NEXUS helpers file

```
# Nexus_DOC.R
#
# Author: Scott Lawrence
# Last Update: 2019-01-15

# ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~#
# 1 Load R Packages ----
# ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~#
library(shiny)
library(DBI)
library(jsonlite)
library(readxl)
library(leaflet)
library(geosphere)
library(ggplot2)
library(ggridges)
library(dplyr)
library(tidyverse)
library(RColorBrewer)

# ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~#
# 2 Directory Management ----
# ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~#

# R escape character uses backslash "\", so directory paths need forward
# slash "/" in place of backslash.
# ++ 2.1 create in_dir ----
# this is the local data path
fs::dir_ls(".")
in_dir <- "data/"



# ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~#
# 3 Load Data Tables ----
# ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~#

# + 3.1 create nexus_ports ----
nexus_ports <- data.frame(
  name = c(
```

```
    "Fort Worth", "Arlington", "Dallas", "DFW Airport", "Frisco",
    "Weatherford", "Tyler", "Waco", "Denton", "Waxahachie"
  ),
  lat = c(
    32.762550, 32.741326, 32.764324, 32.879575, 33.156885,
    32.714701, 32.332695, 31.511785, 33.202340, 32.436070
  ),
  lng = c(
    -97.334502, -97.093387, -96.726364, -97.062307, -96.798145,
    -97.777923, -95.343478, -97.117645, -97.202808, -96.860789
  ),
  stringsAsFactors = FALSE
) %>%
  arrange(name) %>%
  mutate(aadt_rad = 0, aadt_rad_tflag = " ")


# + 3.2 read in txdot_aadt ----
# SOURCE: TxDOT AADT Annuals ----
# Source: http://gis-txdot.opendata.arcgis.com/datasets/txdot-aadt-annuals
txdot_aadt <- read.csv(paste(in_dir, "/TxDOT_AADT_Annuals.csv", sep = ""),
  header = TRUE, stringsAsFactors = FALSE
)

txdot_aadt <- txdot_aadt %>%
  mutate(lat = Y, lng = X) %>%
  select(c("lat", "lng", colnames(txdot_aadt)[!(colnames(txdot_aadt) %in%
c("X", "Y"))]))

for (i in 1:nrow(nexus_ports)) {
  loc_np <- select(filter(nexus_ports, name == nexus_ports$name[i]), c("ln
g", "lat"))
  txt <- paste("txdot_aadt$dist_m_",
    str_replace_all(
      string = nexus_ports$name[i],
      pattern = " ",
      replacement = "_"
    ),
    ' <- as.numeric(distm(select(txdot_aadt, c("lng", "lat")), loc_np, fun
= distHaversine))',
    sep = ""
  )
  eval(parse(text = txt))
}
remove(i)
```

```
remove(loc_np)
remove(txt)


# ++ 3.2.3 create CAP_DMC ----
# locate file
# fs::dir_ls("data")
CAP_DMC <- read_xlsx(paste0(in_dir, "CAP_DMC.xlsx"),
  sheet = 1,
  col_names = TRUE
)
# replace missing
CAP_DMC[is.na(CAP_DMC)] <- 0

# ++ DOC_us data frame ----
# previously (5.4)create DOC_us data frame, this is needed for data frame
# below
n <- 10000
DOC_us <- data.frame(
  Total = rep.int(100, n),
  Maintenance = rep.int(0, n),
  Insurance = rep.int(0, n),
  Fuel = rep.int(0, n),
  Financing_Exp = rep.int(0, n),
  Depreciation = rep.int(0, n),
  Personnel_Exp = rep.int(0, n),
  Training_Gen_Admin = rep.int(0, n),
  Other = rep.int(0, n)
)
# verify
DOC_us %>% dplyr::glimpse(75)
# Observations: 10,000
# Variables: 9
# $ Total              <dbl> 100, 100, 100, 100, 100, 100, 100, 100, 100,
…
# $ Maintenance        <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
…
# $ Insurance          <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
…
# $ Fuel               <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
…
# $ Financing_Exp      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
…
# $ Depreciation       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
…
```

```
# $ Personnel_Exp      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
…
# $ Training_Gen_Admin <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
…
# $ Other              <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
…

# ++ 3.3.1 DMC_model ----
# Add Engine estimates based on total Maintenance DOC
DMC_model <- CAP_DMC %>%
  filter(Model %in% c("206L", "505", "407", "407GX")) %>%
  group_by(Model, Part_System) %>%
  summarize(DMC = sum(Total) * 2) %>%
  spread(Model, DMC)



DMC_model <- rbind(
  DMC_model,
  c(
    "Engine",
    round(mean(DOC_us$Maintenance) - sum(DMC_model$`206L`), 2),

    round(mean(DOC_us$Maintenance) - sum(DMC_model$`407`), 2),

    round(mean(DOC_us$Maintenance) - sum(DMC_model$`407GX`), 2),

    round(mean(DOC_us$Maintenance) - sum(DMC_model$`505`), 2)
  )
)

DMC_model$`505`[DMC_model$Part_System == "Fuel"] <- mean(as.numeric(DMC_mo
del[6, 2:4]))
DMC_model$`505`[DMC_model$Part_System == "Powerplant"] <- mean(as.numeric(
DMC_model[9, 2:4]))
DMC_model$`505`[DMC_model$Part_System == "Engine"] <- round(mean(as.numeri
c(DMC_model[11, 2:4])), 2)

DMC_model <- DMC_model %>% gather(key = "Model", value = "DMC", "206L", "5
05", "407", "407GX")
DMC_model$DMC <- round(as.numeric(DMC_model$DMC), 2)


# SOURCE Electric motor reliability ----
# Source:
```

```
# https://www.cbmconnect.com/large-electric-motor-reliability-what-did-the
-studies-really-say-2/
#
# ~~~~ Electric motor reliability Publication source: ----
#   “Report of Large Motor Reliability Survey of Industrial and Commercial
Installations, Part I”
#               IEEE Transactions on Industry Applications, Vol. IA-21, No
. 4, July/August 1985
#   “Report of Large Motor Reliability Survey of Industrial and Commercial
Installations, Part II”
#               IEEE Transactions on Industry Applications, Vol. IA-21, No
. 4, July/August 1985
#   “Report of Large Motor Reliability Survey of Industrial and Commercial
Installations, Part III”
#               IEEE Transactions on Industry Applications, Vol. IA-23, No
. 1, January/February 1987
#  ~~~~~~~~~~~~~~~~~~~~~~~~
# ~~~~ Electric motor reliability Assumptions: ----
# 500-5000 hp
# 721-1800 RPM
# Level of maintenance and frequency: Excellent, <12 months
# Induction Motor
# Industrial application
#  ~~~~~~~~~~~~~~~~~~~~~~~~
# ~~~~ Electric motor reliability Results: ----
# 0.0730 FPU (Failures per Unit per Year)
# 8 hours Median Downtime per Failure
#  ~~~~~~~~~~~~~~~~~~~~~~~~
# ~~~~ Electric motor reliability Conversion assumptions: ----
# 260 Mdays
# 24 hour operation per Mday
#  ~~~~~~~~~~~~~~~~~~~~~~~~
# ~~~~ Electric motor reliability Conversion expression: ----
# MTBF = (Mdays * hr/Mday) / (FPU * shipset)
# + 3.4.0 create mtbf_motor_elec ----
mtbf_motor_elec <- (260 * 24) / (0.073 * 6)


# ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# US Market ONLY -----
# ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

# SOURCE simplemaps ----
# Source: https://simplemaps.com/data/us-cities
us_cities_geo <- read_csv("data/uscitiesv1.4.csv") %>%
```

```
  mutate(display_name = paste0(city, ", ", state_id)) %>%
  select(display_name, lat, lng)
us_cities_geo %>% glimpse(78)

# SOURCE: datausa.io (housing) ----
# Source:
#   https://datausa.io/profile/geo/dallas-tx/#housing
# Data USA API documentation:
#   https://github.com/DataUSA/datausa-api/wiki/Data-API#ipeds


# ++ 3.4.1 datausa_locations data frame ----
datausa_locations <- fromJSON("http://api.datausa.io/attrs/geo/")
tmp <- as_data_frame(datausa_locations$data)
colnames(tmp) <- datausa_locations$headers
datausa_locations <- tmp

# ++ 3.4.2 create us_metro_rank ----
us_metro_rank <- fromJSON("https://api.datausa.io/api/?sort=desc&force=acs
.yg&show=geo&sumlevel=msa&year=latest&where=pop_rank:<26")
tmp <- as_data_frame(us_metro_rank$data)
colnames(tmp) <- us_metro_rank$headers

us_metro_rank <- tmp %>%
  mutate(
    id = geo, pop = as.integer(pop),
    pop_rank = as.integer(pop_rank)
  ) %>%
  select(id, pop, pop_rank) %>%
  arrange(pop_rank) %>%
  left_join(select(
    datausa_locations,
    id, url_name,
    display_name,
    name, name_long
  ), by = "id")
remove(tmp)


# ++ 3.4.3 create us_metro_children  ----
us_metro_children <- fromJSON(paste0(
  "http://api.datausa.io/attrs/geo/",
  us_metro_rank$id[1], "/children"
))
tmp <- as_data_frame(us_metro_children$data)
```

```
colnames(tmp) <- us_metro_children$headers[1:2]

us_metro_children <- tmp %>%
  mutate(us_metro_id = us_metro_rank$id[1])


for (i in 2:nrow(us_metro_rank)) {
  us_metro_tmp <- fromJSON(paste0(
    "http://api.datausa.io/attrs/geo/",
    us_metro_rank$id[i],
    "/children"
  ))
  tmp <- as_data_frame(us_metro_tmp$data)
  colnames(tmp) <- us_metro_tmp$headers[1:2]
  us_metro_tmp <- tmp %>%
    mutate(us_metro_id = us_metro_rank$id[i])
  us_metro_children <- bind_rows(us_metro_children, us_metro_tmp)
}
remove(us_metro_tmp)
remove(tmp)

# ++ 3.4.4 create datausa_acs_yg_all ----
datausa_acs_yg_all <- fromJSON("https://api.datausa.io/api/?sort=desc&forc
e=acs.yg&show=geo&sumlevel=all&year=all")
tmp <- as_data_frame(datausa_acs_yg_all$data)
colnames(tmp) <- datausa_acs_yg_all$headers
datausa_acs_yg_usmetro <- datausa_locations %>%
  left_join(tmp, by = c("id" = "geo")) %>%
  left_join(select(us_metro_children, id, us_metro_id), by = "id") %>%
  filter(!is.na(us_metro_id)) %>%
  left_join(us_cities_geo, by = "display_name") %>%
  mutate(
    age = as.numeric(age),
    age_moe = as.numeric(age_moe),
    pop = as.numeric(pop),
    pop_moe = as.numeric(pop_moe),
    non_us_citizens = as.numeric(non_us_citizens),
    mean_commute_minutes = as.numeric(mean_commute_minutes),
    income = as.numeric(income),
    income_moe = as.numeric(income_moe),
    owner_occupied_housing_units = as.numeric(owner_occupied_housing_units
),
    median_property_value = as.numeric(median_property_value),
    median_property_value_moe = as.numeric(median_property_value_moe),
    us_citizens = as.numeric(us_citizens),
```

```
    non_eng_speakers_pct = as.numeric(non_eng_speakers_pct)
  )

remove(datausa_acs_yg_all)
remove(tmp)

# ++ 3.4.5 create datausa_acs_yg_usmetro_income_dist ----
datausa_acs_yg_usmetro_income_dist <- fromJSON("https://api.datausa.io/api
/?sort=desc&force=acs.yg_income_distribution&show=geo&sumlevel=all&year=al
l")
tmp <- as_data_frame(datausa_acs_yg_usmetro_income_dist$data)
colnames(tmp) <- datausa_acs_yg_usmetro_income_dist$headers
datausa_acs_yg_usmetro_income_dist <- datausa_acs_yg_usmetro %>%
  select(display_name, image_link, image_meta, image_author, sumlevel, id,
us_metro_id, lat, lng) %>%
  left_join(tmp, by = c("id" = "geo")) %>%
  select(-totalhouseholds, -totalhouseholds_moe) %>%
  gather(key = "range", value = "households", "income_100to125",
         "income_100to125_moe", "income_10to15", "income_10to15_moe",
         "income_125to150", "income_125to150_moe", "income_150to200",
         "income_150to200_moe", "income_15to20", "income_15to20_moe",
         "income_200over", "income_200over_moe", "income_20to25",
         "income_20to25_moe", "income_25to30", "income_25to30_moe",
         "income_30to35", "income_30to35_moe", "income_35to40",
         "income_35to40_moe", "income_40to45", "income_40to45_moe",
         "income_45to50", "income_45to50_moe", "income_50to60",
         "income_50to60_moe", "income_60to75", "income_60to75_moe",
         "income_75to100", "income_75to100_moe", "income_under10",
         "income_under10_moe") %>%
  mutate(
    households = as.numeric(households),
    income = str_replace(
      paste(
        str_replace(
          str_replace(
            str_replace(
              str_replace(range, "_moe", ""),
              "income_", "$"
            ),
            "to", "K - $"
          ),
          "\\$under", "$0K - $"
        ),
        "K",
        sep = ""
```

```
      ),
      "overK", "K +"
    ),
    idx = as.integer(str_sub(income, start = str_locate(income, "\\$")[, 1
] + 1, end = str_locate(income, "K")[, 1] - 1))
  ) %>%
  unique()
remove(tmp)

# ++ 3.4.6 create datausa_acs_yg_usmetro_travel_time ----
datausa_acs_yg_usmetro_travel_time <- fromJSON("https://api.datausa.io/api
/?sort=desc&force=acs.yg_travel_time&show=geo&sumlevel=all&year=all")
tmp <- as_data_frame(datausa_acs_yg_usmetro_travel_time$data)
colnames(tmp) <- datausa_acs_yg_usmetro_travel_time$headers
datausa_acs_yg_usmetro_travel_time <- datausa_acs_yg_usmetro %>%
  select(display_name, image_link, image_meta, image_author, sumlevel, id,
us_metro_id, lat, lng) %>%
  left_join(tmp, by = c("id" = "geo")) %>%
  select(-workers, -workers_moe) %>%
  gather(key = "range",
         value = "households", "travel_10to14", "travel_10to14_moe",
         "travel_15to19", "travel_15to19_moe", "travel_20to24",
         "travel_20to24_moe", "travel_25to29", "travel_25to29_moe",
         "travel_30to34", "travel_30to34_moe", "travel_35to39",
         "travel_35to39_moe", "travel_40to44", "travel_40to44_moe",
         "travel_45to59", "travel_45to59_moe", "travel_5to9",
         "travel_5to9_moe", "travel_60to89", "travel_60to89_moe",
         "travel_90over", "travel_90over_moe", "travel_less5",
         "travel_less5_moe") %>%
  mutate(
    households = as.numeric(households),
    time = str_replace(
      paste(
        str_replace(
          str_replace(
            str_replace(
              str_replace(range, "_moe", ""),
              "travel_", ""
            ),
            "to", " - "
          ),
          "less", "0 - "
        ),
        " min",
        sep = ""
```

```
      ),
      "over", " +"
    ),
    idx = as.integer(str_sub(time, start = 1,
                             end = str_locate(time, " ")[, 1] - 1))
  ) %>%
  unique()
remove(tmp)

# ++ 3.4.7 create us_metro_img ----
us_metro_img <- datausa_acs_yg_usmetro %>%
  group_by(us_metro_id) %>%
  summarize(mx = max(pop)) %>%
  left_join(select(
    datausa_acs_yg_usmetro, pop,
    image_link, image_author
  ),
  by = c("mx" = "pop")
  ) %>%
  select(-mx) %>%
  left_join(select(
    us_metro_rank,
    id, display_name
  ),
  by = c("us_metro_id" = "id")
  )

# ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# *DFW Market ONLY* ----
# ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

# filter(datausa_locations, name == "Texas") %>% select(id)
# ++ 3.5.1 create geo_ids ----
geo_ids <- filter(
  datausa_locations,
  sumlevel == 160 & (
    name == "Arlington, TX" |
      name == "Dallas, TX" |
      name == "Denton, TX" |
      name == "Fort Worth, TX" |
      name == "Frisco, TX" |
      name == "Tyler, TX" |
      name == "Waco, TX" |
      name == "Waxahachie, TX" |
      name == "Weatherford, TX" |
```

```
      name == "Irving, TX" |
      name == "Grapevine, TX" |
      name == "Colleyville, TX" |
      name == "Southlake, TX" |
      name == "Las Colinas, TX" |
      name == "Coppell, TX" |
      name == "Bedford, TX" |
      name == "Euless, TX")
) %>%
  select(name, id)
geo_txt <- geo_ids$id[1]

for (i in 2:nrow(geo_ids)) {
  geo_txt <- paste(geo_txt, ",", geo_ids$id[i], sep = "")
}


# ++ 3.5.2 create datausa_industries ----
datausa_industries <- fromJSON("http://api.datausa.io/attrs/naics/")
tmp <- as_data_frame(datausa_industries$data)
colnames(tmp) <- datausa_industries$headers
datausa_industries <- tmp

# ++ 3.5.3 create datausa_occupations ----
datausa_occupations <- fromJSON("http://api.datausa.io/attrs/soc/")
tmp <- as_data_frame(datausa_occupations$data)
colnames(tmp) <- datausa_occupations$headers
datausa_occupations <- tmp

# ++ 3.5.4 create datausa_acs_yg ----
datausa_acs_yg <- fromJSON(paste(
    "https://api.datausa.io/api/?sort=desc&force=acs.yg&show=geo&sumlevel=
all&year=all&geo=",
    geo_txt, sep = ""))
tmp <- as_data_frame(datausa_acs_yg$data)
colnames(tmp) <- datausa_acs_yg$headers
tmp <- geo_ids %>%
  left_join(tmp, by = c("id" = "geo")) %>%
  select(-id, -estimate, -age_rank, -pop_rank, -income_rank)
datausa_acs_yg <- tmp %>%
  filter(!(name %in% c("Irving, TX", "Grapevine, TX", "Colleyville, TX",
                       "Southlake, TX", "Las Colinas, TX", "Coppell, TX",
                       "Bedford, TX", "Euless, TX"))) %>%
  mutate(
    age = as.numeric(age),
```

```
    age_moe = as.numeric(age_moe),
    pop = as.numeric(pop),
    pop_moe = as.numeric(pop_moe),
    non_us_citizens = as.numeric(non_us_citizens),
    mean_commute_minutes = as.numeric(mean_commute_minutes),
    income = as.numeric(income),
    income_moe = as.numeric(income_moe),
    owner_occupied_housing_units = as.numeric(owner_occupied_housing_units
),
    median_property_value = as.numeric(median_property_value),
    median_property_value_moe = as.numeric(median_property_value_moe),
    us_citizens = as.numeric(us_citizens),
    non_eng_speakers_pct = as.numeric(non_eng_speakers_pct)
  )

# ++ 3.5.5 create dfw ----
dfw <- tmp %>%
  filter(name %in% c(
    "Grapevine, TX",
    "Colleyville, TX",
    "Southlake, TX",
    "Las Colinas, TX",
    "Coppell, TX"
  )) %>%
  # filter(name %in% c("Irving, TX", "Grapevine, TX", "Colleyville, TX", "
Southlake, TX", "Las Colinas, TX", "Coppell, TX", "Bedford, TX", "Euless,
TX")) %>%
  mutate(
    name = "DFW Airport, TX",
    age =
        as.numeric(age) * as.numeric(pop),
    age_moe =
        as.numeric(age_moe) * as.numeric(pop),
    pop =
        as.numeric(pop),
    pop_moe =
        as.numeric(pop_moe),
    non_us_citizens =
        as.numeric(non_us_citizens) * as.numeric(pop),
    mean_commute_minutes =
        as.numeric(mean_commute_minutes) * as.numeric(pop),
    income =
        as.numeric(income) * as.numeric(pop),
    income_moe =
        as.numeric(income_moe) * as.numeric(pop),
```

```
    owner_occupied_housing_units =
        as.numeric(owner_occupied_housing_units) * as.numeric(pop),
    median_property_value =
        as.numeric(median_property_value) * as.numeric(pop),
    median_property_value_moe =
        as.numeric(median_property_value_moe) * as.numeric(pop),
    us_citizens =
        as.numeric(us_citizens) * as.numeric(pop),
    non_eng_speakers_pct =
        as.numeric(non_eng_speakers_pct) * as.numeric(pop)
  ) %>%
  group_by(name, year) %>%
  summarize(
    age = sum(age) / sum(pop),
    age_moe = sum(age_moe) / sum(pop),
    pop = sum(pop),
    pop_moe = sum(pop_moe),
    non_us_citizens = sum(non_us_citizens) / sum(pop),
    mean_commute_minutes = sum(mean_commute_minutes) / sum(pop),
    income = sum(income) / sum(pop),
    income_moe = sum(income_moe) / sum(pop),
    owner_occupied_housing_units = sum(owner_occupied_housing_units) / sum
(pop),
    median_property_value = sum(median_property_value) / sum(pop),
    median_property_value_moe = sum(median_property_value_moe) / sum(pop),
    us_citizens = sum(us_citizens) / sum(pop),
    non_eng_speakers_pct = sum(non_eng_speakers_pct) / sum(pop)
  )

datausa_acs_yg <- bind_rows(datausa_acs_yg, dfw)

# ++ 3.5.6 create datausa_acs_yg_income_dist ----
datausa_acs_yg_income_dist <- fromJSON(paste(
    "https://api.datausa.io/api/?sort=desc&force=acs.yg_income_distributio
n&show=geo&sumlevel=all&year=all&geo=",
    geo_txt, sep = ""))
tmp <- as_data_frame(datausa_acs_yg_income_dist$data)
colnames(tmp) <- datausa_acs_yg_income_dist$headers
tmp <- geo_ids %>%
  left_join(tmp, by = c("id" = "geo")) %>%
  select(-id) %>%
  gather(
    key = "range", value = "households", "income_100to125",
    "income_100to125_moe", "income_10to15", "income_10to15_moe",
    "income_125to150", "income_125to150_moe", "income_150to200",
```

```
    "income_150to200_moe", "income_15to20", "income_15to20_moe",
    "income_200over", "income_200over_moe", "income_20to25",
    "income_20to25_moe", "income_25to30", "income_25to30_moe",
    "income_30to35", "income_30to35_moe", "income_35to40",
    "income_35to40_moe", "income_40to45", "income_40to45_moe",
    "income_45to50", "income_45to50_moe", "income_50to60",
    "income_50to60_moe", "income_60to75", "income_60to75_moe",
    "income_75to100", "income_75to100_moe", "income_under10",
    "income_under10_moe", "totalhouseholds", "totalhouseholds_moe"
  ) %>%
  mutate(households = as.numeric(households))

tmp$income <- "0"
tmp$idx <- 0
for (i in 1:nrow(tmp)) {
  if (str_detect(tmp$range[i], "totalhouseholds")) {
    tmp$income[i] <- "9999"
    tmp$idx[i] <- 9999
  } else {
    tmp$income[i] <- str_replace(
      paste(
        str_replace(
          str_replace(
            str_replace(
              str_replace(tmp$range[i], "_moe", ""),
              "income_", "$"
            ),
            "to", "K - $"
          ),
          "\\$under", "$0K - $"
        ),
        "K",
        sep = ""
      ),
      "overK", "K +"
    )
    tmp$idx[i] <- as.integer(str_sub(tmp$income[i],
                                     start = str_locate(tmp$income[i], "\\$")[1] +
1,
                                     end = str_locate(tmp$income[i], "K")[1] - 1))
  }
}

datausa_acs_yg_income_dist <- tmp %>%
  filter(!(name %in% c("Irving, TX", "Grapevine, TX", "Colleyville, TX",
```

```
                      "Southlake, TX", "Las Colinas, TX", "Coppell, TX",
                      "Bedford, TX", "Euless, TX")))

dfw <- tmp %>%
  filter(name %in% c("Grapevine, TX", "Colleyville, TX", "Southlake, TX",
                     "Las Colinas, TX", "Coppell, TX")) %>%
  # filter(name %in% c("Irving, TX", "Grapevine, TX", "Colleyville, TX",
    # "Southlake, TX", "Las Colinas, TX", "Coppell, TX", "Bedford, TX",
    # "Euless, TX")) %>%
  mutate(name = "DFW Airport, TX") %>%
  group_by(name, year, range, income, idx) %>%
  summarize(households = sum(households))
datausa_acs_yg_income_dist <- bind_rows(datausa_acs_yg_income_dist, dfw) %
>%
  arrange(name, year, idx)

# ++ 3.5.7 create datausa_acs_yg_travel_time ----
datausa_acs_yg_travel_time <- fromJSON(paste(
    "https://api.datausa.io/api/?sort=desc&force=acs.yg_travel_time&show=g
eo&sumlevel=all&year=all&geo=",
  geo_txt,
  sep = ""
))
tmp <- as_data_frame(datausa_acs_yg_travel_time$data)
colnames(tmp) <- datausa_acs_yg_travel_time$headers
tmp <- geo_ids %>%
  left_join(tmp, by = c("id" = "geo")) %>%
  select(-id) %>%
  gather(key = "range", value = "households", "travel_10to14",
         "travel_10to14_moe", "travel_15to19", "travel_15to19_moe",
         "travel_20to24", "travel_20to24_moe", "travel_25to29",
         "travel_25to29_moe", "travel_30to34", "travel_30to34_moe",
         "travel_35to39", "travel_35to39_moe", "travel_40to44",
         "travel_40to44_moe", "travel_45to59", "travel_45to59_moe",
         "travel_5to9", "travel_5to9_moe", "travel_60to89",
         "travel_60to89_moe", "travel_90over", "travel_90over_moe",
         "travel_less5", "travel_less5_moe", "workers", "workers_moe") %>%
  mutate(households = as.numeric(households))

tmp$time <- "0"
tmp$idx <- 0
for (i in 1:nrow(tmp)) {
  if (str_detect(tmp$range[i], "workers")) {
    tmp$time[i] <- "9999"
    tmp$idx[i] <- 9999
```

```
  } else {
    tmp$time[i] <- str_replace(
      paste(
        str_replace(
          str_replace(
            str_replace(
              str_replace(tmp$range[i], "_moe", ""),
              "travel_", ""
            ),
            "to", " - "
          ),
          "less", "0 - "
        ),
        " min",
        sep = ""
      ),
      "over", " +"
    )
    tmp$idx[i] <- as.integer(str_sub(tmp$time[i], start = 1, end = str_loc
ate(tmp$time[i], " ")[1] - 1))
  }
}

# filter datausa_acs_yg_travel_time
datausa_acs_yg_travel_time <- tmp %>%
  filter(!(name %in% c("Irving, TX", "Grapevine, TX", "Colleyville, TX",
                       "Southlake, TX", "Las Colinas, TX", "Coppell, TX",
                       "Bedford, TX", "Euless, TX")))

dfw <- tmp %>%
  filter(name %in% c("Grapevine, TX", "Colleyville, TX", "Southlake, TX",
                     "Las Colinas, TX", "Coppell, TX")) %>%
  # filter(name %in% c("Irving, TX", "Grapevine, TX", "Colleyville, TX",
    # "Southlake, TX", "Las Colinas, TX", "Coppell, TX", "Bedford, TX",
    # "Euless, TX")) %>%
  mutate(name = "DFW Airport, TX") %>%
  group_by(name, year, range, time, idx) %>%
  summarize(households = sum(households))
datausa_acs_yg_travel_time <- bind_rows(datausa_acs_yg_travel_time, dfw) %
>%
  arrange(name, year, idx)

remove(tmp)
remove(dfw)
remove(geo_txt)
```

```
# *MySQL Connection* ----
# Edit dbConnect() call - CSS-IVHM server
# con <- dbConnect(RMySQL::MySQL(),
#                  dbname = "css",
#                  host = "10.224.123.68",
#                  port = 3306,
#                  user = "slawrence",
#                  password = "")
#
# sales <- dbGetQuery(con,
#                     "
#                     SELECT
#                     sales_header_id,
#                     po_number,
#                     customer_id,
#                     part_id,
#                     part_family_id,
#                     part_number,
#                     part_description,
#                     part_segment,
#                     css_reporting.dim_part_segments.`name` AS part_segme
nt_name,
#                     state AS part_state,
#                     price,
#                     `type` AS part_type,
#                     ata_id,
#                     css_reporting.dim_part_ata.`name` AS design_discipli
ne,
#                     sales_type_id,
#                     create_date,
#                     requested_delivery_date,
#                     net_value AS purchase,
#                     order_qty AS qty
#                     FROM css_reporting.sales
#                     LEFT JOIN css_reporting.dim_part_names USING (part_i
d)
#                     LEFT JOIN css_reporting.dim_part_family USING (part_
id)
#                     LEFT JOIN css_reporting.parts USING (part_id)
#                     LEFT JOIN css_reporting.dim_part_ata USING (ata_id)
#                     LEFT JOIN css_reporting.dim_part_segments USING (par
t_segment)
#                     WHERE
#                     (sales_type_id = 20 OR sales_type_id = 33 OR sales_t
```

```
ype_id = 42)
#                     AND YEAR(requested_delivery_date) >= 2013
#                     AND NOT ISNULL(customer_id)
#                     ;
#                     "
# )
#
# # Disconnect
# dbDisconnect(con)
# remove(con)

# ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~#
# *Process Data Sets* ----
# ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~#

# ++ 4.0.1 process txt_aadt ----
txt <- "txdot_aadt_g <- txdot_aadt %>% \n  select(lat, lng, DIST_NM, CNTY_
NM, T_FLAG"
years <- 10
txt_aadt <- ""
for (i in 1:years) {
  txt_aadt <- paste(txt_aadt, ", AADT_", 2017 - i + 1, sep = "")
}
txt <- paste(txt, txt_aadt, sep = "")
for (i in 1:nrow(nexus_ports)) {
  txt <- paste(txt, ", dist_m_",
               str_replace_all(nexus_ports$name[i], " ", "_"), sep = "")
}
txt <- paste(txt, ") %>% gather(key = Year, value = AADT",
             txt_aadt, ') %>% mutate(Year = str_replace(Year, "AADT_", "")
)',
             sep = "")
eval(parse(text = txt))
remove(i)
remove(txt)
remove(txt_aadt)
remove(years)

# ++ 4.0.2 process nexus_ports ----
aadt_rad <- 5
for (i in 1:nrow(nexus_ports)) {
  eval(parse(text =
    paste("nexus_ports$aadt_rad[i] <- txdot_aadt %>%\n    filter(dist_m_",
          str_replace_all(nexus_ports$name[i], " ", "_"),
          " <= aadt_rad * 1609.344) %>%\n    select(AADT_2017) %>%\n    ma
```

```
x()",
          sep = "")))
  eval(parse(text = paste(
      "nexus_ports$aadt_rad_tflag[i] <- txdot_aadt %>%\n    filter(dist_m_
",
  str_replace_all(nexus_ports$name[i], " ", "_"),
  " <= aadt_rad * 1609.344, AADT_2017 == nexus_ports$aadt_rad[i]) %>%\n
select(T_FLAG) %>%\n    as.character()",
                          sep = "")))
}
remove(i)
aadt_rad

# ++ 4.0.3 DOC Simulation ----
n <- 10000
fleet_size <- rweibull(n, shape = 2.93, scale = 29.58968)
fh_us <- 275.2 + 138.5 * log(fleet_size)
fh_nus <- 162.03 + 81.56 * log(fleet_size)
rev_us <- (896.05 + fh_us - 150 * log(fh_us) + 5000 * log(fh_us) / fh_us^2
) * 1.94532 # 1.6211
rev_nus <- (2445.16 + 5 * fh_nus - 450 * log(fh_nus) + 14000 * log(fh_nus)
/ fh_us^2) * 1.94532
profit_us <- rep.int(0, n)
profit_nus <- rep.int(0, n)

DOC_us <- data_frame(
  Total = rep.int(100, n),
  Maintenance = rep.int(0, n),
  Insurance = rep.int(0, n),
  Fuel = rep.int(0, n),
  Financing_Exp = rep.int(0, n),
  Depreciation = rep.int(0, n),
  Personnel_Exp = rep.int(0, n),
  Training_Gen_Admin = rep.int(0, n),
  Other = rep.int(0, n)
)
DOC_nus <- data_frame(
  Total = rep.int(100, n),
  Maintenance = rep.int(0, n),
  Insurance = rep.int(0, n),
  Fuel = rep.int(0, n),
  Financing_Exp = rep.int(0, n),
  Depreciation = rep.int(0, n),
  Personnel_Exp = rep.int(0, n),
  Training_Gen_Admin = rep.int(0, n),
```

```
  Other = rep.int(0, n)
)
for (i in 1:n) {
  if (fleet_size < 1.5) {
    profit_us[i] <- rnorm(1, mean = -0.33, sd = 5)
    profit_nus[i] <- rnorm(1, mean = -0.33, sd = 5)
    DOC_us$Maintenance[i] <- 32.7
    DOC_us$Insurance[i] <- 17.4
    DOC_us$Fuel[i] <- 18.0
    DOC_us$Financing_Exp[i] <- 4.0
    DOC_us$Depreciation[i] <- 3.5
    DOC_us$Personnel_Exp[i] <- 14.9
    DOC_us$Training_Gen_Admin[i] <- 6.2
    DOC_us$Other[i] <- 3.4
    DOC_nus$Maintenance[i] <- 32.2
    DOC_nus$Insurance[i] <- 16.8
    DOC_nus$Fuel[i] <- 17.5
    DOC_nus$Financing_Exp[i] <- 4.7
    DOC_nus$Depreciation[i] <- 4.1
    DOC_nus$Personnel_Exp[i] <- 15.1
    DOC_nus$Training_Gen_Admin[i] <- 6.9
    DOC_nus$Other[i] <- 2.8
  } else if (fleet_size >= 1.5 & fleet_size < 3.5) {
    profit_us[i] <- rnorm(1, mean = -0.33, sd = 3.74)
    profit_nus[i] <- rnorm(1, mean = -0.33, sd = 3.74)
    DOC_us$Maintenance[i] <- 32.3
    DOC_us$Insurance[i] <- 14.7
    DOC_us$Fuel[i] <- 17.9
    DOC_us$Financing_Exp[i] <- 4.4
    DOC_us$Depreciation[i] <- 3.4
    DOC_us$Personnel_Exp[i] <- 17.6
    DOC_us$Training_Gen_Admin[i] <- 6.4
    DOC_us$Other[i] <- 3.2
    DOC_nus$Maintenance[i] <- 31.8
    DOC_nus$Insurance[i] <- 14.1
    DOC_nus$Fuel[i] <- 17.4
    DOC_nus$Financing_Exp[i] <- 5.1
    DOC_nus$Depreciation[i] <- 4.0
    DOC_nus$Personnel_Exp[i] <- 17.9
    DOC_nus$Training_Gen_Admin[i] <- 7.1
    DOC_nus$Other[i] <- 2.6
  } else if (fleet_size >= 3.5 & fleet_size < 7.5) {
    profit_us[i] <- rnorm(1, mean = 3.29, sd = 5)
    profit_nus[i] <- rnorm(1, mean = 3.29, sd = 5)
    DOC_us$Maintenance[i] <- 29.2
```

```
    DOC_us$Insurance[i] <- 15.7
    DOC_us$Fuel[i] <- 16.8
    DOC_us$Financing_Exp[i] <- 4.7
    DOC_us$Depreciation[i] <- 3.4
    DOC_us$Personnel_Exp[i] <- 19.5
    DOC_us$Training_Gen_Admin[i] <- 6.7
    DOC_us$Other[i] <- 4.1
    DOC_nus$Maintenance[i] <- 28.8
    DOC_nus$Insurance[i] <- 15.1
    DOC_nus$Fuel[i] <- 16.3
    DOC_nus$Financing_Exp[i] <- 5.4
    DOC_nus$Depreciation[i] <- 4.0
    DOC_nus$Personnel_Exp[i] <- 19.7
    DOC_nus$Training_Gen_Admin[i] <- 7.4
    DOC_nus$Other[i] <- 3.5
  } else {
    profit_us[i] <- rnorm(1, mean = 7, sd = 4)
    profit_nus[i] <- rnorm(1, mean = 7, sd = 4)
    DOC_us$Maintenance[i] <- 30.5
    DOC_us$Insurance[i] <- 13.4
    DOC_us$Fuel[i] <- 16.3
    DOC_us$Financing_Exp[i] <- 4.9
    DOC_us$Depreciation[i] <- 3.8
    DOC_us$Personnel_Exp[i] <- 19.8
    DOC_us$Training_Gen_Admin[i] <- 7.4
    DOC_us$Other[i] <- 3.9
    DOC_nus$Maintenance[i] <- 30.0
    DOC_nus$Insurance[i] <- 12.8
    DOC_nus$Fuel[i] <- 15.8
    DOC_nus$Financing_Exp[i] <- 5.6
    DOC_nus$Depreciation[i] <- 4.4
    DOC_nus$Personnel_Exp[i] <- 20.1
    DOC_nus$Training_Gen_Admin[i] <- 8.1
    DOC_nus$Other[i] <- 3.3
  }
}
remove(i)

# ++ 4.0.4 DOC_us ----
DOC_us <- rev_us * (1 - profit_us / 100) * (DOC_us / 100)
# DOC_us$Insurance <- DOC_us$Insurance * fh_us / fleet_size
# DOC_us$Financing_Exp <- DOC_us$Financing_Exp * fh_us / fleet_size
# DOC_us$Depreciation <- DOC_us$Depreciation * fh_us / fleet_size

# ++ 4.0.4 DOC_nus ----
```

```
DOC_nus <- rev_nus * (1 - profit_nus / 100) * (DOC_nus / 100)
# DOC_nus$Insurance <- DOC_nus$Insurance * fh_us / fleet_size
# DOC_nus$Financing_Exp <- DOC_nus$Financing_Exp * fh_us / fleet_size
# DOC_nus$Depreciation <- DOC_nus$Depreciation * fh_us / fleet_size


# ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~#
# *Shape files* ----
# ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~#
#
# SECTION 6 Read OGR vector maps into Spatial objects -----
# THE ORIGINAL CODE:
library(rgdal)
# file directory
in_dir <- "data/TxDOT_Congestion/"
# import
shp_cur_cong <- rgdal::readOGR(paste(in_dir,
  "/TxDOT_Congestion.shp",
  sep = ""
),
stringsAsFactors = FALSE
)
# file directory
in_dir <- "data/TxDOT_Future_Congestion/"

shp_fut_cong <- rgdal::readOGR(paste(in_dir,
  "/TxDOT_Future_Congestion.shp",
  sep = ""
),
stringsAsFactors = FALSE
)

library(leaflet)
pal_usmetro <- colorFactor(
  palette =
    colorRampPalette(brewer.pal(
      name = "Paired",
      n = 12
    ))(nrow(us_metro_rank)),
  domain = us_metro_rank$id
)

# pal_cong <- colorFactor(palette = "Reds",
#                         domain = shp_cur_cong$CUR_CONG,
#                         reverse = TRUE)
```

```
# SECTION 7 Export data -----
save.image(file = paste0(
  "data/",
  # timestamp
  base::noquote(lubridate::today()),
  "-nexus-doc-helper.RData"
))
fs::dir_ls("data")
```