Set-V

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.)

(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

(iii)Wirte the R function to find the number of vowels in given string

Sample input c<- “matrix”, output<-2

**Input**:

i.)set.seed(123) # Set a seed for reproducibility

# Generate a random sample from LETTERS

sample\_letters <- sample(LETTERS, 20, replace = TRUE)

# Convert the sample to a factor

sample\_factor <- factor(sample\_letters)

# Extract five levels from the factor

five\_levels <- levels(sample\_factor)[1:5]

# Print the five levels

print(five\_levels)

ii.)find\_range <- function(vector) {

range <- max(vector) - min(vector)

return(range)

}

# Example usage

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

result <- find\_range(C)

print(result)

iii.)count\_vowels <- function(string) {

vowels <- c("a", "e", "i", "o", "u")

count <- sum(strsplit(tolower(string), "")[[1]] %in% vowels)

return(count)

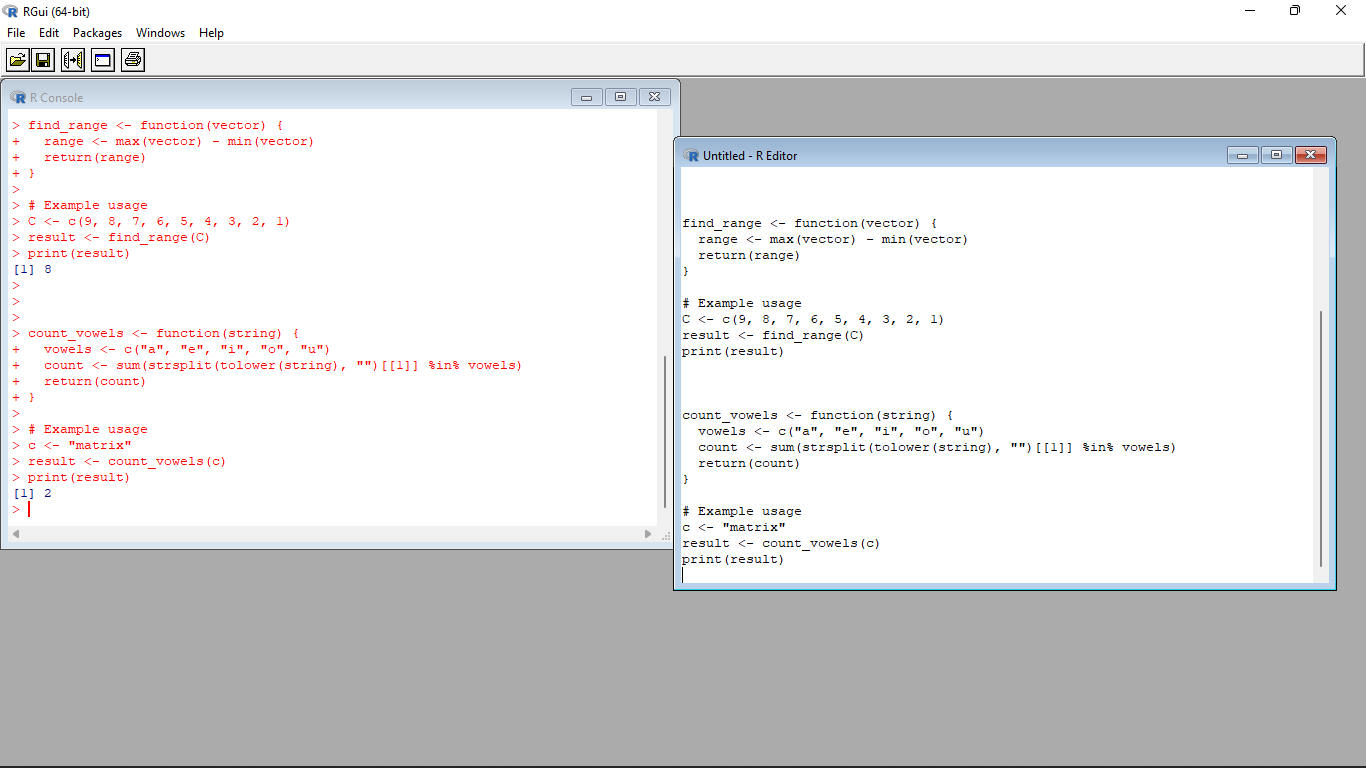
}

# Example usage

c <- "matrix"

result <- count\_vowels(c)

print(result)



2.Load inbuild dataset “ChickWeight” in R

(i) Explore the summary of Data set, like number of Features and its type. Finds the number of records

for each features

(ii)Extract last 6 records of dataset

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

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

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

**Program** :

# (i) Load and explore the dataset

data(ChickWeight)

summary(ChickWeight) # Summary of the dataset

str(ChickWeight) # Information about features and their types

table(ChickWeight$Time) # Number of records for each "Time" feature

table(ChickWeight$Chick) # Number of records for each "Chick" feature

# (ii) Extract the last 6 records of the dataset

last\_six\_records <- tail(ChickWeight, 6)

# (iii) Order the data frame in ascending order by "weight" grouped by "diet"

ordered\_df <- ChickWeight[order(ChickWeight$weight), ]

ordered\_df <- ordered\_df[order(ordered\_df$diet), ]

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

library(reshape2)

melted\_df <- melt(ChickWeight, id.vars = c("Chick", "Time", "Diet"))

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

cast\_df <- dcast(melted\_df, Diet ~ variable, mean(value))

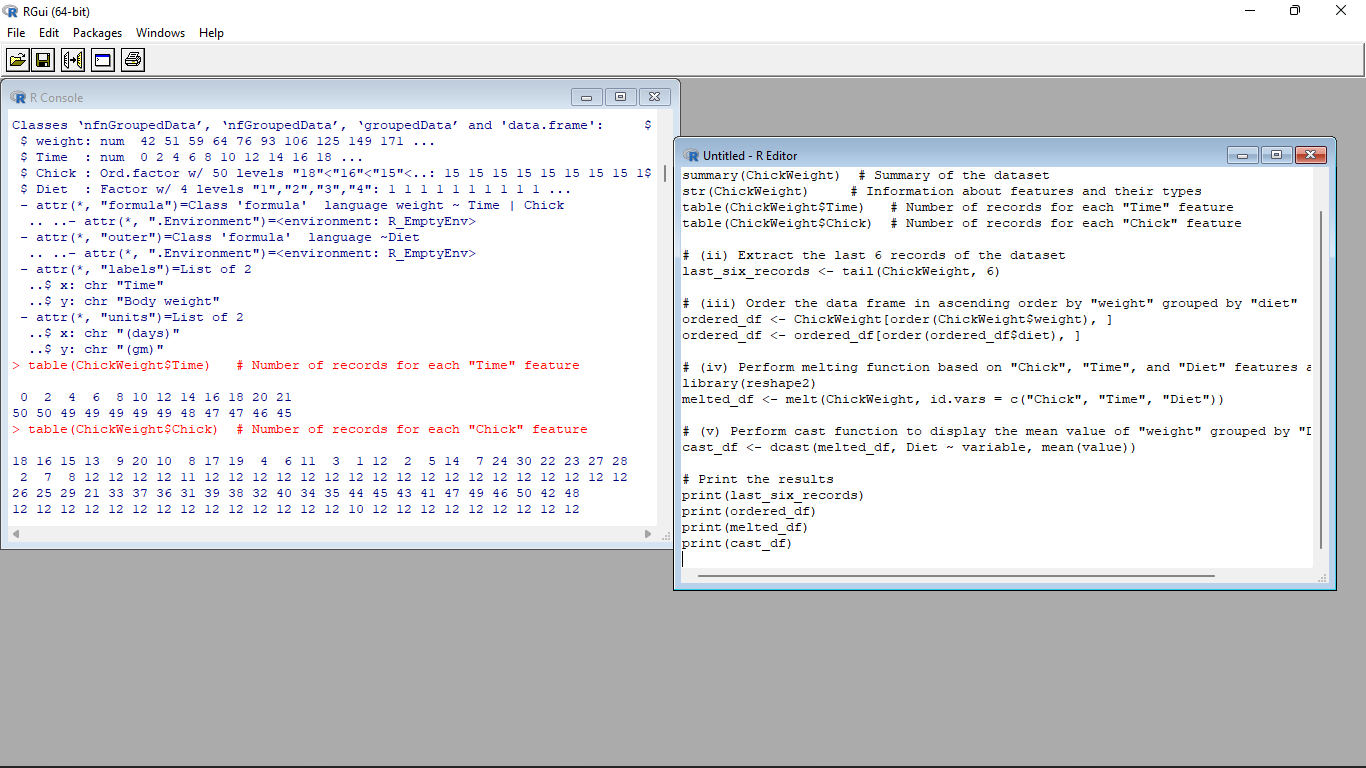
# Print the results

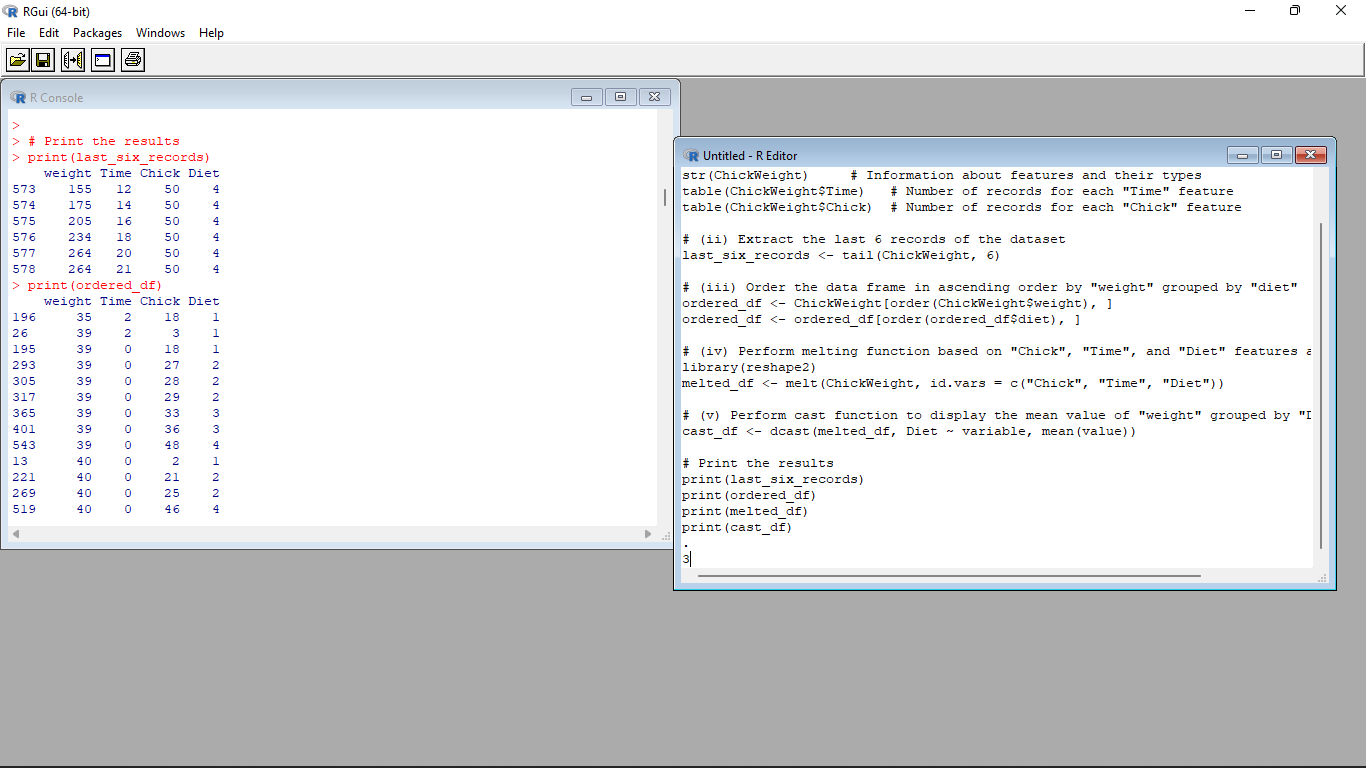
print(last\_six\_records)

print(ordered\_df)

print(melted\_df)

print(cast\_df)





3.(i)Get the Statistical Summary of “ChickWeight” dataset

(ii)Create Box plot for “weight” grouped by “Diet”

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

(iv) Create a Histogram for “Weight” features belong to Diet- 4 category

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

**Input** :

# (i) Get the Statistical Summary of the "ChickWeight" dataset

summary(ChickWeight)

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

boxplot(weight ~ Diet, data = ChickWeight, xlab = "Diet", ylab = "Weight", main = "Weight Distribution by Diet")

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

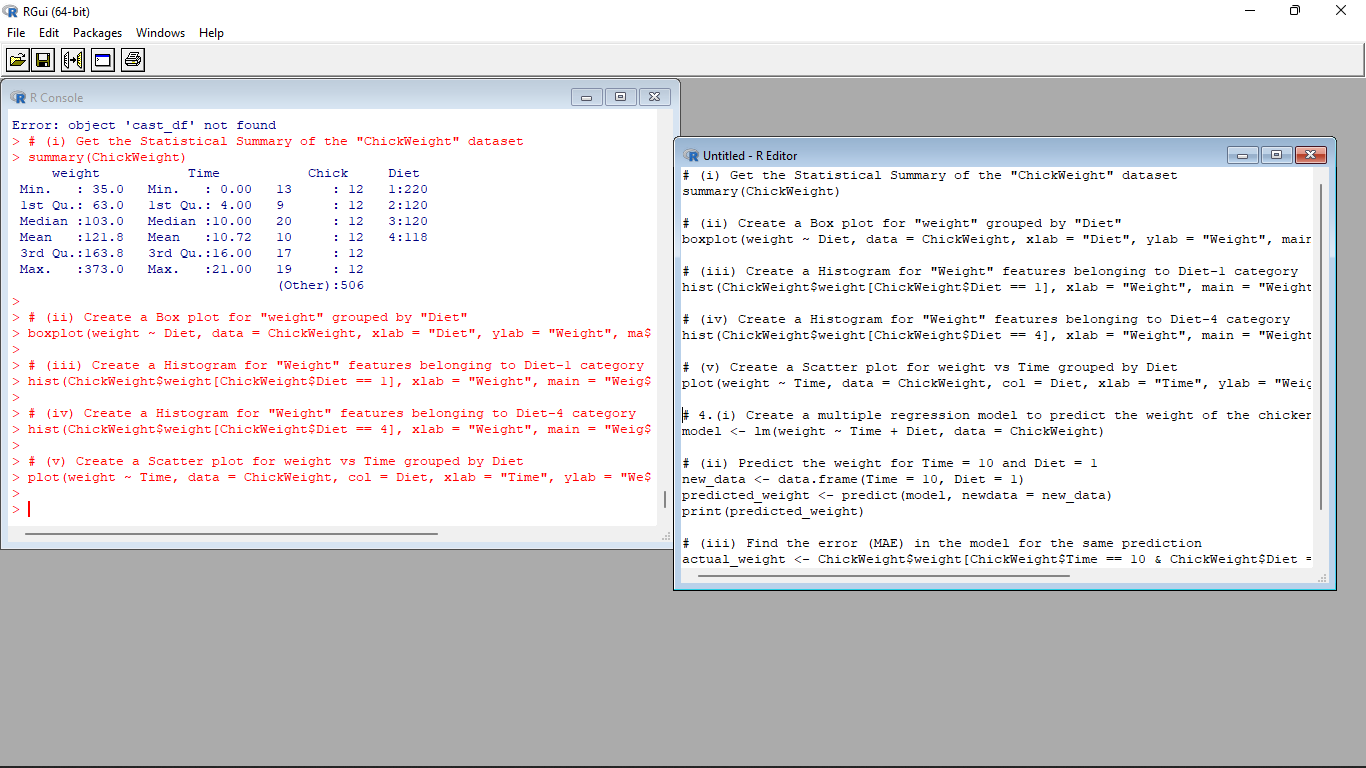
hist(ChickWeight$weight[ChickWeight$Diet == 1], xlab = "Weight", main = "Weight Distribution for Diet-1")

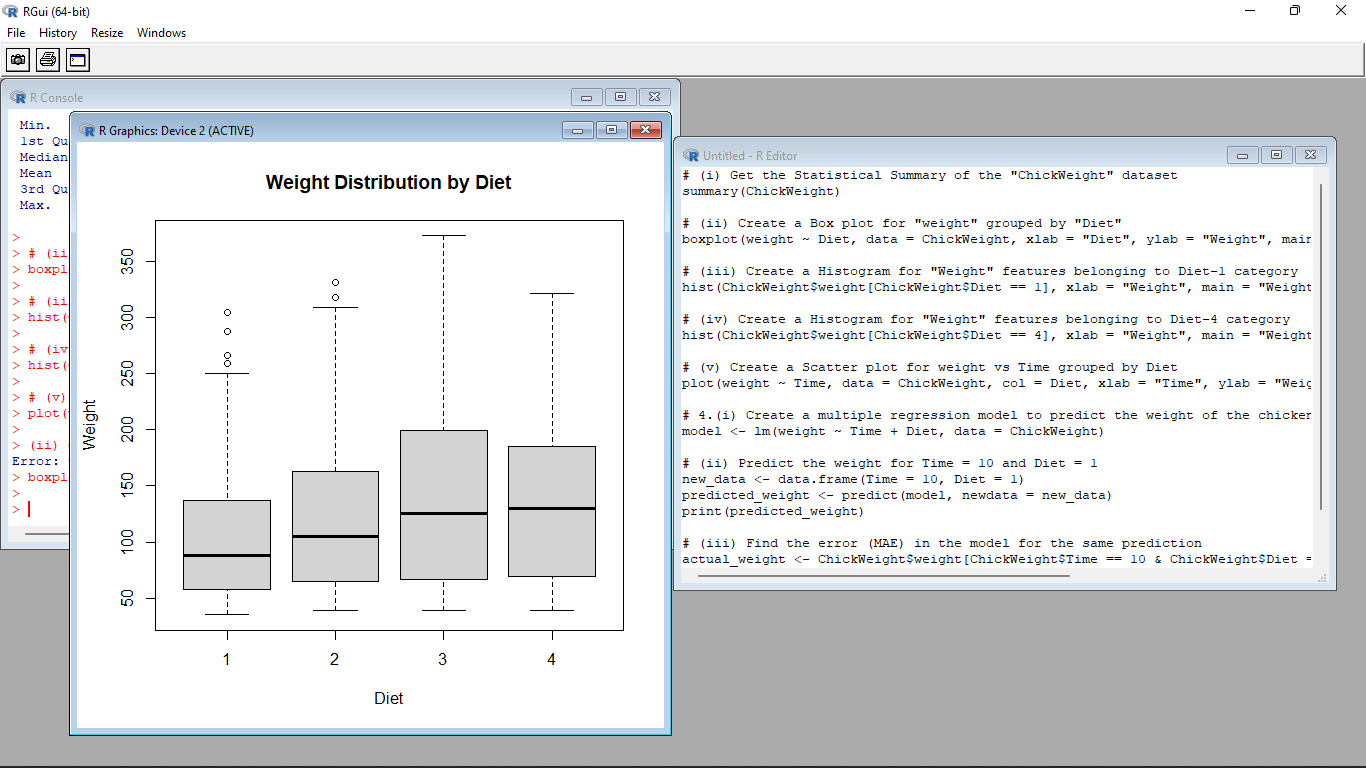
# (iv) Create a Histogram for "Weight" features belonging to Diet-4 category

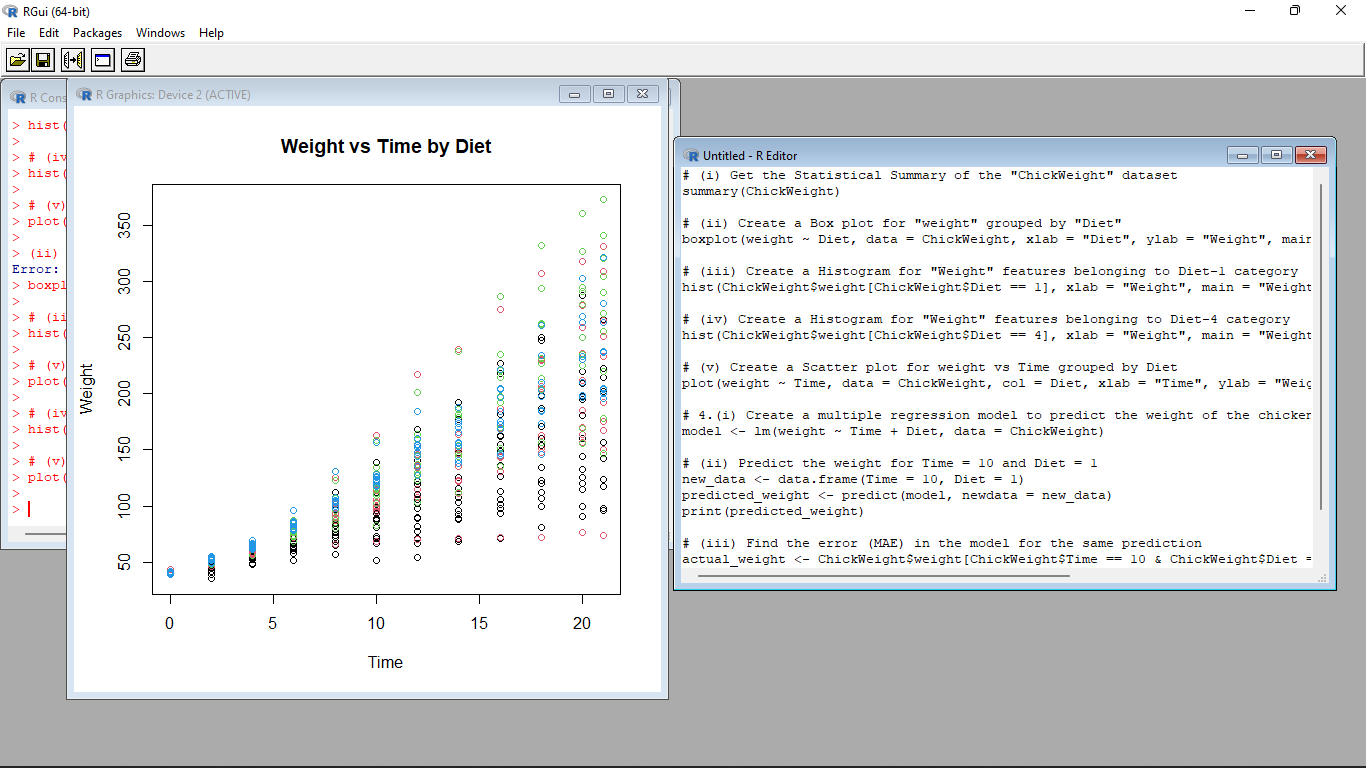
hist(ChickWeight$weight[ChickWeight$Diet == 4], xlab = "Weight", main = "Weight Distribution for Diet-4")

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

plot(weight ~ Time, data = ChickWeight, col = Diet, xlab = "Time", ylab = "Weight", main = "Weight vs Time by Diet")







4.(i) Create multi regression model to find a weight of the chicken , by “Time” and “Diet” as as predictor

variables

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

(iii)Find the error(MAE) in model for same

**Input** :

time <- c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)

diet <- c(1, 1, 1, 1, 1, 2, 2, 2, 2, 2)

weight <- c(2.1, 2.4, 2.6, 2.9, 3.2, 3.5, 3.8, 4.1, 4.4, 4.7)

data <- data.frame(time, diet, weight)

model <- lm(weight ~ time + diet, data=data)

new\_data <- data.frame(time=10, diet=1)

prediction <- predict(model, newdata=new\_data)

cat("Predicted weight: ", prediction, "\n")

actual <- 4.7

error <- actual - prediction

cat("Error: ", error, "\n")

