```
#  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
#  Bus-route creel survey estimation
#  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

# Zhenming Su, IFR, DNR, ANN ARBOR, MI
# Contact Zhenming Su (suz@michigan.gov) for any questions

# ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# Built: Sept 1, 2021
#        Sept 2, 2021 worked
# ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

f_catch_rate_bus_route <- function(CdHd)
{
    # ------------------------------------------------------------
    # Ratio of means estimation of catch rate and variance
    #   Used for completed trips
    # ------------------------------------------------------------
    Cd <- CdHd[,1]
    Hd <- CdHd[,2]

    # ratio of means
    Rd  <- mean(Cd)/mean(Hd)

    SSQ_C   <- crossprod(Cd); #sum of squares
    SSQ_H   <- crossprod(Hd);
    SPD_C_H <- crossprod(Cd, Hd);

    # calculate variance of Rd
    n_int <- length(Hd);

    mHd <- mean(Hd)
    term1 <- 1/(n_int * mHd * mHd);
    term2 <- 2 * Rd * SPD_C_H;
    term3 <- Rd * Rd * SSQ_H;

    # Eq. 2
    V_R <- term1 * (SSQ_C - term2 + term3)/(n_int - 1);

    c(Rd, V_R)
}

bus_route_site_estimation <-  function(BoatCounts_sites, total_route_time, wait_time,
trip_interviews)
{
    # Make daily estimates of access sites on the bus route

    # Total boat counts made at each access site of the daily bus-route
    # BoatCounts_sites <- aggregate(list(FishBoatCounts = boat_counts$FishingBoatCount
s), list(Day = as.factor(boat_counts$Day), AccessSite = boat_counts$AccessSite), sum)

    # Averages of site-specific weighted trip lengths or catch for each sampled day
    #  Q can be the trip length or catch of a species
    Avg_trip_Q <- aggregate(list(m_trip_Q = trip_interviews$Q), list(Day = trip_interv
iews$Day, AccessSite = trip_interviews$AccessSite), mean)
    # Expand the average Q at each site and day by the corresponding total number of t
rips counted
    route_daily_est <- merge(BoatCounts_sites, Avg_trip_Q, by=c("Day","AccessSite"), a
ll.x = TRUE)
    route_daily_est$m_trip_Q[is.na(route_daily_est$m_trip_Q)] <- 0
    route_daily_est$site_total_Q <- route_daily_est$count * route_daily_est$m_trip_Q

    # Creel times at each access site
    route_daily_est$wait_time <- 0
    for (s in wait_time$SiteName){
```

```
      route_daily_est[route_daily_est$AccessSite == s,]$wait_time <- wait_time[wait_ti
me$SiteName == s,]$CreelTime_Min/60
    }
    # Creel estimates per unit of wait time
    route_daily_est$site_Q_per_wait_time <- route_daily_est$site_total_Q/route_daily_e
st$wait_time

    # total daily site estimates
    route_daily_est$daily_site_Q <- total_route_time * route_daily_est$site_Q_per_wait
_time
    route_daily_est
}

f_psu_bus_route <- function(est_by_daytype, interviews, BoatCounts, month)
{
    # ------------------------------------------------
    # psu catch and effort estimation
    #   psu level summary information
    # only for one month
    # ------------------------------------------------
    # ------------------------------------------------------
    # A psu is a day in the bus-route estimation
    # ------------------------------------------------------
    ints$psu <- as.factor(ints$DAY)
    psu_f <-ints$psu
    ints$DAYTYPE <- ifelse(ints$DOW < 6, 1, 2)

    # psu summaries
    psu_summ = aggregate(list(NINTS=ints$DAY), list(MONTH = factor(ints$MONTH), DAYTYP
E = factor(ints$DAYTYPE), DAY = psu_f, psu = psu_f), length)

    n_psu <- nrow(psu_summ)

    # TripHours
    ints$st <- as.POSIXct(paste(ints$SDAY, " ", ints$STIME, sep=""), format = "%m/%d/%
Y %I:%M %p")
    ints$et <- as.POSIXct(paste(ints$EDAY, " ", ints$ETIME, sep=""), format = "%m/%d/%
Y %I:%M %p")

    ints$trip_length <- as.numeric(difftime(ints$et, ints$st, units = "hours"))

    ints$party_ang_hours <- ints$ANGCNT * ints$trip_length

    psu_summ$TripHours   <- tapply(ints$trip_length, list(psu_f), mean)
    psu_summ$V_TripHours    <- tapply(ints$trip_length, list(psu_f), var)/psu_summ$NINT

    # mean anglers/party
    psu_summ$PARTY_SIZE <- tapply(ints$ANGCNT, list(psu_f), mean, na.rm = TRUE)
    psu_summ$V_PARTY_SIZE     <- tapply(ints$ANGCNT, list(psu_f), var, na.rm = TRUE)/ps
u_summ$NINT

    # mean party AngHours
    psu_summ$PartyAngHours <- tapply(ints$party_ang_hours, list(psu_f), mean)
    psu_summ$V_PartyAngHours    <- tapply(ints$party_ang_hours, list(psu_f), var)/psu_s
umm$NINT

    wait_time<- data.frame(AccessSite = c("Bolles", "Sterling"), wait_time = c( 2.6666
66667, 3.833333333))

    # boat hours
    Bd_df <- data.frame(Q = ints$TripLenW, AccessSite = ints$AccessSite, psu = psu_f)
    route_daily_est <- bus_route_site_estimation(boat_counts, total_route_time, wait_t
ime, Bd_df)
    daily_est <- aggregate(list(BH_day = route_daily_est$daily_site_Q),  list(Day = as
.factor(route_daily_est$Day)), sum)
```

```
    psu_summ$BH <- daily_est$BH_day

    mean_monthly_boat_hours <- mean(daily_est$BH_day)
    monthly_boat_hours <- mean_monthly_boat_hours * 31

    ints$WAE[is.na(ints$WAE)] <- 0
    weighted_catch <- ints$WAE/ints$weight_sampling
    Cd_df <- data.frame(Q = weighted_catch, AccessSite = ints$AccessSite, psu = psu_f)
    route_daily_est <- bus_route_site_estimation(boat_counts, total_route_time, wait_t
ime, Cd_df)

    daily_est <- aggregate(list(C_day = route_daily_est$daily_site_Q),  list(Day = as.
factor(route_daily_est$Day)), sum)
    psu_summ$WAE <- daily_est$C_day

    mean_monthly_C <- mean(daily_est$C_day)
    monthly_C <- mean_monthly_C * 31

    psu_summ <- na.omit(psu_summ)
}

week_days <- function(date_str) {
    # use strftime function with "%u" format parameter to return the weekday
    weekdays <- as.numeric(strftime(as.Date(date_str, "%m/%d/%Y"), "%u"))
    weekdays
}

f_bus_route_total_boat_hours <- function(est_by_daytype, ints, boat_counts, total_rout
e_time, wait_times, NumDays)
{
    # ------------------------------------------------
    # monthly effort estimation
    #   only for one month
    # ------------------------------------------------

    # Convert time strings,such as "10:50 AM" to the POSIXct format
    #ints$st <- as.POSIXct(paste(ints$SDAY, " ", ints$STIME, sep=""), format = "%m/%d/
%Y %I:%M %p")
    #ints$et <- as.POSIXct(paste(ints$EDAY, " ", ints$ETIME, sep=""), format = "%m/%d/
%Y %I:%M %p")
    # Calculate trip lengths in hours
    ints$trip_length <- ints$ET - ints$ST #as.numeric(difftime(ints$ET, ints$ST, units
 = "hours"))

    # trip lengths must be greater than zero
    if (any(ints$trip_length <= 0)){
      #cat("trip_lengths are <= 0: ", "\n")
      #print(ints[ints$trip_length <= 0, ])

      ints <- ints[ints$trip_length > 0, ]
    }
    #stopifnot(ints$trip_length >= 0)

    # boat hours
    TripLenW <- ints$trip_length/ints$prob_sampling

    trip_lengths <- data.frame(Q = TripLenW, AccessSite = ints$AccessSite, Day = ints$
DAY)
    route_daily_site_BH_est <- bus_route_site_estimation(boat_counts, total_route_time
, wait_times, trip_lengths)

    daily_BH_est <- aggregate(list(BH_day = route_daily_site_BH_est$daily_site_Q),  li
st(Day = as.factor(route_daily_site_BH_est$Day)), sum)

    if (est_by_daytype){
```

```
        date_str <- paste(ints$MONTH[1], "/", daily_BH_est$Day,"/", ints$YEAR[1], sep="
")

        daily_BH_est$weekday <- week_days(date_str)
        daily_BH_est$DayType <- ifelse(daily_BH_est$weekday<6, "WD", "WE")

        n_sample <- as.numeric(tapply(daily_BH_est$BH_day, list(DayType = as.factor(dai
ly_BH_est$DayType)), length))
        mean_monthly_boat_hours <- as.numeric(tapply(daily_BH_est$BH_day, list(DayType
= as.factor(daily_BH_est$DayType)), mean))
        v_mean_monthly_boat_hours <- as.numeric(tapply(daily_BH_est$BH_day, list(DayTyp
e = as.factor(daily_BH_est$DayType)), var))
        v_mean_monthly_boat_hours <- v_mean_monthly_boat_hours / n_sample

        monthly_boat_hours <- mean_monthly_boat_hours * NumDays
        v_monthly_boat_hours <- v_mean_monthly_boat_hours * NumDays^2
    }
    else
    {
        n_sample <- nrow(daily_BH_est)
        mean_monthly_boat_hours <- mean(daily_BH_est$BH_day)
        v_mean_monthly_boat_hours <- var(daily_BH_est$BH_day)/n_sample
        monthly_boat_hours <- mean_monthly_boat_hours * sum(NumDays)
        v_monthly_boat_hours <- v_mean_monthly_boat_hours * (sum(NumDays))^2
    }
    list(boat_hours = sum(monthly_boat_hours), v_boat_hours = sum(v_monthly_boat_hours
))
}

f_bus_route_total_angler_hours <- function(est_by_daytype, ints, boat_counts, total_ro
ute_time, wait_times, NumDays)
{
    # ------------------------------------------------
    # monthly effort estimation
    #   only for one month
    # ------------------------------------------------

    # Convert time strings,such as "10:50 AM" to the POSIXct format
    #ints$st <- as.POSIXct(paste(ints$SDAY, " ", ints$STIME, sep=""), format = "%m/%d/
%Y %I:%M %p")
    #ints$et <- as.POSIXct(paste(ints$EDAY, " ", ints$ETIME, sep=""), format = "%m/%d/
%Y %I:%M %p")
    # Calculate trip lengths in hours
    ints$trip_length <- ints$ET - ints$ST # as.numeric(difftime(ints$ET, ints$ST, unit
s = "hours"))

    # trip lengths must be greater than zero
    # trip lengths must be greater than zero
    if (any(ints$trip_length <= 0)){
      #cat("trip_lengths are <= 0: ", "\n")
      #print(ints[ints$trip_length <= 0, ])

      ints <- ints[ints$trip_length > 0, ]
    }
    #stopifnot(ints$trip_length >= 0)

    # Total party angler hours
    ints$party_ang_hours <- ints$ANGCNT * ints$trip_length
    ints$party_ang_hours_w <- ints$party_ang_hours/ints$prob_sampling

    # angler hours
    party_ang_hours <- data.frame(Q = ints$party_ang_hours_w, AccessSite = ints$Access
Site, Day = ints$DAY)
    route_daily_site_AH_est <- bus_route_site_estimation(boat_counts, total_route_time
, wait_times, party_ang_hours)
```

```
    daily_AH_est <- aggregate(list(AH_day = route_daily_site_AH_est$daily_site_Q),  li
st(Day = as.factor(route_daily_site_AH_est$Day)), sum)

    if (est_by_daytype){
       date_str <- paste(ints$MONTH[1], "/", daily_AH_est$Day,"/", ints$YEAR[1], sep="
")

       daily_AH_est$weekday <- week_days(date_str)
       daily_AH_est$DayType <- ifelse(daily_AH_est$weekday<6, "WD", "WE")

       n_sample <- as.numeric(tapply(daily_AH_est$AH_day, list(DayType = as.factor(dai
ly_AH_est$DayType)), length))
       mean_monthly_angler_hours <- as.numeric(tapply(daily_AH_est$AH_day, list(DayTyp
e = as.factor(daily_AH_est$DayType)), mean))
       v_mean_monthly_angler_hours <- as.numeric(tapply(daily_AH_est$AH_day, list(DayT
ype = as.factor(daily_AH_est$DayType)), var))
       v_mean_monthly_angler_hours <-  v_mean_monthly_angler_hours / n_sample

       monthly_angler_hours <- mean_monthly_angler_hours * NumDays
       v_monthly_angler_hours <- v_mean_monthly_angler_hours * NumDays^2
    }
    else
    {
       n_sample <- nrow(daily_AH_est)
       monthly_angler_hours <- mean(daily_AH_est$AH_day) * sum(NumDays)
       v_mean_monthly_angler_hours <- var(daily_AH_est$AH_day)/n_sample
       v_monthly_angler_hours <- v_mean_monthly_angler_hours * (sum(NumDays))^2
    }


    list(angler_hours = sum(monthly_angler_hours), v_angler_hours = sum(v_monthly_angl
er_hours))
}

f_bus_route_total_catch <- function(ints, boat_counts, total_route_time, wait_times, N
umDays)
{
    # Total catch for a multiple day period

    #species = c("WAE","NOP","LMB","SMB","MUS","WHB","YEP","LWF","LHR","BLG","CCF","CW
S","RWF","RKB","PSF","BKT","WHP","SPL","BCR","TMU","DRU","PKS","ATS","CAR","SMT","STN"
,"OTH","COS_REL","CHS_REL","RBT_REL","BNT_REL","LAT_REL","FAT_REL","LAT.FAT.UNK_REL","
WAE_REL","NOP_REL","LMB_REL","SMB_REL","MUS_REL","WHB_REL","YEP_REL","LWF_REL","LHR_RE
L","BLG_REL","CCF_REL","CWS_REL","RWF_REL","RKB_REL","PSF_REL","BKT_REL","WHP_REL","SP
L_REL","BCR_REL","TMU_REL","DRU_REL","PKS_REL","ATS_REL","SMT_REL","STN_REL","OTH_REL"
,"COS_NLEG","CHS_NLEG","RBT_NLEG","BNT_NLEG","LAT_NLEG","FAT_NLEG","LAT.FAT.UNK_NLEG",
"WAE_NLEG","NOP_NLEG","LMB_NLEG","SMB_NLEG","MUS_NLEG","WHB_NLEG")

    species = c("WAE")

    monthly_C <- rep(0, length(species))
    names(monthly_C) <- species
    v_monthly_C <- rep(0, length(species))
    names(v_monthly_C) <- species
    i <- 1
    for (s in species) {
      catch <- ints[, s]
      catch[is.na(catch)] <- 0
      if (sum(catch) > 0){
        weighted_catch <- catch/ints$prob_sampling

        Cd_df <- data.frame(Q = weighted_catch, AccessSite = ints$AccessSite, Day = in
ts$DAY)
        route_daily_est <- bus_route_site_estimation(boat_counts, total_route_time, wa
```

```
it_times, Cd_df)

        daily_Catch_est <- aggregate(list(C_day = route_daily_est$daily_site_C),  list
(Day = as.factor(route_daily_est$Day)), sum)

        if (est_by_daytype){
            date_str <- paste(ints$MONTH[1], "/", daily_Catch_est$Day,"/", ints$YEAR[1]
, sep="")
            daily_Catch_est$weekday <- week_days(date_str)
            daily_Catch_est$DayType <- ifelse(daily_Catch_est$weekday<6, "WD", "WE")

            n_sample <- as.numeric(tapply(daily_Catch_est$C_day, list(DayType = as.fact
or(daily_Catch_est$DayType)), length))
            mean_monthly_catch <- as.numeric(tapply(daily_Catch_est$C_day, list(DayType
 = as.factor(daily_Catch_est$DayType)), mean))
            v_mean_monthly_catch <- as.numeric(tapply(daily_Catch_est$C_day, list(DayTy
pe = as.factor(daily_Catch_est$DayType)), var))
            v_mean_monthly_catch <-  v_mean_monthly_catch / n_sample

            monthly_C[i] <- sum(mean_monthly_catch * NumDays)
            v_monthly_C[i] <- sum(v_mean_monthly_catch * NumDays^2)
        }
        else
        {
            mean_monthly_catch <- mean(daily_Catch_est$C_day)
            n_sample <- nrow(daily_Catch_est)
            v_mean_monthly_catch <- var(daily_Catch_est$C_day)/n_sample

            monthly_C[i] <- mean_monthly_catch * sum(NumDays)
            v_monthly_C[i] <- v_mean_monthly_catch * (sum(NumDays))^2
        }
      }
      i <- i + 1
    }

    data.frame(catch = monthly_C[monthly_C!=0], v_catch = v_monthly_C[monthly_C!=0])
}

f_bus_route_creel_estimation <- function(est_by_daytype, interviews, BoatCounts, total
_route_time, wait_times, NumDays)
{
    if ((!est_by_daytype))
    {
        NumDays <- sum(NumDays)
    }

    interviews$DAYTYPE <- ifelse(interviews$DOW < 6, 1, 2)

    # est_by_daytype is only applied to daily effort estimation, not catch rate estima
tion
    # ------------------------------------------------
    # psu is (1) day for the daily estimator, and
    #        (2) a period of multiple days of a daytype for the multiple-day estimator
    #        (3) a month for the multiple-day estimator if daytype is not considered
    #psu_stats <- f_psu_stat(est_by_daytype, interviews, simu_Month)

    tot_boat_hours <- f_bus_route_total_boat_hours(est_by_daytype, interviews, BoatCou
nts, total_route_time, wait_times, NumDays)
    tot_angler_hours <- f_bus_route_total_angler_hours(est_by_daytype, interviews, Boa
tCounts, total_route_time, wait_times, NumDays)
    total_catch <- f_bus_route_total_catch(interviews, BoatCounts, total_route_time, w
ait_times, NumDays)

    list(angler_hours = tot_angler_hours$angler_hours, v_angler_hours = tot_angler_hou
rs$v_angler_hours, boat_hours = tot_boat_hours$boat_hours, v_boat_hours = tot_boat_hou
```

```
rs$v_boat_hours, catch = total_catch$catch[1], v_catch = total_catch$v_catch[1])
}
```