Data Wrangling Final Project

Jeffrey Orcutt

2021-05-17

Use this RMarkdown file to create the final\_tourism data frame. There are two data sets, country\_info and tourism, in the data folder of this project.

Have fun!

library(readxl)  
library(writexl)  
library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.1 ──

## ✓ ggplot2 3.3.3 ✓ purrr 0.3.4  
## ✓ tibble 3.1.1 ✓ dplyr 1.0.5  
## ✓ tidyr 1.1.3 ✓ stringr 1.4.0  
## ✓ readr 1.4.0 ✓ forcats 0.5.1

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(dplyr, warn.conflicts = FALSE)  
library(zoo)

##   
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

library(DescTools)  
library(openxlsx)  
  
#disables scientific notation  
options(scipen=999)  
  
  
# bring data into two tables  
country\_info2 <- read\_xlsx("/cloud/project/data/country\_info.xlsx")  
country\_info <- NULL  
  
country\_info2$temp\_val <- NULL  
country\_info$Country <- NULL  
country\_info$Continent <- NULL  
  
country\_info <- country\_info2 %>% mutate(temp\_val=case\_when(Continent==1 ~ "North America", Continent==2 ~ "South America", Continent==3 ~ "Europe", Continent==4 ~ "Africa", Continent==5 ~ "Asia", Continent == 6 ~ "Oceania", Continent==7 ~ "Antartica")) %>% select(Country, Continent = temp\_val)  
  
#country\_info <- select(country\_info2$temp\_val, country\_info2$Country)  
  
tourism\_temp <-read\_xlsx("/cloud/project/data/tourism.xlsx")  
  
# replace ".." values with NA values  
tourism\_temp[tourism\_temp == ".."] <- NA  
  
# this also works  
#tourism2 <- tourism2 %>% mutate\_all(~ str\_replace(., "\\.\\.", ""))  
  
#replace \_XXXX with YXXX in column names  
tourism\_temp <- rename\_with(tourism\_temp, ~ gsub("\_", "Y",.x, fixed=F))  
  
# write out the modified file in order to bring in the data as numeric, now that the char ".." are removed  
write\_xlsx(tourism\_temp, "/cloud/project/data/tourism2.xlsx")   
  
#remove tourism\_temp, country\_info2 from memory  
rm(tourism\_temp)  
rm(country\_info2)  
  
#read in the updated file specifying columns, suppress the warnings because each cell that is blank throws a warning.  
tourism <-suppressWarnings(read\_xlsx("/cloud/project/data/tourism2.xlsx" , col\_types=c("numeric", "text", "text","numeric","numeric","numeric","numeric","numeric","numeric","numeric","numeric","numeric","numeric","numeric","numeric","numeric","numeric","numeric","numeric","numeric","numeric","numeric","numeric" )))  
  
  
  
# create a primary key column, just in case we need to split data later.  
tourism$KEY\_COL <- NA  
for(i in 1:dim(tourism)[1])  
{  
 tourism$KEY\_COL[i] <- i  
 next;  
}  
  
  
#create a couple of new columns   
tourism$Country\_Name <- ""  
tourism$Tourism\_Type <- ""  
  
#convert Series to upper case letters  
tourism$Series <- toupper(tourism$Series)  
  
  
  
#function to standardize the data within COUNTRY  
type\_of\_tourism <- function(data) {  
 if(data == "Arrivals - Thousands") { return("Arrivals")}  
 else if(data == "Departures - Thousands") {return("Departures")}  
 else if(data == "Passenger transport - US$ Mn") { return( "Passenger Transport - US$") }  
 else if(data =="Tourism expenditure in other countries - US$ Mn") { return("Tourism expenditure in other countries - US$")}  
 else if(data == "Tourism expenditure in the country - US$ Mn") { return("Tourism expenditure in the country - US$")}  
 else if(data == "Travel - US$ Mn") {return("Travel - US$")}  
 else(return(NA))  
}  
  
#creates Country\_Name if the value in COUNTRY is actually a country, NA otherwise  
tourism$Country\_Name <- NA  
  
for(i in 1:length(tourism$COUNTRY)) {   
 if(tourism$COUNTRY[i] %in% country\_info$Country) {  
 tourism$Country\_Name[i] <- tourism$COUNTRY[i] }  
 else { tourism$Country\_Name[i] <- NA  
 }  
}  
  
  
#does vector-wise multiplication to correct for thousands or millions  
for(i in 1:(dim(tourism)-1)[1])  
 {  
  
 if(str\_detect(tourism$COUNTRY[i], "Thousands"))  
 {  
# print("Detected")  
 tourism[i,4:23] <-tourism[i,4:23]\*1000  
 next  
 }  
 if(str\_detect(tourism$COUNTRY[i], "US\\$ Mn"))  
 {  
# print("Detected million")  
 tourism[i,4:23] <- as.list(tourism[i,4:23]\*1000000)  
 next  
 }   
}  
  
#changes the data in Year columns to characters  
tourism[,4:23] <- sapply(tourism[,4:23], as.character)  
  
#copies country name downwards   
tourism <- transform(tourism, Country\_Name = na.locf(Country\_Name))  
  
max\_val <- 0  
max\_length <- 0  
#max\_length <- max(sapply(names(final\_tourism), nchar))  
  
  
#formats the Year Columns using prettynum  
for(i in 1:dim(tourism)[1])   
{  
 for(j in 4:24)  
 {  
  
 if(is.na(tourism[i,j])) {next}  
   
# if(tourism[i,j] >= max\_val)   
# {  
# max\_val <- tourism[i,j]  
# max\_length <- str\_length(max\_val)  
# }  
   
 str\_trim(tourism[i,j], side="both")  
# print(paste(i," ", j, tourism[i,j], prettyNum(tourism[i,j], format="d", big.mark=",", big.interval=3, preserve.width = "none") ))  
 tourism[i,j] <- StrAlign(str\_trim(prettyNum(tourism[i,j], format="d", big.mark=",", big.interval=3, width=max\_length+3 ), side="right"), sep="\r")  
   
 }  
}  
  
  
#format(final\_tourism$Y2019,width=name.width,justify="right")  
  
  
  
#create a copy of tourism2  
tourism2 <- tourism  
  
#use the custom function from ~line 80 to generate the Category of tourism  
tourism2 <- cbind(tourism2, "Category" = as.data.frame(sapply(tourism2$COUNTRY, type\_of\_tourism )))  
  
#for some reason the name wasn't taking, so manually assigned it again  
names(tourism2)[27] <-"Category"  
  
#create the Tourism\_type column  
tourism2$Tourism\_Type <- NA  
  
#generates the inbound/outbound tourism for the tourism\_type column   
for(i in 1:length(tourism2$COUNTRY))   
 {   
 if(tourism2$COUNTRY[i] == "Inbound tourism")   
 {  
 tourism2$Tourism\_Type[i] <- "Inbound tourism"   
 }  
 else if(tourism2$COUNTRY[i] == "Outbound tourism")   
 { tourism2$Tourism\_Type[i] <- "Outbound tourism"   
 }  
}  
  
#fills the tourism\_type column   
tourism2$Tourism\_Type[tourism2$Tourism\_Type == ""] <- NA  
tourism2 <- transform(tourism2, Tourism\_Type = na.locf(Tourism\_Type, na.rm=F))  
  
#begins to filter down the tourism2 data.frame   
tourism2 <- subset(tourism2, is.na(A))  
tourism2 <- subset(tourism2, COUNTRY != "Inbound tourism")   
tourism2 <- subset(tourism2, COUNTRY != "Outbound tourism")  
  
  
#create final data frames  
cleaned\_tourism <- NULL  
final\_tourism <- NULL  
final\_tourism\_xlsx <- createWorkbook()  
addWorksheet(final\_tourism\_xlsx, "Y2019")  
  
cleaned\_tourism <- select(tourism2, "Country\_Name", "Tourism\_Type", "Category", "Series", "Y2019")  
  
  
#uses an inner\_join to join the tables  
final\_tourism <- cleaned\_tourism %>% inner\_join(country\_info, by = c("Country\_Name"="Country"))  
  
  
#create a style, use the style to fill the datatable, keeping NA values, removing the filter option  
right\_just\_style <- createStyle(halign="right")  
addStyle(final\_tourism\_xlsx, 1, style=right\_just\_style, rows=1:dim(cleaned\_tourism)[1], cols=5 )  
writeDataTable(final\_tourism\_xlsx, 1, final\_tourism, tableStyle = "none", withFilter = FALSE, keepNA = TRUE)  
  
#write out final\_tourism data frame to a file  
saveWorkbook(final\_tourism\_xlsx, "/cloud/project/data/final\_tourism.xlsx", overwrite=TRUE)  
  
  
#remove old objects from memory  
rm(i)  
rm(j)  
rm(tourism)  
rm(tourism2)