**Data science**

**Name:** Pallavi Hemant Khandare

**Roll no:** 2095

**Subject:** data science

**Class:** TYCS

|  |  |  |
| --- | --- | --- |
| **INDEX** | | |
| **NO** | **TITLE** | **DATE** |
| 1 | Practical of data collection, data curation and management for unstructured data (NoSQL) | 24/1/23 |
| 2 | Practical of data collection, data curation and management for large scale data system (MongoDB) | 24/1/23 |
| 3 | Practical of simple/multiple linear regression | 24/1/23 |
| 4 | Practical of time-series forecasting | 24/1/23 |
| 5 | Practical of time series analysis | 24/1/23 |
| 6 | Practical of k-means clustering | 24/1/23 |
| 7 | Practical of decision tree | 24/1/23 |
| 8 | Practical of hypothesis testing | 24/1/23 |
| 9 | Practical of analysis of variance | 24/1/23 |

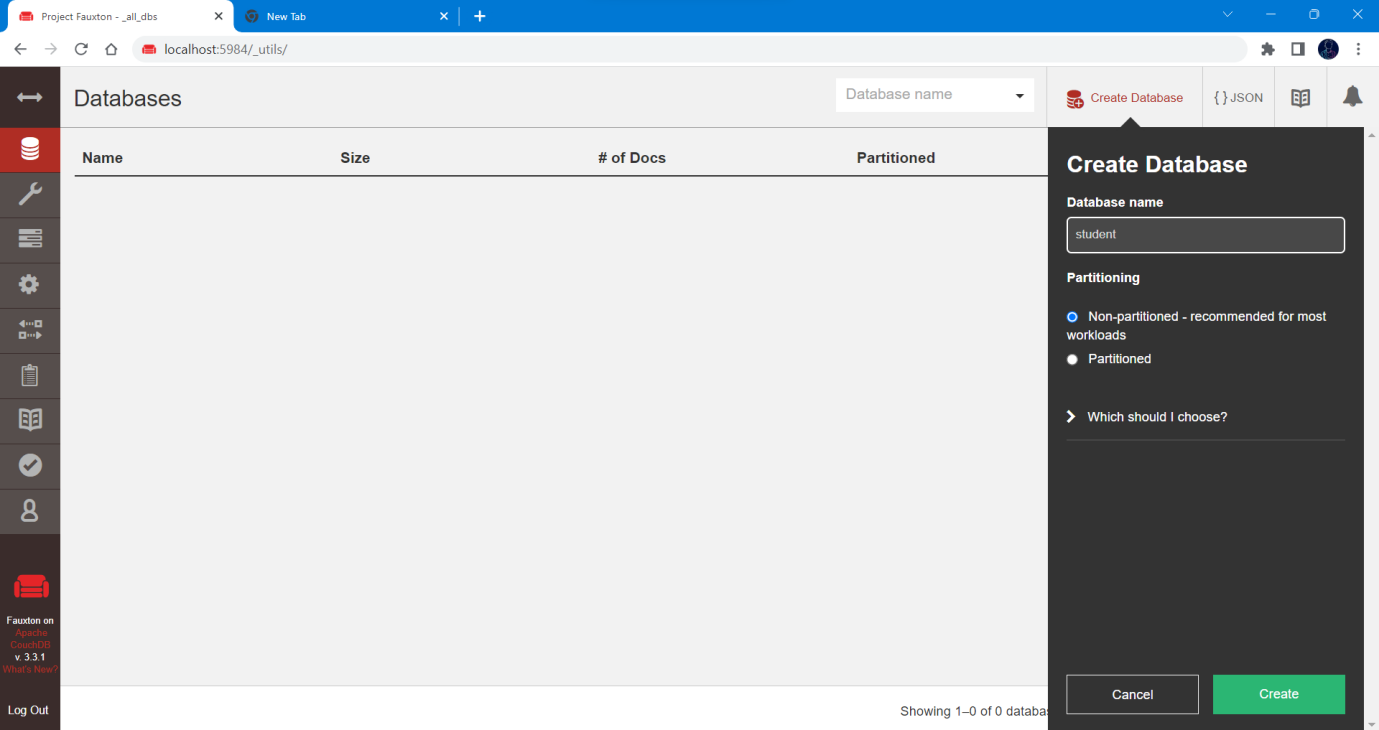
**Practical 1**

**Aim:** Practical of Data collection, Data curation and management for Unstructured data(NoSQL)

**Steps:**

Step-1: Install CouchDB. After installation go to your browser and type below url: <http://localhost:5984/_utils/> and login with username and password.

Step-2: After login go to Databases and click on “Create Database” and give name to your database(name should be in lowercase) and click on create.

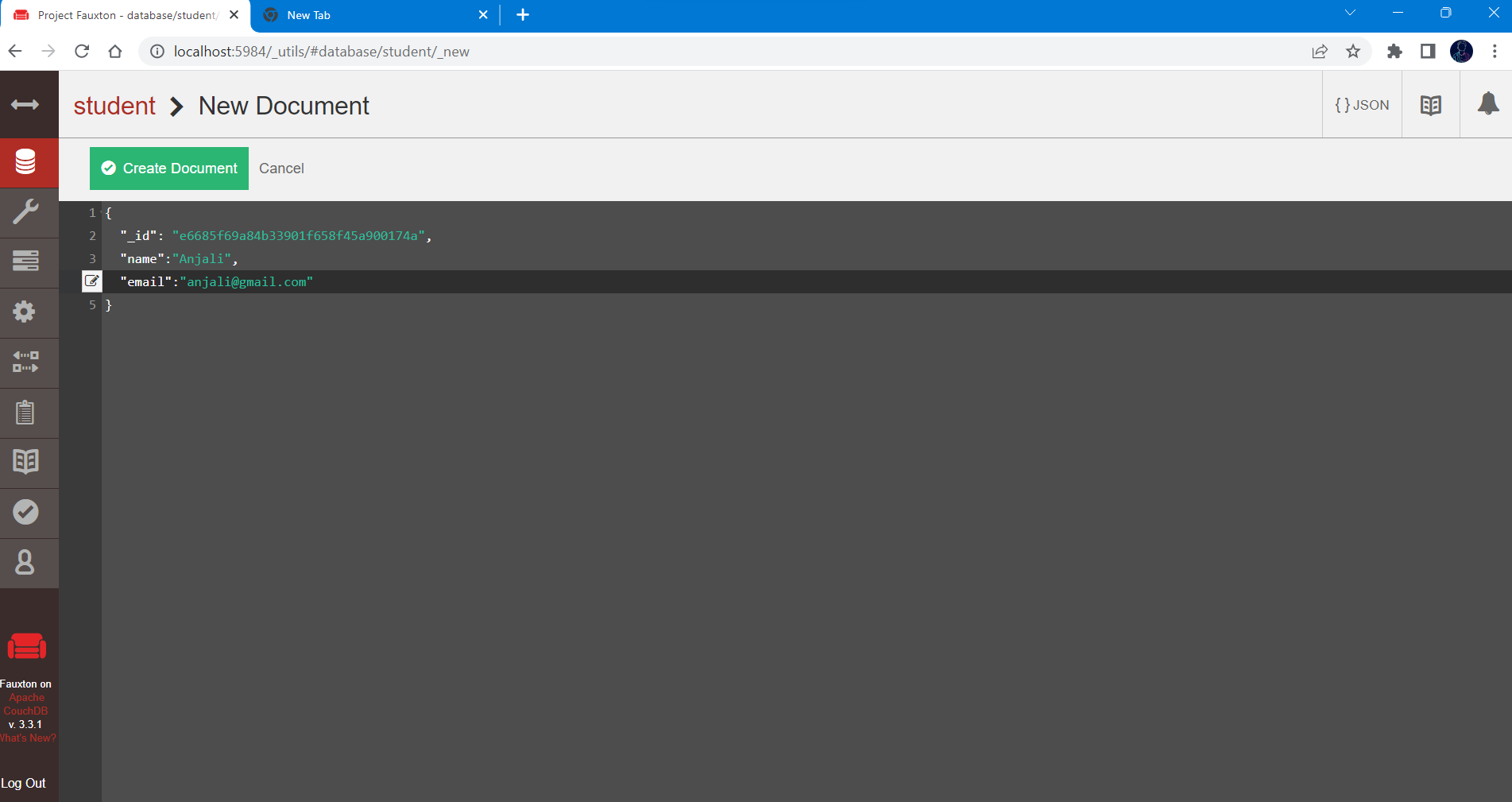


Step-3: Now in your created database go into the **Design Documents** and click on **Create Document.** Now add columns into your database accordingly as shown in image:

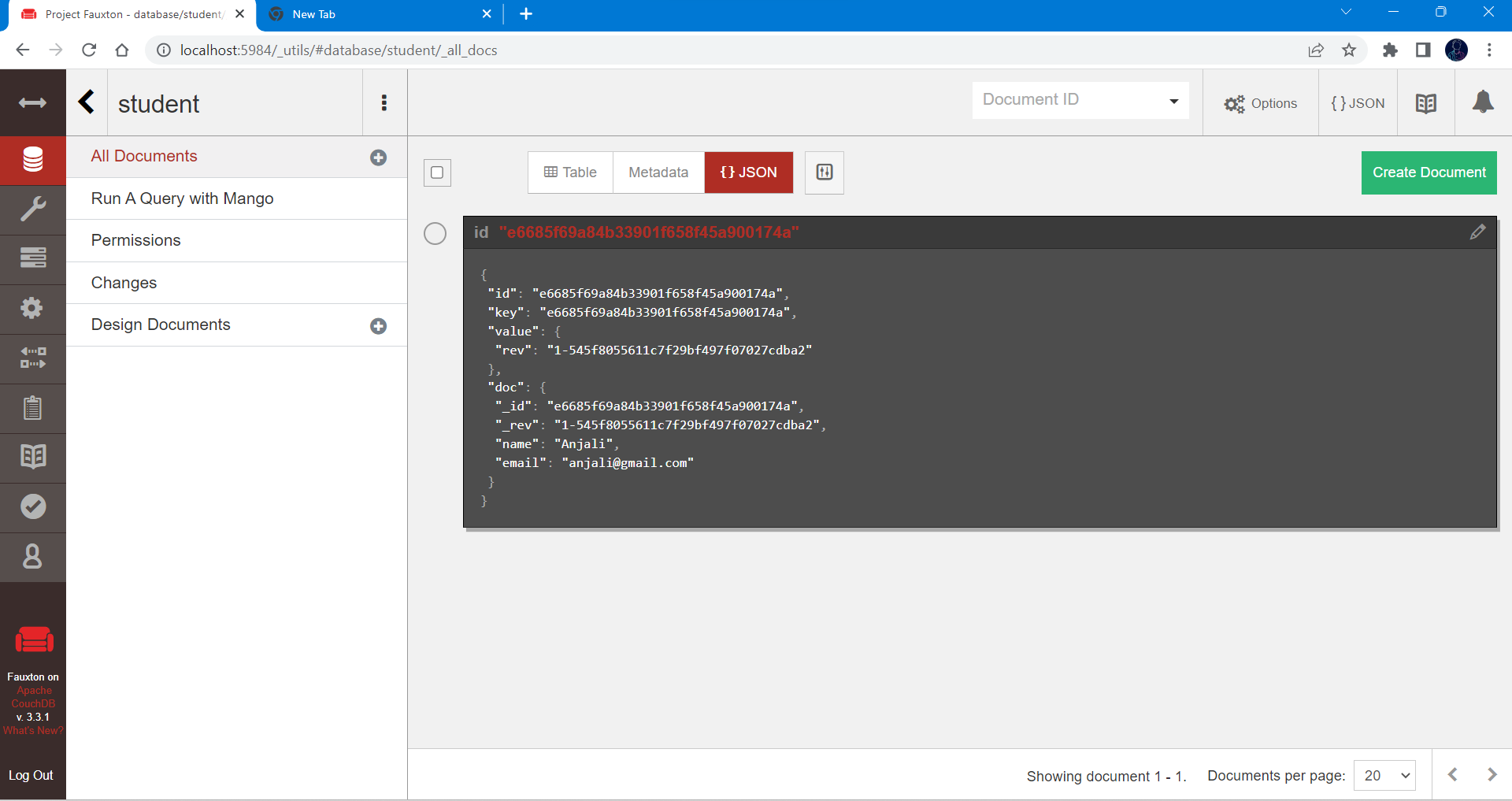
You can add multiple columns.

You can also create new document for new data with additional columns.

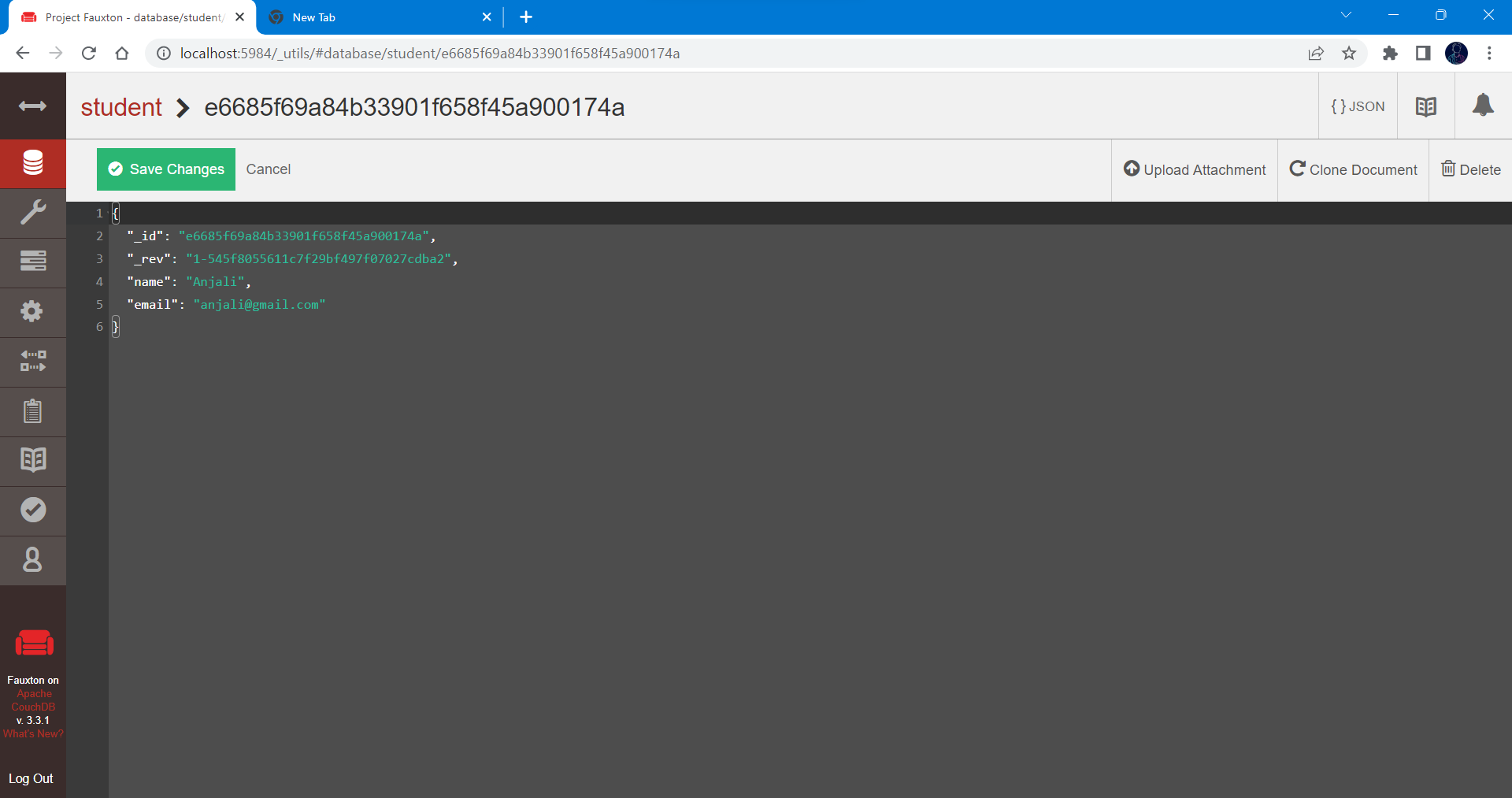
For saving document click on create document.

****

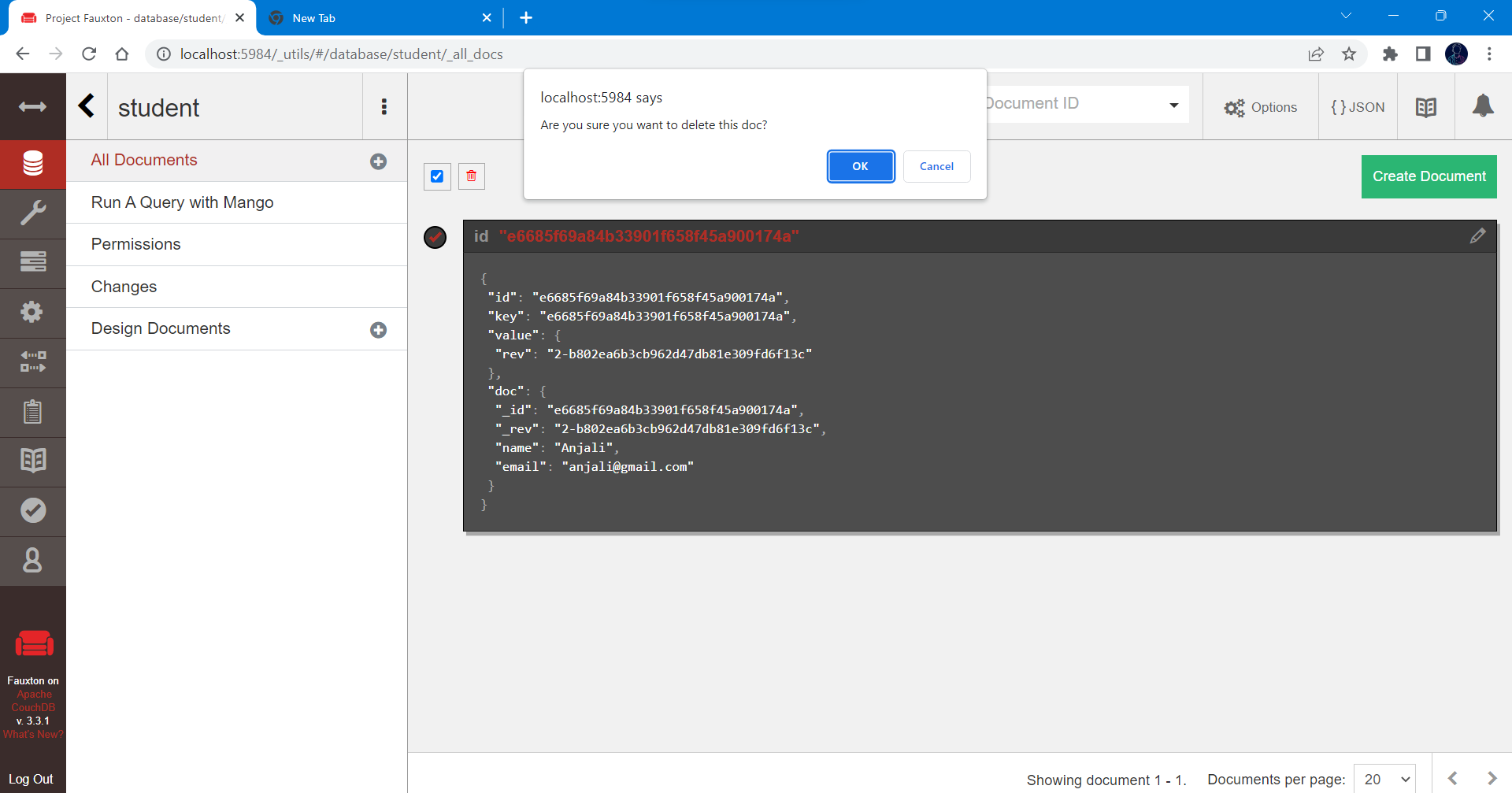
Step-4: Now to view data in json format click on JSON. For table format click on TABLE.



Step-5: To edit data click on id number in red and it will redirect your to window where you can make changes. To save data click on save changes.



Step-6: To delete column click on check button and click delete button.



**Practical 2**

How to Start MongoDB

Search on Chrome: nosql client for mongodb

Link: <https://nosqlbooster.com/downloads>

Select on download< then select Windows 64 bit.

To start the MongoDB select start option from collect.

Practical:

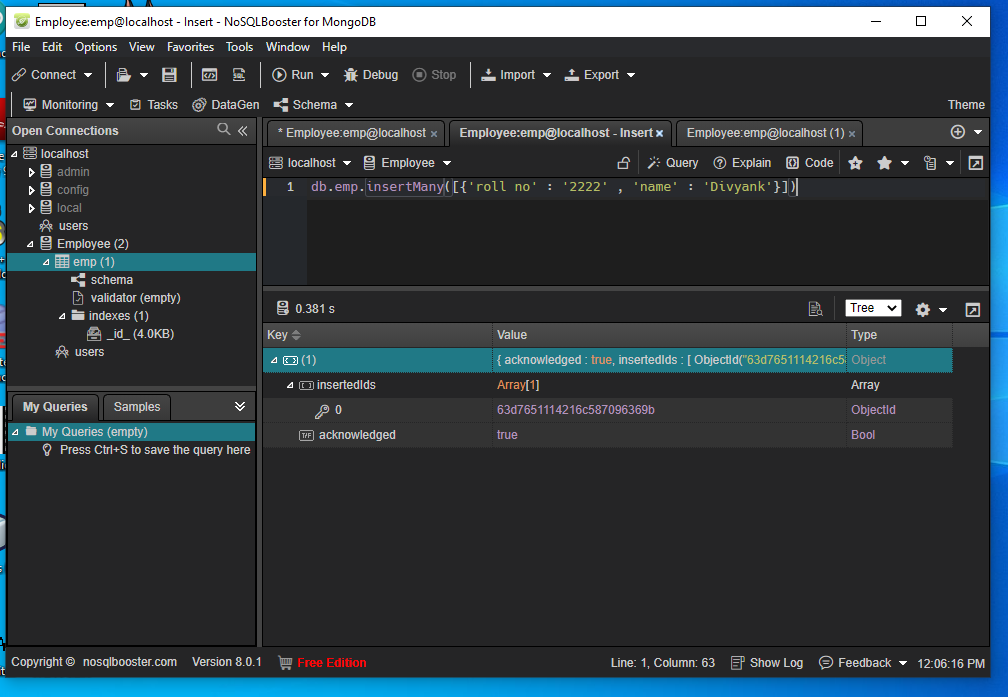
Click on Localhost<Rc : Create Database< Name: Employee<ok<Rc: on Employee<Create Collection< Name: Emp<ok< Rc: Emp < Create Document.    (We can create multiple documents.)

In first Document Enter key value pair.

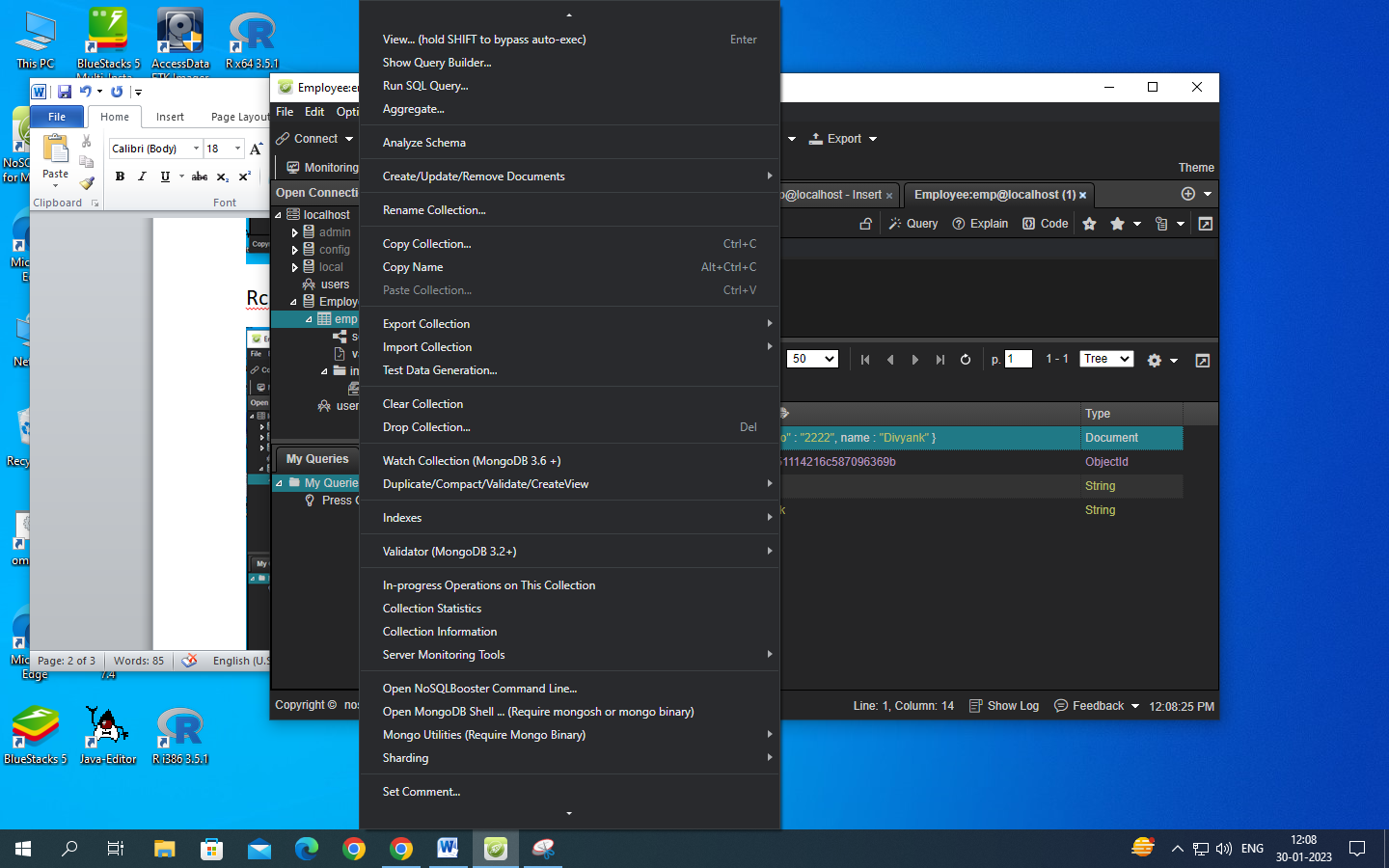
Write code now:

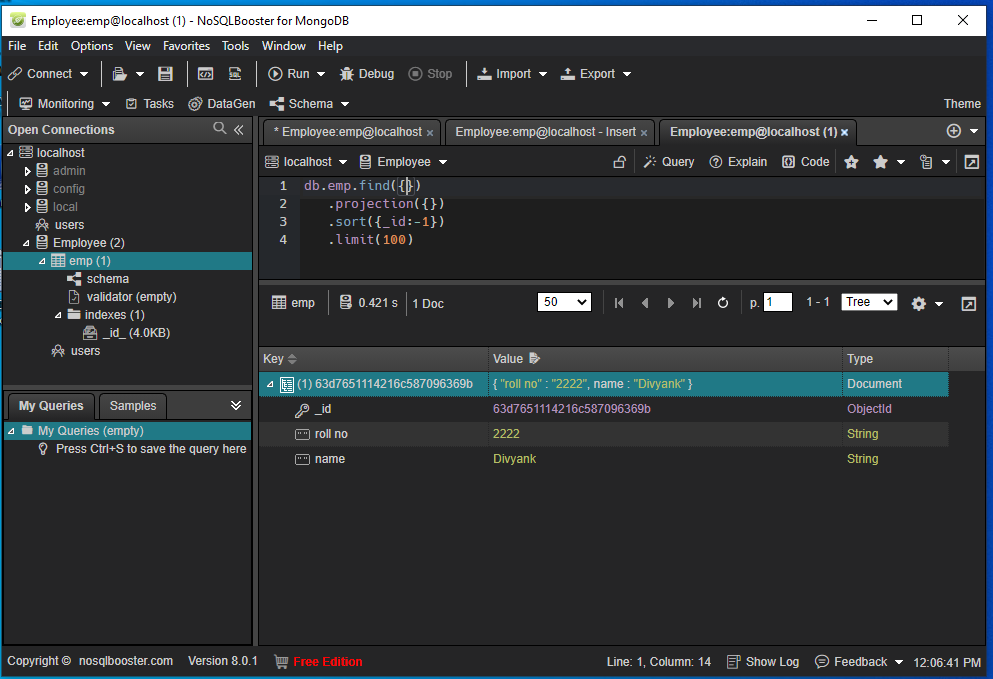
db.emp.insertMany([{‘name’ : ’ABCD’ , ’roll no’ : ’2222’}])

Rc on Emp and Choose View.

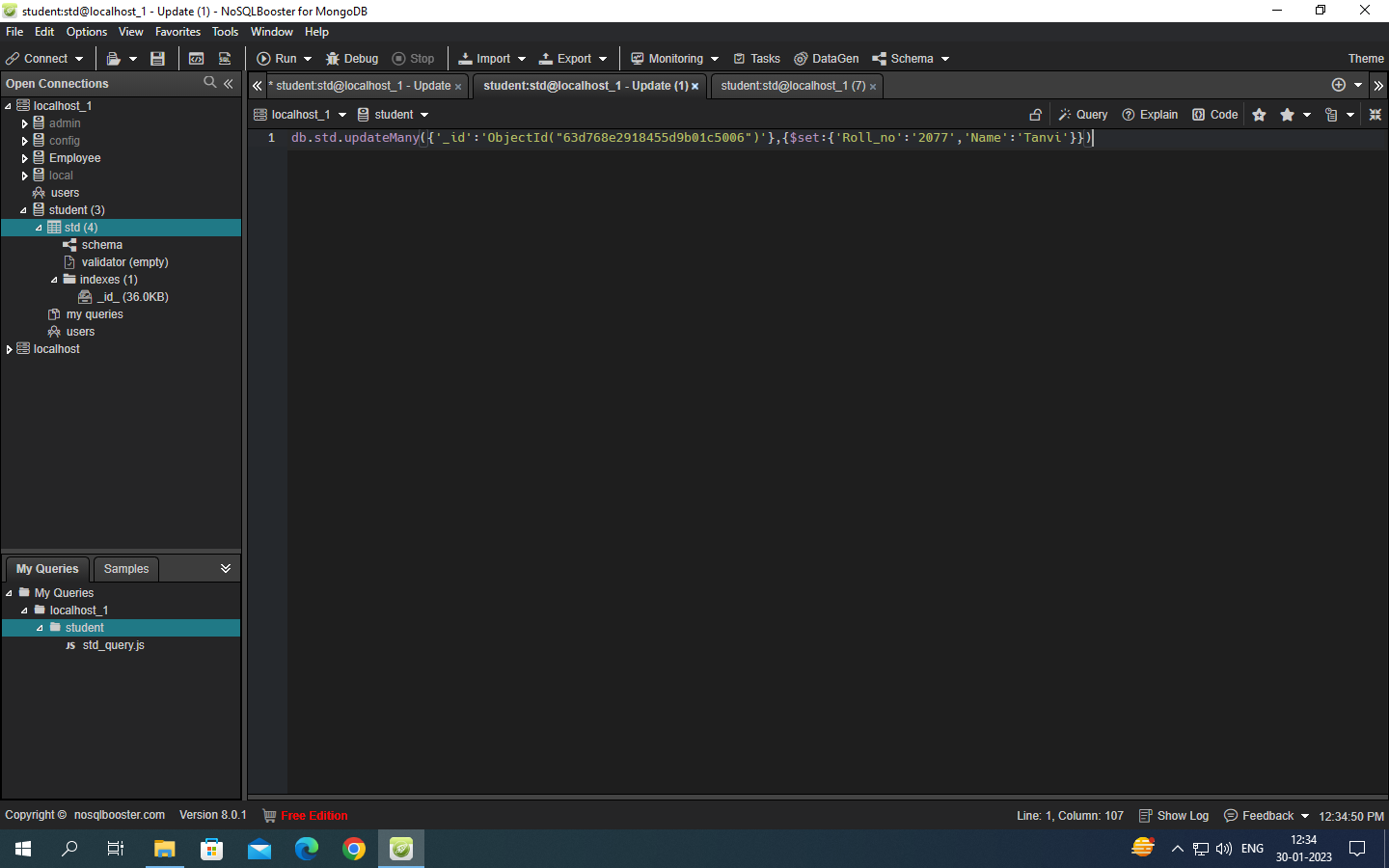


Rc on emp< Then select the 1st option

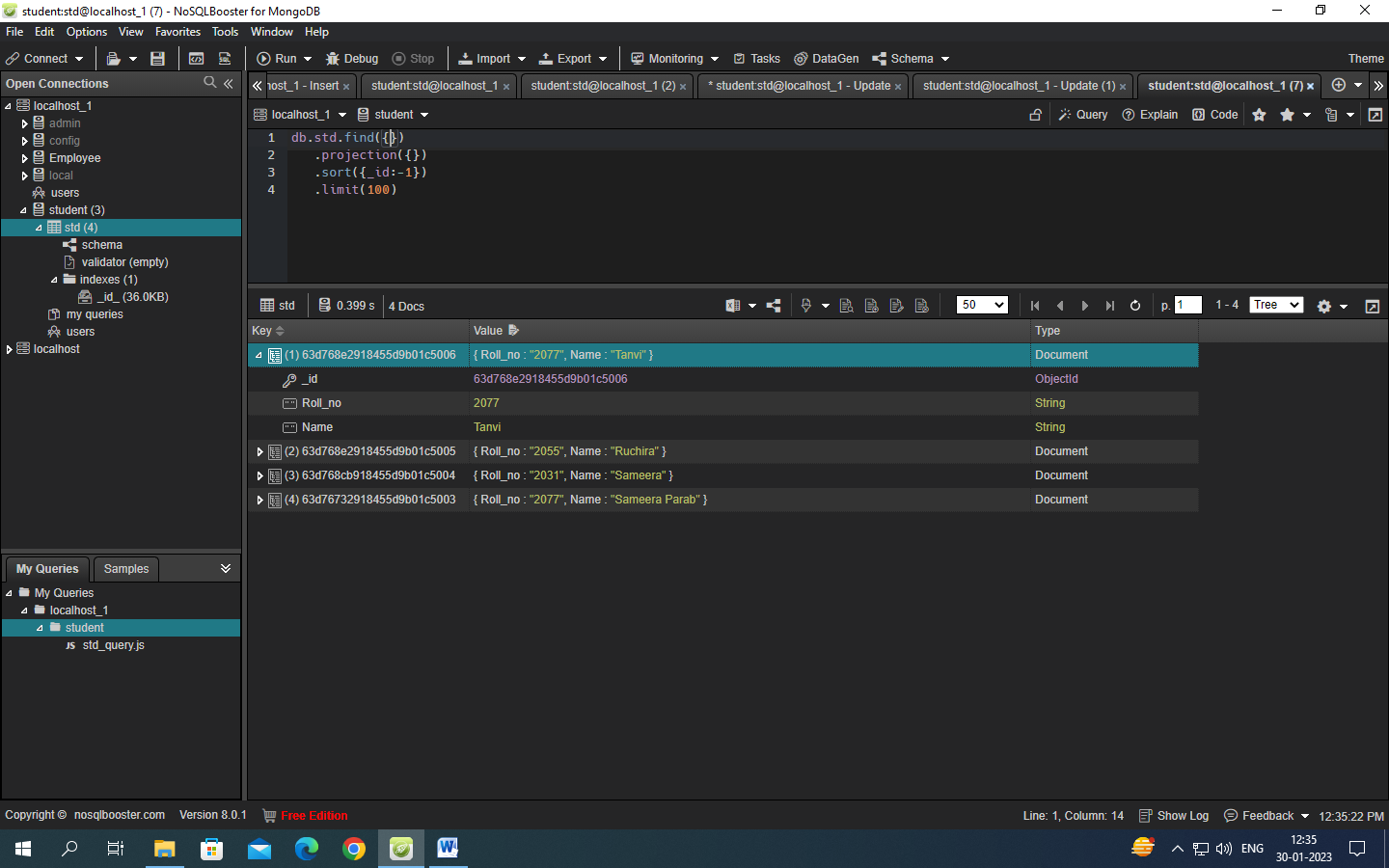




For update:



Output:



**Practical 3**

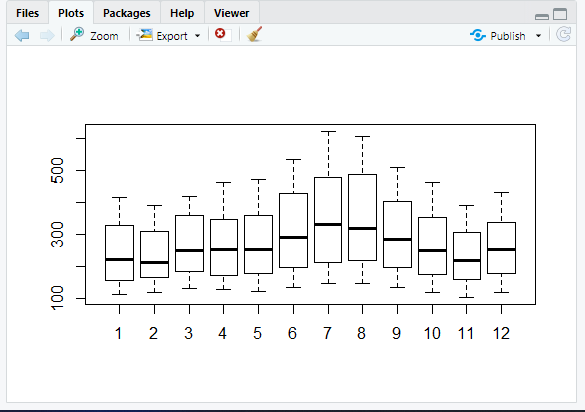
Simple/multiple linear regression

|  |
| --- |
| height <- c(102, 117, 105, 141,135,115, 138, 144, 157, 100, 131,119, 115,121, 113)  weight <- c(61, 46, 62, 54, 60, 69, 51, 50, 46, 64, 48, 56, 64, 48,59)  student <- lm(weight~height)  student  predict(student, data, frame(height=199), interval="confidence")  plot(student) |

**Practical 4**

Time-series forecasting

|  |
| --- |
| data('AirPassengers')  class(AirPassengers)  start(AirPassengers)  end(AirPassengers)  frequency(AirPassengers)  summary(AirPassengers)  plot(AirPassengers)  abline(reg=lm(AirPassengers~time(AirPassengers)))  cycle(AirPassengers)  plot(aggregate(AirPassengers, FUN=mean))  boxplot(AirPassengers ~ cycle(AirPassengers)) |



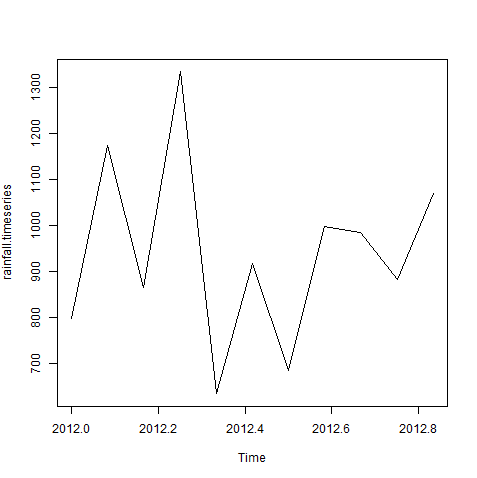
**Practical 5**

Time series analysis

|  |
| --- |
| rainfall <- c(799, 1174, 865.1, 1334.6, 635.4, 918.5, 685.5, 998.6, 985, 882.8, 1071)  rainfall.timeseries <- ts(rainfall, start = c(2012, 1), frequency = 12)  print(rainfall.timeseries)  png(file="rainfall.png")  plot(rainfall.timeseries)  dev.off() |

Output:

Files-> rainfall.png

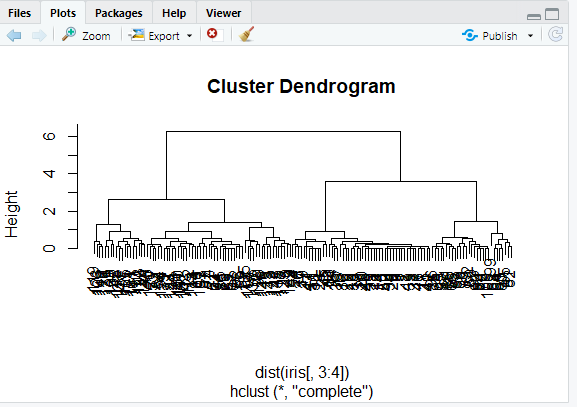


**Practical -6**

K-means clustering

|  |
| --- |
| data(iris)  names(iris)  new\_data<-subset(iris,select=c(-Species))  new\_data  c1<-kmeans(new\_data,3)  c1  data<-new\_data  wss<-sapply(1:15,function(k){kmeans(data,k)$tot.withinss})  wss  plot(1:15,wss,type="b",pch=19,frame=FALSE,xlab="Number of clusters K",ylab="Total within-clusters sums of squares")  library(cluster)  clusplot(new\_data,c1$cluster,color=TRUE,shade=TRUE,labels=2,lines=0)  c1$cluster  c1$centers  "agglomarative clustring"  clusters<-hclust(dist(iris[,3:4]))  plot(clusters)  clusterCut<-cutree(cluster,3)  table(clusterCut,iris$Species) |

Ouput:-



**Practical 7**

Decision tree

|  |
| --- |
| insatll.packages("party")  library(party)  input.dat<-readingSkills[c(1:15),]  png(file="decision\_tree.png")  output.tree<-ctree(nativeSpeaker~age+shoeSize+score,data=input.dat)  plot(output.tree)  dev.off() |

**Practical 8**

Hypothesis testing

|  |
| --- |
| dataf<-seq(1,20,by=1)  dataf  mean(dataf)  sd(dataf)  a<-t.test(dataf,alternate="two.sided",mu=10,conf.int=0.95)  a  a$p.value  a$statistic  (10.5-10)/(sd(dataf)/sqrt(length(dataf)))  length(dataf)=1  length(dataf)  dataf  dataf<-seq(1,20,by=1)  length(dataf)-1 |

**Practical -9**

Analysis of variance

|  |
| --- |
| data("warpbreaks")  head(warpbreaks)  summary(warpbreaks)  Model\_1<-aov(breaks~wool+tension,data=warpbreaks)  summary(Model\_1)  plot(Model\_1)  Model\_2<-aov(breaks~wool+tension+wool:tension,data=warpbreaks)  summary(Model\_2)  plot(Model\_2) |

