**Edits to HB1110 script (to reflect HB1110-S2-E)**

# Script for data analysis related to HB1110 data request

#

# It attaches various dummies to the input parcels file

# and writes two output parcels file, one with all parcels included

# in the bill analysis and one with parcels likely to develop.

# It then creates several city-level summaries and exports

# the into csv files.

#

# Last update: 03/08/2023

# Drew Hanson & Hana Sevcikova

if(! "data.table" %in% installed.packages())

install.packages("data.table")

library(data.table)

# Run this script from the directory of this file, unless data\_dir is set as an absolute path

setwd("J:/Projects/Bill-Analysis/2023/scripts")

#setwd("~/psrc/R/bill-analysis/scripts")

# Settings

write.parcels.file <- TRUE

write.summary.files.to.csv <- FALSE

write.summary.files.to.excel <- TRUE

data\_dir <- "../data" # directory where the data files below live

# (it's a relative path to the script location; can be also set as an absolute path)

parcels\_file\_name <- "parcels\_for\_bill\_analysis.csv"

#parcel\_vision\_hct\_file\_name <- "parcel\_vision\_hct.csv"

parcel\_vision\_hct\_file\_name <- "revised\_buffers\_1110.csv"

cities\_file\_name <- "cities.csv"

tier\_file\_name <- "cities\_coded\_all\_20230307.csv"

# tier definitions

tier\_constraints <- list(`1` = c(4, 2), # in the form c(hct\_constraint, non-hct\_constraint)

`2` = c(6, 4)

#`1` = c(6, 4) # original constraint

)

#tier\_column <- "Original"

tier\_column <- "Substitute"

tier\_column <- "Final"

# name of the output files; should include "XXX" which will be replaced by "in\_bill" and "to\_develop" to distinguish the two files

output\_parcels\_file\_name <- paste0("selected\_parcels\_for\_mapping\_XXX-", Sys.Date(), ".csv") # will be written into data\_dir

# directory name where results should be written

output\_dir <- paste0("HB1110\_results-", Sys.Date())

# Read input files

parcels\_for\_bill\_analysis <- fread(file.path(data\_dir, parcels\_file\_name)) # parcels

parcel\_vision\_hct <- fread(file.path(data\_dir, parcel\_vision\_hct\_file\_name)) # HCT locations

cities <- fread(file.path(data\_dir, cities\_file\_name)) # cities table

tiers\_by\_city <- fread(file.path(data\_dir, tier\_file\_name))[, c("city\_id", tier\_column), with = FALSE] # table containing tier assignment to cities

setnames(tiers\_by\_city, tier\_column, "tier") # rename the tier column to "tier" for simpler access

cities <- merge(cities, tiers\_by\_city, all = TRUE)

cities[is.na(tier), tier := 0]

# Get all non-zero tiers and check if there is a definition for them

tiers <- as.character(unique(cities[tier > 0, tier]))

if(any(! tiers %in% names(tier\_constraints)))

stop("Missing definition of tier(s) ", paste(tiers[! tiers %in% names(tier\_constraints)], collapse = ", "), " in the tier\_constraints object.")

#Fixes City ID error in the input dataset (might not be needed after it's corrected in the input file)

parcels\_for\_bill\_analysis[city\_id==95, city\_id := 96]

# Codes records that fall in cities/geographies excluded from HB1110 bill requirements with 0's and and everything else with 1's

parcels\_for\_bill\_analysis[cities, city\_tier := i.tier, on = "city\_id"]

# Creates updated parcel table with "hct\_vision" field added

#parcel\_vision\_hct[, vision\_hct := any(hct\_quarter\_mile, parks, schools)]

#parcels\_updated <- merge(parcels\_for\_bill\_analysis, parcel\_vision\_hct[, .(parcel\_id, vision\_hct)], all=TRUE)

parcels\_updated <- copy(parcels\_for\_bill\_analysis)

parcels\_updated[, vision\_hct := 0]

parcels\_updated[parcel\_id %in% parcel\_vision\_hct[hct\_quarter\_mile == 1 | parks == 1 | schools == 1, parcel\_id], vision\_hct := 1]

#Creates "res\_zone" field that denotes residential zoned parcels

parcels\_updated[, res\_zone := 0]

parcels\_updated[DUcap > 0 & is\_mixed\_cap == 0, res\_zone := 1]

#Creates "mix\_zone" field that denotes mixed-use zoned parcels

parcels\_updated[, mixed\_zone := 0]

parcels\_updated[DUcap > 0 & is\_mixed\_cap == 1, mixed\_zone := 1]

# exclude all parks

parcels\_updated[land\_use\_type\_id == 19, `:=`(res\_zone = 0, mixed\_zone = 0)]

#Creates "already\_zoned" field to denote parcels that are already zoned to meet requirements of bill (Step 3 in Methodology document)

parcels\_updated[, already\_zoned := 0]

for(tier in tiers) { # iterate over the non-zero city tiers

parcels\_updated[(city\_tier == tier) & ((vision\_hct == 1 & DUcap >= tier\_constraints[[tier]][1]) |

(vision\_hct == 0 & DUcap >= tier\_constraints[[tier]][2])), already\_zoned := 1]

}

#Creates "sq\_ft\_less\_2500" field denoting records with parcel square footage of less then 2,500

parcels\_updated[, sq\_ft\_less\_2500 := 0]

parcels\_updated[parcel\_sqft < 2500, sq\_ft\_less\_2500 := 1]

#Creates "sf\_use" field denoting single family parcels

parcels\_updated[, sf\_use := 0]

parcels\_updated[residential\_units == 1, sf\_use := 1]

#Creates "vacant" field denoting vacant parcels

parcels\_updated[, vacant := 0]

parcels\_updated[Nblds == 0, vacant := 1]

#Creates "land\_greater\_improvement" field to denotes parcels that have land value that is greater than the improvement value

parcels\_updated[is.na(improvement\_value), improvement\_value := 0]

parcels\_updated[, land\_greater\_improvement := 0]

parcels\_updated[land\_value > improvement\_value, land\_greater\_improvement := 1]

#Creates "built\_sqft\_less\_1400" field to denotes parcels that have a built square footage of less than 1,400

parcels\_updated[, built\_sqft\_less\_1400 := 0]

parcels\_updated[residential\_sqft < 1400, built\_sqft\_less\_1400 := 1]

#Adds city\_name field to final parcel table

parcels\_final <- merge(parcels\_updated, cities[, .(city\_id, city\_name)], by = "city\_id")

# split parcels into two sets: 1. all parcels included in the bill, 2. parcels likely to develop

parcels\_in\_bill <- parcels\_final[city\_tier > 0 & already\_zoned == 0 & (res\_zone == 1 | mixed\_zone == 1)]

parcels\_likely\_to\_develop <- parcels\_final[city\_tier > 0 & already\_zoned == 0 & (res\_zone == 1 | mixed\_zone == 1) & sq\_ft\_less\_2500 == 0 & land\_greater\_improvement == 1 & built\_sqft\_less\_1400 == 1 & (vacant == 1 | sf\_use == 1), ]

#Writes csv output files

if(write.parcels.file) {

#remove all fields from parcel tables except parcel\_id,vision\_hct,and city\_tier

parcels\_in\_bill\_to\_save <- parcels\_in\_bill[, c("parcel\_id", "vision\_hct","city\_tier")]

parcels\_likely\_to\_develop\_to\_save <- parcels\_likely\_to\_develop[, c("parcel\_id", "vision\_hct","city\_tier")]

# write to disk into a subdirectory of output\_dir

pcl\_dir <- file.path(data\_dir, output\_dir, "parcels")

if(!dir.exists(pcl\_dir)) dir.create(pcl\_dir, recursive = TRUE)

fwrite(parcels\_in\_bill\_to\_save, file.path(pcl\_dir, gsub("XXX", "in\_bill", output\_parcels\_file\_name)))

fwrite(parcels\_likely\_to\_develop\_to\_save, file.path(pcl\_dir, gsub("XXX", "to\_develop", output\_parcels\_file\_name)))

cat("\nParcels written into ", file.path(pcl\_dir, output\_parcels\_file\_name), "\n")

}

# Functions for generating summaries

create\_summary\_detail <- function(dt, col\_prefix){

detail <- dt[, .(

total\_parcels = .N,

res\_vacant = sum(res\_zone == 1 & vacant == 1),

res\_sf\_use = sum(res\_zone == 1 & sf\_use == 1),

res\_other\_use = sum(res\_zone == 1 & vacant == 0 & sf\_use == 0),

mix\_vacant = sum(mixed\_zone == 1 & vacant == 1),

mix\_sf\_use = sum(mixed\_zone == 1 & sf\_use == 1),

mix\_other\_use = sum(mixed\_zone == 1 & vacant == 0 & sf\_use == 0),

other = sum(res\_zone == 0 & mixed\_zone == 0)

), by = "city\_id"][order(city\_id)]

# add prefix to column names (excluding city\_id which is first)

setnames(detail, colnames(detail)[-1], paste0(col\_prefix, colnames(detail)[-1]))

return(detail)

}

create\_summary <- function(dt){

# part that involves all parcels

summary\_all <- dt[, .(

total\_parcels = .N,

already\_zoned = sum(already\_zoned)

), by = "city\_id"][order(city\_id)]

# generate HCT and nonHCT parts of the summary

summary\_hct <- create\_summary\_detail(dt[vision\_hct == 1 & already\_zoned == 0], col\_prefix = "hct\_")

summary\_nonhct <- create\_summary\_detail(dt[vision\_hct == 0 & already\_zoned == 0], col\_prefix = "nhct\_")

# merge together and add city\_name

summary\_final <- merge(merge(cities[, .(city\_id, city\_name, tier)], summary\_all, by = "city\_id", all = TRUE),

merge(summary\_hct, summary\_nonhct, by = "city\_id", all = TRUE),

by = "city\_id", all = TRUE)[order(-tier, city\_id)]

return(summary\_final)

}

# Create summaries

summaries <- list()

summaries[["all\_parcels"]] <- create\_summary(parcels\_final)

summaries[["filter\_parcel\_sqft"]] <- create\_summary(parcels\_final[sq\_ft\_less\_2500 == 0])

summaries[["filter\_parcel\_sqft\_under1400"]] <- create\_summary(parcels\_final[sq\_ft\_less\_2500 == 0 & built\_sqft\_less\_1400 == 1])

summaries[["filter\_parcel\_sqft\_land\_value"]] <- create\_summary(parcels\_final[sq\_ft\_less\_2500 == 0 & land\_greater\_improvement == 1])

summaries[["filter\_all"]] <- create\_summary(parcels\_final[sq\_ft\_less\_2500 == 0 & land\_greater\_improvement == 1 & built\_sqft\_less\_1400 == 1])

# create a dataset of existing units

existing\_units <- merge(cities[, .(city\_id, tier, city\_name)],

parcels\_in\_bill[, .(total = sum(residential\_units),

HCT = sum(residential\_units \* vision\_hct == 1),

nonHCT = sum(residential\_units \* (vision\_hct == 0))),

by = "city\_id"],

by = "city\_id")

# add regional totals as the first row

existing\_units <- rbind(data.table(city\_id = 0, tier = 0, city\_name = "Region",

total = existing\_units[tier > 0, sum(total)],

HCT = existing\_units[tier > 0, sum(HCT)],

nonHCT = existing\_units[tier > 0, sum(nonHCT)]),

existing\_units)

# create top page with regional summaries

top\_page <- top\_page\_total <- NULL

for(sheet in names(summaries)){

top\_page\_total <- rbind(top\_page\_total, data.table(indicator = sheet,

tier = -1,

summaries[[sheet]][tier > 0, lapply(.SD, sum, na.rm = TRUE),

.SDcols = setdiff(colnames(summaries[[sheet]]), c("city\_id", "city\_name", "tier"))]),

fill = TRUE)

}

description <- list(

all\_parcels = "Total number of parcels",

filter\_parcel\_sqft = "Parcels larger than 2500 sqft",

filter\_parcel\_sqft\_under1400 = "Parcels larger than 2500 sqft that have built residential sqft smaller than 1400",

filter\_parcel\_sqft\_land\_value = "Parcels larger than 2500 sqft with land value > improvement value",

filter\_all = "Parcels larger than 2500 sqft passing both market criteria"

)

descr <- cbind(data.table(description), indicator = names(description))

top\_page\_total <- merge(descr, top\_page\_total, by = "indicator", sort = FALSE)

# summaries by tier

for(sheet in names(summaries)){

top\_page <- rbind(top\_page, top\_page\_total[indicator == sheet])

top\_page <- rbind(top\_page, data.table(indicator = "",

description = "",

summaries[[sheet]][, lapply(.SD, sum, na.rm = TRUE),

.SDcols = setdiff(colnames(summaries[[sheet]]), c("city\_id", "city\_name", "tier")), by = "tier"]),

fill = TRUE)

}

top\_page[,tier := as.character(tier)][tier == -1, tier := "1,2"]

summaries[["existing\_units"]] <- existing\_units

summaries <- c(list(Region = top\_page), summaries) # set the regional summaries as the first sheet

if(write.summary.files.to.csv || write.summary.files.to.excel){

summary\_dir <- file.path(data\_dir, output\_dir)

if(!dir.exists(summary\_dir)) dir.create(summary\_dir) # create directory if not exists

if(write.summary.files.to.csv) {

csvdir <- file.path(summary\_dir, "csv")

if(!dir.exists(csvdir)) dir.create(csvdir) # create sub-directory if not exists

for(table in names(summaries))

fwrite(summaries[[table]], file = file.path(csvdir, paste0(table, ".csv")))

}

if(write.summary.files.to.excel) {

library(openxlsx)

# style of the header

style <- createStyle(

textDecoration = "BOLD", fontColour = "#FFFFFF", fontSize = 12, fgFill = "#4F80BD"

)

# set the width of columns and how many columns should be freezed

colwidths <- list()

firstcol <- list()

for(sheet in names(summaries)){

colwidths[[sheet]] <- rep(10, ncol(summaries[[sheet]]))

colwidths[[sheet]][colnames(summaries[[sheet]]) == "city\_id"] <- 7

colwidths[[sheet]][colnames(summaries[[sheet]]) == "tier"] <- 5

colwidths[[sheet]][colnames(summaries[[sheet]]) == "city\_name"] <- 20

firstcol[[sheet]] <- 4

}

colwidths[["Region"]][1:2] <- c(25, 40)

colwidths[["Region"]][-(1:3)] <- 15

write.xlsx(summaries, file = file.path(summary\_dir, "HB1110\_all\_tables.xlsx"),

headerStyle = style, colWidths = colwidths, firstActiveRow = 2, firstActiveCol = firstcol)

}

cat("\nSummary files written into ", summary\_dir, "\n")

}