Untitled

Miao Cai

5/28/2020

Read and clean data

```
pacman::p_load(data.table, dplyr, lubridate, fst)

# 1. original ping -> d
d = fread("data/original/Meeeting-11-16-18%2Fpings_500drivers_11.csv") %>%
  .[,.(driver = gsub("\"", "", V7), ping_time = ymd_hms(V2),
       speed = V3, lat = V4, lon = V6)]  %>%
  .[!is.na(ping_time)] %>%
  setkey(driver, ping_time) %>%
  .[,diff := as.integer(difftime(ping_time,
                shift(ping_time, type = "lag",
                      fill = 0), units = "mins")), driver] %>%
  .[,diff := {diff[1] = 0L; diff}, driver]
fst::write_fst(d, "data/cleaned/01a_ping_original_500drivers.fst", compress = 100)

# 1b. create driver list -> d_list
d_list = d[,.(n_ping = .N), driver]
fst::write_fst(d_list, "data/cleaned/00driver_list.fst", compress = 100)

# 2. weather -> w
w = vroom::vroom(dir("data/weather", "\\.csv$", full.names = T)) %>%
  as.data.table() %>%
  .[,DATIME := lubridate::mdy_hm(DATIME)] %>%
  .[,.(DATIME, LATITUDE, LONGITUDE, PRECIP_INTENSITY, PRECIP_PROBABILITY,
       WIND_SPEED, VISIBILITY, SUNRISE_TIME, SUNSET_TIME,
       DURING_SUNSET, DURING_DUSK, DURING_SUNRISE, DURING_DAWN)]
fst::write_fst(w, "data/cleaned/02weather.fst", compress = 100)

# 3. driver demographic -> dinfo
alldr = fread("data/original/ALL_DRIVERS_DATA2016-09-30 10-53-42.csv") %>%
  .[,EMPLID := stringr::str_replace_all(EMPLID, " ", "")]
d_list = fst::read_fst("data/cleaned/driver_list.fst")
dinfo = fread("data/original/ALPHA_TO_EMPLID2016-10-21 14-00-24.csv") %>%
  .[,driver := tolower(ALPHA)] %>%
  .[,`:=`(driver = stringr::str_replace_all(driver, " ", ""),
          EMPLID = stringr::str_replace_all(EMPLID, " ", ""))] %>%
  .[,.(driver, EMPLID)] %>%
  setkey(driver) %>%
  merge(d_list, by = "driver", all.y = TRUE) %>%
  merge(alldr, by = "EMPLID", all.x = TRUE) %>%
  .[,BIRTHDATE := ymd(BIRTHDATE)] %>%
```

```
  .[!is.na(BIRTHDATE),] %>%
  setkey(driver) %>%
  .[,n_missing := rowSums(is.na(.))] %>%
  .[order(driver, -n_missing)] %>%
  .[,head(.SD, 1), by = driver] %>%
  .[, age := 2015 - year(BIRTHDATE)] %>%
  #.[!(driver %in% c("kisi", "codc"))] %>%
  .[,.(driver, EMPLID, age, race = ETHNIC_GROUP, gender = GENDER)] %>%
  .[,race := case_when(race == "BLACK     " ~ "Black",
                       race == "WHITE     " ~ "White",
                       TRUE ~ "Other")]
fst::write_fst(dinfo, "data/cleaned/03driver_information.fst", compress = 100)

# 4. SCEs -> ce
dinfo = fst::read_fst("data/cleaned/03driver_information.fst")
ce = fread("data/original/CRITICAL_EVENT_QUERY2016-09-30 10-58-28.csv") %>%
  .[,`:=`(EMPLID = stringr::str_replace_all(EMPLID, " ", ""),
          EVT_TYP = stringr::str_replace_all(EVT_TYP, " ", ""))] %>%
  .[,event_time := ymd_hms(paste(EVENT_DATE, EVENT_HOUR, sep = " "))] %>%
  .[,.(EMPLID, event_time, event_type = EVT_TYP)] %>%
  merge(dinfo, by = "EMPLID", all.y = TRUE) %>%
  .[!is.na(event_time),.(driver, event_time, event_type)] %>%
  .[,event_type := case_when(event_type == "HEADWAY" ~ "HW",
                             event_type == "HARD_BRAKING" ~ "HB",
                             event_type == "COLLISION_MITIGATION" ~ "CM",
                             event_type == "ROLL_STABILITY" ~ "RS")] %>%
  unique()
fst::write_fst(ce, "data/cleaned/04safety_critical_events.fst", compress = 100)
```

Mark shift and trip ID

```
pacman::p_load(data.table, dplyr, lubridate, fst)
d = fst::read_fst("data/cleaned/01a_ping_original_500drivers.fst") %>%
  as.data.table()

## PART 1: mark all ping with shift_id
threshold_shift = 8*60
s1 = d %>%
  .[diff >= threshold_shift|speed <= 10, speed := 0] %>%
  .[,rleid := rleid(speed != 0), driver] %>%
  .[,`:=`(speed1 = speed)] %>%
  .[,`:=`(sum_speed = sum(speed), sum_time = sum(diff)), .(driver, rleid)] %>%
  .[sum_speed == 0 & sum_time < threshold_shift, speed1 := 3] %>%
  .[,`:=`(sum_speed = sum(speed1)), .(driver, rleid)] %>%
  .[,shift_id := fifelse(sum_speed == 0, 0, rleid(speed1 != 0)), driver] %>%
  .[,`:=`(rev_cums_sp = rev(cumsum(rev(speed))),
          cums_sp = cumsum(speed)), .(driver, shift_id)] %>%
  .[rev_cums_sp == 0|cums_sp == 0, shift_id := 0]

# STATS: pings
s1[shift_id == 0,.N] # 3,550,935 -> shift_id == 0
s1[,round(sum(shift_id == 0)*100/.N, 2)] # percent of stopping pings: 26.93%
s1[shift_id != 0,.N] # 9,636,349 -> shift_id != 0
```

```
s1[,round(sum(shift_id != 0)*100/.N, 2)] # 73.07%

s_len = s1 %>%
  .[shift_id != 0] %>%
  .[,.(driver, shift_id, shift_length = sum(diff)), .(driver, shift_id)]

# STATS: shifts
s_len[,.N] # 77,870 unique shifts
s_len[shift_length > 14*60, .N] # 2,198 very long shifts
s_len[, round(sum(shift_length > 14*60)*100/.N, 2)] # 1.72%
s_len[shift_length <= 0.5*60, .N] # 773 very short shifts
s_len[, round(sum(shift_length <= 0.5*60)*100/.N, 2)] # 0.99%
s_len[shift_length > 0.5*60 & shift_length <= 14*60, .N] # 75,760 eligible shifts
s_len[, round(sum(shift_length > 0.5*60 & shift_length <= 14*60)*100/.N, 2)] # 97.29%


# filter eligible shifts
s2 = s1 %>%
  .[shift_id != 0] %>%
  .[,shift_length := sum(diff), .(driver, shift_id)] %>%
  .[,.(driver, ping_time, speed, lat, lon, shift_id, shift_length)] %>%
  .[shift_length > 30 & shift_length <= 14*60] %>%
  .[,shift_length := NULL] %>%
  setkey(driver, ping_time)

# STATS: pings
s1[shift_id == 0,.N] # 3,550,935 -> shift_id == 0
s1[,sum(shift_id == 0)/.N] # percent of stopping pings: 26.93%
s2[,.N] # 9,349,312
round(s2[,.N]*100/s1[,.N], 2) # 70.9%

fst::write_fst(s2, "data/cleaned/01b_ping_shift_id_500drivers.fst")


## PART 2: mark all ping with trip_id ##
threshold_trip = 30
t1 = s2 %>%
  setkey(driver, ping_time) %>%
  .[,diff := as.integer(difftime(ping_time, shift(ping_time,
              type = "lag",fill = 0), units = "mins")), driver] %>%
  .[,diff := {diff[1] = 0L; diff}, driver] %>%
  .[diff >= threshold_trip, speed := 0] %>%
  .[,rleid := rleid(speed != 0), driver] %>%
  .[,`:=`(rleid1 = rleid, speed1 = speed)] %>%
  .[,`:=`(sum_speed = sum(speed), sum_time = sum(diff)), .(driver, rleid)] %>%
  .[sum_speed == 0 & sum_time < threshold_trip, speed1 := 3] %>%
  .[,`:=`(sum_speed = sum(speed1)), .(driver, rleid)] %>%
  .[,trip_id := data.table::fifelse(sum_speed == 0, 0,
              rleid(speed1 != 0)), driver] %>%
  .[trip_id != 0,trip_length := sum(diff), .(driver, trip_id)] %>%
  .[,.(driver, ping_time, speed, lat, lon, diff, shift_id, trip_id)]

t2 = t1 %>%
```

```
  .[trip_id != 0,] %>%
  .[,`:=`(lon1 = shift(lon, type = "lag", fill = NA),
          lat1 = shift(lat, type = "lag", fill = NA)),
    by = .(driver, shift_id, trip_id)] %>%
  .[,distance := geosphere::distHaversine(cbind(lon, lat), cbind(lon1, lat1))] %>%
  .[,distance := round(distance/1609.344, 3)] %>%
  .[,distance := {distance[1] = 0; distance}, .(driver, shift_id, trip_id)] %>%
  .[,c("lon1", "lat1") := NULL] %>%
  setkey(driver, ping_time)

fst::write_fst(t2, "data/cleaned/01c_ping_trip_id_500drivers.fst")
```

Aggregate data

```
pacman::p_load(data.table, dplyr, lubridate, fst)

d = fst::read_fst("data/cleaned/01c_ping_trip_id_500drivers.fst") %>%
  as.data.table() %>%
  setkey(driver, ping_time)
w = fst::read_fst("data/cleaned/02weather.fst") %>% as.data.table()
dinfo = fst::read_fst("data/cleaned/03driver_information.fst") %>% as.data.table()
ce = fst::read_fst("data/cleaned/04safety_critical_events.fst") %>% as.data.table()

# Merge weather & driver information to ping
d1 = d %>%
  .[,`:=`(ping_id = 1:.N,
          DATIME = lubridate::floor_date(ping_time, "hours"),
          LATITUDE = as.numeric(gsub("([0-9]+\\.[0-9]{2})(.*)", "\\1", lat)),
          LONGITUDE = as.numeric(gsub("([0-9]+\\.[0-9]{2})(.*)", "\\1", lon)))] %>%
  merge(w, by = c("DATIME", "LATITUDE", "LONGITUDE"), all.x = TRUE) %>%
  merge(dinfo[,.(driver, age, race, gender)], by = "driver", all.x = TRUE)

########################  trip  ########################
dtrip = d1 %>%
  .[trip_id != 0,] %>%
  setkey(driver, ping_time) %>%
  .[,.(start_time = ping_time[1], end_time = ping_time[.N],
       start_lat = LATITUDE[1], start_lon = LONGITUDE[1],
       end_lat = LATITUDE[.N], end_lon = LONGITUDE[.N],
       speed_mean = mean(speed, na.rm = TRUE),
       speed_sd = sd(speed, na.rm = TRUE),
       distance = sum(distance, na.rm = TRUE),
       age = age[1], race = race[1], gender = gender[1],
       prep_inten = mean(PRECIP_INTENSITY, na.rm = TRUE),
       prep_prob = mean(PRECIP_PROBABILITY, na.rm = TRUE),
       wind_speed = mean(WIND_SPEED, na.rm = TRUE),
       visibility = mean(VISIBILITY, na.rm = TRUE),
       sunrise = fifelse(sum(DURING_SUNRISE == "Y", na.rm = TRUE) > 0, 1, 0),
       sunset = fifelse(sum(DURING_SUNSET == "Y", na.rm = TRUE) > 0, 1, 0),
       dusk = fifelse(sum(DURING_DUSK == "Y", na.rm = TRUE) > 0, 1, 0),
       dawn = fifelse(sum(DURING_DAWN == "Y", na.rm = TRUE) > 0, 1, 0)),
    .(driver, shift_id, trip_id)] %>%
  .[,`:=`(trip_time = as.integer(difftime(end_time, start_time,
```

```
                                          units = "mins")),
          speed_sd  = fifelse(is.na(speed_sd), 0, speed_sd),
          prep_inten = fifelse(is.na(prep_inten), 0, prep_inten),
          prep_prob = fifelse(is.na(prep_prob), 0, prep_prob),
          wind_speed = fifelse(is.na(wind_speed), mean(wind_speed, na.rm = TRUE),
                               wind_speed),
          visibility = fifelse(is.na(visibility), mean(visibility, na.rm = TRUE),
                               visibility))] %>%
  .[order(driver, trip_id)]

trip_range = dtrip %>%
  .[,.(driver, shift_id, trip_id, start_time, end_time)] %>%
  setkey(driver, start_time, end_time)

n_CE_trip = ce %>%
  .[,dummy := event_time] %>%
  setkey(driver, event_time, dummy) %>%
  foverlaps(trip_range, mult = "all", type = "within", nomatch = NA) %>%
  .[!is.na(shift_id)|!is.na(trip_id),] %>%
  .[,.(driver, shift_id, trip_id, event_time, event_type)] %>%
  .[,.(nCE = .N), .(driver, shift_id, trip_id)] %>%
  setkey(driver, shift_id, trip_id)

dtrip1 = dtrip %>%
  merge(n_CE_trip, by = c('driver', 'shift_id', 'trip_id'), all.x = TRUE) %>%
  .[, nCE := fifelse(is.na(nCE), 0, nCE)]
fst::write_fst(dtrip1, "data/cleaned/12dtrip.fst", compress = 100)

########################  shift  ########################
dshift = d1 %>%
  .[,.(start_time = ping_time[1], end_time = ping_time[.N],
       start_lat = LATITUDE[1], start_lon = LONGITUDE[1],
       end_lat = LATITUDE[.N], end_lon = LONGITUDE[.N]),
    .(driver, shift_id)] %>%
  .[,shift_time := as.integer(difftime(end_time, start_time,
                                       units = "mins"))] %>%
  .[order(driver, shift_id)]
fst::write_fst(dshift, "data/cleaned/13dshift.fst", compress = 100)

######################## intervals ########################
dtrip = fst::read_fst("data/cleaned/12dtrip.fst") %>% as.data.table()
dshift = fst::read_fst("data/cleaned/13dshift.fst") %>% as.data.table()

source("data/function_mkint.R")
agg_int30 = mkint(30)
agg_int60 = mkint(60)

# add critical events
ce = fst::read_fst("data/cleaned/04safety_critical_events.fst") %>% as.data.table()
source("data/function_indexce.R")
ceindexed30 = indexce(agg_int30, ce)
ceindexed60 = indexce(agg_int60, ce)
```

```
ce_int30 = agg_int30 %>%
  merge(ceindexed30[,.(nCE = .N), .(driver, interval_id)],
        by = c("driver", "interval_id"), all.x = TRUE) %>%
  .[,nCE := fifelse(is.na(nCE), 0, nCE)]
ce_int60 = agg_int60 %>%
  merge(ceindexed60[,.(nCE = .N), .(driver, interval_id)],
        by = c("driver", "interval_id"), all.x = TRUE) %>%
  .[,nCE := fifelse(is.na(nCE), 0, nCE)]

# delete shifts with more than 11 hours of driving time
ce_int30_11 = ce_int30 %>%
  .[,max_drive := sum(interval_time), .(driver, shift_id)] %>%
  .[max_drive <= 11*60]
ce_int60_11 = ce_int60 %>%
  .[,max_drive := sum(interval_time), .(driver, shift_id)] %>%
  .[max_drive <= 11*60]
```

Hierarchical logisitc and negative binomial models

```
pacman::p_load(data.table, dplyr, ggplot2, fst, MASS)
d = fst::read_fst("data/cleaned/32interval30_CE_11hours_limit.fst") %>% as.data.table()

z = d %>%
  .[,`:=`(cumdrive = cumdrive/60,
          CE_binary = fifelse(nCE > 0, 1, 0),
          race = factor(race, levels = c("White", "Black", "Other")),
          gender = factor(gender, levels = c("M", "F", "U")))]

f_logit = glm(CE_binary ~ cumdrive + speed_mean + speed_sd + age + race + gender +
                prep_inten + prep_prob + wind_speed + visibility + interval_time,
              family = "binomial", data = z)
saveRDS(f_logit, "fit/f_logit.rds")

f_nb = glm.nb(nCE ~ cumdrive + speed_mean + speed_sd + age + race + gender +
                prep_inten + prep_prob + wind_speed + visibility +
                offset(log(interval_time)),
                data = z)
saveRDS(f_nb, "fit/f_nb.rds")
```