Student Name = Yogesh Haresh Bojja

Student Number = s3789918

URL = https://rmiteduau-my.sharepoint.com/:p:/g/personal/s3789918_student_rmit_edu_au/EdwIJXr-3hlIuURGEaEhqyUBOaCeBbrQ3f7vkwUw6n4plQ?e=jqwwJW

Raw Code =

Datascraping.R = R code for scraping data.

Visualization.R = R code for visualization.

**************** ************ **Datascraping.R** **********************************

```
library(dplyr)

library(rvest)

library(ggplot2)

url =  c("https://www.imdb.com/search/title/?title_type=feature&release_date=2015-01-01,2021-01-01&countries=in&languages=hi&count=250",

        "https://www.imdb.com/search/title/?title_type=feature&release_date=2015-01-01,2021-01-01&countries=in&languages=hi&count=250&start=251&ref_=adv_nxt",

        "https://www.imdb.com/search/title/?title_type=feature&release_date=2015-01-01,2021-01-01&countries=in&languages=hi&count=250&start=501&ref_=adv_nxt",

        "https://www.imdb.com/search/title/?title_type=feature&release_date=2015-01-01,2021-01-01&countries=in&languages=hi&count=250&start=751&ref_=adv_nxt",

        "https://www.imdb.com/search/title/?title_type=feature&release_date=2015-01-01,2021-01-01&countries=in&languages=hi&count=250&start=1001&ref_=adv_nxt",

        "https://www.imdb.com/search/title/?title_type=feature&release_date=2015-01-01,2021-01-01&countries=in&languages=hi&count=250&start=1251&ref_=adv_nxt",

        "https://www.imdb.com/search/title/?title_type=feature&release_date=2015-01-01,2021-01-01&countries=in&languages=hi&count=250&start=1501&ref_=adv_nxt",

        "https://www.imdb.com/search/title/?title_type=feature&release_date=2015-01-01,2021-01-01&countries=in&languages=hi&count=250&start=1751&ref_=adv_nxt")


movies = data.frame()

for(u in url){

  data = read_html(u)

  df <- data %>%
html_nodes('.mode-advanced') %>%    # select enclosing nodes
# iterate over each, pulling out desired parts and coerce to data.frame
map_df(~list(name = html_nodes(.x, '.lister-item-header a') %>%
          html_text() %>%
          {if(length(.) == 0) NA else .},    # replace length-0 elements with NA
        year = html_nodes(.x, '.text-muted.unbold') %>%
          html_text() %>%
          {if(length(.) == 0) NA else .},
        rating = html_nodes(.x, '.ratings-imdb-rating strong') %>%
          html_text() %>%
          {if(length(.) == 0) NA else .},
        genre = html_nodes(.x, '.genre') %>%
          html_text() %>%
          {if(length(.) == 0) NA else .},
        votes = html_nodes(.x, '.sort-num_votes-visible span:nth-child(2)') %>%
          html_text() %>%
          {if(length(.) == 0) NA else .},
        cast = html_nodes(.x, '.text-muted+ p') %>%
          html_text() %>%
          {if(length(.) == 0) NA else .},
        runtime = html_nodes(.x, '.runtime') %>%
          html_text() %>%
          {if(length(.) == 0) NA else .},
        image_url = html_nodes(.x, '.loadlate') %>%
          html_attr("loadlate") %>%
          {if(length(.) == 0) NA else .},
        next_link = html_nodes(.x, '.lister-item-header a') %>%
          html_attr("href") %>%
          {if(length(.) == 0) NA else paste("https://www.imdb.com/",.,sep = "")},
        certificate = html_nodes(.x, '.certificate') %>%
          html_text() %>%
                  {if(length(.) == 0) NA else .}
    ))
  movies = rbind(movies,df)
}


library(sjmisc)
library(polite)
production = data.frame()

for (link in movies$next_link){
  dat = read_html(link)
  df <- dat %>% html_nodes("#titleDetails")
  k = df %>% html_nodes(".txt-block")
  prods_name = NULL
  budget = NULL
  box_office  =NULL
  date = NULL
  for(h in k){
    h4 = h %>% html_nodes("h4") %>% html_text() %>% trim()
    a = h %>% html_nodes("a")
    #cat(h %>% html_nodes("h4") %>% html_text() %>% trim())
    #cat("\t",h %>% html_nodes("a") %>% html_text() %>% trim()," ----------\n")
    if(str_contains(h4, "Production", ignore.case = TRUE)){
      prods_name = a %>% html_text() %>% trimws(., which = "both", whitespace = "[\t\n' ']")
     }
    if(str_contains(h4, "budget", ignore.case = TRUE)){
      budget = h %>% html_text() %>% trimws(., which = "both", whitespace = "[\t\n' ']")
    }
    if(str_contains(h4, "gross", ignore.case = TRUE)){
      box_office = h %>% html_text() %>% trim()
  }
  if(str_contains(h4, "date", ignore.case = TRUE)){
   date = h %>% html_text() %>% trimws(., which = "both", whitespace = "[\t\n' ']")
  }

 }
 if(is.null(prods_name)){
  prods_name = NA
 }
 if(is.null(budget)){
  budget = NA
 }
 if(is.null(box_office)){
  box_office = NA
 }
 if(is.null(date)){
  date = NA
 }
 table = data.frame(prods_name[-length(prods_name)] %>% paste(collapse = ", "), budget, date,
box_office)
 production = rbind(production,table)
}
names(production) = c("studio","budget","date","box_office")

for(i in seq(1:1926)){
 std = production[i,]$studio
 bd = production[i,]$budget
 dt = production[i,]$date
 bo = production[i,]$box_office

 if(std=="" && is.na(bd) && is.na(dt) && is.na(bo)){
```

```
dat = read_html(movies$next_link[i])
df <- dat %>% html_nodes("#titleDetails")
k = df %>% html_nodes(".txt-block")
prods_name = NULL
budget = NULL
box_office  =NULL
date = NULL
for(h in k){
  h4 = h %>% html_nodes("h4") %>% html_text() %>% trim()
  a = h %>% html_nodes("a")
  #cat(h %>% html_nodes("h4") %>% html_text() %>% trim())
  #cat("\t",h %>% html_nodes("a") %>% html_text() %>% trim()," ----------\n")
  if(str_contains(h4, "Production", ignore.case = TRUE)){
    prods_name = a %>% html_text() %>% trimws(., which = "both", whitespace = "[\t\n' ']")
  }
  if(str_contains(h4, "budget", ignore.case = TRUE)){
    budget = h %>% html_text() %>% trimws(., which = "both", whitespace = "[\t\n' ']")
  }
  if(str_contains(h4, "gross", ignore.case = TRUE)){
    box_office = h %>% html_text() %>% trim()
  }
  if(str_contains(h4, "date", ignore.case = TRUE)){
    date = h %>% html_text() %>% trimws(., which = "both", whitespace = "[\t\n' ']")
  }

}
if(is.null(prods_name)){
  prods_name = NA
}
if(is.null(budget)){
  budget = NA
```

```
    }
    if(is.null(box_office)){
      box_office = NA
    }
    if(is.null(date)){
      date = NA
    }

    production[i,]$studio = prods_name[-length(prods_name)] %>% paste(collapse = ", ")
    production[i,]$budget = budget
    production[i,]$box_office = box_office
    production[i,]$date = date
  }
}
create = function(l){
  cat("----",l)
  return(strsplit(l, '\\?')[[1]][1] %>% paste(.,"companycredits?ref_=tt_dt_co",sep = ""))
}
movies$dist_link = lapply(movies$next_link, FUN=create)

df1 = data.frame()
for(i in seq(1,1926)){
  link = movies$dist_link[i][[1]][1]
  cat("\n###### -----",i,"----",movies$name[i])
  dat = read_html(link)
  df <- dat %>%
    html_nodes('#distributors+ .simpleList') %>%
    map_df(~list(distributers = html_nodes(.x, 'a')
                 %>% html_text() %>% paste(collapse = ", ")%>%
                 {if(length(.) == 0) "null" else .}
    ))
  if(is_empty(dat %>% html_nodes('#distributors+ .simpleList'))){
    df = data.frame(distributers="NA")
  }
  df1 = rbind(df1, df)
  print(df1[i,1])
}


Movie_df = cbind(movies, production, df1)
list_of_datasets <- list("movies_list" = Movie_df)
write.xlsx(list_of_datasets, file = "movies_2015_2021.xlsx")

url = c("https://www.imdb.com/search/title/?title_type=tv_series,tv_miniseries&release_date=2019-01-01,2021-01-01&countries=in&languages=hi&count=250",
        "https://www.imdb.com/search/title/?title_type=tv_series,tv_miniseries&release_date=2019-01-01,2021-01-01&countries=in&languages=hi&count=250&start=251&ref_=adv_nxt",
        "https://www.imdb.com/search/title/?title_type=tv_series,tv_miniseries&release_date=2019-01-01,2021-01-01&countries=in&languages=hi&count=250&start=501&ref_=adv_nxt",
        "https://www.imdb.com/search/title/?title_type=tv_series,tv_miniseries&release_date=2019-01-01,2021-01-01&countries=in&languages=hi&count=250&start=751&ref_=adv_nxt")

webseries_1 = data.frame()
for(u in url){
  data = read_html(u)
  df <- data %>%
    html_nodes('.mode-advanced') %>%    # select enclosing nodes
    # iterate over each, pulling out desired parts and coerce to data.frame
    map_df(~list(name = html_nodes(.x, '.lister-item-header a') %>%
                   html_text() %>%
                   {if(length(.) == 0) NA else .},    # replace length-0 elements with NA
                 year = html_nodes(.x, '.text-muted.unbold') %>%
                   html_text() %>%
                   {if(length(.) == 0) NA else .},
                 rating = html_nodes(.x, '.ratings-imdb-rating strong') %>%
                   html_text() %>%
                   {if(length(.) == 0) NA else .},
                 genre = html_nodes(.x, '.genre') %>%
                   html_text() %>%
                   {if(length(.) == 0) NA else .},
                 votes = html_nodes(.x, '.sort-num_votes-visible span:nth-child(2)') %>%
                   html_text() %>%
                   {if(length(.) == 0) NA else .},
                 cast = html_nodes(.x, '.text-muted+ p') %>%
                   html_text() %>%
                   {if(length(.) == 0) NA else .},
                 runtime = html_nodes(.x, '.runtime') %>%
                   html_text() %>%
                   {if(length(.) == 0) NA else .},
                 image_url = html_nodes(.x, '.loadlate') %>%
                   html_attr("loadlate") %>%
                   {if(length(.) == 0) NA else .},
                 next_link = html_nodes(.x, '.lister-item-header a') %>%
                   html_attr("href") %>%
                   {if(length(.) == 0) NA else paste("https://www.imdb.com/",.,sep = "")},
                 certificate = html_nodes(.x, '.certificate') %>%
                   html_text() %>%
                   {if(length(.) == 0) NA else .}
    ))
  webseries_1 = rbind(webseries_1,df)
}


production_w = data.frame()
```

```
for (link in webseries_1$next_link){
  dat = read_html(link)
  df <- dat %>% html_nodes("#titleDetails")
  k = df %>% html_nodes(".txt-block")
  prods_name = NULL
  budget = NULL
  box_office  =NULL
  date = NULL
  for(h in k){
    h4 = h %>% html_nodes("h4") %>% html_text() %>% trim()
    a = h %>% html_nodes("a")
    #cat(h %>% html_nodes("h4") %>% html_text() %>% trim())
    #cat("\t",h %>% html_nodes("a") %>% html_text() %>% trim()," ----------\n")
    if(str_contains(h4, "Production", ignore.case = TRUE)){
      prods_name = a %>% html_text() %>% trimws(., which = "both", whitespace = "[\t\n' ']")
    }
    if(str_contains(h4, "budget", ignore.case = TRUE)){
      budget = h %>% html_text() %>% trimws(., which = "both", whitespace = "[\t\n' ']")
    }
    if(str_contains(h4, "gross", ignore.case = TRUE)){
      box_office = h %>% html_text() %>% trim()
    }
    if(str_contains(h4, "date", ignore.case = TRUE)){
      date = h %>% html_text() %>% trimws(., which = "both", whitespace = "[\t\n' ']")
    }

  }
  if(is.null(prods_name)){
    prods_name = NA
  }
  if(is.null(budget)){
    budget = NA
  }
  if(is.null(box_office)){
    box_office = NA
  }
  if(is.null(date)){
    date = NA
  }
  table = data.frame(prods_name[-length(prods_name)] %>% paste(collapse = ", "), budget, date, box_office)
  production_w = rbind(production_w,table)
}
names(production_w) = c("studio","budget","date","box_office")


for(i in seq(1:752)){
  std = production_w[i,]$studio
  bd = production_w[i,]$budget
  dt = production_w[i,]$date
  bo = production_w[i,]$box_office

  if(std=="" && is.na(bd) && is.na(dt) && is.na(bo)){
    dat = read_html(webseries_1$next_link[i])
    df <- dat %>% html_nodes("#titleDetails")
    k = df %>% html_nodes(".txt-block")
    prods_name = NULL
    budget = NULL
    box_office  =NULL
    date = NULL
    for(h in k){
```

```
  h4 = h %>% html_nodes("h4") %>% html_text() %>% trim()
  a = h %>% html_nodes("a")
  #cat(h %>% html_nodes("h4") %>% html_text() %>% trim())
  #cat("\t",h %>% html_nodes("a") %>% html_text() %>% trim()," ----------\n")
  if(str_contains(h4, "Production", ignore.case = TRUE)){
    prods_name = a %>% html_text() %>% trimws(., which = "both", whitespace = "[\t\n' ']")
  }
  if(str_contains(h4, "budget", ignore.case = TRUE)){
    budget = h %>% html_text() %>% trimws(., which = "both", whitespace = "[\t\n' ']")
  }
  if(str_contains(h4, "gross", ignore.case = TRUE)){
    box_office = h %>% html_text() %>% trim()
  }
  if(str_contains(h4, "date", ignore.case = TRUE)){
    date = h %>% html_text() %>% trimws(., which = "both", whitespace = "[\t\n' ']")
  }

}
if(is.null(prods_name)){
  prods_name = NA
}
if(is.null(budget)){
  budget = NA
}
if(is.null(box_office)){
  box_office = NA
}
if(is.null(date)){
  date = NA
}
```

```
    production_w[i,]$studio = prods_name[-length(prods_name)] %>% paste(collapse = ", ")
    production_w[i,]$budget = budget
    production_w[i,]$box_office = box_office
    production_w[i,]$date = date

  }
}

create = function(l){
  cat("----",l)
  return(strsplit(l, '\\?')[[1]][1] %>% paste(.,"companycredits?ref_=tt_dt_co",sep = ""))
}
webseries_1$dist_link = lapply(webseries_1$next_link, FUN=create)

df1 = data.frame()
for(i in seq(1,752)){
  link = webseries_1$dist_link[i][[1]][1]
  cat("\n###### -----",i,"----",webseries_1$name[i])
  dat = read_html(link)
  df <- dat %>%
    html_nodes('#distributors+ .simpleList') %>%
    map_df(~list(distributers = html_nodes(.x, 'a')
                %>% html_text() %>% paste(collapse = ", ")%>%
                {if(length(.) == 0) "null" else .}
    ))
  if(is_empty(dat %>% html_nodes('#distributors+ .simpleList'))){
    df = data.frame(distributers="NA")
  }
  df1 = rbind(df1, df)
  print(df1[i,1])
}
```

```
Webseries_df = cbind(webseries_1, production_w, df1)

list_of_datasets <- list("webseries_list" = Webseries_df)

write.xlsx(list_of_datasets, file = "webseries_2019_2021.xlsx")



url =
c("https://www.imdb.com/search/title/?title_type=tv_series,tv_miniseries&release_date=2015-01-01,2019-01-01&countries=in&languages=hi&count=250",

        "https://www.imdb.com/search/title/?title_type=tv_series,tv_miniseries&release_date=2015-01-01,2019-01-01&countries=in&languages=hi&count=250&start=251&ref_=adv_nxt",

        "https://www.imdb.com/search/title/?title_type=tv_series,tv_miniseries&release_date=2015-01-01,2019-01-01&countries=in&languages=hi&count=250&start=501&ref_=adv_nxt",

        "https://www.imdb.com/search/title/?title_type=tv_series,tv_miniseries&release_date=2015-01-01,2019-01-01&countries=in&languages=hi&count=250&start=751&ref_=adv_nxt")


webseries_1 = data.frame()

for(u in url){

  data = read_html(u)

  df <- data %>%

    html_nodes('.mode-advanced') %>%    # select enclosing nodes

    # iterate over each, pulling out desired parts and coerce to data.frame

    map_df(~list(name = html_nodes(.x, '.lister-item-header a') %>%

                   html_text() %>%

                   {if(length(.) == 0) NA else .},    # replace length-0 elements with NA

                 year = html_nodes(.x, '.text-muted.unbold') %>%

                   html_text() %>%

                   {if(length(.) == 0) NA else .},

                 rating = html_nodes(.x, '.ratings-imdb-rating strong') %>%

                   html_text() %>%

                   {if(length(.) == 0) NA else .},

                 genre = html_nodes(.x, '.genre') %>%
                   html_text() %>%
                   {if(length(.) == 0) NA else .},
                 votes = html_nodes(.x, '.sort-num_votes-visible span:nth-child(2)') %>%
                   html_text() %>%
                   {if(length(.) == 0) NA else .},
                 cast = html_nodes(.x, '.text-muted+ p') %>%
                   html_text() %>%
                   {if(length(.) == 0) NA else .},
                 runtime = html_nodes(.x, '.runtime') %>%
                   html_text() %>%
                   {if(length(.) == 0) NA else .},
                 image_url = html_nodes(.x, '.loadlate') %>%
                   html_attr("loadlate") %>%
                   {if(length(.) == 0) NA else .},
                 next_link = html_nodes(.x, '.lister-item-header a') %>%
                   html_attr("href") %>%
                   {if(length(.) == 0) NA else paste("https://www.imdb.com/",.,sep = "")},
                 certificate = html_nodes(.x, '.certificate') %>%
                   html_text() %>%
                   {if(length(.) == 0) NA else .}
    ))
  webseries_1 = rbind(webseries_1,df)
}



production_w = data.frame()


for (link in webseries_1$next_link){
  dat = read_html(link)
  df <- dat %>% html_nodes("#titleDetails")
  k = df %>% html_nodes(".txt-block")
```

```
prods_name = NULL

budget = NULL

box_office  =NULL

date = NULL

for(h in k){

  h4 = h %>% html_nodes("h4") %>% html_text() %>% trim()

  a = h %>% html_nodes("a")

  if(str_contains(h4, "Production", ignore.case = TRUE)){

    prods_name = a %>% html_text() %>% trimws(., which = "both", whitespace = "[\t\n' ']")

  }

  if(str_contains(h4, "budget", ignore.case = TRUE)){

    budget = h %>% html_text() %>% trimws(., which = "both", whitespace = "[\t\n' ']")

  }

  if(str_contains(h4, "gross", ignore.case = TRUE)){

    box_office = h %>% html_text() %>% trim()

  }

  if(str_contains(h4, "date", ignore.case = TRUE)){

    date = h %>% html_text() %>% trimws(., which = "both", whitespace = "[\t\n' ']")

  }


}

if(is.null(prods_name)){

  prods_name = NA

}

if(is.null(budget)){

  budget = NA

}

if(is.null(box_office)){

  box_office = NA

}

if(is.null(date)){
    date = NA
  }
  cat("\n#####################################################")
  cat("\nprod ---",prods_name[-length(prods_name)] %>% paste(collapse = ", "))
  cat("\nbudget ----",budget)
  cat("\ndate ----",date)
  cat("\nbox ----",box_office)
  table = data.frame(prods_name[-length(prods_name)] %>% paste(collapse = ", "), budget, date,
box_office)
  production_w = rbind(production_w,table)
}
names(production_w) = c("studio","budget","date","box_office")



for(i in seq(1:848)){
  std = production_w[i,]$studio
  bd = production_w[i,]$budget
  dt = production_w[i,]$date
  bo = production_w[i,]$box_office

  if(std=="" && is.na(bd) && is.na(dt) && is.na(bo)){
    dat = read_html(webseries_1$next_link[i])
    df <- dat %>% html_nodes("#titleDetails")
    k = df %>% html_nodes(".txt-block")
    prods_name = NULL
    budget = NULL
    box_office  =NULL
    date = NULL
    for(h in k){
      h4 = h %>% html_nodes("h4") %>% html_text() %>% trim()
      a = h %>% html_nodes("a")
  #cat(h %>% html_nodes("h4") %>% html_text() %>% trim())
  #cat("\t",h %>% html_nodes("a") %>% html_text() %>% trim()," ----------\n")
  if(str_contains(h4, "Production", ignore.case = TRUE)){
    prods_name = a %>% html_text() %>% trimws(., which = "both", whitespace = "[\t\n' ']")
  }
  if(str_contains(h4, "budget", ignore.case = TRUE)){
    budget = h %>% html_text() %>% trimws(., which = "both", whitespace = "[\t\n' ']")
  }
  if(str_contains(h4, "gross", ignore.case = TRUE)){
    box_office = h %>% html_text() %>% trim()
  }
  if(str_contains(h4, "date", ignore.case = TRUE)){
    date = h %>% html_text() %>% trimws(., which = "both", whitespace = "[\t\n' ']")
  }

}
if(is.null(prods_name)){
  prods_name = NA
}
if(is.null(budget)){
  budget = NA
}
if(is.null(box_office)){
  box_office = NA
}
if(is.null(date)){
  date = NA
}

production_w[i,]$studio = prods_name[-length(prods_name)] %>% paste(collapse = ", ")
production_w[i,]$budget = budget
```

```
    production_w[i,]$box_office = box_office
    production_w[i,]$date = date

    cat("\n#########################################################")
    cat("\nprod ---",i,prods_name[-length(prods_name)] %>% paste(collapse = ", "))
    cat("\nbudget ----",budget)
    cat("\ndate ----",date)
    cat("\nbox ----",box_office)

  }
}

create = function(l){
  cat("----",l)
  return(strsplit(l, '\\?')[[1]][1] %>% paste(.,"companycredits?ref_=tt_dt_co",sep = ""))
}
webseries_1$dist_link = lapply(webseries_1$next_link, FUN=create)

df1 = data.frame()
for(i in seq(1,848)){
  link = webseries_1$dist_link[i][[1]][1]
  cat("\n###### -----",i,"----",webseries_1$name[i])
  dat = read_html(link)
  df <- dat %>%
    html_nodes('#distributors+ .simpleList') %>%
    map_df(~list(distributers = html_nodes(.x, 'a')
            %>% html_text() %>% paste(collapse = ", ")%>%
            {if(length(.) == 0) "null" else .}
    ))
  if(is_empty(dat %>% html_nodes('#distributors+ .simpleList'))){
    df = data.frame(distributers="NA")
  }
  df1 = rbind(df1, df)
  print(df1[i,1])
}



Webseries_df = cbind(webseries_1, production_w, df1)
list_of_datasets <- list("webseries_list" = Webseries_df)
write.xlsx(list_of_datasets, file = "webseries_2015_2019.xlsx")
```

************************ **Visualization.R** ********************************

```
movies = read.xlsx("movies_2015_2021.xlsx", sheet = 1)
movies$date = movies$date %>% trimws(.,which = c("both","left","right"), whitespace = "['Release Date:']")
split = function(x){
  x = strsplit(x, " ")[[1]][1:3] %>% paste(collapse=" ")
  return(x)
}
movies$date = movies$date %>% lapply(., FUN = split)
extract = function(x){
  x = regmatches(x, gregexpr(pattern = "2...",text = x))
  return(x[[1]][1])
}
movies$year = movies$year %>% lapply(., FUN = extract)
extract = function(x){
  x = regmatches(x, gregexpr(pattern = ":.* ",text = x))[[1]][1]
  x = trimws(x, which = c("both","left","right"), whitespace = "[':']")
  if(str_contains(x, "INR")){
    x = trimws(x, which = c("both","left","right"), whitespace = "['INR'[:space:]]")
    x = strsplit(x,",")[[1]] %>% paste(collapse = "")
    x = as.numeric(x)/10000000
  }
  else{
    x = trimws(x, which = c("both","left","right"), whitespace = "['\n'[:space:]]")
    x = trimws(x, which = c("both","left","right"), whitespace = "['$'[:space:]]")
    x = strsplit(x,",")[[1]] %>% paste(collapse = "")
    x = as.numeric(as.integer(x) * 72)/10000000
  }

  return(x)
}
movies$budget = movies$budget %>% lapply(., FUN = extract)

extract = function(x){
  x = regmatches(x, gregexpr(pattern = ":.*",text = x))[[1]][1]
  x = trimws(x, which = c("both","left","right"), whitespace = "[': ']")
  if(str_contains(x, "INR")){
    x = trimws(x, which = c("both","left","right"), whitespace = "['INR'[:space:]]")
    x = strsplit(x,",")[[1]] %>% paste(collapse = "")
    x = as.numeric(x)/10000000
  }
  else{
    x = trimws(x, which = c("both","left","right"), whitespace = "['\n'[:space:]]")
    x = trimws(x, which = c("both","left","right"), whitespace = "['$'[:space:]]")
    x = strsplit(x,",")[[1]] %>% paste(collapse = "")
    x = as.numeric(as.integer(x) * 72)/10000000
  }

  return(x)
}
```

```
movies$box_office = movies$box_office %>% lapply(., FUN = extract)



####### how many films actor did #############

salman = data.frame()
for (y in c("2015", "2016", "2017", "2018", "2019", "2020")){
  a = movies$cast[movies$year==y] %>% sapply(.,FUN = function(x){
    bool = str_contains(x,"salman khan",ignore.case = TRUE)
    return(bool)
  })
  sum = table(a)["TRUE"]
  if(is.na(sum)) sum = 0
  salman = rbind(salman, data.frame(year = y, actor="Salman Khan", releases = sum))
}

sharukh = data.frame()
for (y in c("2015", "2016", "2017", "2018", "2019", "2020")){
  a = movies$cast[movies$year==y] %>% sapply(.,FUN = function(x){
    bool = str_contains(x,"Shah rukh khan",ignore.case = TRUE)
    return(bool)
  })
  sum = table(a)["TRUE"]
  if(is.na(sum)) sum = 0
  sharukh = rbind(sharukh, data.frame(year = y, actor="Shah rukh Khan", releases = sum))
}

akshay = c()
for (y in c("2015", "2016", "2017", "2018", "2019", "2020")){
  a = movies$cast[movies$year==y] %>% sapply(.,FUN = function(x){
    bool = str_contains(x,"akshay kumar",ignore.case = TRUE)
    return(bool)
  })
  sum = table(a)["TRUE"]
  if(is.na(sum)) sum = 0
  akshay = rbind(akshay,data.frame(year = y, actor="Akshay Kumar", releases = sum))
}

deepika = data.frame()
for (y in c("2015", "2016", "2017", "2018", "2019", "2020")){
  a = movies$cast[movies$year==y] %>% sapply(.,FUN = function(x){
    bool = str_contains(x,"Deepika padukone",ignore.case = TRUE)
    return(bool)
  })
  sum = table(a)["TRUE"]
  if(is.na(sum)) sum = 0
  deepika = rbind(deepika, data.frame(year = y, actor="Deepika Padukone", releases = sum))
}

anushka = data.frame()
for (y in c("2015", "2016", "2017", "2018", "2019", "2020")){
  a = movies$cast[movies$year==y] %>% sapply(.,FUN = function(x){
    bool = str_contains(x,"anushka sharma",ignore.case = TRUE)
    return(bool)
  })
  sum = table(a)["TRUE"]
  if(is.na(sum)) sum = 0
  anushka = rbind(anushka, data.frame(year = y, actor="Anushka Sharma", releases = sum))
}

alia = data.frame()
for (y in c("2015", "2016", "2017", "2018", "2019", "2020")){
  a = movies$cast[movies$year==y] %>% sapply(.,FUN = function(x){
    bool = str_contains(x,"alia bhatt",ignore.case = TRUE)
    return(bool)
  })
  sum = table(a)["TRUE"]
  if(is.na(sum)) sum = 0
  alia = rbind(alia, data.frame(year = y, actor="Alia Bhatt", releases = sum))
}


########### total movies #######
releases = c()
for (y in c("2015", "2016", "2017", "2018", "2019", "2020")){
  a = movies$year[movies$year == y]
  releases = append(releases, length(a))
}
year = c("2015", "2016", "2017", "2018", "2019", "2020")
x = c(1,2,3,4,5,6)
df = data.frame(x = x, year=year, releases = releases)
predict = lm(releases~x, data = df[1:5,])
predict = round(6*predict$coefficients['x'] + predict$coefficients['(Intercept)'])



library(ggplot2)
library(hrbrthemes)
df1 = data.frame(x=c(5,6), year = c("2019", "2020"),releases=c(df$releases[5], predict))
movies_trend = ggplot(df, mapping = aes(x=year, y=releases, group=2)) +
  geom_vline(xintercept="2019", color="orange", size=.9) +
  geom_text(aes(x="2019", y=300, label = "Start of COVID-19"), angle = 90, vjust = 1.1, colour = "red")+
  geom_line(color="#e60000", size=2, alpha=0.9, linetype=1) +
 geom_line(df1,mapping = aes(x=year, y=releases, group=2), color="#69b3a2", size=2, alpha=0.9, linetype=2) +
 geom_point(aes(x=df1$year[2], y=df1$releases[2])) +
 theme(axis.text.x = element_text(face = "bold", color = "black",
                                  size = 10, angle = 0),
       axis.text.y = element_text(face = "bold", color = "black",
                                  size = 10, angle = 0),
       axis.title = element_text(face = "bold", color = "black",
                                 size = 15, angle = 0),
       axis.title.x = element_text(face = "bold", color = "black",
                                   size = 12),
       axis.title.y = element_text(face = "bold", color = "black",
                                   size = 12),
       title = element_text(face = "bold", color = "black",
                            size = 12, angle = 0)) +
 annotate(geom="point", x=df$year[6], y=df$releases[6], size=3, shape=21, fill="black") +
 annotate(geom="text", x=df$year[6], y=df$releases[6],
          label=paste("\n    Actual releases-",df$releases[6])) +
 annotate(geom="point", x=df$year[5], y=df$releases[5], size=3, shape=21, fill="black") +
 annotate(geom="text", x=df$year[5], y=df$releases[5],
          label=paste("\n          ",df$releases[5])) +
 annotate(geom="point", x=df$year[4], y=df$releases[4], size=3, shape=21, fill="black") +
 annotate(geom="text", x=df$year[4], y=df$releases[4],
          label=paste("\n    ",df$releases[4])) +
 annotate(geom="point", x=df$year[3], y=df$releases[3], size=3, shape=21, fill="black") +
 annotate(geom="text", x=df$year[3], y=df$releases[3],
          label=paste("\n    ",df$releases[3])) +
 annotate(geom="point", x=df$year[2], y=df$releases[2], size=3, shape=21, fill="black") +
 annotate(geom="text", x=df$year[2], y=df$releases[2],
          label=paste("\n    ",df$releases[2])) +
 annotate(geom="point", x=df$year[1], y=df$releases[1], size=3, shape=21, fill="black") +
 annotate(geom="text", x=df$year[1], y=df$releases[1],
        label=paste("       ",df$releases[1])) +
 annotate(geom="point", x=df1$year[2], y=df1$releases[2], size=2, shape=21, fill="black") +
 annotate(geom="point", x=df1$year[2], y=df1$releases[2], size=5, shape=21, fill="transparent") +
 annotate(geom="text", x=df1$year[2], y=df1$releases[2],
        label=paste("\n\n\n       Predicted \n            releases-",df1$releases[2])) +
 labs (title = "Number of films released",
      y="Count of films ",
      x="Year",
      subtitle="(2015-2020)"
 )

ggsave(movies_trend, filename = "Movies_trend.png", width = 12, height = 6)
################# number of films by actors ##########
df = rbind(anushka,alia,deepika,salman,sharukh,akshay)

levels(df$year) = c("2015","2016","2017","2018","2019","2020")
actor_releases = ggplot(df, mapping = aes(x=year, y=releases, fill=actor)) +
 geom_bar(stat = "identity", position = "dodge")+
 facet_wrap(~actor) +
 theme_ipsum() +
 theme(axis.text.x = element_text(face = "bold", color = "black",
                          size = 10, angle = 45),
      axis.text.y = element_text(face = "bold", color = "black",
                          size = 10, angle = 0),
      axis.title = element_text(face = "bold", color = "black",
                         size = 11, angle = 0),
      title = element_text(face = "bold", color = "black",
                      size = 12, angle = 0)) +
 labs (title = "Release of top actors",
      y="Count of films ",
        x="",
        subtitle="(2015-2020)"
  )
ggsave(actor_releases, filename = "Actor_releases.png", width = 12, height = 6)
############### movies delayed ###############
library(tidyverse)
df = filter(movies, (year=="2019"))
df = filter(df,df$date %>% sapply(.,FUN=function(x){
  if(!str_contains(x,"2019", ignore.case = TRUE)){
    if(str_contains(x,"na",ignore.case = TRUE)){
      return(FALSE)
    }
    return(TRUE)
  }
  return(FALSE)
}))
films_not_released = dim(df)[1]

total = 0
ott = 0
for(i in 1:23){
  x = df$distributers[i]
  if(!is.na(x)){
    total = total + 1
    if(str_contains(x,"hotstar",ignore.case = TRUE)){
      ott = ott + 1
    }
    if(str_contains(x,"jio",ignore.case = TRUE)){
      ott = ott + 1
    }
    if(str_contains(x,"netflix",ignore.case = TRUE)){
      ott = ott + 1
    }
    if(str_contains(x,"prime",ignore.case = TRUE)){
      ott = ott + 1
    }
    if(str_contains(x,"zee",ignore.case = TRUE)){
      ott = ott + 1
    }
    if(str_contains(x,"sonyliv",ignore.case = TRUE)){
      ott = ott + 1
    }
  }
}

percentage_of_ott_release = ott/total*100

library(ggplot2)
df_2019 = filter(movies, (year=="2019"))
df = data.frame(released=c("Released", "Not released"), value = c(dim(df_2019)[1]-
films_not_released, films_not_released))
movie_2019_status = ggplot(df, aes(x="", y=value, fill=released)) + geom_bar(stat="identity",
width=1) +
  coord_polar("y", start=0) + geom_text(aes(label = paste0(value)), position = position_stack(vjust =
0.5))+
  scale_fill_manual(values=c("#e60000", "#3fe04c")) +
  labs(x = NULL, y = NULL, fill = NULL, title = "Status of movies in 2019")+
   theme_bw()+ theme(axis.line = element_blank(),
                                axis.text = element_blank(),
                                axis.ticks = element_blank(),
                                plot.title = element_text(hjust = 0.5, color = "black"))
ggsave(movie_2019_status, filename="movie_2019_status.png", width = 8, height = 8)
```

```
df = data.frame(released = c("Released on OTT", "Released in theatre"), value = c((ott/total)*100, 100-(ott/total)*100))
unreleased_status = ggplot(df, aes(x="", y=value, fill=released)) + geom_bar(stat="identity", width=1) +
  coord_polar("y", start=0) + geom_text(aes(label = paste0(round(value,3), "%")), position = position_stack(vjust = 0.5))+
  scale_fill_manual(values=c("#e60000", "#3fe04c")) +
  labs(x = NULL, y = NULL, fill = NULL, title = "Platform of delayed movies")+
  theme_bw()+ theme(axis.line = element_blank(),
                    axis.text = element_blank(),
                    axis.ticks = element_blank(),
                    plot.title = element_text(hjust = 0.5, color = "black"))

ggsave(unreleased_status, filename = "Unreleased_of_2019.png", width = 8, height = 8)
############### webseries data prep######################
webseries_2015_2019 = read.xlsx("webseries_2015_2019.xlsx", sheet = 1)
webseries_2019_2021 = read.xlsx("webseries_2019_2021.xlsx", sheet = 1)
webseries = rbind(webseries_2015_2019, webseries_2019_2021)

webseries$date = webseries$date %>% trimws(.,which = c("both","left","right"), whitespace = "['Release Date:']")
split = function(x){
  x = strsplit(x, " ")[[1]][1:3] %>% paste(collapse=" ")
  return(x)
}
webseries$date = webseries$date %>% lapply(., FUN = split)

extract = function(x){
  x = regmatches(x, gregexpr(pattern = "2...",text = x))
  return(x[[1]][1])
}
webseries$year = webseries$year %>% lapply(., FUN = extract)
```

```
releases = c()

for (y in c("2015", "2016", "2017", "2018", "2019", "2020")){

  a = webseries$year[webseries$year == y]

  releases = append(releases, length(a))

}

x = c(1,2,3,4,5,6)

year = c("2015", "2016", "2017", "2018", "2019", "2020")

df = data.frame(x = x, year=year, releases = releases)

predict = lm(releases~x, data = df[1:5,])

predict = round(6*predict$coefficients['x'] + predict$coefficients['(Intercept)'])



library(ggplot2)

library(hrbrthemes)

df1 = data.frame(x=c(5,6), year=c("2019","2020"), releases=c(df$releases[5], predict))

webseries_trend = ggplot(df, mapping = aes(x=year, y=releases, group = 2)) +
geom_line(color="#e60000", size=2, alpha=0.9, linetype=1) +

  geom_line(df1, mapping = aes(x=year, y=releases, group=2), color="#69a3b2", size=2, alpha=0.9,
linetype=2) +

  theme(axis.text.x = element_text(face = "bold", color = "black",

                                    size = 10, angle = 0),

        axis.text.y = element_text(face = "bold", color = "black",

                                    size = 10, angle = 0),

        axis.title = element_text(face = "bold", color = "black",

                                   size = 15, angle = 0),

        axis.title.x = element_text(face = "bold", color = "black",

                                     size = 12),

        axis.title.y = element_text(face = "bold", color = "black",

                                     size = 12),

        title = element_text(face = "bold", color = "black",

                              size = 12, angle = 0)) +
  geom_vline(xintercept="2019", color="orange", size=.9) +
  geom_text(aes(x="2019", y=210, label = "Start of COVID-19"), angle = 90, vjust = 1.1, colour="red")+
  geom_vline(xintercept="2016", color="green", size=.9) +
  geom_text(aes(x="2016", y=315, label = "Jio Telecomm publicly available (5th sept 2016)"), angle = 90, vjust = 1.1, colour="red")+
  annotate(geom="point", x=df$year[6], y=df$releases[6], size=3, shape=21, fill="black") +
  annotate(geom="text", x=df$year[6], y=df$releases[6],
         label=paste("\n    Actual releases-",df$releases[6])) +
  annotate(geom="point", x=df$year[5], y=df$releases[5], size=3, shape=21, fill="black") +
  annotate(geom="text", x=df$year[5], y=df$releases[5],
         label=paste("\n    ",df$releases[5])) +
  annotate(geom="point", x=df$year[4], y=df$releases[4], size=3, shape=21, fill="black") +
  annotate(geom="text", x=df$year[4], y=df$releases[4],
         label=paste("\n    ",df$releases[4])) +
  annotate(geom="point", x=df$year[3], y=df$releases[3], size=3, shape=21, fill="black") +
  annotate(geom="text", x=df$year[3], y=df$releases[3],
         label=paste("\n    ",df$releases[3])) +
  annotate(geom="point", x=df$year[2], y=df$releases[2], size=3, shape=21, fill="black") +
  annotate(geom="text", x=df$year[2], y=df$releases[2],
         label=paste("\n    ",df$releases[2])) +
  annotate(geom="point", x=df$year[1], y=df$releases[1], size=3, shape=21, fill="black") +
  annotate(geom="text", x=df$year[1], y=df$releases[1],
         label=paste("       ",df$releases[1])) +
  annotate(geom="point", x=df1$year[2], y=df1$releases[2], size=2, shape=21, fill="black") +
  annotate(geom="point", x=df1$year[2], y=df1$releases[2], size=5, shape=21, fill="transparent") +
  annotate(geom="text", x=df1$year[2], y=df1$releases[2],
         label=paste("\n\n\n       Predicted \n           releases-",df1$releases[2])) +
  labs (title = "Number of Webseries released",
      y="Count of webseries ",
      x="Year",
      subtitle="(2015-2020)"
  )
ggsave(webseries_trend, filename = "Webseries_trend.png", width = 12, height = 6)




############### ott count ##################
netflix = data.frame()
for (y in c("2015", "2016", "2017", "2018", "2019", "2020")){
  a = webseries$distributers[webseries$year==y] %>% sapply(.,FUN = function(x){
    bool = str_contains(x,"netflix",ignore.case = TRUE)
    return(bool)
  })
  sum = table(a)["TRUE"]
  if(is.na(sum)) sum = 0
  netflix = rbind(netflix, data.frame(year = y, ott="Netflix", releases = sum))
}
lab = paste("Netflix\t(+",mean(netflix[5:6,3])-mean(netflix[1:4,3]),")",sep="")
netflix = data.frame(x=c("Pre-lockdown", "Post-lockdown"), ott=c(lab,lab),
average=c(mean(netflix[1:4,3]), mean(netflix[5:6,3])))

hotstar = data.frame()
for (y in c("2015", "2016", "2017", "2018", "2019", "2020")){
  a = webseries$distributers[webseries$year==y] %>% sapply(.,FUN = function(x){
    bool = str_contains(x,"hotstar",ignore.case = TRUE)
    return(bool)
  })
  sum = table(a)["TRUE"]
  if(is.na(sum)) sum = 0
  hotstar = rbind(hotstar, data.frame(year = y, ott="Hotstar", releases = sum))
}
lab = paste("Hotstar\t(+",mean(hotstar[5:6,3])-mean(hotstar[1:4,3]),")",sep="")
```

```
hotstar = data.frame(x=c("Pre-lockdown", "Post-lockdown"), ott=c(lab,lab),
average=c(mean(hotstar[1:4,3]), mean(hotstar[5:6,3])))


prime = data.frame()
for (y in c("2015", "2016", "2017", "2018", "2019", "2020")){
  a = webseries$distributers[webseries$year==y] %>% sapply(.,FUN = function(x){
    bool = str_contains(x,"prime",ignore.case = TRUE)
    return(bool)
  })
  sum = table(a)["TRUE"]
  if(is.na(sum)) sum = 0
  prime = rbind(prime, data.frame(year = y, ott="Amazon Prime", releases = sum))
}
lab = paste("Amazon Prime\t(+",mean(prime[5:6,3])-mean(prime[1:4,3]),")",sep="")
prime = data.frame(x=c("Pre-lockdown", "Post-lockdown"), ott=c(lab,lab),
average=c(mean(prime[1:4,3]), mean(prime[5:6,3])))

jio = data.frame()
for (y in c("2015", "2016", "2017", "2018", "2019", "2020")){
  a = webseries$distributers[webseries$year==y] %>% sapply(.,FUN = function(x){
    bool = str_contains(x,"jio",ignore.case = TRUE)
    return(bool)
  })
  sum = table(a)["TRUE"]
  if(is.na(sum)) sum = 0
  jio = rbind(jio, data.frame(year = y, ott="Jio Studio", releases = sum))
}
lab = paste("Jio Studio\t(+",mean(jio[5:6,3])-mean(jio[1:4,3]),")",sep="")
jio = data.frame(x=c("Pre-lockdown", "Post-lockdown"), ott=c(lab,lab), average=c(mean(jio[1:4,3]),
mean(jio[5:6,3])))
```

```
alt = data.frame()

for (y in c("2015", "2016", "2017", "2018", "2019", "2020")){

  a = webseries$distributers[webseries$year==y] %>% sapply(.,FUN = function(x){

    bool = str_contains(x,"alt",ignore.case = TRUE)

    return(bool)

  })

  sum = table(a)["TRUE"]

  if(is.na(sum)) sum = 0

  alt = rbind(alt, data.frame(year = y, ott="ALT Balaji", releases = sum))

}

lab = paste("ALT Balaji\t(+",mean(alt[5:6,3])-mean(alt[1:4,3]),")",sep="")

alt = data.frame(x=c("Pre-lockdown", "Post-lockdown"), ott=c(lab,lab), average=c(mean(alt[1:4,3]),
mean(alt[5:6,3])))


zee = data.frame()

for (y in c("2015", "2016", "2017", "2018", "2019", "2020")){

  a = webseries$distributers[webseries$year==y] %>% sapply(.,FUN = function(x){

    bool = str_contains(x,"zee",ignore.case = TRUE)

    return(bool)

  })

  sum = table(a)["TRUE"]

  if(is.na(sum)) sum = 0

  zee = rbind(zee, data.frame(year = y, ott="ZEE 5", releases = sum))

}

lab = paste("ZEE 5\t(+",mean(zee[5:6,3])-mean(zee[1:4,3]),")",sep="")

zee = data.frame(x=c("Pre-lockdown", "Post-lockdown"), ott=c(lab,lab), average=c(mean(zee[1:4,3]),
mean(zee[5:6,3])))


sony = data.frame()

for (y in c("2015", "2016", "2017", "2018", "2019", "2020")){

  a = webseries$distributers[webseries$year==y] %>% sapply(.,FUN = function(x){

    bool = str_contains(x,"sonyliv",ignore.case = TRUE)
    return(bool)
  })
  sum = table(a)["TRUE"]
  if(is.na(sum)) sum = 0
  sony = rbind(sony, data.frame(year = y, ott="Sony Liv", releases = sum))
}
lab = paste("Sony Liv\t(+",mean(sony[5:6,3])-mean(sony[1:4,3]),")",sep="")
sony = data.frame(x=c("Pre-lockdown", "Post-lockdown"), ott=c(lab,lab),
average=c(mean(sony[1:4,3]), mean(sony[5:6,3])))

ullu = data.frame()
for (y in c("2015", "2016", "2017", "2018", "2019", "2020")){
  a = webseries$distributers[webseries$year==y] %>% sapply(.,FUN = function(x){
    bool = str_contains(x,"ullu",ignore.case = TRUE)
    return(bool)
  })
  sum = table(a)["TRUE"]
  if(is.na(sum)) sum = 0
  ullu = rbind(ullu, data.frame(year = y, ott="ULLU", releases = sum))
}
lab = paste("ULLU\t(+",mean(ullu[5:6,3])-mean(ullu[1:4,3]),")",sep="")
ullu = data.frame(x=c("Pre-lockdown", "Post-lockdown"), ott=c(lab, lab),
average=c(mean(ullu[1:4,3]), mean(ullu[5:6,3])))

df = rbind(netflix, prime, jio, ullu, hotstar, zee, alt)

levels(df$ott) = c("Netflix","Amazon Prime","Jio Studio","ULLU","ZEE 5", "ALT Balaji")
ott_comparison = ggplot(df, mapping = aes(x=reorder(x,average), y=average, label=average)) +
  geom_bar(stat = "identity", position = "identity", fill="#e60000")+
  geom_text(aes(label = average), vjust= -0.2)+
  ylim(0,35)+
  theme_ipsum() +
  theme(axis.text.x = element_text(face = "bold", color = "black",
                                    size = 10, angle = 0),
        axis.text.y = element_text(face = "bold", color = "black",
                                    size = 10, angle = 0),
        axis.title = element_text(face = "bold", color = "black",
                                   size = 11, angle = 0),
        title = element_text(face = "bold", color = "black",
                              size = 12, angle = 0)
        ) +
  facet_wrap(~ott) +
  labs (title = "OTT comparison(Average count of releases)",
        y=" Average count of Webseries released ",
        x="",
        subtitle="Pre-lockdown vs Post-lockdown"
  )
ggsave(ott_comparison, filename = "OTT_comparison.png", width = 12, height = 6)

############ JIO inception ############
#https://borgenproject.org/internet-access-india/

df = data.frame(x=c("Before inception of Jio", "After inception of Jio"), usage=c(0.7,11))
jio_inception = ggplot(df, aes(x=reorder(x,usage), y=usage, group=1))+
  geom_line(color="dark green", size=2, alpha=0.9, linetype=1)+
  annotate("point",x="Before inception of Jio",y=0.7,size=4,shape=21,fill="black") +
  annotate("text", x="Before inception of Jio",y=0.7,label="                700 MB")+
  annotate("point",x="After inception of Jio",y=11,size=4,shape=21,fill="black")+
  annotate("text", x="After inception of Jio",y=11,label="\n          11 GB")+
  theme_ipsum() +
  theme(axis.text.x = element_text(face = "bold", color = "black",
                                    size = 12, angle = 0),
        axis.text.y = element_text(face = "bold", color = "black",
                                    size = 12, angle = 0),
        axis.title = element_text(face = "bold", color = "black",
                                   size = 20, angle = 0),
        title = element_text(face = "bold", color = "black",
                              size = 12, angle = 0),
        axis.title.y = element_text(face = "bold", color = "black",
                                    size = 15),
        panel.border = element_blank(),
        axis.line= element_line(colour="black")
  ) +
  labs (title = "Rise in internet usage since inception of JIO",
        y=" Internet Usage (GB)                              ",
        x="",
        caption = "Source : https://borgenproject.org/internet-access-india/"
  )
ggsave(jio_inception, filename = "Jio_inception.png", width = 8, height = 8)



######## jio price #######
#https://restofworld.org/2020/how-india-mobile-data-became-worlds-
cheapest/#:~:text=At%20one%20point%20in%202016,approximately%20225%20rupees%20(%243).


df = data.frame(x=c("Post-Jio Launch", "Pre-Jio Launch"), price=c(18.5,225))
jio_prices = ggplot(df, aes(x=reorder(x,desc(price)), y=price, group=1))+
  geom_line(color="red", size=2, alpha=0.9, linetype=1)+
  annotate("point",x="Pre-Jio Launch",y=225,size=4,shape=21,fill="black") +
  annotate("text", x="Pre-Jio Launch",y=225,label="                      Rs. 225/GB")+
  annotate("point",x="Post-Jio Launch",y=18.5,size=4,shape=21,fill="black")+
  annotate("text", x="Post-Jio Launch",y=18.5,label="\n          Rs.18.5/GB")+
  theme_ipsum() +
  theme(axis.text.x = element_text(face = "bold", color = "black",
                                    size = 12, angle = 0),
        axis.text.y = element_text(face = "bold", color = "black",
                                    size = 12, angle = 0),
        axis.title = element_text(face = "bold", color = "black",
                                   size = 20, angle = 0),
        title = element_text(face = "bold", color = "black",
                       size = 12, angle = 0),
        axis.title.y = element_text(face = "bold", color = "black",
                                    size = 15),
        panel.border = element_blank(),
        axis.line= element_line(colour="black")
  ) +
  labs (title = "Decreases in internet rates after inception of JIO",
        y=" Price (Rs.)                              ",
        x="",
        caption = "Source : (Deck and Deck, 2021)"
  )
ggsave(jio_prices, filename = "Jio_prices.png", width = 8, height = 8)

##########################################
```