HW8

Aidan Pizzo

10/6/21

source(“configuration.R”)

#data.munged.R

```r
library(plyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
get_files <- function()
{
  yt_dir <- "../../../datasets/youtubeData/"
  read_files <- Map(function(file){
    path <- paste(yt_dir, file, sep="")
    fileCountry <- substr(file, 1, 2)
    theFile <- read.csv(path, header = T, stringsAsFactors = F)
    theFile$country = rep(fileCountry, length(theFile$video_id))
    return(theFile)
  }, dir(yt_dir))
  do.call(rbind, read_files)
}


make_df <- function()
{
  df <- get_files()
  get_info <- plyr::ddply(df, .(country), function(frame) {
    popTitle <- frame$title[which.max(frame$views)]
    dayViews <- Map(function(theDay){
      day_df <- dplyr::filter(frame, trending_date == theDay)
      totalViews <- day_df$views %>% sum
      day <- theDay
      out_df <- data.frame(day, totalViews, stringsAsFactors=F)
      return(out_df)
    }, frame$trending_date %>% unique)
    dayViews <- do.call(rbind, dayViews)
    busiestDay <- dayViews$day[which.max(dayViews$totalViews)]
    theCountry <- frame$country[1]
    nonCountry_df <- dplyr::filter(df, country != theCountry)
    unique_title <- setdiff(frame$title, nonCountry_df$title)
    unique_df <- dplyr::filter(frame, is.element(title, unique_title))
    uniquePopTitle <- unique_df$title[which.max(unique_df$views)]
    out_df <- data.frame(theCountry, popTitle, busiestDay, uniquePopTitle, stringsAsFactors=F)
    return(out_df)
  })
  return(get_info)
}


num_trends <- function()
{
  df <- get_files()
  get_info <- plyr::ddply(df, .(video_id), function(frame) {
    video_id <- frame$video_id[1]
    nCountry <- length(unique(frame$country))
    out_df <- data.frame(video_id, nCountry, stringsAsFactors=F)
    return(out_df)
  })
  return(get_info)
}


trend_all <- function()
{
  df <- get_files()
  df2 <- num_trends()
  all_countries <- list()
  for (i in 1:length(df2$nCountry))
  {
    if (df2$nCountry[i] == 10)
    {
      all_countries <- c(all_countries, df2$video_id[i])
    }
  }
  all_countries <- do.call(rbind, all_countries)
  all_countries <- as.vector(all_countries)
  all_countries <- setdiff(all_countries, "#NAME?")
  out_df <- dplyr::filter(df, is.element(video_id, all_countries))
  out_df <- dplyr::select(out_df, title)
  out_vec <- out_df$title %>% unique
}
```

#analysis.R

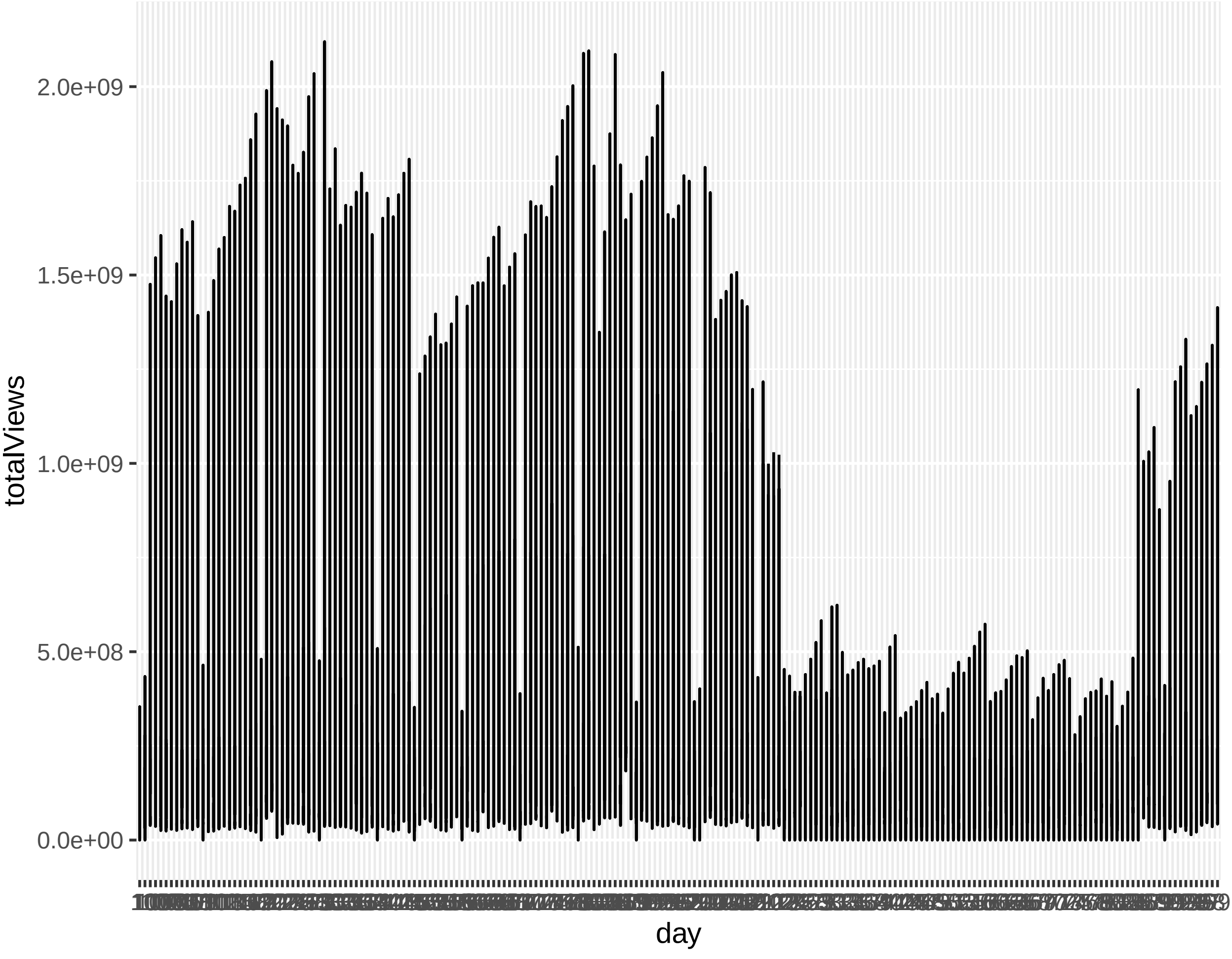
```
library(lubridate)
```

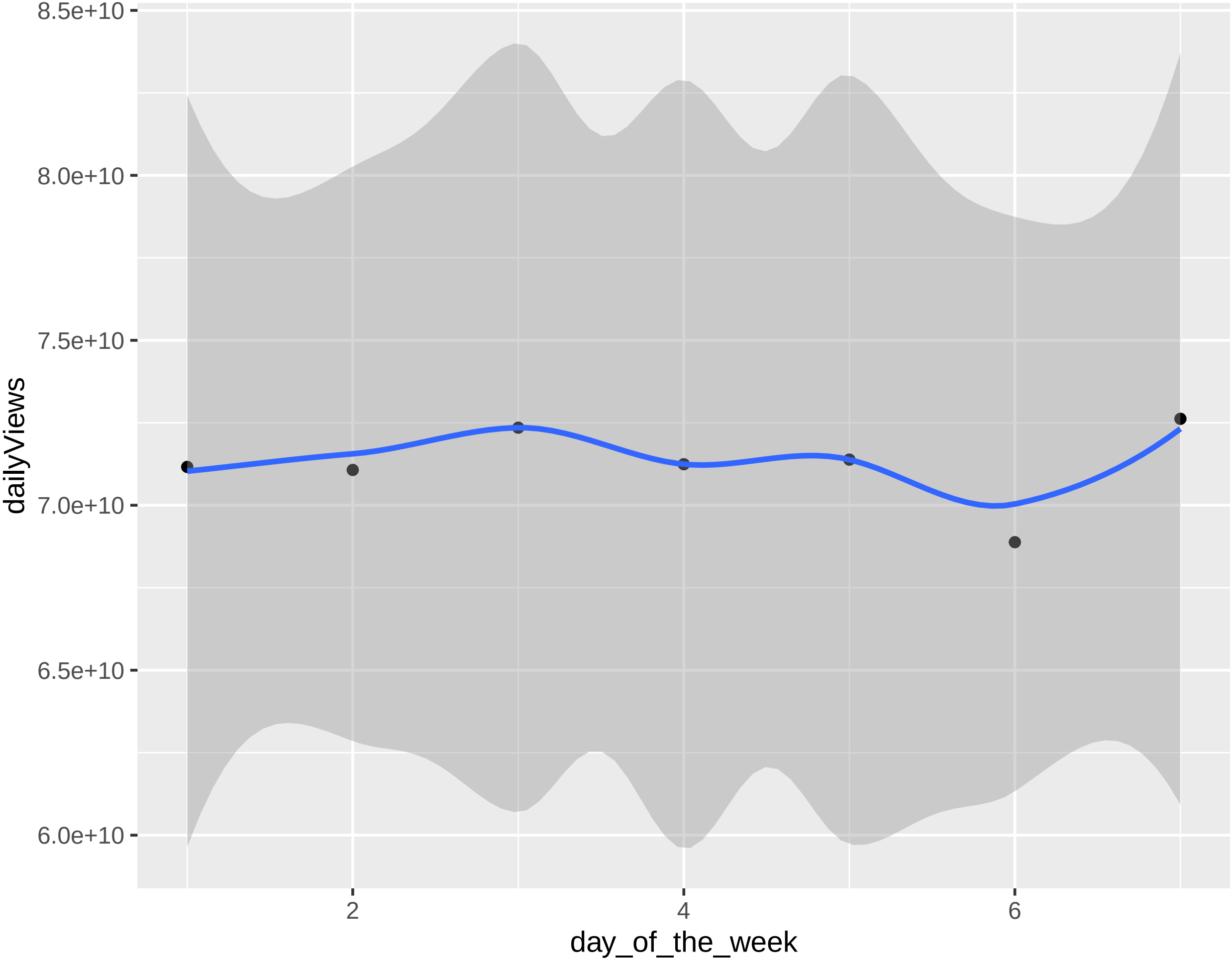
```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(ggplot2)

make_plot_1 <- function()
{
  df <- get_files()
  df <- dplyr::select(df, trending_date, views, country)

  days <- df$trending_date %>% unique
  countries <- df$country %>% unique
  plot_list <- list()
  for (i in 1:length(countries))
  {
    country_df <- dplyr::filter(df, country == countries[i])
    totalViews <- Map(function(date){
      day_df <- dplyr:: filter(country_df, trending_date == date)
      totalViews <- sum(day_df$views)
      return(totalViews)
    }, days)
    totalViews <- do.call(rbind, totalViews)
    plot_df <- data.frame(day = days, totalViews, country = rep(countries[i], length(totalViews)), strin
    plot_list[[i]] <- plot_df
    names(plot_list)[i] <- countries[i]
  }
  plot_df <- do.call(rbind, plot_list)
  for (i in 1:length(plot_df$day))
  {
    plot_df$day[i] <- clean_date_num(plot_df$day[i])
  }
  g1 <- ggplot(data=plot_df) + geom_line(mapping=aes(x=day,
                                                     y=totalViews))
  return(g1)
}


make_plot_2 <- function()
{
  df <- get_files()
  df <- dplyr::select(df, trending_date, views, country)

  days <- df$trending_date %>% unique
  countries <- df$country %>% unique
  plot_list <- list()
  for (i in 1:length(countries))
  {
    country_df <- dplyr::filter(df, country == countries[i])
    totalViews <- Map(function(date){
      day_df <- dplyr:: filter(country_df, trending_date == date)
      totalViews <- sum(day_df$views)
      return(totalViews)
    }, days)
    totalViews <- do.call(rbind, totalViews)
    plot_df <- data.frame(day = days, totalViews, country = rep(countries[i], length(totalViews)), stri
    plot_list[[i]] <- plot_df
    names(plot_list)[i] <- countries[i]
  }
  plot_df <- do.call(rbind, plot_list)
  for (i in 1:length(plot_df$day))
  {
    plot_df$day[i] <- clean_date_mdy(plot_df$day[i])
  }
  day_of_the_week <- 1:7
  dailyViews <- vector("numeric")
  for (i in 1:length(day_of_the_week))
  {
    day_df <- dplyr::filter(plot_df, day == day_of_the_week[i])
    dailyViews[i] <- sum(day_df$totalViews)
  }
  out_df <- data.frame(day_of_the_week, dailyViews, stringsAsFactors = F)
  g2 <- ggplot(data=out_df) + geom_point(mapping=aes(x=day_of_the_week,
                                                     y=dailyViews))+
  geom_smooth(mapping=aes(x=day_of_the_week, y=dailyViews))
  return(g2)
}

clean_date_num <- function(date)
{
    yr <- substr(date, 1, 2)
    dy <- substr(date, 4, 5)
    mo <- substr(date, 7, 8)
    out_date <- paste(mo, "/", dy, "/", yr, sep="")
    out_date <- out_date %>% mdy
    num_date <- out_date %>% as.numeric - 17483
    return(num_date)
}

clean_date_mdy <- function(date)
{
    yr <- substr(date, 1, 2)
    dy <- substr(date, 4, 5)
    mo <- substr(date, 7, 8)
    out_date <- paste(mo, "/", dy, "/", yr, sep="")
    out_date <- out_date %>% mdy
    out_date <- wday(out_date)
    return(out_date)
}
```

#presentation.R

```
g1 <- make_plot_1()
g2 <- make_plot_2()
```


2.0e+09
1.5e+09
1.0e+09
5.0e+08
0.0e+00
totalViews
day


8.5e+10
8.0e+10
7.5e+10
7.0e+10
6.5e+10
6.0e+10
dailyViews
2
4
6
day_of_the_week