Data Science Center Stats 2019

Ian May UID: 304663145

October 2, 2019

```r
library(readr)
library(dplyr)
```

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

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

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

```r
library(stringr)
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.1.1     v tidyr   0.8.3
## v tibble  2.1.1     v purrr   0.3.2
## v ggplot2 3.1.1     v forcats 0.4.0

## -- Conflicts ------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
events<- read_csv("lc_events_20191001012339.csv")
```

```
## Warning: Missing column names filled in: 'X10' [10]

## Parsed with column specification:
## cols(
##   .default = col_character(),
##   `Event ID` = col_double(),
##   Date = col_date(format = ""),
##   `End Time` = col_time(format = ""),
##   Setup = col_time(format = ""),
##   Teardown = col_time(format = ""),
##   X10 = col_logical(),
##   Seats = col_double(),
##   `Confirmed Registrations` = col_double(),
##   `Waiting-List Registrations` = col_double(),
##   `Cancelled Registrations` = col_double(),
##   `Anticipated Attendance` = col_double(),
##   `Actual Attendance` = col_double(),
##   `Confirmed Attendance` = col_double(),
##   `Phone Number` = col_logical(),
##   Barcode = col_logical(),
##   Created = col_datetime(format = "")
## )
## See spec(...) for full column specifications.
```

```
## Warning: 300 parsing failures.
## row col   expected     actual                           file
##   1  -- 39 columns 25 columns 'lc_events_20191001012339.csv'
##   2  -- 39 columns 25 columns 'lc_events_20191001012339.csv'
##   3  -- 39 columns 25 columns 'lc_events_20191001012339.csv'
##   4  -- 39 columns 25 columns 'lc_events_20191001012339.csv'
##   5  -- 39 columns 25 columns 'lc_events_20191001012339.csv'
## ... ... .......... .......... ..............................
## See problems(...) for more details.
```

```
events$Attend.All.Days <- events$`This workshop is held over 2 weeks on 4 days (Oct 16-17 & 23-24, 2019
events$`This workshop is held over 2 weeks on 4 days (Oct 16-17 & 23-24, 2019) from 9am-noon. Can you a

event <- read_csv("lc_events_20191001012339.csv")
```

```
## Warning: Missing column names filled in: 'X10' [10]
```

```
## Parsed with column specification:
## cols(
##   .default = col_character(),
##   `Event ID` = col_double(),
##   Date = col_date(format = ""),
##   `End Time` = col_time(format = ""),
##   Setup = col_time(format = ""),
##   Teardown = col_time(format = ""),
##   X10 = col_logical(),
##   Seats = col_double(),
##   `Confirmed Registrations` = col_double(),
##   `Waiting-List Registrations` = col_double(),
##   `Cancelled Registrations` = col_double(),
##   `Anticipated Attendance` = col_double(),
##   `Actual Attendance` = col_double(),
##   `Confirmed Attendance` = col_double(),
##   `Phone Number` = col_logical(),
##   Barcode = col_logical(),
##   Created = col_datetime(format = "")
## )
## See spec(...) for full column specifications.
```

```
## Warning: 300 parsing failures.
## row col   expected     actual                           file
##   1  -- 39 columns 25 columns 'lc_events_20191001012339.csv'
##   2  -- 39 columns 25 columns 'lc_events_20191001012339.csv'
##   3  -- 39 columns 25 columns 'lc_events_20191001012339.csv'
##   4  -- 39 columns 25 columns 'lc_events_20191001012339.csv'
##   5  -- 39 columns 25 columns 'lc_events_20191001012339.csv'
## ... ... .......... .......... ..............................
## See problems(...) for more details.
```

```
head(event)
```

```
## # A tibble: 6 x 39
##   `Event ID` Title Description Date       `Start Time` `End Time` Setup
##        <dbl> <chr> <chr>       <date>     <chr>        <time>     <tim>
## 1    4320496 Deve~ "Please jo~ 2018-07-26 10:00am      11:30         NA
## 2    4621279 ICPS~ "The Libra~ 2018-10-01 9:00am       13:00         NA
## 3    4626430 ICPS~ "The Libra~ 2018-10-02 9:00am       14:00         NA
## 4    4626437 ICPS~ "The Libra~ 2018-10-03 9:00am       14:00         NA
## 5    4626440 ICPS~ "The Libra~ 2018-10-04 9:00am       14:00         NA
## 6    4626443 ICPS~ "The Libra~ 2018-10-05 9:00am       13:00         NA
## # ... with 32 more variables: Teardown <time>, Location <chr>, X10 <lgl>,
## #   `Event Organizer` <chr>, Presenter <chr>, Audiences <chr>,
## #   Categories <chr>, `Publishing Status` <chr>, `Internal Tags` <chr>,
## #   Notes <chr>, `Registration Enabled` <chr>, Seats <dbl>, `Confirmed
## #   Registrations` <dbl>, `Waiting-List Registrations` <dbl>, `Cancelled
## #   Registrations` <dbl>, `Anticipated Attendance` <dbl>, `Actual
## #   Attendance` <dbl>, `Confirmed Attendance` <dbl>, Status <chr>,
## #   Attended <chr>, `First Name` <chr>, `Last Name` <chr>, Email <chr>,
## #   `Phone Number` <lgl>, Barcode <lgl>, Created <dttm>, `Status (Graduate
## #   Student, Faculty, Post Doc.)` <chr>, `Department or Program` <chr>,
## #   `Can we email you about future Data Science Center events &
## #   workshops?` <chr>, `What would you like to get out of this
## #   course?` <chr>, `Note:` <chr>, `This workshop is held over 2 weeks on
## #   4 days (Oct 16-17 & 23-24, 2019) from 9am-noon. Can you attend attend
## #   all 4 days? Please select all that apply:` <chr>
```

```
DSC <- event %>% filter(`Last Name` == "Jamison"|`First Name` == "Kristian" | `First Name` == "Leigh")

event$`Status (Graduate Student, Faculty, Post Doc.)`[event$`Last Name` == "Jamison"|event$`First Name`

#Ian's practice using functions, Found online

moveme <- function (invec, movecommand) {
  movecommand <- lapply(strsplit(strsplit(movecommand, ";")[[1]],
                               ",|\\s+"), function(x) x[x != ""])
  movelist <- lapply(movecommand, function(x) {
    Where <- x[which(x %in% c("before", "after", "first",
                             "last")):length(x)]
    ToMove <- setdiff(x, Where)
    list(ToMove, Where)
  })
  myVec <- invec
  for (i in seq_along(movelist)) {
    temp <- setdiff(myVec, movelist[[i]][[1]])
    A <- movelist[[i]][[2]][1]
    if (A %in% c("before", "after")) {
      ba <- movelist[[i]][[2]][2]
      if (A == "before") {
        after <- match(ba, temp) - 1
      }
      else if (A == "after") {
        after <- match(ba, temp)
      }
    }
    else if (A == "first") {
      after <- 0
    }
    else if (A == "last") {
      after <- length(myVec)
```

```r
    }
    myVec <- append(temp, values = movelist[[i]][[1]], after = after)
  }
  myVec
}

event$Attend.All.Days <- event$`This workshop is held over 2 weeks on 4 days (Oct 16-17 & 23-24, 2019)
event$`This workshop is held over 2 weeks on 4 days (Oct 16-17 & 23-24, 2019) from 9am-noon. Can you att
event$Organizer <- event$`Event Organizer`
event$`Event Organizer` <- NULL
event$ID <- event$`Event ID`
event$`Event ID` <- NULL
event$Start <- event$`Start Time`
event$`Start Time` <- NULL
event$End <- event$`End Time`
event$`End Time`<- NULL
event$Published <- event$`Publishing Status`
event$`Publishing Status` <- NULL
event$Registration.Enabled <- event$`Registration Enabled`
event$`Registration Enabled` <- NULL
event$First <- event$`First Name`
event$`First Name` <- NULL
event$Last <- event$`Last Name`
event$`Last Name` <- NULL
event$Phone <- event$`Phone Number`
event$`Phone Number` <- NULL
event$Attendee.Occupation <- event$`Status (Graduate Student, Faculty, Post Doc.)`
event$`Status (Graduate Student, Faculty, Post Doc.)` <-NULL
event$Attendee.Dept.program <- event$`Department or Program`
event$`Department or Program` <- NULL
event$`Email.opt.in?` <-  event$`Can we email you about future Data Science Center events & workshops?`
event$`Can we email you about future Data Science Center events & workshops?` <- NULL
event$Tags <- event$`Internal Tags`
event$`Internal Tags` <- NULL
event$Confirm.Reg <- event$`Confirmed Registrations`
event$`Confirmed Registrations` <-NULL
event$Wait.Reg <- event$`Waiting-List Registrations`
event$`Waiting-List Registrations`<- NULL
event$Cancel.Reg <- event$`Cancelled Registrations`
event$`Cancelled Registrations` <- NULL
event$Anticipated.Attend <- event$`Anticipated Attendance`
event$`Anticipated Attendance` <- NULL
event$Actual.Attend <- event$`Actual Attendance`
event$`Actual Attendance` <- NULL
event$Confirm.Attend <- event$`Confirmed Attendance`
event$`Confirmed Attendance` <- NULL
event$Event.Status <- event$Status
event$Status <- NULL
event$What.would.you.like.to.get.out.of.this.course <- event$`What would you like to get out of this cou
event$`What would you like to get out of this course?` <- NULL

#rearrange to reflect original dataset
```

```r
event <- event[moveme(names(event), "ID first; Start before Setup; End before Setup; Tags before Notes"
event <- event[moveme(names(event), "Published before Tags")]
event <- event[moveme(names(event)," Registration.Enabled before Seats")]
event <- event[moveme(names(event), "Confirm.Reg before Event.Status")]
event <- event[moveme(names(event), "Wait.Reg before Event.Status")]
event <- event[moveme(names(event), "Actual.Attend before Event.Status")]
event <- event[moveme(names(event), "Confirm.Attend before Event.Status")]
event <- event[moveme(names(event), "Event.Status before Attended")]
event <- event[moveme(names(event), "Phone before Barcode")]
event <- event[moveme(names(event), "Attendee.Dept.program before Barcode")]
event <- event[moveme(names(event), "Email.opt.in? before What.would.you.like.to.get.out.of.this.course
event <- event[moveme(names(event), "Attend.All.Days last")]
event <- event[moveme(names(event), "Note: before Attend.All.Days")]

names(events)
```

```
##  [1] "Event ID"
##  [2] "Title"
##  [3] "Description"
##  [4] "Date"
##  [5] "Start Time"
##  [6] "End Time"
##  [7] "Setup"
##  [8] "Teardown"
##  [9] "Location"
## [10] "X10"
## [11] "Event Organizer"
## [12] "Presenter"
## [13] "Audiences"
## [14] "Categories"
## [15] "Publishing Status"
## [16] "Internal Tags"
## [17] "Notes"
## [18] "Registration Enabled"
## [19] "Seats"
## [20] "Confirmed Registrations"
## [21] "Waiting-List Registrations"
## [22] "Cancelled Registrations"
## [23] "Anticipated Attendance"
## [24] "Actual Attendance"
## [25] "Confirmed Attendance"
## [26] "Status"
## [27] "Attended"
## [28] "First Name"
## [29] "Last Name"
## [30] "Email"
## [31] "Phone Number"
## [32] "Barcode"
## [33] "Created"
## [34] "Status (Graduate Student, Faculty, Post Doc.)"
## [35] "Department or Program"
## [36] "Can we email you about future Data Science Center events & workshops?"
## [37] "What would you like to get out of this course?"
## [38] "Note:"
## [39] "Attend.All.Days"
```

```
names(event)
```

```
##  [1] "ID"
##  [2] "Title"
##  [3] "Description"
##  [4] "Date"
##  [5] "Start"
##  [6] "End"
##  [7] "Setup"
##  [8] "Teardown"
##  [9] "Location"
## [10] "X10"
## [11] "Presenter"
## [12] "Audiences"
## [13] "Categories"
## [14] "Published"
## [15] "Tags"
## [16] "Notes"
## [17] "Registration.Enabled"
## [18] "Seats"
## [19] "Event.Status"
## [20] "Attended"
## [21] "Email"
## [22] "Phone"
## [23] "Attendee.Dept.program"
## [24] "Barcode"
## [25] "Created"
## [26] "Organizer"
## [27] "First"
## [28] "Last"
## [29] "Attendee.Occupation"
## [30] "Cancel.Reg"
## [31] "Anticipated.Attend"
## [32] "Confirm.Reg"
## [33] "Wait.Reg"
## [34] "Actual.Attend"
## [35] "Confirm.Attend"
## [36] "Email.opt.in?"
## [37] "What.would.you.like.to.get.out.of.this.course"
## [38] "Note:"
## [39] "Attend.All.Days"
```

```
event <- event[c(1:10, 26, 11:18, 32, 33, 30, 31, 34, 35, 19, 20, 27, 28, 21, 22, 24, 25, 29, 23, 36, 37

names(event) # Should reflect variable order in events
```

```
##  [1] "ID"
##  [2] "Title"
##  [3] "Description"
##  [4] "Date"
##  [5] "Start"
##  [6] "End"
##  [7] "Setup"
##  [8] "Teardown"
##  [9] "Location"
## [10] "X10"
## [11] "Organizer"
## [12] "Presenter"
## [13] "Audiences"
## [14] "Categories"
## [15] "Published"
## [16] "Tags"
## [17] "Notes"
## [18] "Registration.Enabled"
## [19] "Seats"
## [20] "Confirm.Reg"
## [21] "Wait.Reg"
## [22] "Cancel.Reg"
## [23] "Anticipated.Attend"
## [24] "Actual.Attend"
## [25] "Confirm.Attend"
## [26] "Event.Status"
## [27] "Attended"
## [28] "First"
## [29] "Last"
## [30] "Email"
## [31] "Phone"
## [32] "Barcode"
## [33] "Created"
## [34] "Attendee.Occupation"
## [35] "Attendee.Dept.program"
## [36] "Email.opt.in?"
## [37] "What.would.you.like.to.get.out.of.this.course"
## [38] "Note:"
## [39] "Attend.All.Days"
```

```r
dim(event) #Should be 300, 39
```

```
## [1] 300  39
```

```r
rm(events) # when finished
```

```r
event2 <- event
library(dplyr)


event2 <- select(event2,-c(Setup, Teardown, Tags, X10, Notes, Organizer, Phone, Barcode, `Note:`, First
#organizer deleted because only DSC related people were mentioned

#Make Start time the same format as End.

event2$Start <- strptime(event2$Start, "%I:%M %p")
event2$Start <- format(event2$Start, format="%H:%M:%S")
head(event2$Start, 10)
```

```
##  [1] "10:00:00" "09:00:00" "09:00:00" "09:00:00" "09:00:00" "09:00:00"
##  [7] "14:00:00" "14:00:00" "14:00:00" "14:00:00"
```

```r
library(hms)
event2$Start <- as.hms(event2$Start)

#Replace Start and End times with Event Duration

event2$Duration <- (event2$End - event2$Start)/60
event2 <- event2[moveme(names(event2), "Duration before Start")]
event2$Start <- NULL
event2$End <- NULL

#Rename locations

event2$Location[event2$Location == "WORKSHOPS: Data Science Center, 21536 Young Research Library"] <-"YR
event2$Location[event2$Location == "Research Library 11630F - RC Classroom"] <-"YRL- RC Classroom"
event2$Location[event2$Location == "Young Research Library, West Classroom"] <-"YRL- West Classroom"
event2$Location[event2$Location == "Research Library Conference Center"] <-"YRL- Conference Center"

#Create variables that show whether a programming language was taught or not

event2$Rprogramming <- ifelse(event2$Categories == "R (programming language)" | event2$Categories == "Da

event2$Python <- ifelse(event2$Categories == "Carpentries, Python" | event2$Categories == "Python" | eve

event2$OpenRefine <- ifelse(event2$Categories == "Data, Open Refine" | event2$Categories == "Open Refine

event2 <- event2[moveme(names(event2), "Rprogramming after Categories")]
event2 <- event2[moveme(names(event2), "Python after Rprogramming")]
event2 <- event2[moveme(names(event2), "OpenRefine after Python")]

#Amend Attended variable

event2$Attended[event2$Attended == "-"] <- "No" #28 variables left blank

#Simplify Occupation values

event2$Attendee.Occupation[event2$Attendee.Occupation == "doctoral student" | event2$Attendee.Occupation
event2$Attendee.Occupation[event2$Attendee.Occupation == "Faculty (Undergraduate)"] <- "Faculty"
event2$Attendee.Occupation[event2$Attendee.Occupation == "grad" | event2$Attendee.Occupation == "grad st
event2$Attendee.Occupation[event2$Attendee.Occupation == "grad student & staff" | event2$Attendee.Occupa
event2$Attendee.Occupation[event2$Attendee.Occupation == "Graduate Student and Faculty"] <- "Grad Studen
event2$Attendee.Occupation[event2$Attendee.Occupation == "staff" | event2$Attendee.Occupation == "Staff
event2$Attendee.Occupation[event2$Attendee.Occupation == "student" | event2$Attendee.Occupation == "Stud
event2$Attendee.Occupation[event2$Attendee.Occupation == "undergrad" | event2$Attendee.Occupation == "Un
event2$Attendee.Occupation[event2$Attendee.Occupation == "Visiting Graduate Student" | event2$Attendee.O
event2$Attendee.Occupation[event2$Attendee.Occupation == "Alum of UCR"] <- "Visitor"
event2$Attendee.Occupation[event2$Attendee.Dept.program == "UCLA Extension Applications Programming"] <-

#Rename "Building interactive data applications with Shiny" category
event2$Categories[event2$Title == "Building interactive data applications with Shiny"] <- "Data, R (prog

for(i in 1:ncol(event2)){
  cat(sum(is.na(event2[,i])), i, "\n")
}
```

```
## 0 1
## 0 2
## 3 3
## 0 4
## 8 5
## 51 6
## 24 7
## 158 8
## 34 9
## 34 10
## 34 11
## 34 12
## 0 13
## 0 14
## 0 15
## 0 16
## 0 17
## 0 18
## 0 19
## 218 20
## 0 21
## 28 22
## 28 23
## 28 24
## 37 25
## 39 26
## 39 27
## 117 28
## 300 29
```

```r
#eliminate variables with high levels of missing values ()
event2 <- event2[,-c(8, 20, 28, 29)]

#eliminate na's from rest of dataset

#Description
event2$Description[is.na(event2$Description)] <- "No Description"

#Categories
event2$Categories[event2$Title == "Python Book Club"] <- "Python"
event2$Categories[event2$Title == "Introduction to Tableau Data Visualization"] <- "Data"
event2$Categories[event2$Title == "R Working Group - MURP"] <- "R (programming language)"
event2$Categories[event2$Title == "Introduction to the Bash Shell Command Line"] <- "Bash"

#Python
event2$Python[event2$Title == "Python Book Club"] <- 1

#Location
event2$Location[grep("Presentation Room", event2$Description)] <- "YRL- Presentation Room"
event2$Location[grep("(FAIR)", event2$Description)] <- "Powell"
event2$Location[grep("IDRE Portal", event2$Description)] <- "IDRE Portal"

#Rprogramming, Python, Open Refine
r <- c("Shiny", "R programming", "RStudio", "tidyr", "dplyr", "stringr", "ggplot2")
```

```r
event2$Rprogramming[event2$Description %in% r ] <- 1
event2$Rprogramming[grep("R Working", event2$Title)] <- 1
py <- c("Jupyter", "Python")
event2$Python[event2$Description %in% py] <- 1
op_ref <- c("Open Refine")
event2$OpenRefine[event2$Description %in% op_ref] <- 1

#Personal Data
event2 <- event2 %>% filter(!is.na(Event.Status)) #eliminate 28 people who put down personal informatio
event2 <- event2 %>% mutate_at(vars("Attendee.Occupation", "Attendee.Dept.program", "Email.opt.in?"), ~

#Duration
#Eliminate day-long/multi-day events
event2<-event2[!(event2$Title=="GIS Day"|event2$Title=="Love Data Week 2019"|event2$Title=="Software Ca




for(i in 1:ncol(event2)){
  cat(sum(is.na(event2[,i])), i, "\n")
}
```

```
## 0 1
## 0 2
## 0 3
## 0 4
## 0 5
## 0 6
## 10 7
## 0 8
## 27 9
## 12 10
## 27 11
## 0 12
## 0 13
## 0 14
## 0 15
## 0 16
## 0 17
## 0 18
## 0 19
## 0 20
## 0 21
## 0 22
## 0 23
## 0 24
## 0 25
```

```r
#Only NA's left given to Python Fundamentals event

event2$`Email.opt.in?`<- ifelse(event2$`Email.opt.in?` == "Yes", "Yes", "No")
head(event2$`Email.opt.in?`)
```

```
## [1] "No" "No" "No" "No" "No" "No"
```

```r
# Questionable variables: Created, Published
event3 <- select(event2, -c("Created"))

library(xlsx)
write.xlsx(event3, file="clean_data.xlsx",
           sheetName="DSC_Stats_18_19")
```