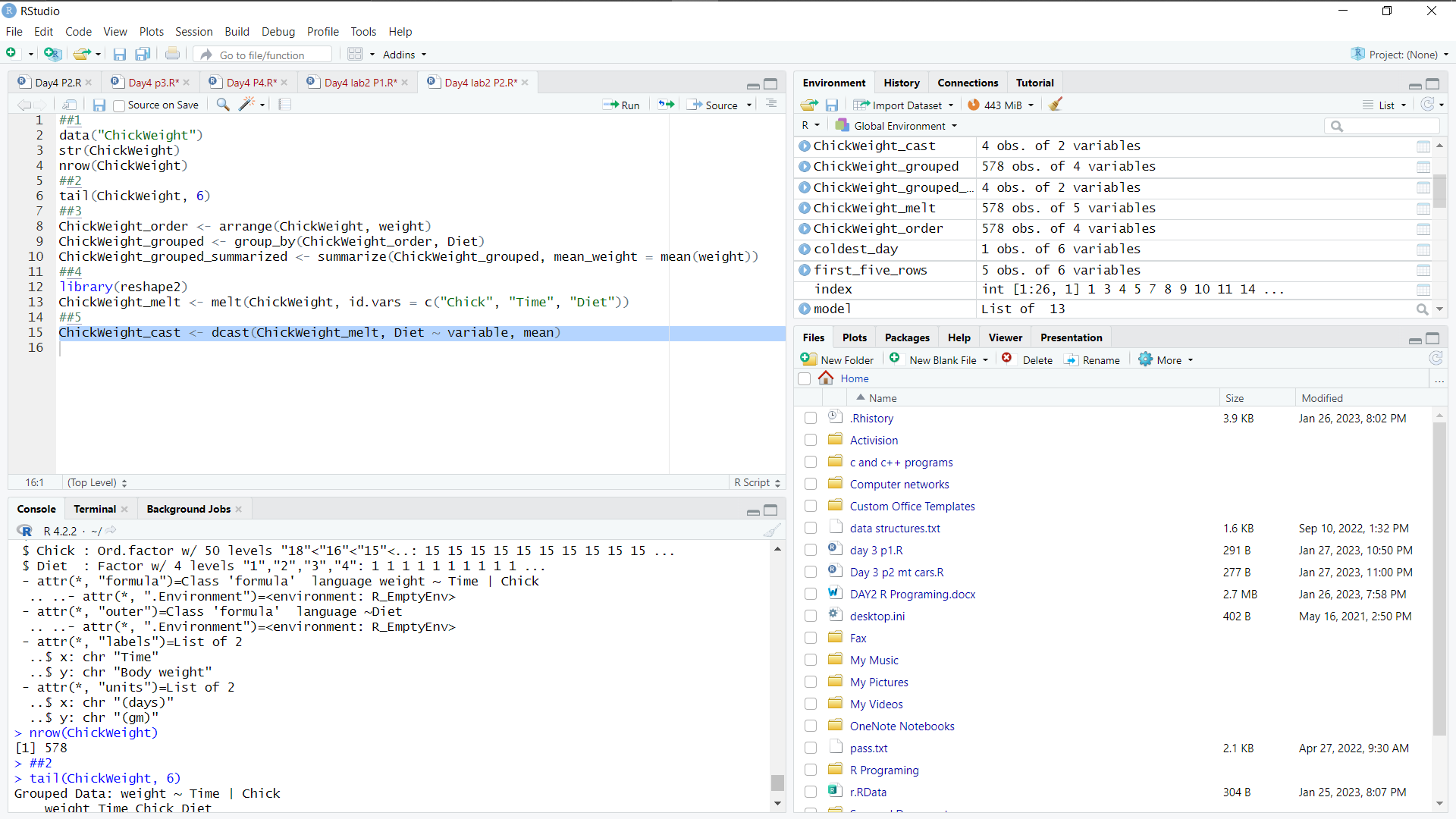
ChickWeight\_melt <- melt(ChickWeight, id.vars = c("Chick", "Time", "Diet"))  
OUTPUT:



(v)Perform cast function to display the mean value of weight grouped by Diet

CODE:

ChickWeight\_cast <- dcast(ChickWeight\_melt, Diet ~ variable, mean)  
OUTPUT:

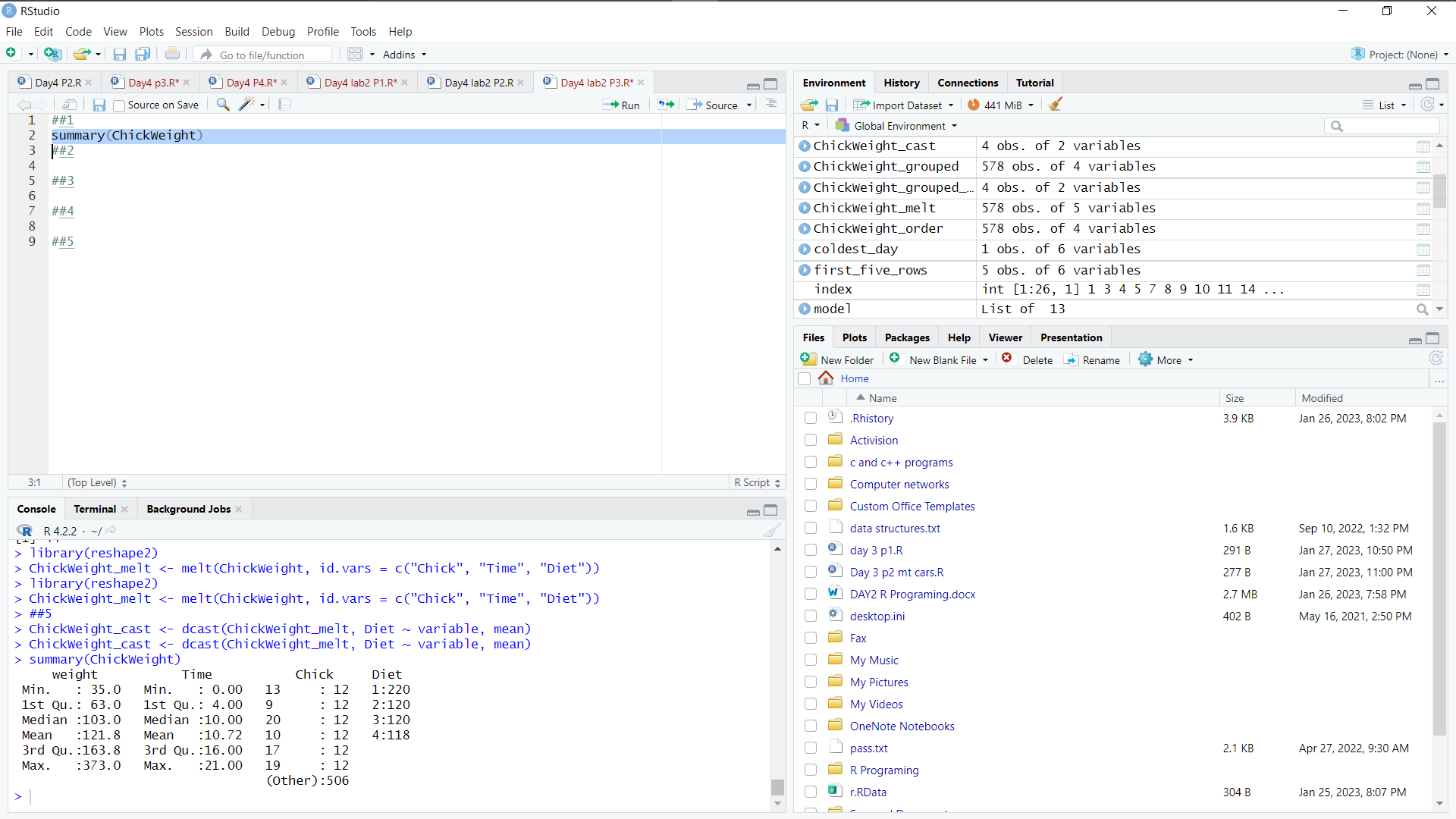


3.(1)Get the Statistical Summary of "ChickWeight" dataset

CODE:

summary(ChickWeight)

OUTPUT:



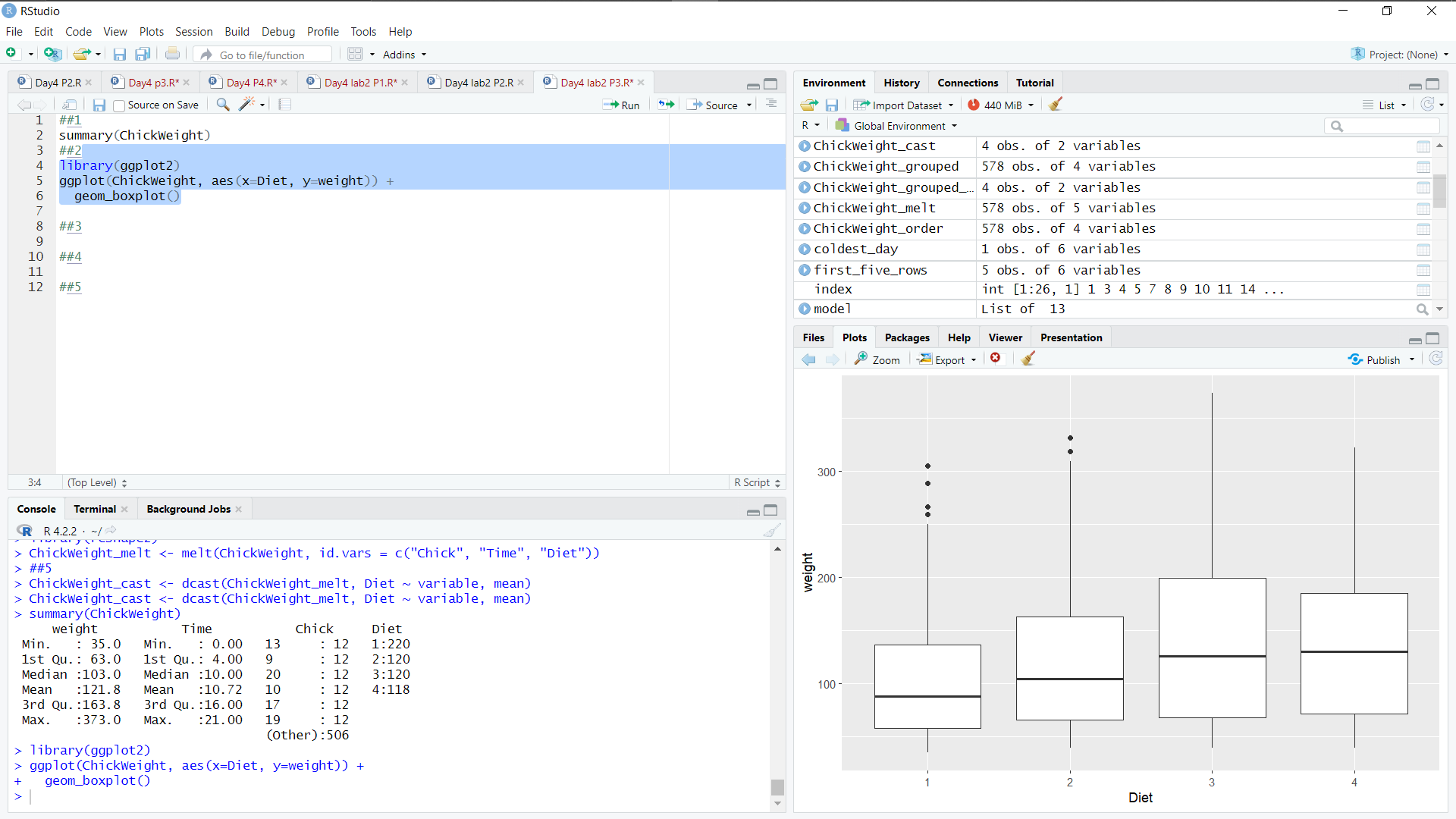
(ii) Create Box plot for "weight" grouped by "Diet"

CODE:

library(ggplot2)

ggplot(ChickWeight, aes(x=Diet, y=weight)) +

geom\_boxplot()  
OUTPUT:

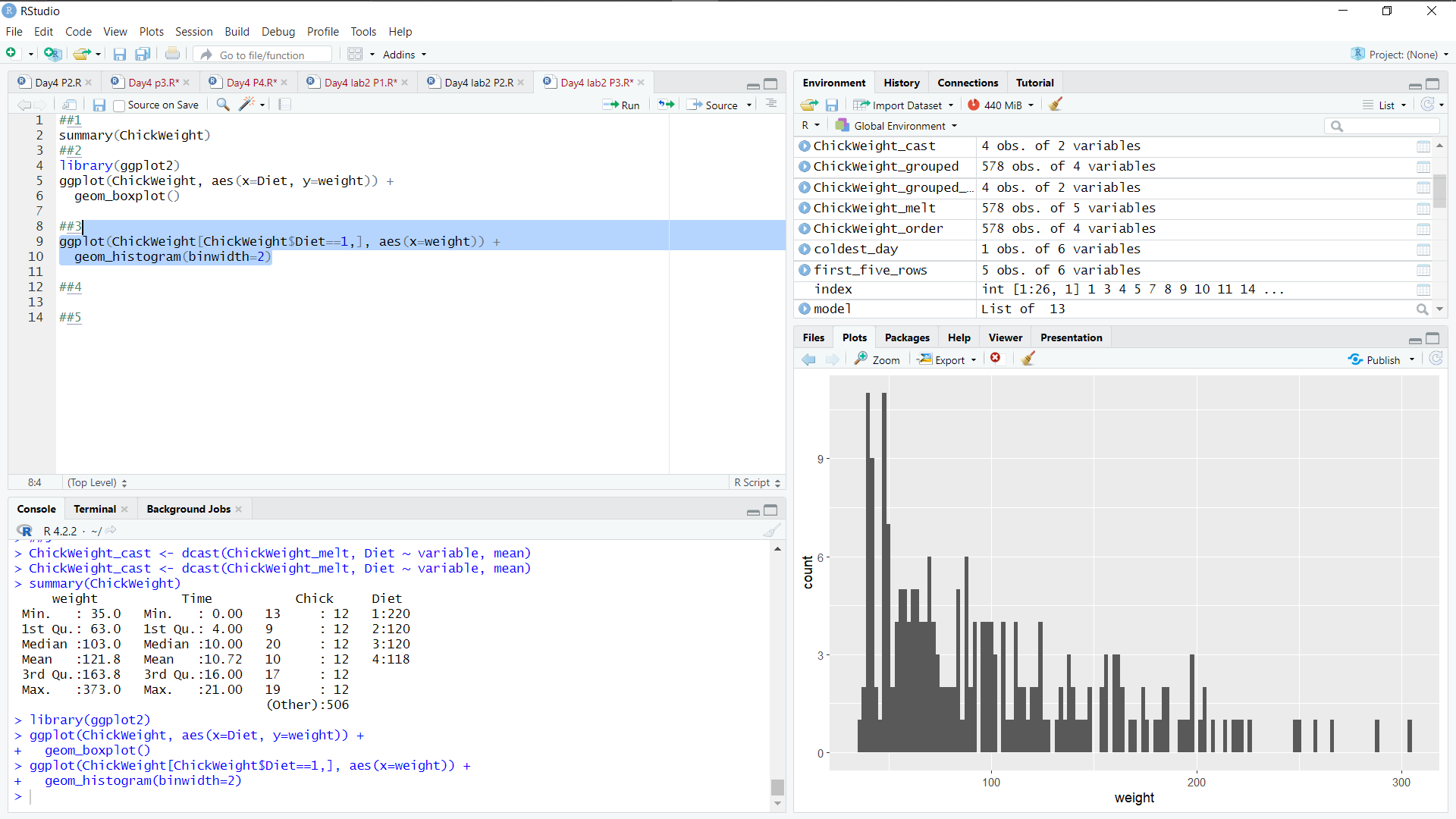


(iii)Create a Histogram for "Weight" features belong to Diet-1 category

CODE:

ggplot(ChickWeight[ChickWeight$Diet==1,], aes(x=weight)) +

geom\_histogram(binwidth=2)  
OUTPUT:

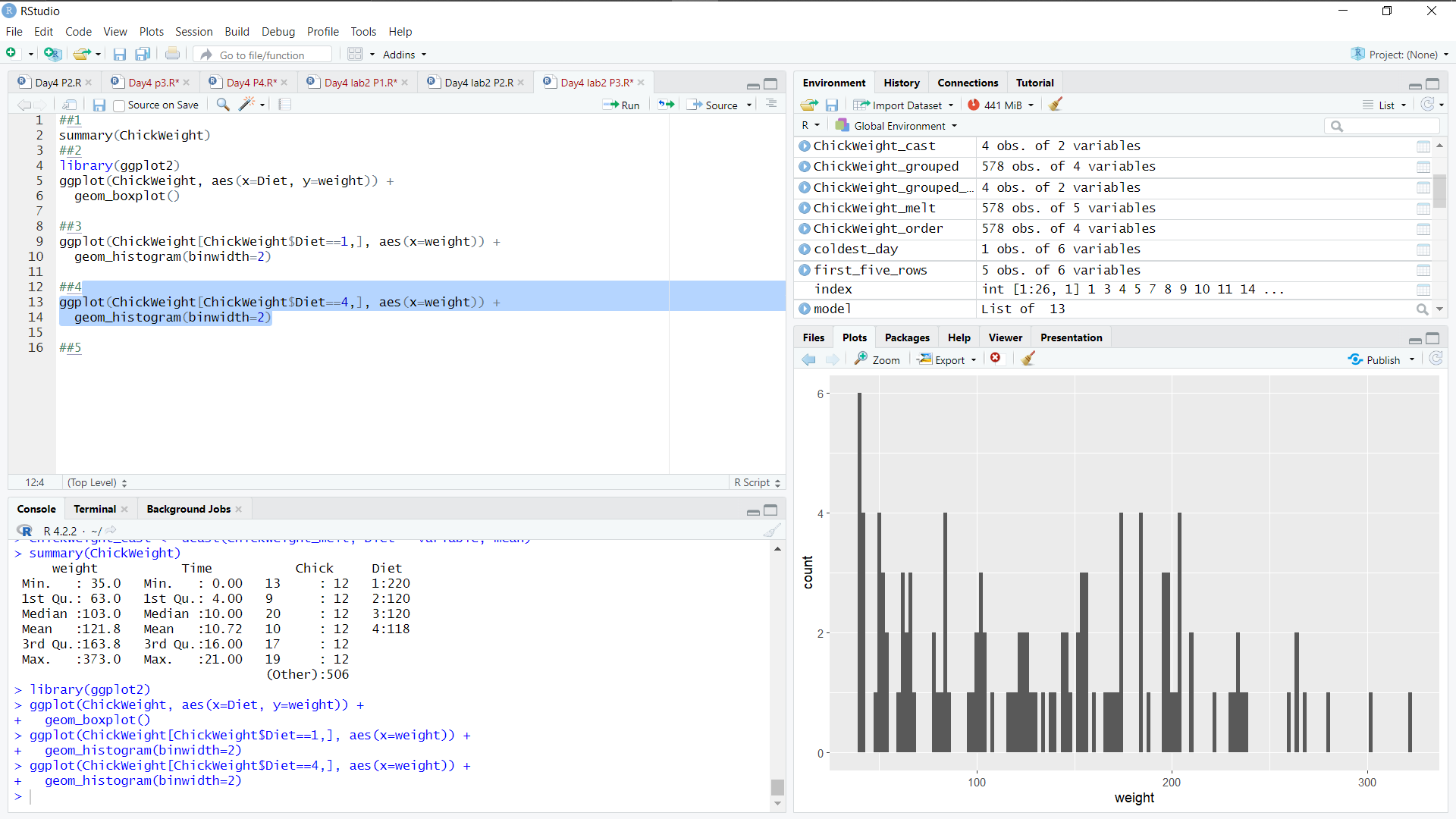


(iv) Create a Histogram for "Weight" features belong to Dist- 4 category

CODE:

ggplot(ChickWeight[ChickWeight$Diet==4,], aes(x=weight)) +

geom\_histogram(binwidth=2)  
OUTPUT:



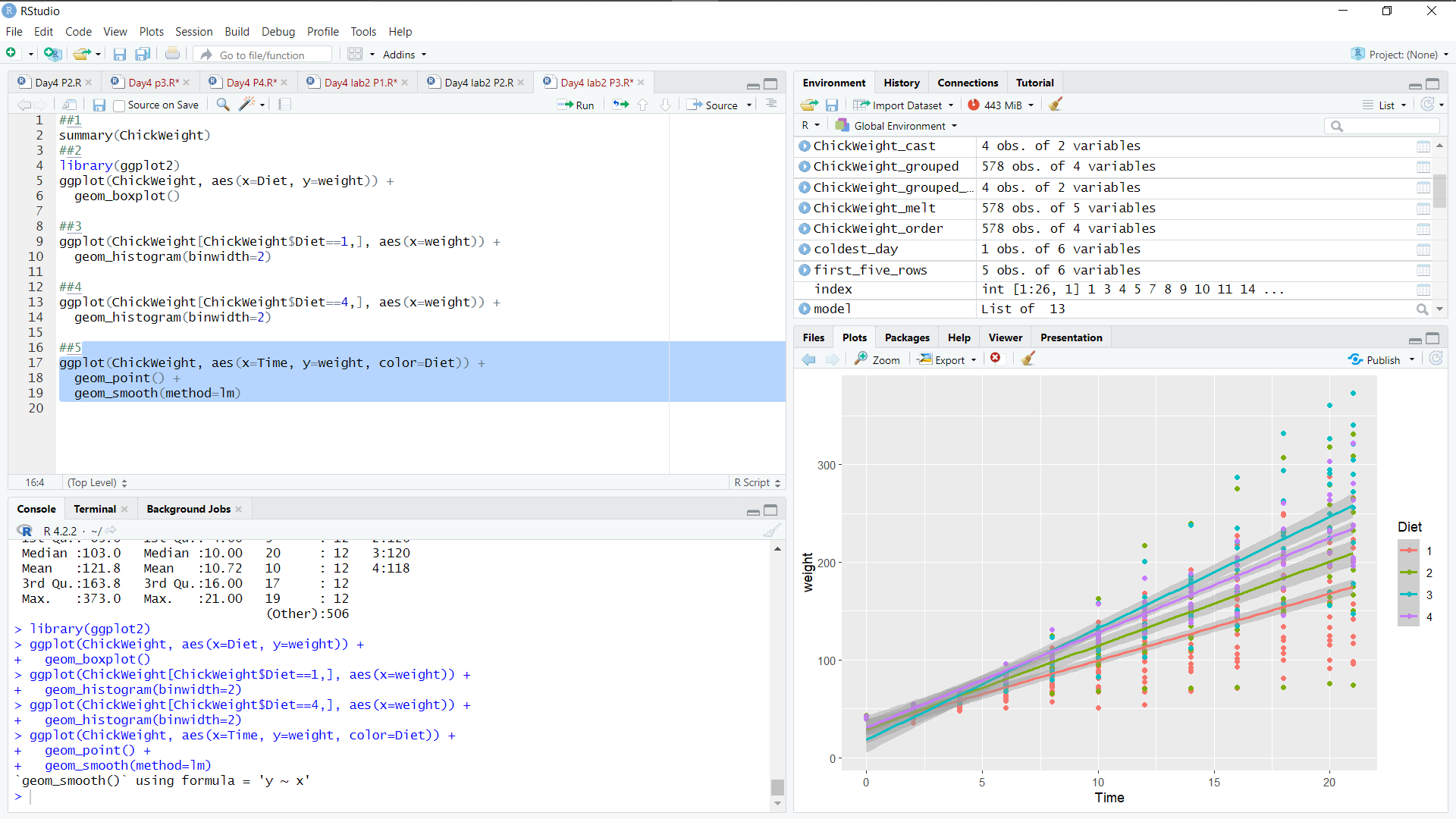
(v) Create Scatter plot for weight vs Time grouped by Diet

CODE:

ggplot(ChickWeight, aes(x=Time, y=weight, color=Diet)) +

geom\_point() +

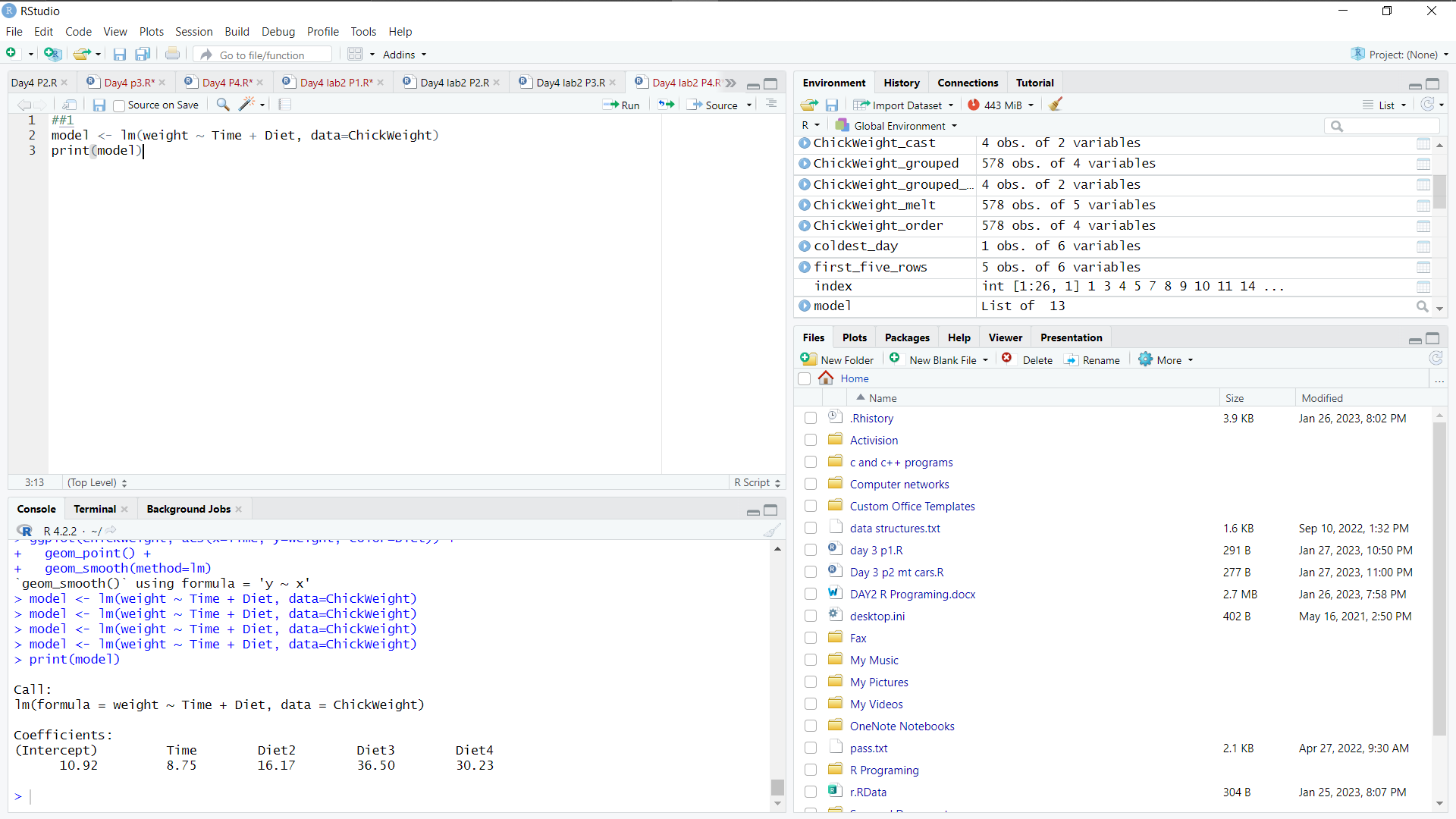
geom\_smooth(method=lm)  
OUTPUT:



4.(1) Create multiregression model to find a weight of the chicken, by "Time" and "Diet" as as predictor variables

CODE:

model <- lm(weight ~ Time + Diet, data=ChickWeight)  
OUTPUT:



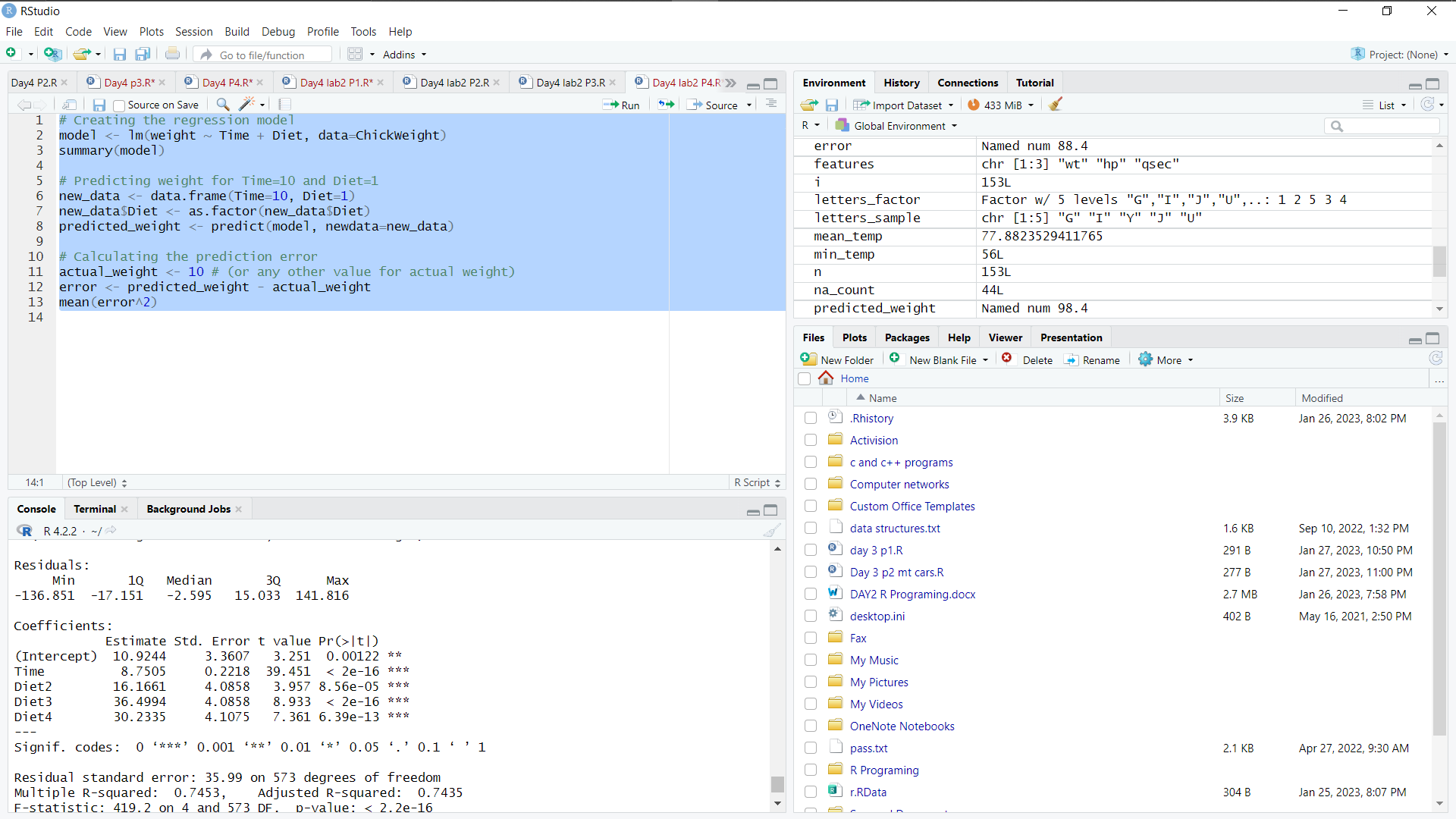
(ii) Predict weight for Time 10 and Diet=1

CODE:

new\_data <- data.frame(Time=10, Diet=1)

new\_data$Diet <- as.factor(new\_data$Diet)

predicted\_weight <- predict(model, newdata=new\_data)  
OUTPUT:



(iii)Find the error in model for same

CODE:

actual\_weight <- 10 # (or any other value for actual weight)

error <- predicted\_weight - actual\_weight

mean(error^2)  
OUTPUT:

