Data Wrangling

Rei, Alex, Andrés

4/22/2022

```
####################
### DATA LOADING ###
####################

acems <- read.csv("app/data/acems_metadata_2016_2022.csv") %>%
  select(
    "time_entered" = Timestamp,
    Date.of.Call,
    Time.of.Call,
    Location,
    Chief.Complaint,
    Result
  ) %>%
  janitor::clean_names()

acems <- acems %>%
  mutate(
    time_entered = parse_date_time(time_entered, "mdy HMS"),
    date_of_call = parse_date_time(date_of_call, "mdy"))

###################
### DATA FIXING ###
###################

# one call had date entered wrong (apparent from difference in time_entered
# and date_of_call)
acems$date_of_call[acems$date_of_call == ymd("2017-01-07")] <- ymd("2017-04-02")

# three calls had their years entered wrong
acems$date_of_call[acems$date_of_call == ymd("2016-03-01")] <- ymd("2017-03-01")
acems$date_of_call[acems$date_of_call == ymd("1999-04-29")] <- ymd("2019-04-29")
acems$date_of_call[acems$date_of_call == ymd("1998-09-22")] <- ymd("2019-09-22")

# one call has chief complaint "test run"; likely not real call
acems <- acems %>%
  filter(chief_complaint != "test run")

#################
### WRANGLING ###
#################

# returns the days as a factor
```

```r
acems <- acems %>%
  mutate(day_of_week = lubridate::wday(acems$date_of_call, label=TRUE, abbr = FALSE))

# weekend or not
acems <- acems %>%
  mutate(is_weekend = ifelse(day_of_week == "Sunday" |
                               day_of_week == "Saturday",
                               "Weekend", "Weekday"))

# convert is_weekend to factor
acems$is_weekend <- as.factor(acems$is_weekend)

# parse time and convert to military time
acems <- acems %>%
  separate(time_of_call, c('time_of_call', 'AM_PM'), sep = " ") %>%
  mutate(time_of_call = hms(time_of_call)) %>%
  mutate(time_of_call = case_when(
    AM_PM == "PM" & lubridate::hour(time_of_call) != 1 ~
      time_of_call + hours(12),
    TRUE ~
      time_of_call
  )) %>%
  select(-AM_PM)

acems$time_of_call <- hms(acems$time_of_call)

# day or night shift
acems <- acems %>%
  mutate(hour_of_call = lubridate::hour(time_of_call)) %>%
  mutate(shift_type = ifelse(
    hour_of_call >= 8 & hour_of_call < 20,
    "AM", "PM")) %>%
  select(-hour_of_call)

# convert shift_type to factor
acems$shift_type <- as.factor(acems$shift_type)

# extract month of call
acems <- acems %>%
  mutate(month_of_call = lubridate::month(date_of_call, label = TRUE, abbr = FALSE))

# month vectors based on semester
spring = c("January", "February", "March", "April", "May", "June")
fall = c("August", "September", "October", "November", "December")

# creating semester column
acems <- acems %>%
mutate(semester = case_when(
         month_of_call %in% spring ~ "Spring" ,
         month_of_call %in% fall ~ "Fall" ,
         TRUE ~ "NA"))

# creating academic year column
```

```r
acems <- acems %>%
mutate(academic_year = case_when(
         (semester == "Fall" & year(date_of_call) == 2016) |
         (semester == "Spring" & year(date_of_call) == 2017) ~ "2016-2017",
         (semester == "Fall" & year(date_of_call) == 2017) |
         (semester == "Spring" & year(date_of_call) == 2018) ~ "2017-2018",
         (semester == "Fall" & year(date_of_call) == 2018) |
         (semester == "Spring" & year(date_of_call) == 2019) ~ "2018-2019",
         (semester == "Fall" & year(date_of_call) == 2019) |
         (semester == "Spring" & year(date_of_call) == 2020) ~ "2019-2020",
         (semester == "Fall" & year(date_of_call) == 2020) |
         (semester == "Spring" & year(date_of_call) == 2021) ~ "2020-2021",
         (semester == "Fall" & year(date_of_call) == 2021) |
         (semester == "Spring" & year(date_of_call) == 2022) ~ "2021-2022",
         TRUE ~ "NA"))

# converting both columns to factors
acems$semester <- as.factor(acems$semester)
acems$academic_year <- as.factor(acems$academic_year)

# deselecting the time when call was entered
acems <- acems %>%
  select(-time_entered)
```

```r
### wrangling LOCATION column

# vector for correct location names
location_vector = c("Valentine Quad (Chapin/Fayerweather/Moore/Valentine)",
                    "Noah Webster Circle (Arms/Converse/Grosvenor/Morris Pratt/Morrow)")

acems$location[acems$location == "Freshman Quad"] <- "First-year Quad"
acems$location[acems$location == "Gym/Sports Fields/Chapman"] <- "Gym/Athletic Fields"
acems$location[acems$location == "Keefe/Beneski" |
               acems$location == "Keefe"] <- "Keefe/Beneski/Mead"
acems$location[acems$location == "Valentine/Moore"] <- location_vector[1]
acems$location[acems$location == "Powerhouse"] <- "Powerhouse/Police Station"
acems$location[acems$location == "Neport/Seligman"] <- "Newport/Seligman"
acems$location[acems$location == "Humphries"] <- "Chapman/Humphries/Lincoln/Rice"
acems$location[acems$location == "Noah Webster Circle (Arms/Grosvenor/Morris Pratt/Morrow)" |
acems$location == "Morris Pratt/Morrow" |
acems$location == "Noah Webster Circle (Grosvenor/Morris Pratt/Morrow)" |
acems$location == "Noah Webster Circle (Converse/Arms/Grosvenor/Morris Pratt/Morrow)" |
acems$location == "Converse" ] <- location_vector[2]
```

```r
### wrangling RESULT column
acems$result[str_detect(acems$result, "^AFD")] <- "AFD"
acems$result[str_detect(acems$result, "AFD$")] <- "AFD"
acems$result[acems$result == "Transport to UMASS"] <- "UMass transport"
acems$result[acems$result == "Refusal (taxi to CDH)"] <- "Taxi to CDH"
acems$result[acems$result == "AHS transport" |
             acems$result == "UHS"] <- "AHS/UHS Transport"
acems$result[grepl("Refusal", acems$result, fixed=TRUE) |
             acems$result == "refusal"] <- "Refusal"
```

```
acems$result[acems$result == "Transport to Counseling Center" |
             acems$result == "This didn't have to be a call"] <- "Other"
```

Chief Complaint

When metadata is entered into the spreadsheet for a call, the ACEMS member must specify a chief complaint. This can be selected from a dropdown menu or entered manually. Thus, our data has a lot of misspellings and different wordings for similar complaints. In this section, we group all of these into uniform spellings and wordings.

```
### wrangling chief_complaint column

acems$chief_complaint[grepl("chest", acems$chief_complaint, fixed=TRUE) |
                      grepl("Chest", acems$chief_complaint, fixed=TRUE)] <- "Chest Pain"

acems$chief_complaint[grepl("dizziness", acems$chief_complaint, fixed=TRUE) |
                    grepl("Dizziness", acems$chief_complaint, fixed=TRUE)] <- "Dizziness"

acems$chief_complaint[grepl("ear", acems$chief_complaint, fixed=TRUE) |
                     grepl("Ear", acems$chief_complaint, fixed=TRUE)] <- "Ear Infection"

acems$chief_complaint[grepl("eye", acems$chief_complaint, fixed=TRUE) |
                      grepl("Eye", acems$chief_complaint, fixed=TRUE) |
        acems$chief_complaint == "ocular inflammation"]  <- "Eye Infection/Inflammation"

acems$chief_complaint[grepl("bite", acems$chief_complaint, fixed=TRUE) |
                      grepl("Bite", acems$chief_complaint, fixed=TRUE) |
                      grepl("Tick", acems$chief_complaint, fixed=TRUE) |
                      grepl("sting", acems$chief_complaint, fixed=TRUE) |
                    grepl("Insect", acems$chief_complaint, fixed=TRUE)] <- "Insect"

acems$chief_complaint[grepl("panic", acems$chief_complaint, fixed=TRUE) |
                    grepl("Panic", acems$chief_complaint, fixed=TRUE)] <- "Panic Attack"

acems$chief_complaint[grepl("burn", acems$chief_complaint, fixed=TRUE) |
                      grepl("Burn", acems$chief_complaint, fixed=TRUE)] <- "Burn"

acems$chief_complaint[grepl("abdominal", acems$chief_complaint, fixed=TRUE) |
                      grepl("Abdominal", acems$chief_complaint, fixed=TRUE) |
                      acems$chief_complaint == "kidney pain"] <- "Acute Abdominal Pain"

acems$chief_complaint[acems$chief_complaint == "leg pain" |
                      acems$chief_complaint == "Neck pain" |
                      grepl("ache", acems$chief_complaint, fixed=TRUE) |
                      acems$chief_complaint == "Arm & Shoulder Pain"] <- "Body Aches"

acems$chief_complaint[grepl("allergic", acems$chief_complaint, fixed=TRUE) |
      grepl("Allergic", acems$chief_complaint, fixed=TRUE)] <- "Allergic Reaction"

acems$chief_complaint[grepl("laceration", acems$chief_complaint, fixed=TRUE) |
                      acems$chief_complaint == "Hemorrhage" |
      grepl("Laceration", acems$chief_complaint, fixed=TRUE)] <- "Laceration/Hemorrhage"
```

```
acems$chief_complaint[grepl("food", acems$chief_complaint, fixed=TRUE) |
                      grepl("Food", acems$chief_complaint, fixed=TRUE) |
                      acems$chief_complaint == "nutrition"] <- "Food Poisoning"

acems$chief_complaint[acems$chief_complaint == "Anaphalaxsis" |
                      acems$chief_complaint == "Anaphylaxsis" |
                      acems$chief_complaint == "shock" ] <- "Anaphylaxis"

acems$chief_complaint[grepl("rash", acems$chief_complaint, fixed=TRUE) |
                      grepl("Rash", acems$chief_complaint, fixed=TRUE)] <- "Rash"

acems$chief_complaint[acems$chief_complaint == "Mental Health" |
                      acems$chief_complaint == "Mental Health Related" |
                      acems$chief_complaint == "anxiety" |
                      acems$chief_complaint == "Fatigue" |
acems$chief_complaint == "Psychiatric Crisis"] <- "Behavioral/Altered Mental Status"

acems$chief_complaint[acems$chief_complaint == "nausea/vomitting" |
      acems$chief_complaint == "Gastrointestinal (Nausea, Comitting, and/or Diarrhea)" |
      acems$chief_complaint == "Nausea" |
      acems$chief_complaint == "stomach pain and general malaise" |
      acems$chief_complaint == "Gastrointestinal/Syncope"] <- "Gastrointestinal"

acems$chief_complaint[grepl("head", acems$chief_complaint, fixed=TRUE) |
                      grepl("Head", acems$chief_complaint, fixed=TRUE)] <- "Head Injury"

acems$chief_complaint[acems$chief_complaint == "Asthma"] <- "Respiratory Distress"

acems$chief_complaint[acems$chief_complaint == "Scrape"] <- "Abrasion"

acems$chief_complaint[acems$chief_complaint == "Rash" |
                      acems$chief_complaint == "Hives"] <- "Rash/Hives"

acems$chief_complaint[acems$chief_complaint == "Flu-like symptoms"] <- "Flu-Like Symptoms"

acems$chief_complaint[acems$chief_complaint == "Hypoglycemia"] <- "Diabetic Emergency"

acems$chief_complaint[grepl("injury", acems$chief_complaint, fixed=TRUE) |
                      grepl("Injury", acems$chief_complaint, fixed=TRUE) &
                      acems$chief_complaint != "Musculoskeletal Injury" &
                      acems$chief_complaint != "Soft Tissue Injury"] <- "Other Injury"


acems$chief_complaint[grepl("no patient found", acems$chief_complaint, fixed =
                            TRUE)] <- "No Patient Found"

acems$chief_complaint[grepl("Icepack", acems$chief_complaint, fixed = TRUE)] <-
  "Icepack Requested"
```

```
# Rename "other" chief complaints (ones with only one occurrence) into something
# more general or coinciding with another category
renames <- read_csv("app/data/chief_complaints_other_renames.csv")
acems <- acems %>%
```

```
  left_join(renames) %>%
  mutate(chief_complaint = case_when(
    is.na(rename) ~ chief_complaint,
    TRUE ~ rename
  )) %>%
  select(-rename)

# This file was used to create grouping categories in chief_complaints_grouping
acems %>%
  group_by(chief_complaint) %>%
  summarise(N = n()) %>%
  arrange(desc(N)) %>%
  write_csv("app/data/chief_complaints.csv")

grouping <- read_csv("app/data/chief_complaints_grouping.csv")

acems <- acems %>%
  left_join(grouping)

acems %>%
  group_by(category) %>%
  summarize(calls = n()) %>%
  arrange(desc(calls)) %>%
  kable(booktabs=TRUE)
```

category	calls
General Injury	389
Alcohol/Drugs	321
Pathogenic	94
Neurological	72
Metabolic/Gastrointestinal	61
Cardiovascular/Respiratory	48
Internal Pain	45
Allergic Reaction	35
Behavorial/Psychiatric	25
Environmental	21
Other	12
Dermatological	5

```
# write the wrangled data set to csv file
acems %>% write_csv("app/data/acems.csv")
```