Diabetes Prevalence in Brownsville, Texas

Simulated Demo

Gabby Novak

```
#### Packages ####
library(tigris)
library(knitr)
library(kableExtra)
library(viridis)
library(extrafont)
library(tidyverse)
library(lubridate)
library(rgdal)
library(survey)
library(INLA)
```

This project used data from the Cameron County Hispanic Cohort (CCHC) gathered by the University of Texas School of Public Health. The cohort collects health information on Hispanic individuals living in far-Southern Texas through a battery of questionaires, physical examinations, and labratory tests. My team sought to apply the techniques of geospatial anslysis to examine the variation of diabetes prevalence by census tract in the center Brownsville area in service of the identification of risk factors and better targetted interventions. This is not public data and results have not yet been published. For these reasons, I will be preforming similar analysis on a simulated data set. This simulation is additionally once removed from the original data set in that it only includes variables relevant to the actual analysis preformed. For this reason, much of the code used for the processing and cleaning of the cohort data is omitted from this.

The data we were working with spanned 2004 to 2018. However, we were examining prevalence on a Census Tract level (a geographic unit used by the US Census Beauru), and these vary according to population fluctuation. This meant that the 2010 census included tract numbers that did not exist in the 2000 census and vice versa. Additionally, a few tract borders shifted. All credit goes to Yunyun Jiang at University of Texas School of Pulic Health for developing the tract conversion procedure.

Writing Mapping Objects

The code below was only run once as it writes mapping files. These files were shared with the team and imported for use in the analysis.

Tract mapping files were downloaded using the `tigris` package which sources them directly from the Census Beauru's website.

```
#### Import base map, NOT RUN ####

tractmap2000<-tracts("TX",county="Cameron",year=2000)
tractmap2010<-tracts("TX",county="Cameron",year=2010)

# Get rid of problematic trailing zero
tractmap2000@data$NAME00[tractmap2000@data$NAME00=="126.10"]<-126.1
```

The study area was chosen based on data collection and then slightly modified to maintain the exterior borders between time periods (2004-2009 and 2010-2018).

```
#### Define study area ####
focus00<-c("125.04","125.07","126.04","126.05","126.06","126.07","126.08","126.09",
           "126.1","126.11","126.12","126.13","128","129","130.02","130.03","130.04",
           "131.02","131.04","131.06","132.03","132.04","132.05","132.06","132.07",
           "132.08","133.03","133.04","133.05","133.06","133.07","133.08","133.09",
           "134.01","134.02","135","136","137","138.01","138.02","139.01","139.02",
           "139.03","140.01","140.02","141")
# n tracts 2000 = 46

focus10<-c("125.04","125.07","126.07","126.08","126.09","126.12","126.13","128","129",
           "130.02","130.03","130.04","131.02","131.04","131.06","132.03","132.04",
           "132.05","132.06","132.07","133.03","133.05","133.06","133.07","133.08",
           "133.09","134.01","134.02","135","136","137","138.01","138.02","139.01",
           "139.02","139.03","140.01","140.02","141","143","144","145","9801")
# n tracts 2010 = 43
```

```
#### Subset mapping objects, NOT RUN ####
focusmap00<-tractmap2000[tractmap2000@data$NAME00 %in% focus00,]
focusmap10<-tractmap2010[tractmap2010@data$NAME10 %in% focus10,]

#### Write mapping files, NOT RUN ####
writeOGR(obj = focusmap00,layer="focusmap00",
         dsn = "focusmap00.shp",
         driver="ESRI Shapefile")
writeOGR(obj = focusmap10,layer="focusmap10",
         dsn = "focusmap10.shp",
         driver="ESRI Shapefile")
```

`writeOGR` function creates 4 mapping objects, all are referenced later.

Preparing Mapping Objects

```r
#### Read in Spatial Objects ####
# For source, see code above
focusmap2000<-readOGR("focusmap00.shp")
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "D:\Simulated-Research-Demos\DiabetesPrevalenceInBrownsvilleTexas\focusmap00.shp", layer: "fo
## with 46 features
## It has 14 fields
```

```r
focusmap2010<-readOGR("focusmap10.shp")
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "D:\Simulated-Research-Demos\DiabetesPrevalenceInBrownsvilleTexas\focusmap10.shp", layer: "fo
## with 43 features
## It has 14 fields
```

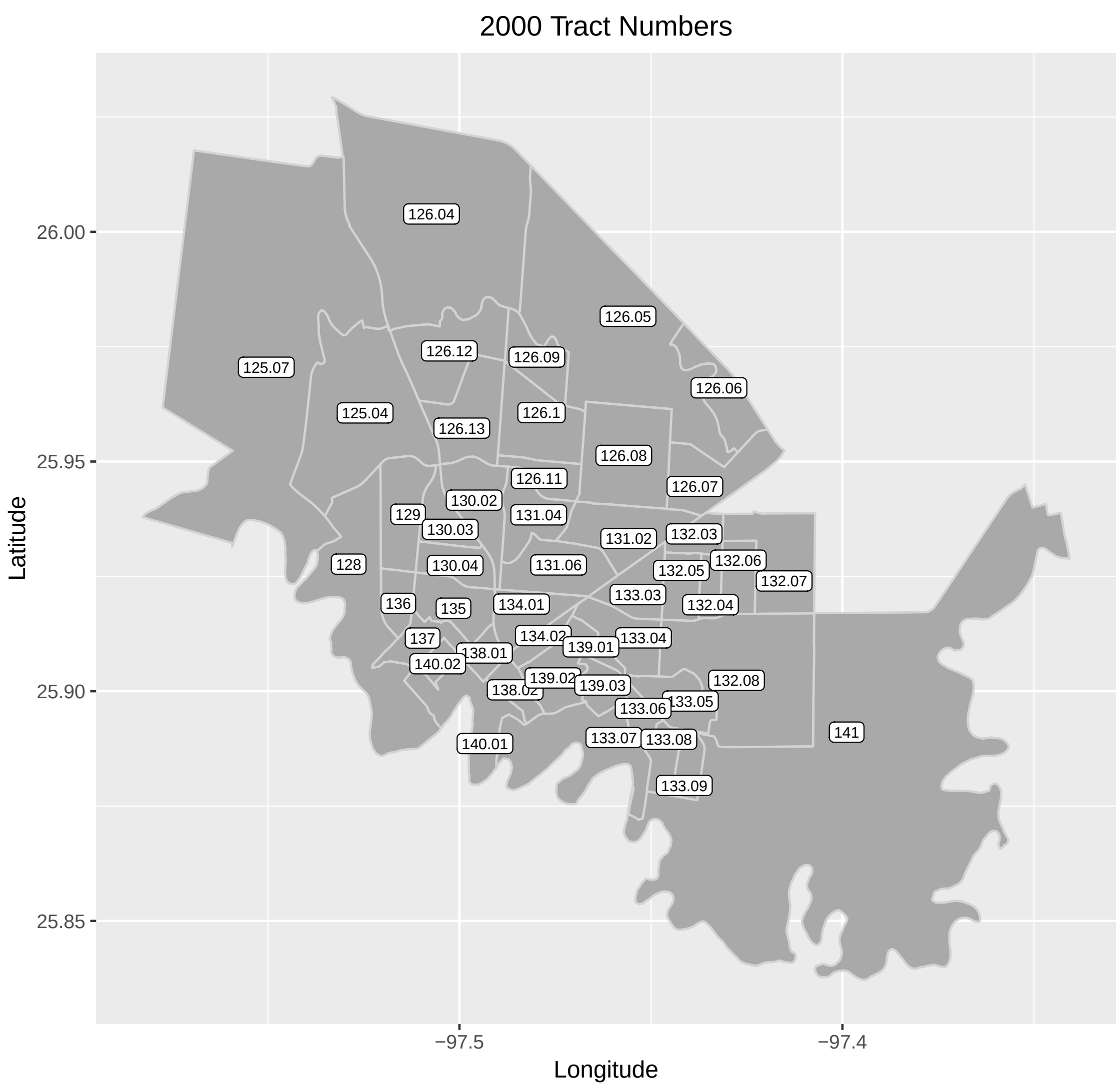
```r
#### Create ggplot-able objects ####
focusmap00<-fortify(focusmap2000, region="NAME00")
focusmap10<-fortify(focusmap2010, region="NAME10")

#### Ensure correct plotting order ####

focusmap00<-focusmap00[order(focusmap00$order),]
focusmap10<-focusmap10[order(focusmap10$order),]

#### Labeling Centroids ####

# 2000
# Simplifies each geographic region into 1 coordinate at the rought center of each group
names00 <- aggregate(cbind(long, lat) ~ id, data=focusmap00,
                     FUN=function(x)mean(range(x)))
# Adjusts for more legible labels
names00$long[names00$id==133.04]<-names00$long[names00$id==133.04]+.007
names00$long[names00$id==133.07]<-names00$long[names00$id==133.07]+.0105
names00$lat[names00$id==133.07]<-names00$lat[names00$id==133.07]+.005
names00$lat[names00$id==133.09]<-names00$lat[names00$id==133.09]-.004
names00$lat[names00$id==133.08]<-names00$lat[names00$id==133.08]+.004
names00$long[names00$id==132.07]<-names00$long[names00$id==132.07]+.004
names00$lat[names00$id==132.07]<-names00$lat[names00$id==132.07]-.004
names00$lat[names00$id==132.06]<-names00$lat[names00$id==132.06]+.004
names00$lat[names00$id==132.04]<-names00$lat[names00$id==132.04]-.004
names00$lat[names00$id==132.05]<-names00$lat[names00$id==132.05]+.0035
names00$long[names00$id==133.03]<-names00$long[names00$id==133.03]+.003
names00$lat[names00$id==133.03]<-names00$lat[names00$id==133.03]-.003
names00$long[names00$id==128]<-names00$long[names00$id==128]-.005
names00$lat[names00$id==128]<-names00$lat[names00$id==128]+.01
names00$lat[names00$id==130.02]<-names00$lat[names00$id==130.02]+.005
names00$lat[names00$id==130.03]<-names00$lat[names00$id==130.03]-.005
names00$long[names00$id==134.01]<-names00$long[names00$id==134.01]-.005
```

```r
names00$lat[names00$id==134.01]<-names00$lat[names00$id==134.01]+.004
names00$long[names00$id==133.05]<-names00$long[names00$id==133.05]+.005
names00$lat[names00$id==126.13]<-names00$lat[names00$id==126.13]-.004
names00$long[names00$id==126.07]<-names00$long[names00$id==126.07]-.008
names00$lat[names00$id==126.07]<-names00$lat[names00$id==126.07]-.003
names00$long[names00$id==140.01]<-names00$long[names00$id==140.01]+.005
names00$lat[names00$id==140.01]<-names00$lat[names00$id==140.01]-.005

# 2010
names10 <- aggregate(cbind(long, lat) ~ id, data=focusmap10,
                     FUN=function(x)mean(range(x)))

names10$long[names10$id==133.07]<-names10$long[names10$id==133.07]+.007
names10$lat[names10$id==133.07]<-names10$lat[names10$id==133.07]+.005
names10$lat[names10$id==133.09]<-names10$lat[names10$id==133.09]-.004
names10$lat[names10$id==133.08]<-names10$lat[names10$id==133.08]+.004
names10$long[names10$id==132.07]<-names10$long[names10$id==132.07]+.004
names10$lat[names10$id==132.07]<-names10$lat[names10$id==132.07]-.004
names10$lat[names10$id==132.06]<-names10$lat[names10$id==132.06]+.004
names10$lat[names10$id==132.04]<-names10$lat[names10$id==132.04]-.004
names10$lat[names10$id==132.05]<-names10$lat[names10$id==132.05]+.0035
names10$long[names10$id==133.03]<-names10$long[names10$id==133.03]+.003
names10$lat[names10$id==133.03]<-names10$lat[names10$id==133.03]-.003
names10$long[names10$id==128]<-names10$long[names10$id==128]-.005
names10$lat[names10$id==128]<-names10$lat[names10$id==128]+.01
names10$lat[names10$id==130.02]<-names10$lat[names10$id==130.02]+.005
names10$lat[names10$id==130.03]<-names10$lat[names10$id==130.03]-.005
names10$long[names10$id==134.01]<-names10$long[names10$id==134.01]-.005
names10$lat[names10$id==134.01]<-names10$lat[names10$id==134.01]+.004
names10$long[names10$id==133.05]<-names10$long[names10$id==133.05]+.005
names10$lat[names10$id==126.13]<-names10$lat[names10$id==126.13]-.004
names10$long[names10$id==126.07]<-names10$long[names10$id==126.07]-.008
names10$lat[names10$id==126.07]<-names10$lat[names10$id==126.07]-.003
names10$long[names10$id==140.01]<-names10$long[names10$id==140.01]+.005
names10$lat[names10$id==140.01]<-names10$lat[names10$id==140.01]-.005
```

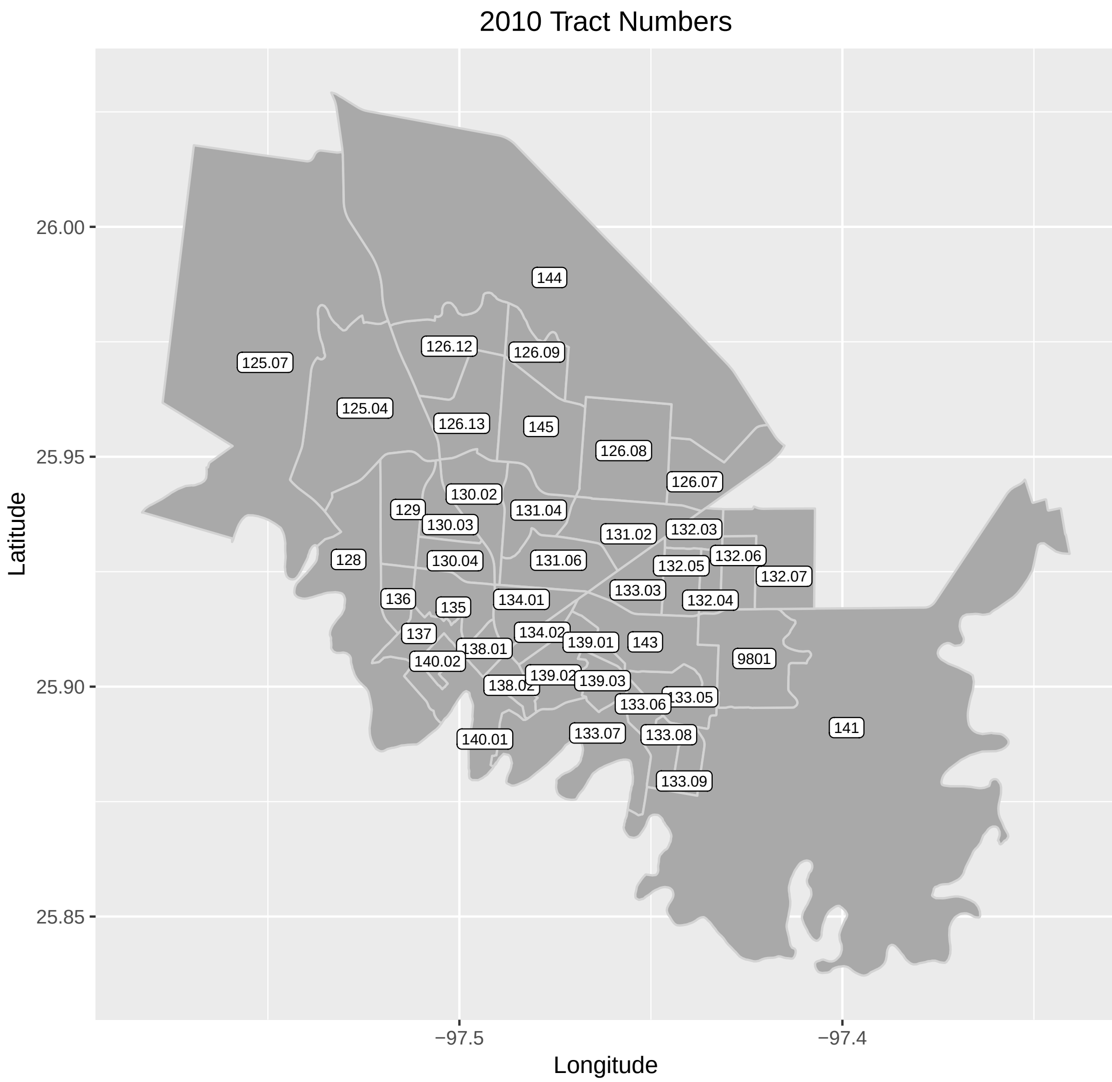
At this point, objects are ready to be mapping in `ggplot2` and label locations are stored and legible. Even without data, we can still plot the map and label them with the tract numbers. This was a useful reference throughout the analysis.

Mapping Tract Numbers

```
# 2000
ggplot()+
    geom_polygon(data=focusmap00,aes(long,lat,group=group)
                 ,color="light grey", fill="dark grey")+
     coord_fixed(1.2)+
     labs(x="Longitude",y="Latitude",title="2000 Tract Numbers")+
     theme(plot.title = element_text(hjust = 0.5))+
     geom_label(data=names00,aes(long,lat,label=id),size=2.5,
                label.padding = unit(.15,"lines"))
```



```r
# 2010
ggplot()+
    geom_polygon(data=focusmap10,aes(long,lat,group=group),
                 color="light grey", fill="dark grey")+
    coord_fixed(1.2)+
    labs(x="Longitude",y="Latitude",title="2010 Tract Numbers")+
    theme(plot.title = element_text(hjust = 0.5))+
    geom_label(data=names10,aes(long,lat,label=id),size=2.5,
               label.padding = unit(.15,"lines"))
```



Census Data Processing

Data was pulled from the US Census Beurau website for use in the weighting procedure. The census numbers are, of course, estimates in and of themselves. They are also based on a sample design and may not account for undocumented individuals who may be particularly prevalent so close to the US/Mexico border.

```
#### Import census data ####

# Source: https://factfinder.census.gov/faces/nav/jsf/pages/index.xhtml
# Form: 2010 100% Data Short Form 1 Sex by Age (Hispanic or Latino) (DEC 10 SF1 P12H)
# Geography: Census Tracts in Cameron County, Texas
brownsville2010virgin<-read_csv("DEC_10_SF1_P12H_with_ann.csv", col_names=TRUE,skip=1)

# Source: https://factfinder.census.gov/faces/nav/jsf/pages/index.xhtml
# Form: 2000 100% Data Short Form 1 Sex by Age (Hispanic or Latino) (DEC 00 SF1 P012H)
# Geography: Census Tracts in Cameron County, Texas
brownsville2000virgin<-read_csv("DEC_00_SF1_P012H_with_ann.csv", col_names=TRUE,skip=1)
```

The Census Beurau uses much narrower age categories than we needed. Weighting for this analysis was based on two genders and three age groups (18-34,35-64,65+) for a total of six age-gender strata.The following function takes the census data and restructures it to provide populations for our strata of interest.

```
#### Function ####
# Input: Read in variations of census SF1 P12 (Sex by Age)
# Output: Data frame with population by stratum per tract
P12cleaning<-function(dataset){
  # Restructuring geography column
  geonum<-unique(map_dbl(.x=dataset$Geography,.f=~str_count(.x,",")+1))
  dataset<-dataset%>%
    mutate(Geography=str_remove_all(Geography, " "),
           Geography=str_remove_all(Geography,"[a-zA-Z]"),
           Geography=str_remove_all(Geography,"[[:punct:]-[.]]"))%>%
    # Gives character number coumn names from 1 to number of geographies specified
    separate(Geography,into=as.character(seq(from=1,to=geonum,by=1)),sep=",")%>%
    # Removes id and total columns
    select(-Id,-Id2,-`Total:`,-`Female:`,-`Male:`)%>%
    # Removes extraneous former-geography columns
    select(tract=1,last_col(offset=0:45))
  # Creating legible column names
  vars<-list()
  for(x in 1:length(colnames(dataset))){
    if(str_detect(colnames(dataset[x]),"[[:punct:]]"))
    {newname<-str_remove(colnames(dataset[x]),"[[:punct:]]")}
    else{newname<-colnames(dataset[x])}
    vars[x]<-newname
    if(str_detect(vars[x],"^Male - "))
    {newname<-str_replace(vars[x],"Male - ","m_")}
    else{newname<-vars[x]}
    vars[x]<-newname
    if(str_detect(vars[x],"^Female - "))
    {newname<-str_replace(vars[x],"Female - ","f_")}
    else{newname<-vars[x]}
    vars[x]<-newname
```

```r
    if(str_detect(vars[x]," to "))
    {newname<-str_replace(vars[x]," to ","_")}
    else{newname<-vars[x]}
    vars[x]<-newname
    if(str_detect(vars[x]," and "))
    {newname<-str_replace(vars[x]," and ","_")}
    else{newname<-vars[x]}
    vars[x]<-newname
    if(str_detect(vars[x]," years"))
    {newname<-str_remove(vars[x]," years")}
    else{newname<-vars[x]}
    vars[x]<-newname
    if(str_detect(vars[x],"Under "))
    {newname<-str_replace(vars[x],"Under ","0_")}
    else{newname<-vars[x]}
    vars[x]<-newname}
  colnames(dataset)<-vars
  # Stratify
  dataset<-dataset%>%
    group_by(tract)%>%
    summarize(m_18_34=sum(m_18_19,m_20,m_21,m_22_24,m_25_29,m_30_34),
              f_18_34=sum(f_18_19,f_20,f_21,f_22_24,f_25_29,f_30_34),
              m_35_64=sum(m_35_39,m_40_44,m_45_49,m_50_54,m_55_59,m_60_61,m_62_64),
              f_35_64=sum(f_35_39,f_40_44,m_45_49,f_50_54,f_55_59,f_60_61,f_62_64),
              m_65_over=sum(m_65_66,m_67_69,m_70_74,m_75_79,m_80_84,m_85_over),
              f_65_over=sum(f_65_66,f_67_69,f_70_74,f_75_79,f_80_84,f_85_over))
  return(dataset)
}

#### Manipulate Format ####
bvl2000<-P12cleaning(brownsville2000virgin)
# The trailing zero causes matching problems when converted to numeric
bvl2000$tract[bvl2000$tract=="126.10"]<-126.1
# Filter to only include data on the tracts we are interested in
bvl2000<-bvl2000%>%
  filter(tract %in% focus00)

bvl2010<-P12cleaning(brownsville2010virgin)
bvl2010<-bvl2010%>%
  filter(tract %in% focus10)

#### Illustration of output ####
head(bvl2000)
```

```
## # A tibble: 6 x 7
##   tract  m_18_34 f_18_34 m_35_64 f_35_64 m_65_over f_65_over
##   <chr>    <dbl>   <dbl>   <dbl>   <dbl>     <dbl>     <dbl>
## 1 125.04     500     619     608     668        71       114
## 2 125.07     512     560     551     658       130       129
## 3 126.04     109     127     132     154        34        30
## 4 126.05      89     107     131     159        29        28
## 5 126.06     170     201     206     232        30        32
## 6 126.07     305     369     200     255        36        45
```

Simulating Data

```
set.seed(09062019)
#### Multiple Visit Data ####
datamult<-data.frame(
  id=sample(0:5000,10000,replace=T),
  visitdate=sample(seq(as.Date("2004-01-01"),as.Date("2018-12-31"),by="day"),
                   10000,replace=T),
  age=sample(18:100,10000,replace=T),
  insure=sample(0:1,10000,replace=T),
  employ=sample(0:1,10000,replace=T),
  diabetes=sample(0:1,10000,replace=T))

datamult$time<-map_chr(.x=datamult$visitdate,
                       .f=~if(.x>as.Date("2010-01-01"))
                         {return("post")}else{return("pre")})
datamult$id2<-paste0(datamult$id,datamult$time)

demos <- data.frame(id=unique(datamult$id),
                    gender=sample(c("male","female"),
                                  length(unique(datamult$id)),replace=T),
                    tract=sample(focus00,length(unique(datamult$id)),replace=T),
                    edu=sample(c("Less than High School","High School / GED",
                                 "Post-Secondary School"),
                               length(unique(datamult$id)),replace=T))

datamult <- datamult%>%
  left_join(demos,by="id")%>%
  select(id,id2,time,visitdate,tract,gender,age,edu,insure,employ,diabetes)

#### Single ID Data ####
data <- datamult%>%
  group_by(id2,time,gender,edu,tract)%>%
  summarize(age=mean(age),
            diabetes=map_dbl(.x=mean(diabetes),
                             .f=~ifelse(.x==0, 0,1)),
            insure=map_chr(.x=mean(insure),
                           .f=~ifelse(.x==1, "With Insurance",
                                      ifelse(.x==0, "No Insurance", "Mixed Coverage"))),
            employ=map_chr(.x=mean(employ),
                           .f=~ifelse(.x==1, "Employed",
                                      ifelse(.x==0, "Unemployed", "Mixed Status"))))%>%
  ungroup()%>%
  mutate(gender=map_chr(.x=gender,~ifelse(.x=="female","f_","m_")),
         age=map_chr(.x=age,~ifelse(.x>=18 & .x<35, "18_34",
                                    ifelse(.x>=35 & .x<65, "35_64",
                                           ifelse(.x>=65, "65_over", NA)))),
         stratum=paste0(gender,age))%>%
  select(id2,time,tract,stratum,employ,insure,edu,diabetes)

head(data)
```

```
## # A tibble: 6 x 8
```

```
##   id2     time  tract  stratum  employ    insure     edu           diabetes
##   <chr>   <chr> <fct>  <chr>    <chr>     <chr>      <fct>            <dbl>
## 1 1000po~ post  126.07 f_35_64  Mixed St~ Mixed Cov~ Less than Hi~        1
## 2 1000pre pre   126.07 f_35_64  Unemploy~ With Insu~ Less than Hi~        0
## 3 1001po~ post  140.02 f_35_64  Mixed St~ Mixed Cov~ Less than Hi~        1
## 4 1001pre pre   140.02 f_65_ov~ Mixed St~ No Insura~ Less than Hi~        0
## 5 1002po~ post  126.04 f_65_ov~ Employed  With Insu~ High School ~        1
## 6 1002pre pre   126.04 f_65_ov~ Mixed St~ No Insura~ High School ~        1
```

Data Cleaning

Tract Conversion

Data was collected between 2004 and 2018. As seen before with the maps, Some tract boundaries were redrawn for the 2010 census. However, all observations were recorded with their 2000 census tract. The following code converts to the appropriate tract. Conversions were calculated by Yunyun Jiang, at the time a graduate student at the University of Texas School of Public Health. Be aware that this is a simplified version of the true conversion, which utalized exact addresses to ensure proper tract selection.

```
data <- data%>%
  mutate(tract=ifelse(time=="pre",tract,
                    ifelse(tract==126.04,144,
                           ifelse(tract==126.06,144,
                                  ifelse(tract==126.1,145,
                                         ifelse(tract==126.11,145,
                                                ifelse(tract==132.08,141,
                                                       ifelse(tract==133.04,143,
                                                              tract))))))))
```

Constant information per ID

```
#### Function ####
# Input: Dataset and the grouping variable and the information variable
# Output: TRUE if information variable is constant for each level of grouping variable,
#         FALSE if if information variable differs for each level of grouping variable
constant <- function(data, variable, grouping){
  grouping <- enquo(grouping)
  variable <- enquo(variable)
  tab <- data%>%
    group_by(!! grouping)%>%
    summarize(identical=n_distinct(!! variable))%>%
    filter(identical>1)
  if (nrow(tab)>0){return(FALSE)}
  else {return(TRUE)}
}

#### Usage ####
constant(data, tract, id2)
```

```
## [1] TRUE
```

```
constant(data, stratum, id2)
```

```
## [1] TRUE
```

```
constant(data, employ,id2)
```

```
## [1] TRUE
```

```r
constant(data,insure,id2)
```

```
## [1] TRUE
```

```r
constant(data,diabetes, id2)
```

```
## [1] TRUE
```

These all return true because this data was created clean. `x` demonstrates the function returning FALSE.

```r
x <- data.frame(col1=c(1,1,1,1),
                col2=c(1,2,3,4),
                col3=c(1,1,2,2))

constant(x, col3, col1)
```

```
## [1] FALSE
```

In this case the function returns FALSE because for each value in `col1` (1), there are multiple values of `col3` (1 and 2).

Follow Up

Follow up has to be calcuated from the `multdata` data set because `data` drops the individual visit dates.

```
followup <- datamult%>%
  #### Overall follow up ####
  group_by(id)%>%
  summarize(visits=n_distinct(visitdate),
            first=min(visitdate),
            last=max(visitdate))%>%
  mutate(followup=time_length(interval(start=first,end=last,tzone="Etc/GMT-5"),
                              unit="months"),
         followup2=as.period(interval(start=first,end=last,tzone="Etc/GMT-5"),
                             unit="months"))%>%
  summarize(no.part=n(), # total subjects
            mean.visit=mean(visits),
            sd.visit=sd(visits),
            min.visit=min(visits),
            max.visit=max(visits),
            mean.follow=mean(followup),
            sd.follow=sd(followup),
            max.follow=max(followup),
            min.follow=min(followup))%>%
  bind_cols(datamult%>%summarize(tot.visit=n()))%>% # total visits
  #### follow up by time frame ####
  bind_rows(datamult%>%
              group_by(id2)%>%
              summarize(visits=max(n_distinct(visitdate)), # total subjects
                        first=min(visitdate),
                        last=max(visitdate))%>%
              mutate(followup=time_length(interval(start=first,end=last,tzone="Etc/GMT-5"),
                                          unit="months"),
                     followup2=as.period(interval(start=first,end=last,tzone="Etc/GMT-5"),
                                         unit="months"),
                     time=str_extract(id2,".{3}$"))%>%
              group_by(time)%>%
              summarize(no.part=n(), # total visits
                        mean.visit=mean(visits),
                        sd.visit=sd(visits),
                        min.visit=min(visits),
                        max.visit=max(visits),
                        mean.follow=mean(followup),
                        sd.follow=sd(followup),
                        max.follow=max(followup),
                        min.follow=min(followup))%>%
              # total visits
              bind_cols(datamult%>%group_by(time)%>%summarize(tot.visit=n())))%>%
  mutate(var=c("Overall","2010-2018","2004-2009"),
         range.visit=paste0("[",min.visit,", ",max.visit,"]"),
         range.follow=paste0("[",round(min.follow,2),", ",
                             round(max.follow,2),"]"," (~",round(max.follow/12,1)," years)"),
         mean.follow=paste0(round(mean.follow,2)," (~",round(mean.follow/12,1)," years)"),
         sd.follow=paste0(round(sd.follow,2)," (~",round(sd.follow/12,1)," years)"),
         order=c(1,3,2))%>%
```

```r
  arrange(order)%>%
  select(var,no.part,tot.visit,mean.visit,sd.visit,range.visit,
         mean.follow,sd.follow,range.follow)

#### Formatting ####

# Adding footnote symbols
followup[2,1]<-paste(followup[2,1],footnote_marker_symbol(1))
followup[3,1]<-paste(followup[3,1],footnote_marker_symbol(1))

followup%>%
  mutate(no.part=prettyNum(no.part,big.mark=","),
         tot.visit=prettyNum(tot.visit,big.mark=","))%>%
  kable(booktabs=T,digits=2, caption="Summary of Follow-Up",
        escape=F, col.names=c(" ","No. Subjects","Tot. Visits","Ave. Visits","SD Visits",
                              "Range No. Visits","Ave Follow-up Months",
                              "SD Follow-up Months","Range Follow-up Months"))%>%
          kable_styling(latex_options=c("HOLD_position","scale_down"),position="center")%>%
  footnote(symbol="Determined from the visits that contributed to the estimates for the
           specified time period. Some individual participants have visits in, and thus are
           included in, both time periods.",
           threeparttable=T)%>%
  column_spec(2:6,width="1cm")%>%
  column_spec(7:8,width="3cm")%>%
  column_spec(9,width="4cm")%>%
  column_spec(1,width="2cm")%>%
  row_spec(0,align="c")
```

Table 1: Summary of Follow-Up

	No. Subjects	Tot. Visits	Ave. Visits	SD Visits	Range No. Visits	Ave Follow-up Months	SD Follow-up Months	Range Follow-up Months
Overall	4,299	10,000	2.33	1.27	[1, 8]	57.23 (4.8 years)	54.17 (4.5 years)	[0, 178.94] (14.9 years)
2004-2009 *	2,797	4,075	1.46	0.72	[1, 6]	9.75 (0.8 years)	17.22 (1.4 years)	[0, 71.55] (6 years)
2010-2018 *	3,476	5,925	1.70	0.90	[1, 7]	20.66 (1.7 years)	28.89 (2.4 years)	[0, 107.87] (9 years)

* makecell[l]Determined from the visits that contributed to the estimates for the specified time period. Some individual participants have visits in, and thus are included in, both time periods.

Table 1

A table 1 is included in the vast majority of pubic health papers. It describes select demographics of the study population.

```
#### Table ####
  #### Gender ####
  data%>%
    mutate(var=factor(str_extract(stratum,"^."),levels=c("f","m"),
                      label=c("Female","Male")))%>%
    group_by(var,time)%>%
    summarize(n=n(),prev=mean(diabetes,na.rm=T),cil=t.test(diabetes)$conf.int[1],
              ciu=t.test(diabetes)$conf.int[2])%>%
    mutate(ci=paste0("(",round(cil,3),", ",round(ciu,3),")"))%>%
    select(-cil,-ciu)%>%
    ungroup()%>%
    mutate(vals=map2_chr(.x=n,.y=prev,.f=~paste(.x,.y,sep=";",collapse=";")),
           vals=map2_chr(.x=vals,.y=ci,.f=~paste(.x,.y,sep=";",collapse=";")))%>%
    select(var,time,vals)%>%
    spread(time,vals,2:3)%>%
    select(var,pre,post)%>%
    separate(pre,into=c("n.pre","prev.pre","ci.pre"),sep=";")%>%
    separate(post,into=c("n.post","prev.post","ci.post"),sep=";")%>%
      mutate(var=as.character(var))%>%
  #### Age ####
    bind_rows(data%>%
      mutate(var=factor(str_remove(stratum,"^.."),levels=c("18_34","35_64","65_over"),
                      labels=c("18 to 34 years","35 to 64 years","65 years and over")))%>%
      group_by(var,time)%>%
      summarize(n=n(),prev=mean(diabetes,na.rm=T),cil=t.test(diabetes)$conf.int[1],
                ciu=t.test(diabetes)$conf.int[2])%>%
      mutate(ci=paste0("(",round(cil,3),", ",round(ciu,3),")"))%>%
      select(-cil,-ciu)%>%
      ungroup()%>%
      mutate(vals=map2_chr(.x=n,.y=prev,.f=~paste(.x,.y,sep=";",collapse=";")),
             vals=map2_chr(.x=vals,.y=ci,.f=~paste(.x,.y,sep=";",collapse=";")))%>%
      select(var,time,vals)%>%
      spread(time,vals,2:3)%>%
      select(var,pre,post)%>%
      separate(pre,into=c("n.pre","prev.pre","ci.pre"),sep=";")%>%
      separate(post,into=c("n.post","prev.post","ci.post"),sep=";")%>%
      mutate(var=as.character(var)))%>%
  #### Education ####
      bind_rows(data%>%
        mutate(var=factor(edu,
                          levels=c("Less than High School","High School / GED",
                                   "Post-Secondary School")))%>%
        group_by(var,time)%>%
        summarize(n=n(),prev=mean(diabetes,na.rm=T),
                  cil=t.test(diabetes,na.rm=T)$conf.int[1],
                  ciu=t.test(diabetes,na.rm=T)$conf.int[2])%>%
        mutate(ci=paste0("(",round(cil,3),", ",round(ciu,3),")"))%>%
        select(-cil,-ciu)%>%
        ungroup()%>%
```

```
        mutate(vals=map2_chr(.x=n,.y=prev,.f=~paste(.x,.y,sep=";",collapse=";")),
               vals=map2_chr(.x=vals,.y=ci,.f=~paste(.x,.y,sep=";",collapse=";")))%>%
        select(var,time,vals)%>%
        spread(time,vals,2:3)%>%
        select(var,pre,post)%>%
        separate(pre,into=c("n.pre","prev.pre","ci.pre"),sep=";")%>%
        separate(post,into=c("n.post","prev.post","ci.post"),sep=";")%>%
        mutate(var=as.character(var)))%>%
  #### Employment ####
      bind_rows(data%>%
        mutate(var=factor(employ,
                          levels=c("Employed","Unemployed","Mixed Status")))%>%
        filter(!is.na(var))%>%
        group_by(var,time)%>%
        summarize(n=n(),prev=mean(diabetes,na.rm=T),
                  cil=t.test(diabetes,na.rm=T)$conf.int[1],
                  ciu=t.test(diabetes,na.rm=T)$conf.int[2])%>%
        mutate(ci=paste0("(",round(cil,3),", ",round(ciu,3),")"))%>%
        select(-cil,-ciu)%>%
        ungroup()%>%
        mutate(vals=map2_chr(.x=n,.y=prev,.f=~paste(.x,.y,sep=";",collapse=";")),
               vals=map2_chr(.x=vals,.y=ci,.f=~paste(.x,.y,sep=";",collapse=";")))%>%
        select(var,time,vals)%>%
        spread(time,vals,2:3)%>%
        select(var,pre,post)%>%
        separate(pre,into=c("n.pre","prev.pre","ci.pre"),sep=";")%>%
        separate(post,into=c("n.post","prev.post","ci.post"),sep=";")%>%
        mutate(var=as.character(var)))%>%
  #### Insurance ####
        bind_rows(data%>%
          rename(var=insure)%>%
          filter(!is.na(var))%>%
          mutate(var=factor(var,
                            levels=c("With Insurance","No Insurance",
                                     "Mixed Coverage")))%>%
          group_by(var,time)%>%
          summarize(n=n(),prev=mean(diabetes,na.rm=T),
                    cil=t.test(diabetes,na.rm=T)$conf.int[1],
                    ciu=t.test(diabetes,na.rm=T)$conf.int[2])%>%
          mutate(ci=paste0("(",round(cil,3),", ",round(ciu,3),")"))%>%
          select(-cil,-ciu)%>%
          ungroup()%>%
          mutate(vals=map2_chr(.x=n,.y=prev,.f=~paste(.x,.y,sep=";",collapse=";")),
                 vals=map2_chr(.x=vals,.y=ci,.f=~paste(.x,.y,sep=";",collapse=";")))%>%
          select(var,time,vals)%>%
          spread(key=time,value=vals,2:3)%>%
          separate(pre,into=c("n.pre","prev.pre","ci.pre"),sep=";")%>%
          separate(post,into=c("n.post","prev.post","ci.post"),sep=";")%>%
          mutate(var=as.character(var)))%>%
    mutate(prev.pre=as.numeric(prev.pre),prev.post=as.numeric(prev.post))%>%
#### Kable styling ####
    kable(booktabs=T,digits=3,caption="Select Demographics of Study Population",
          col.names=c("",rep(c("n","Crude Prev.","95% C.I."),2)))%>%
```

```
kable_styling(latex_options="HOLD_position",position="center")%>%
add_header_above(c(" "=1, "2004-2009"=3, "2010-2018"=3))%>%
pack_rows(index=c("Gender"=2, "Age"=3, "Education"=3, "Employment"=3,
                  "Insurance Coverage"=3))%>%
column_spec(c(4,7), width="2.5cm")%>%
column_spec(c(3,6), width="1cm")%>%
column_spec(5,border_left=T)%>%
row_spec(0, align="c")%>%
footnote(general="Crude Prevalence estimate calculated by cases /
         observations for participants of the indicated description.",
         threeparttable=T)
```

Table 2: Select Demographics of Study Population

	2004-2009			2010-2018		
	n	Crude Prev.	95% C.I.	n	Crude Prev.	95% C.I.
Gender						
Female	1417	0.614	(0.589, 0.639)	1788	0.636	(0.614, 0.659)
Male	1380	0.591	(0.565, 0.617)	1688	0.643	(0.62, 0.666)
Age						
18 to 34 years	452	0.538	(0.491, 0.584)	502	0.556	(0.512, 0.599)
35 to 64 years	1204	0.638	(0.611, 0.665)	1610	0.676	(0.654, 0.699)
65 years and over	1141	0.592	(0.563, 0.62)	1364	0.627	(0.601, 0.653)
Education						
Less than High School	951	0.596	(0.565, 0.627)	1182	0.636	(0.609, 0.664)
High School / GED	881	0.602	(0.569, 0.634)	1139	0.634	(0.606, 0.662)
Post-Secondary School	965	0.610	(0.58, 0.641)	1155	0.648	(0.621, 0.676)
Employment						
Employed	1112	0.555	(0.526, 0.584)	1268	0.565	(0.538, 0.593)
Unemployed	1127	0.555	(0.526, 0.585)	1212	0.569	(0.541, 0.597)
Mixed Status	558	0.794	(0.76, 0.828)	996	0.819	(0.795, 0.843)
Insurance Coverage						
With Insurance	1132	0.564	(0.535, 0.593)	1262	0.572	(0.545, 0.599)
No Insurance	1115	0.550	(0.521, 0.579)	1231	0.574	(0.547, 0.602)
Mixed Coverage	550	0.791	(0.757, 0.825)	983	0.808	(0.783, 0.832)

Note:
makecell[l]Crude Prevalence estimate calculated by cases / observations for participants of the indicated description.

Weighting

Modelling

Mapping