```
# Nevada Department of Employment, Training and Rehabilitation
# Research and Analysis Bureau
# Legislative District Profiles

# This script requires an API key from the Census Bureau.
# Getting and installing this is documented in the help for the tidycensus package


loader <- function(x){
  for(i in x){
    if(!require(i, character.only = TRUE) ){
      install.packages(i, dependencies = TRUE)
      require(i , character.only = TRUE)
    }
  }
}

loader(c("tigris", "spatial.tools","sf","sp","purrr", "tidyquant",
         "DBI","tidycensus","tigris","spdplyr","rgdal","readxl","tmap",
         "tmaptools", "lubridate", "magrittr","zoo","Rcpp","RcppRoll", "ggforce",
         "openxlsx", "scales", "rgeos", "raster", "rmapzen", "tidyverse"))
select = dplyr::select
extract = tidyr::extract

############################
### Set State Variable ###
############################

Abb_ST <- "AK"          # This is used in the code to get data
Name_State <- "Alaska" # This is used in labelling

################################
### Build ACS Variable List ###
################################

var1 <- load_variables(2017, "acs5")
var2 <- load_variables(2017, "acs5/subject")
var <- rbind(var1, var2)
rm(var1)
rm(var2)

var_filtered <- var %>%
  filter(str_detect(name, "2301_")) %>%

  separate(label,
c(NA,"Data","Demographic_Population","Demographic_Type","Demographic_Group","Demographic_Subgroup
_1","Demographic_Subgroup_2"), sep="!!") %>%
  mutate(Group = paste0(
    Demographic_Population,
    ifelse(!is.na(Demographic_Type),paste0(" ",Demographic_Type),""),
    ifelse(!is.na(Demographic_Group),paste0(": ",Demographic_Group),""),
    ifelse(!is.na(Demographic_Subgroup_1),paste0(", ",Demographic_Subgroup_1),""),
    ifelse(!is.na(Demographic_Subgroup_2),paste0(", ",Demographic_Subgroup_2),"")
  )) %>%
  mutate(Group = str_to_title(Group)) %>%
  mutate(Group = str_replace_all(Group,c(" And "=" and ", " Or " = " or ", " To " = " to "))) %>%

  mutate(Group_Alone = paste0(

    ifelse(!is.na(Demographic_Group),paste0(Demographic_Group),""),
    ifelse(!is.na(Demographic_Subgroup_1),paste0(",\n ",Demographic_Subgroup_1),""),
    ifelse(!is.na(Demographic_Subgroup_2),paste0(",\n ",Demographic_Subgroup_2),"")
  )) %>%

  mutate(Group_Alone = str_to_title(Group_Alone)) %>%
  mutate(Group_Alone = str_replace_all(Group_Alone,c(" And "=" and ", " Or " = " or ", " To " = "
to ")))


########################
### Get TIGRIS Data ###
########################

options(tigris_use_cache = TRUE)
```

```
# Primary and Secondary Roads
STroads <- primary_secondary_roads(Abb_ST,year=2017)

STroads_Primary <- STroads[STroads$RTTYP == "I",]
STroads_Secondary <- STroads[STroads$RTTYP == "U",]

# Counties lines for statewide map
STCounties <- counties(Abb_ST)

# Census Blocks
#STBlocks <- blocks(Abb_ST)

# Zip Code Areas
#STZips <- zctas(state=Abb_ST)

# State Assembly and Senate Districts
ST_Districts_Lower <- state_legislative_districts(Abb_ST, house = "lower") %>%
  rename(SLDST = SLDLST)
ST_Districts_Upper <- state_legislative_districts(Abb_ST, house = "upper")%>%
  rename(SLDST = SLDUST)
ST_Districts <- rbind(ST_Districts_Lower,ST_Districts_Upper)

ST_Districts <- ST_Districts %>%
  select(NAMELSAD) %>%
  mutate(rowid = 1:nrow(ST_Districts))


####################
### Get ACS Data ###
####################

# Use variable list to call data from ACS.

data1 <- get_acs(geography="tract", state=Abb_ST, variables = var_filtered$name, output = "tidy",
geometry = TRUE, cache_table=TRUE, survey = "acs5")

ST_Tract_Data <- data1 %>%
  select(-moe) %>%
  left_join(select(var_filtered,name,Data), by = c("variable"="name")) %>%
  separate(variable,into=c("AAA",NA,"BBB"),sep="_") %>%
  mutate(variable = paste0(AAA,"_C01_",BBB)) %>%
  select(-c(AAA,BBB)) %>%
  spread(Data, estimate) %>%
  left_join(var_filtered, by = c("variable"="name")) %>%
  select(-c(Data, Demographic_Group, Demographic_Subgroup_1, Demographic_Subgroup_2, concept))
%>%
  mutate(Group_Alone = ifelse(Group_Alone == "",Group,Group_Alone)) %>%
  rename(LFPR = "Labor Force Participation Rate", UR = "Unemployment rate", Population = Total,
EPR = "Employment/Population Ratio") %>%
  mutate(Unemployed = Population*LFPR*UR/10000,
         LF = LFPR*Population/100,
         Employed = LF - Unemployed)

Demographic_List <- as.data.frame(unique(ST_Tract_Data$Group_Alone))
Demographic_List <- rowid_to_column(Demographic_List, "ID")
Demographic_List <- rename(Demographic_List,"Group" = "unique(ST_Tract_Data$Group_Alone)")
Demographic_List$Ordered_Groups <- paste0("Group ",str_pad(Demographic_List$ID,2,pad="0"),":
",Demographic_List$Group)

ST_Tract_Data <- left_join(ST_Tract_Data, select(Demographic_List, Group, Ordered_Groups), by =
c("Group_Alone" = "Group"))


Population2064 <- c("Group 21: Population 20 to 64 Years",
                    "Group 22: Male",
                    "Group 23: Female",
                    "Group 24: Female,\n With Own Children Under 18 Years",
                    "Group 25: Female,\n With Own Children Under 18 Years,\n With Own Children
Under 6 Years Only",
                    "Group 26: Female,\n With Own Children Under 18 Years,\n With Own Children
Under 6 Years and 6 to 17 Years",
                    "Group 27: Female,\n With Own Children Under 18 Years,\n With Own Children
Under 6 to 17 Years Only",
                    "Group 28: Below Poverty Level",
                    "Group 29: At or Above The Poverty Level",
```

```
                    "Group 30: With Any Disability")
Population2564 <- c("Group 31: Educational Attainment Population 25 to 64 Years",
                    "Group 32: Less Than High School Graduate",
                    "Group 33: High School Graduate (Includes Equivalency)",
                    "Group 34: Some College or Associate's Degree",
                    "Group 35: Bachelor's Degree or Higher")




##############################
### Loop Through Districts ###
##############################

# Clear previous spreadsheets, if any
rm(Intersection_Results)

### Pre-Loop Setup ###
# Map Values Pre-Loop
st_crs(ST_Tract_Data) <- 4269

map_name1 <- "Overlapping Tracts"
legend_units_text <- "Population"
footer <- "Data by Census Tract from U.S. Census Bureau, 2013-2017 American Community
Survey\nReport Prepared by Nevada Department of Employment, Training, and Rehabilitation Research
and Analysis Bureau"
ST_Tract_Data$mapping_data <- ST_Tract_Data$Population

tm_Area <- list()

### Excel Workbook Creation Pre-Loop
# Workforce Demographics ACS Table
Intersection_Results <- createWorkbook()
percent_style <- createStyle(numFmt = "0.0%")
header_style <- createStyle(wrapText = TRUE, border = "bottom", borderStyle = "double", halign =
"center", valign = "center")

EQUI_header_style <- createStyle(wrapText = TRUE, border = "bottom", borderStyle = "double",
halign = "center", valign = "top")
summary_style <- createStyle(fontSize = 14, textDecoration = "bold", halign = "left")
disclaimer_style <- createStyle(fontSize = 10, textDecoration = "italic", wrapText = TRUE, halign
= "left")

HUExplanation <- "This table lists those demographic groups with an unemployment rate four
percentage points higher than the district as a whole, or more than double the rate of the
district as a whole."

# Create Buffer Inside Perimiter
Buffer_Meters <- -50
Buffer_Degrees <- Buffer_Meters/90000

# Create folder for district maps
 dir.create("district_maps")

### Loop Starts ###
for (i in 1:nrow(ST_Districts)) {

############################
### Define Local Area(s) ###
############################

# Define Buffer around Area to Use
# Lat/Long uses Buffer in Meters, Polygons use degrees
# @ 36 degrees latitude, 1 degree is approximately 90,000m
# Converting m and degrees to miles, for conveneince



# Tell the code below what to use for comparisons
Comparison_Area <- gBuffer(ST_Districts[i,], width = Buffer_Degrees)
Area_Name <- as.character(ST_Districts@data[i,1])

#Test Outputs for Sanity
```

```
tm_shape(STCounties)+tm_polygons()+
  tm_shape(Comparison_Area)+tm_polygons(border.col="red", alpha=0)+
  tm_shape(STroads_Primary) +
  tm_lines(col="black", lwd = 3)+
  tm_shape(STroads_Secondary) +
  tm_lines(col="black", lwd = 2)


########################
### Identify Overlap ###
########################

#Define Coordinate System for data
st_crs(ST_Tract_Data) <- 4269

#Find Intersecting data elements
Intersecting_Tracts <- st_intersects(st_as_sf(Comparison_Area), ST_Tract_Data)

#Structure data to extract matches (on row name / record basis)
Intersection_Table <- ST_Tract_Data %>%
  rownames_to_column(var="DataRow") %>%
  mutate(DataRow = as.integer(DataRow)) %>%
  filter(DataRow %in% Intersecting_Tracts[[1]])

###################
### Data Tables ###
###################
SummaryHeader <- paste0("Workforce Summary for ",Area_Name)
SummaryDisclaimer <- "Workforce data by census tract from the U.S. Census Bureau, American
Community Survey five-year average data."

#Summarize Grouped Data
Intersection_Data <- st_drop_geometry(Intersection_Table) %>%
  group_by(Ordered_Groups) %>%
  summarize(Employment = round(sum(Employed, na.rm = TRUE)),
            Unemployment = round(sum(Unemployed, na.rm = TRUE)),
            Labor_Force = round(sum(LF, na.rm = TRUE)),
            Population = round(sum(Population, na.rm = TRUE)),
            Unemployment_Rate = Unemployment / Labor_Force,
            'Labor Force Participation Rate' = Labor_Force / Population,
            'Employment Population Ratio' = Employment / Population)
Intersection_Data %<>%
  mutate(
    Comparison_Rate = case_when(
     Ordered_Groups %in% Population2064 ~ as.numeric(Intersection_Data[21,6]),
     Ordered_Groups %in% Population2564 ~ as.numeric(Intersection_Data[31,6]),
     TRUE ~ as.numeric(Intersection_Data[1,6])),
    High_Unemployment_Group = case_when(
      Unemployment_Rate - Comparison_Rate >=0.04 ~ TRUE,
      Unemployment_Rate / Comparison_Rate >= 2.00 ~ TRUE,
      TRUE ~ FALSE)
    ) %>%
  rename('Labor Force' = Labor_Force,
         'Demographic Group' = Ordered_Groups,
         'Unemployment Rate' = Unemployment_Rate) %>%
  select(1,5,4,2,3,6,7,8,10)

HU_Data <- filter(Intersection_Data, High_Unemployment_Group == TRUE) %>%
  rename('High Unemployment Groups' = 'Demographic Group')


###Create Output files
## ACS Results
addWorksheet(Intersection_Results, Area_Name)

class(Intersection_Data$Employment) <- "comma"
class(Intersection_Data$Unemployment) <- "comma"
class(Intersection_Data$'Labor Force') <- "comma"
class(Intersection_Data$Population) <- "comma"

class(HU_Data$Employment) <- "comma"
class(HU_Data$Unemployment) <- "comma"
class(HU_Data$'Labor Force') <- "comma"
class(HU_Data$Population) <- "comma"
```

```
# Data for Whole Area
writeDataTable(Intersection_Results, Area_Name, select(Intersection_Data, 1:8), startRow = 3,
tableStyle = "TableStyleMedium5")

addStyle(Intersection_Results, sheet = Area_Name, style = percent_style, cols = 6, rows = 4:38)
addStyle(Intersection_Results, sheet = Area_Name, style = percent_style, cols = 7, rows = 4:38)
addStyle(Intersection_Results, sheet = Area_Name, style = percent_style, cols = 8, rows = 4:38)
addStyle(Intersection_Results, sheet = Area_Name, style = header_style, cols = 1:8, rows = 3)

# Data for High Unemployment Groups
writeDataTable(Intersection_Results, Area_Name, select(HU_Data,1:8), startRow = 41, tableStyle =
"TableStyleMedium3")

addStyle(Intersection_Results, sheet = Area_Name, style = percent_style, cols = 6, rows = 42:77)
addStyle(Intersection_Results, sheet = Area_Name, style = percent_style, cols = 7, rows = 42:77)
addStyle(Intersection_Results, sheet = Area_Name, style = percent_style, cols = 8, rows = 42:77)
addStyle(Intersection_Results, sheet = Area_Name, style = header_style, cols = 1:8, rows = 41)

setColWidths(Intersection_Results, sheet = Area_Name, cols=1:8, widths=14)
setColWidths(Intersection_Results, sheet = Area_Name, cols=1, widths=60)
setRowHeights(Intersection_Results, sheet = Area_Name, rows=3, heights=45)
setRowHeights(Intersection_Results, sheet = Area_Name, rows=41, heights=45)

mergeCells(Intersection_Results,sheet = Area_Name, cols = 1:8, rows = 1)
mergeCells(Intersection_Results,sheet = Area_Name, cols = 1:8, rows = 2)
mergeCells(Intersection_Results,sheet = Area_Name, cols = 1:8, rows = 40)

writeData(Intersection_Results, sheet = Area_Name, xy = c(1,1), SummaryHeader)
writeData(Intersection_Results, sheet = Area_Name, xy = c(1,2), SummaryDisclaimer)
writeData(Intersection_Results, sheet = Area_Name, xy = c(1,40), HUExplanation)

addStyle(Intersection_Results, sheet = Area_Name, cols=1, rows=1, summary_style)


writeComment(Intersection_Results, sheet = Area_Name, xy = c(3,3),
             comment = createComment("Individuals working or looking for work at the time of the
survey.", author = "DETR", style = NULL,
                                     visible = FALSE, width = 4, height = 3))
writeComment(Intersection_Results, sheet = Area_Name, xy = c(4,3),
             comment = createComment("Individuals who are working at the time of the survey.",
author = "DETR", style = NULL,
                                     visible = FALSE, width = 4, height = 3))
writeComment(Intersection_Results, sheet = Area_Name, xy = c(5,3),
             comment = createComment("Individuals not working, who have looked in the past four
weeks at the time of the survey.", author = "DETR", style = NULL,
                                     visible = FALSE, width = 4, height = 3))
writeComment(Intersection_Results, sheet = Area_Name, xy = c(6,3),
             comment = createComment("Total Unemployment divided by Total Labor Force", author =
"DETR", style = NULL,
                                     visible = FALSE, width = 4, height = 3))
writeComment(Intersection_Results, sheet = Area_Name, xy = c(7,3),
             comment = createComment("Total Labor Force divided by Total Population", author =
"DETR", style = NULL,
                                     visible = FALSE, width = 4, height = 3))
writeComment(Intersection_Results, sheet = Area_Name, xy = c(8,3),
             comment = createComment("Total Employment divided by Total Population", author =
"DETR", style = NULL,
                                     visible = FALSE, width = 4, height = 3))


freezePane(Intersection_Results, sheet = Area_Name, firstActiveRow = 3, firstActiveCol = 2)
pageSetup(Intersection_Results, sheet = Area_Name, orientation="landscape", fitToWidth=TRUE,
fitToHeight=TRUE)

####################
### Create Maps ###
####################

bb_Area <- bb(Comparison_Area, ylim=c(-2,3), xlim=c(-2,3),relative=TRUE)
bb_AreaCrop <- bbox_to_SpatialPolygons(bb(Comparison_Area, ylim=c(-2,3), xlim=c(-
2,3),relative=TRUE, output="extent"), CRS("+proj=longlat +ellps=GRS80 +towgs84=0,0,0,0,0,0,0
+no_defs"))

tm_Area[[i]] <-
  tm_layout(main.title=paste0("Census Tracts Overlapping ",Area_Name),
```

```
            legend.outside=TRUE,
            legend.bg.color="white",
            legend.bg.alpha=0.5,
            legend.frame="black",
            legend.outside.position="right",
            attr.outside=TRUE
  )+

  #Shape gets data from ST_Tract_Data, bounded by the Area parameters set above
  tm_shape(filter(Intersection_Table, Ordered_Groups == "Group 01: Population 16 Years and
Over"), bbox=bb_Area) +

  tm_fill(col = "Population",
          title = "Population",
          palette = "Blues",
          n=25,
          textNA = "No Data",
          style="quantile"
  )+

  #Adding Road data for reference

#  tm_shape(crop_shape(STroads_Primary,bb_AreaCrop), bbox=bb_Area) +
#  tm_lines(col="black", lwd = 3)+

#  tm_shape(crop_shape(STroads_Secondary,bb_AreaCrop)) +
#  tm_lines(col="black", lwd = 2)+

  tm_shape(crop_shape(STroads,bb_AreaCrop)) +
  tm_lines(col="black", lwd = 0.5)+

  tm_shape(Comparison_Area) +
  tm_polygons(alpha = 0,
              border.col = "red",
              lwd = 3.0
  )+


  #Credits / Annotation
  tm_credits(
    text = footer,
    position = c("left","bottom"),
    size = 1.2
 )

# PDFs of counted tracts for district.  Comment out if maps are not needed.
#tmap_save(tm_Area[[i]], paste0("district_maps/",Name_State," ",ST_Districts@data[i,1],".pdf"),
width = 8.5, height = 11, units = "in")


###################
### Close Loop ###
###################
}
# End Loop
# Close PDF

# Save Workbook
saveWorkbook(Intersection_Results, paste0("Workforce Details, ",Name_State," Legislative
Districts.xlsx"), overwrite = TRUE)
```