Project Check in 1

2024-10-03

My Github

Below you will find the template for Project Check in 1. Each group member should submit a project check in document. Work with your team to ensure you limit duplicated work. The example code below is instructional and should not appear in your submission. Please do feel free to adapt it to your own dataset.

1. ***If your dataset has changed*** from what you submitted for Project Check in 0, please record:

- number of observations (rows)
- number of variables (columns)
- number of missing values
- names of particular columns of interest (if there are too many to print all of them!)
- data source and links to any accompanying documentation

```
library(dplyr)
library(lubridate)

intakes <- read.csv("dat/Austin_Animal_Center_Intakes_20241014.csv", header=T)
outcomes <- read.csv("dat/Austin_Animal_Center_Outcomes_20241014.csv", header=T)
aac_dataset <- merge(intakes, outcomes, by="Animal.ID", all.x=TRUE)

# here we will:
# - restrict the data down to intakes in the last 5 years only
# - sanitize the data of NA values in the identifier and intake age column
# - reduce the data to non-duplicated data only (see commented out code below)
aac_dataset <- aac_dataset[as.numeric(gsub("\\D", "", aac_dataset$MonthYear.x)) >= 2019,]
aac_dataset <- aac_dataset[as.numeric(gsub("\\D", "", aac_dataset$MonthYear.y)) >= 2019,]
aac_dataset <- subset(aac_dataset, !is.na(Animal.ID) & !is.na(Age.upon.Intake))
aac_dataset <- aac_dataset %>% distinct(Animal.ID, .keep_all=T)

# the commented out code below is a so far unsuccessful attempt at forming
# the duplicated entries into usable data. doing so is quite challenging
# as the amount of duplicates vary and then grouping each animal to apply a
# true/false scheme to each animal ID is rather challenging. however, I would
# like to return to this problem in the near future.

#keep_schemes <- list()
#for (i in 1:9) {
#  dupes <- i+1
#  scheme <- NULL
#  for (i in 1:i) {
#    scheme <- c(scheme, rep(F, dupes), T)
#  }
#  keep_schemes[i] <- list(scheme)
```

```r
#}
#
#filter_dupes <- function(dList) {
#  scheme <- keep_schemes[sqrt(length(dList)+1)-2]
#  print(subset(dList, scheme[[1]]))
#  subset(dList, scheme[[1]])
#}
#
#dupes <- aac_dataset[duplicated(aac_dataset$Animal.ID),]
#dupes %>% group_by(Animal.ID) %>% group_map(~ filter_dupes(.x))

# now we'll remove some duplicated/redundant columns
aac_dataset$Name.y <- NULL
aac_dataset$Animal.Type.y <- NULL
aac_dataset$Breed.y <- NULL
aac_dataset$Color.y <- NULL

# next, let's set proper typing
aac_dataset$DateTime.x <- mdy_hms(aac_dataset$DateTime.x)
aac_dataset$DateTime.y <- mdy_hms(aac_dataset$DateTime.y)
aac_dataset$Date.of.Birth <- mdy(aac_dataset$Date.of.Birth)

# last, we'll convert all ages to a decimal format representing years
convert_ages <- function(var) {
  for (i in 1:nrow(aac_dataset)) {
    split_age <- strsplit(aac_dataset[i, var], " ")

    # we check to make sure length is 2 as this is the expected format for this variable.
    if (length(split_age[[1]]) == 2) {
      age_num <- as.numeric(split_age[[1]][1])
      age_class <- split_age[[1]][2]
      if (age_class == "years") {
        aac_dataset[i, var] <- age_num
      } else if (age_class == "months") {
        aac_dataset[i, var] <- age_num / 12
      } else if (age_class == "weeks") {
        aac_dataset[i, var] <- age_num / 52.143
      } else if (age_class == "days") {
        aac_dataset[i, var] <- age_num / 365
      } else { # this case shouldn't occur, but if it does, add neg. val. for removal
        aac_dataset[i, var] <- -1
      }
    } else { # if length is not 2 as is expected, we mark for removal
      aac_dataset[i, var] <- -1
    }
  }
  return(aac_dataset)
}

aac_dataset <- convert_ages("Age.upon.Intake")
aac_dataset$Age.upon.Intake <- as.numeric(aac_dataset$Age.upon.Intake)
aac_dataset <- subset(aac_dataset, Age.upon.Intake >= 0)
aac_dataset <- convert_ages("Age.upon.Outcome")
```

```r
aac_dataset$Age.upon.Outcome <- as.numeric(aac_dataset$Age.upon.Outcome)
aac_dataset <- subset(aac_dataset, Age.upon.Outcome >= 0)

nrow(aac_dataset)
```

```
## [1] 47251
```

```r
ncol(aac_dataset)
```

```
## [1] 19
```

```r
sum(is.na(aac_dataset)) # No values missing
```

```
## [1] 0
```

```r
colnames(aac_dataset)[c(3, 5, 6, 7, 10, 11, 14, 16)]
```

```
## [1] "DateTime.x"       "Found.Location"   "Intake.Type"      "Intake.Condition"
## [5] "Age.upon.Intake"  "Breed.x"          "MonthYear.y"      "Outcome.Type"
```

Sources: Intake Dataset, Outcome Dataset

2. Show summary statistics for the five variables of the most interest.

```
range(aac_dataset$DateTime.x)
```

```
## [1] "2019-01-01 11:10:00 UTC" "2024-10-14 14:02:00 UTC"
```

```
median(aac_dataset$DateTime.x)
```

```
## [1] "2021-08-27 15:18:00 UTC"
```

```
table(aac_dataset$Intake.Condition)
```

```
##
##       Aged     Agonal   Behavior Congenital      Feral    Injured   Med Attn
##        163          2         47          1         30       4047         48
## Med Urgent    Medical   Neonatal Neurologic     Normal    Nursing      Other
##          9        329       1203          6      37268       1045        135
##    Panleuk      Parvo   Pregnant       Sick      Space    Unknown
##          1          6         61       2827          4         19
```

```
mean(aac_dataset$Age.upon.Intake)
```

```
## [1] 2.220368
```

```
sd(aac_dataset$Age.upon.Intake)
```

```
## [1] 3.021934
```

```
range(aac_dataset$Age.upon.Intake)
```

```
## [1]  0 30
```

```
bTbl <- data.frame(table(aac_dataset$Breed.x))
bTbl[bTbl$Freq > 500,]
```

```
##                         Var1  Freq
## 198                      Bat   873
## 499      Chihuahua Shorthair  1177
## 500  Chihuahua Shorthair Mix   903
## 667     Domestic Medium Hair  1075
## 669       Domestic Shorthair 13663
## 670   Domestic Shorthair Mix  3850
## 747          German Shepherd   901
## 748      German Shepherd Mix   938
## 917       Labrador Retriever  1012
## 918   Labrador Retriever Mix  1718
## 1153               Pit Bull  1865
## 1154           Pit Bull Mix  1586
```

```r
table(aac_dataset$Outcome.Type)
```

```
##
##                    Adoption            Died        Disposal      Euthanasia
##          22           24022             449             318            2540
##        Lost         Missing        Relocate Return to Owner       Rto-Adopt
##           3              21               4            5770             547
##      Stolen        Transfer
##           5           13550
```

3. Show another set of summary statistics by filtering on a column of interest. This could be years, teams, genres. Dig a little deeper here! Think about what might have a different distribution!

```
stray_df <- aac_dataset[aac_dataset$Intake.Type == "Stray",]
table(stray_df$Intake.Condition)
```

```
##
##       Aged     Agonal   Behavior Congenital      Feral    Injured   Med Attn
##        103          1         16          1         28       3300         36
## Med Urgent    Medical   Neonatal Neurologic     Normal    Nursing      Other
##          8        225       1006          4      24032        859         59
##      Parvo   Pregnant       Sick    Unknown
##          6         42       1803          6
```

```
table(stray_df$Outcome.Type)
```

```
##
##                         Adoption            Died        Disposal      Euthanasia
##               13            15670             326             137             816
##             Lost          Missing        Relocate Return to Owner       Rto-Adopt
##                2               15               4            3704             309
##           Stolen         Transfer
##                1            10538
```

```
aggregate(Age.upon.Intake ~ Animal.Type.x, data=stray_df, mean)
```

```
##   Animal.Type.x Age.upon.Intake
## 1          Bird        1.444721
## 2           Cat        1.201400
## 3           Dog        2.616585
## 4     Livestock        1.725000
## 5         Other        1.853271
```

```
aggregate(Age.upon.Outcome ~ Animal.Type.x, data=stray_df, mean)
```

```
##   Animal.Type.x Age.upon.Outcome
## 1          Bird         1.462441
## 2           Cat         1.249787
## 3           Dog         2.660891
## 4     Livestock         1.725000
## 5         Other         1.890574
```

4. Visualize the distribution of at least three variables. For example, below I've made a histogram to investigate the distribution of Petal Length. I've colored them by species. This is the base R way. You'll notice that we've had to be clever about how we set the x-axis here, it has to encompass the full range of Petal.Length.

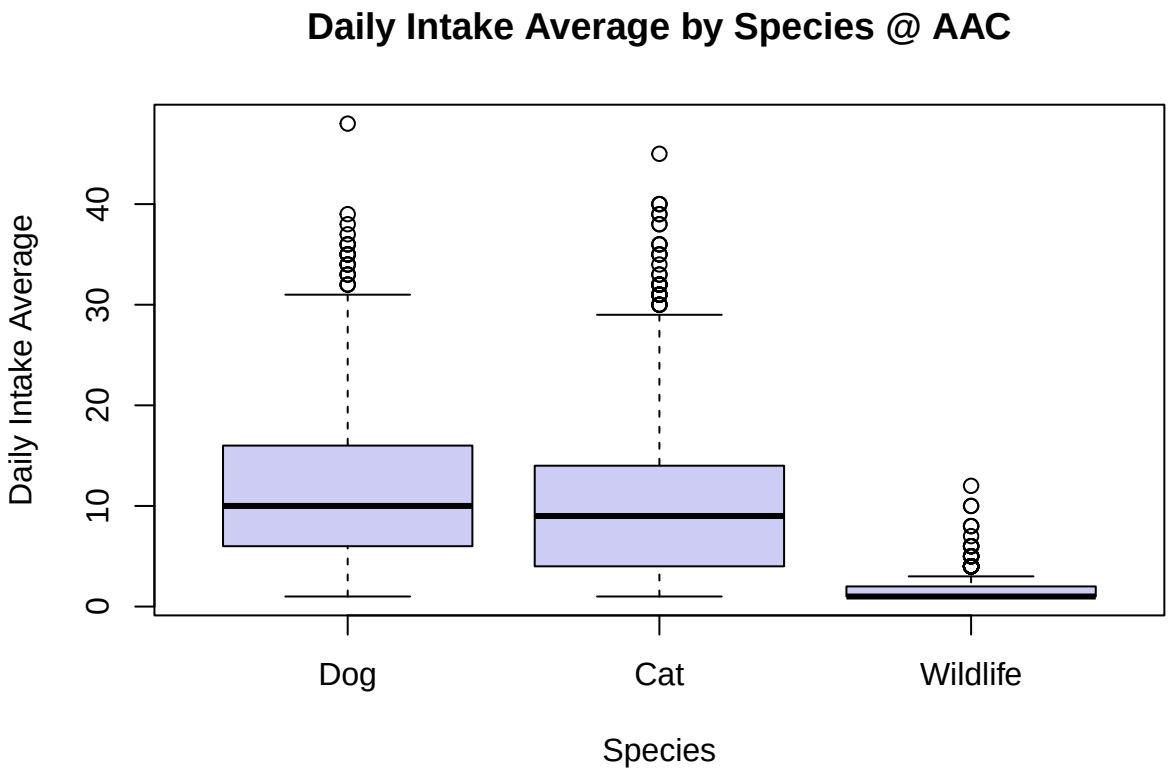
I expect plots to have sensible, nice-looking labels.

```
# 3 box plots (dog, cat, wildlife) demonstrating the daily intake averages
dog_df <- subset(aac_dataset, Animal.Type.x == "Dog")
cat_df <- subset(aac_dataset, Animal.Type.x == "Cat")
wdlf_df <- subset(aac_dataset, Intake.Type == "Wildlife")

dog_freq <- data.frame(table(date(dog_df$DateTime.x)))$Freq
cat_freq <- data.frame(table(date(cat_df$DateTime.x)))$Freq
wdlf_freq <- data.frame(table(date(wdlf_df$DateTime.x)))$Freq
max_freq <- max(length(dog_freq), length(cat_freq), length(wdlf_freq))
length(dog_freq) <- length(cat_freq) <- length(wdlf_freq) <- max_freq

freq_data = data.frame(Dog=dog_freq, Cat=cat_freq, Wildlife=wdlf_freq)

boxplot(freq_data, main="Daily Intake Average by Species @ AAC",
        xlab="Species", ylab="Daily Intake Average", col=rgb(0, 0, 0.8, 0.2))
```



5. Show three scatterplots that show the relationship between variables. Coloring data is a useful dimension to add here! There are many different ways to generate color palettes in R, but the general process here is the same:

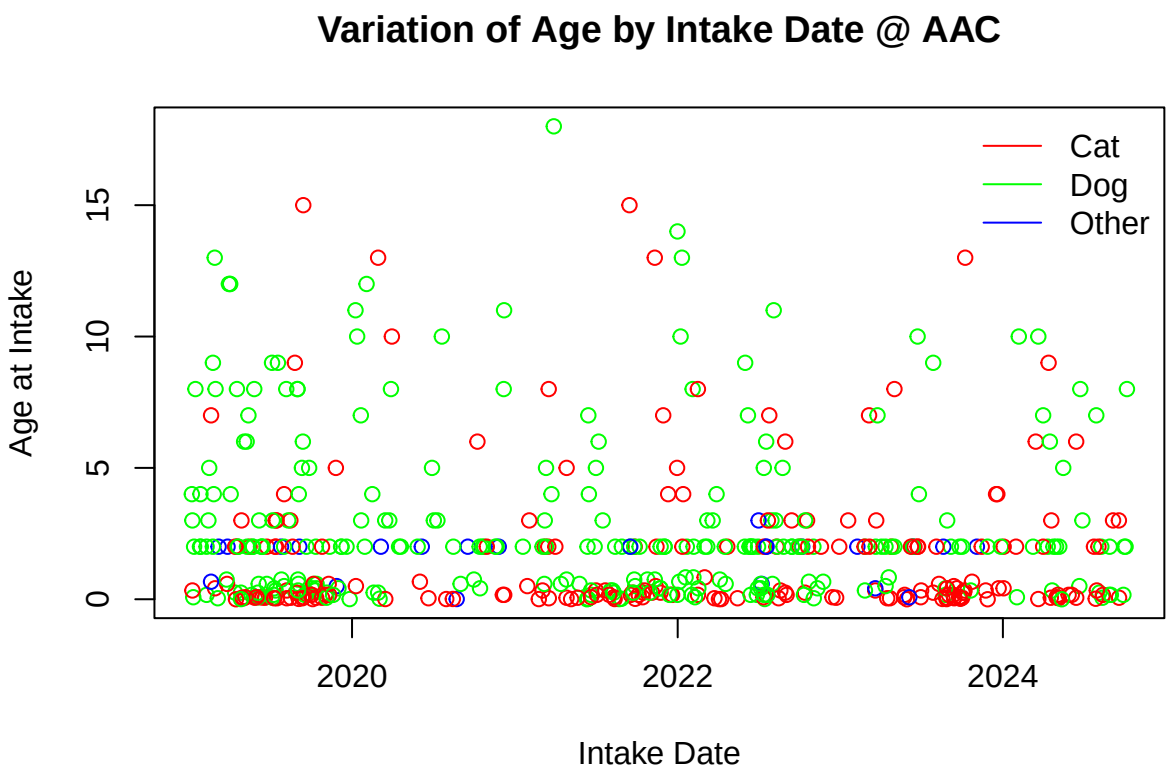
- define color palette, here my_colors (or pick your own hex codes or a fancy color brewer package)
- map the variable to the variables. We've done this using fields::color.scale() in class. Or get fancy with for loops and overplotting
- generate plot!

Extremely optional: You can turn your points into emojis with the emojifont package

```
library(fields)
set.seed(159)

# I used a sample here because the scatterplot with all data is barely readable
sample <- subset(
  sample_n(aac_dataset, 500),
  Animal.Type.x == "Cat" | Animal.Type.x == "Dog" | Animal.Type.x == "Other"
)

base_colors <- c("red", "green", "blue")
point_colors <- color.scale(sample$Animal.Type.x, col=base_colors)
plot(sample$DateTime.x, sample$Age.upon.Intake, col=point_colors,
     main="Variation of Age by Intake Date @ AAC", xlab="Intake Date",
     ylab="Age at Intake")
legend("topright", c("Cat", "Dog", "Other"), lty=rep(1, 3), col=base_colors, bty="n")
```



6. Write a few sentences about the observations you see in the above plots. Provide any context where necessary.

In the box plots, it's indicated that dogs generally have the highest daily intake average, while cats follow just shortly behind. This data is personally very surprising for me, as I expected the daily intake average of cats to be significantly lower. On the contrary, wildlife has a very low daily average, and outliers only reaching into the 10s, a stark contrast to the outliers in the 30s and 40s for dogs and cats. In the scatterplot, the data has less significance as animal shelters tend to operate their age system on a yearly basis if over a year, resulting in no variation from 1 to 2, 2 to 3, etc. However, the clearest trend is that the large majority of animals intake to the Austin Animal Center are younger than 5, with most being younger or at 2. Some outliers can be observed, the most extreme one being a dog marked at 18. As a side note, it must be taken into account that this data is only a sample size of 500 from a dataset of over 40000 entries, resulting in a large amount of outliers not being observed.