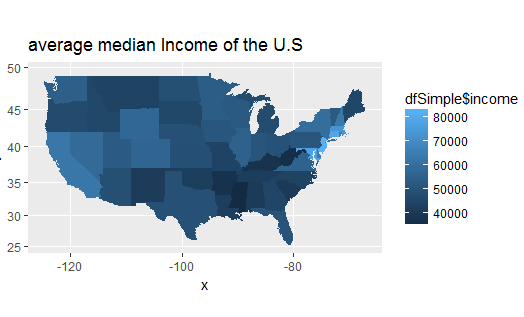
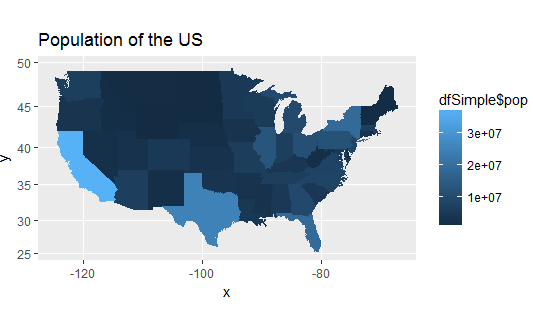
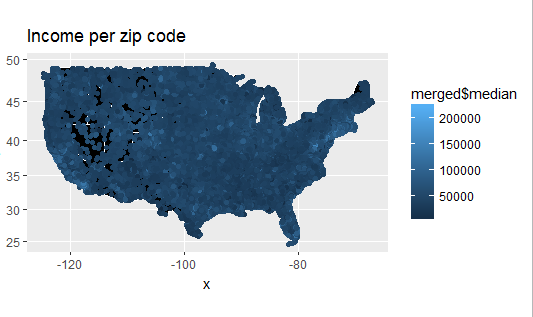
Jacob Dineen

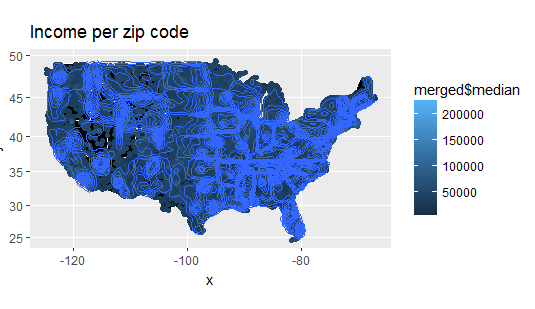
**#HW 7**

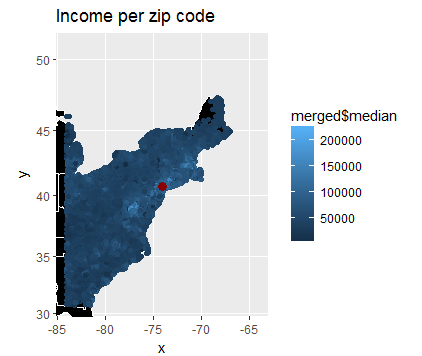
**Map Outputs:**

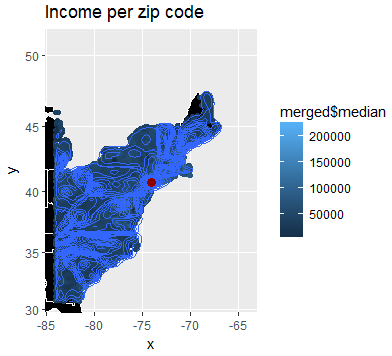












**Unexecuted:**

#ADS IST 687

#Jacob Dineen

#Homework 7

#Due 9/3/2017

#CLEAR ENVIRONMENT AND INSTALL INITIAL PACKAGES

rm(list = ls(all = TRUE))#Clear Enviroment

#specify the packages of interest

packages=c("maps","zipcode","mapproj","ggmap","ggplot2","gdata")

#use this function to check if each package is on the local machine if a package is installed, it will be loaded if any are not, the missing package(s) will be installed and loaded

package.check <- lapply(packages, FUN = function(x) {

if (!require(x, character.only = TRUE)) {

install.packages(x, dependencies = TRUE)

library(x, character.only = TRUE)

}

})

#Perl Directory for GDATA read.xls

installXLSXsupport(perl = 'C:/Strawberry/perl/bin/perl.exe')

perl <- "C:/Strawberry/perl/bin/perl.exe"

#######################################################STEP 1 Load the Data##################################################################

#Save As DF

medianincome <- read.xls(file.choose()) #File Choose let's me manually choose from taskviewer. 32635 Obs, 4 Variables

str(medianincome)

#Change column names

colnames(medianincome) <- c("zip", "median", "mean", "population")

#Remove row 1 with dupe column Names

medianincome <- medianincome [-1,]

#remove all commas

medianincome$median <- gsub(",", "", medianincome$median)

medianincome$mean <- gsub(",", "", medianincome$mean)

medianincome$population <- gsub(",","",medianincome$population)

#Put zips into standard format- Add a zero to the front of each int string

medianincome$zip <- clean.zipcodes(medianincome$zip)

str(medianincome)

#store zipcode package as a new dataframe

zipcodes <- data(zipcode) #saved as zipcode

merged <- merge(medianincome, zipcode, by="zip")

str(merged)

#Remove HW and Al

merged <- merged[!grepl("HI", merged$state),]

merged <- merged[!grepl("AK", merged$state),]

#Change medianincome and population to numeric

merged$median <- as.numeric(merged$median)

merged$population <- as.numeric(merged$population)

str(merged)

########################################################STEP 2 Show the Income and Population per State##################################################################

income <- tapply(merged$median, merged$state, mean) # calc mean of median by state

str(income)

state <- rownames(income) # place rownames from income into state variable

medianIncome <- data.frame(state, income) # create a df with state variable & income variable

str(medianIncome)

head(medianIncome)

pop <- tapply(merged$population, merged$state, sum ) # sum up population for each state

str(pop)

head(pop)

state <- rownames(pop)

statePop <- data.frame(state, pop)

dfSimple <- merge(medianIncome, statePop, by="state") # create new df by merging df's medianIncome, stateIncome

str(dfSimple)

head(dfSimple)

#Match statenames per state abbreviations

str(state.abb)

head(state.abb)

match(dfSimple$state,state.abb) # the relative position of state.abb in dfSimple$state

dfSimple$state

dfSimple$stateName <- state.name[match(dfSimple$state,state.abb)]

str(dfSimple)

head(dfSimple)

#State names to lowercase

dfSimple$stateName <- tolower(dfSimple$stateName)

head(dfSimple)

#Show the US map with avg median income

us <- map\_data("state") # performed above, not adding anything new

mapIncome <- ggplot(dfSimple, aes(map\_id = stateName))

mapIncome <- mapIncome + geom\_map(map = us, aes(fill = dfSimple$income))

mapIncome <- mapIncome + expand\_limits(x = us$long, y = us$lat)

mapIncome <- mapIncome + coord\_map()

mapIncome <- mapIncome + ggtitle("average median Income of the U.S")

mapIncome

#Show the US map with color representing state pop

us <- map\_data("state") # performed above, not adding anything new

mapPop <- ggplot(dfSimple, aes(map\_id = stateName))

mapPop <- mapPop + geom\_map(map = us, aes(fill = dfSimple$pop))

mapPop <- mapPop + expand\_limits(x = us$long, y = us$lat)

mapPop <- mapPop + coord\_map()

mapPop <- mapPop + ggtitle("Population of the US")

mapPop

##############################################STEP 3 Show the Income per Zip Code##################################################################

head(merged)

merged$stateName <- state.name[match(merged$state,state.abb)]

merged$stateName <- tolower(merged$stateName)

head(merged)

mapZip <- ggplot(merged, aes(map\_id = stateName))

mapZip <- mapZip + geom\_map(map=us, fill="black", color="white")

mapZip <- mapZip + expand\_limits(x =us$long, y = us$lat)

mapZip <- mapZip + geom\_point(data = merged, aes(x = merged$longitude, y = merged$latitude, color=merged$median))

mapZip <- mapZip + coord\_map() + ggtitle("Income per zip code")

mapZip

##############################################STEP 4 Show Zipcode density##################################################################

# remove al & HI from dfNew$state - Did this above, but doing it again here

merged$state<-merged[merged$state!="AK" & merged$state!="HI",]

str(merged)

mapZip <- ggplot(merged, aes(map\_id = stateName))

mapZip <- mapZip + geom\_map(map=us, fill="black", color="white")

mapZip <- mapZip + expand\_limits(x =us$long, y = us$lat)

mapZip <- mapZip + geom\_point(data = merged,aes(x = merged$longitude, y = merged$latitude, color=merged$median))

mapZip <- mapZip + coord\_map() + ggtitle("Income per zip code")

mapD <- mapZip + geom\_density\_2d(data = merged, aes(x = merged$longitude, y = merged$latitude))

mapD

##############################################STEP 5 Zoom in to the region around NYC##################################################################

latlon <- geocode("NYC, ny")

mapZipZoomed <- mapZip + geom\_point(aes(x = latlon$lon, y = latlon$lat), color="darkred", size = 3)

mapZipZoomed <- mapZipZoomed + xlim(latlon$lon-10, latlon$lon+10) + ylim(latlon$lat-10,latlon$lat+10) + coord\_map()

mapZipZoomed

mapDZoomed <- mapD + geom\_point(aes(x = latlon$lon, y = latlon$lat), color="darkred", size = 3)

mapDZoomed <- mapDZoomed + xlim(latlon$lon-10, latlon$lon+10) + ylim(latlon$lat-10,latlon$lat+10) + coord\_map()

mapDZoomed

**EXECUTED:**

#ADS IST 687

> #Jacob Dineen

> #Homework 7

> #Due 9/3/2017

>

>

> #CLEAR ENVIRONMENT AND INSTALL INITIAL PACKAGES

> rm(list = ls(all = TRUE))#Clear Enviroment

>

>

> #specify the packages of interest

> packages=c("maps","zipcode","mapproj","ggmap","ggplot2","gdata")

>

> #use this function to check if each package is on the local machine if a package is installed, it will be loaded if any are not, the missing package(s) will be installed and loaded

> package.check <- lapply(packages, FUN = function(x) {

+ if (!require(x, character.only = TRUE)) {

+ install.packages(x, dependencies = TRUE)

+ library(x, character.only = TRUE)

+ }

+ })

Loading required package: maps

Loading required package: zipcode

Loading required package: mapproj

Loading required package: ggmap

Loading required package: ggplot2

Google Maps API Terms of Service: http://developers.google.com/maps/terms.

Please cite ggmap if you use it: see citation('ggmap') for details.

Loading required package: gdata

gdata: read.xls support for 'XLS' (Excel 97-2004) files ENABLED.

gdata: read.xls support for 'XLSX' (Excel 2007+) files ENABLED.

Attaching package: ‘gdata’

The following object is masked from ‘package:stats’:

nobs

The following object is masked from ‘package:utils’:

object.size

The following object is masked from ‘package:base’:

startsWith

Warning messages:

1: package ‘maps’ was built under R version 3.3.3

2: package ‘mapproj’ was built under R version 3.3.3

3: package ‘ggmap’ was built under R version 3.3.3

4: package ‘ggplot2’ was built under R version 3.3.3

5: package ‘gdata’ was built under R version 3.3.3

>

>

> #Perl Directory for GDATA read.xls

> installXLSXsupport(perl = 'C:/Strawberry/perl/bin/perl.exe')

Perl XLSX support libraries successfully installed.

> perl <- "C:/Strawberry/perl/bin/perl.exe"

>

>

> #######################################################STEP 1 Load the Data##################################################################

>

> #Save As DF

> medianincome <- read.xls(file.choose()) #File Choose let's me manually choose from taskviewer. 32635 Obs, 4 Variables

> str(medianincome)

'data.frame': 32635 obs. of 4 variables:

$ Data.from..http...www.psc.isr.umich.edu.dis.census.Features.tract2zip.: Factor w/ 32635 levels "10001","10002",..: 32635 9 19 30 43 46 48 49 50 51 ...

$ X : Factor w/ 23074 levels "1,018","1,038",..: 23074 16322 13238 2456 20891 21422 18630 14068 20928 18496 ...

$ X.1 : Factor w/ 23568 levels ".","1,022","1,104",..: 23568 17442 19971 3303 21414 21943 20690 17459 21166 20513 ...

$ X.2 : Factor w/ 15173 levels "1","1,000","1,001",..: 15173 3199 7094 14260 10455 2097 245 14597 7785 324 ...

>

> #Change column names

> colnames(medianincome) <- c("zip", "median", "mean", "population")

>

> #Remove row 1 with dupe column Names

> medianincome <- medianincome [-1,]

>

> #remove all commas

> medianincome$median <- gsub(",", "", medianincome$median)

> medianincome$mean <- gsub(",", "", medianincome$mean)

> medianincome$population <- gsub(",","",medianincome$population)

>

>

> #Put zips into standard format- Add a zero to the front of each int string

> medianincome$zip <- clean.zipcodes(medianincome$zip)

> str(medianincome)

'data.frame': 32634 obs. of 4 variables:

$ zip : chr "01001" "01002" "01003" "01005" ...

$ median : chr "56663" "49853" "28462" "75423" ...

$ mean : chr "66688" "75063" "35121" "82442" ...

$ population: chr "16445" "28069" "8491" "4798" ...

>

>

> #store zipcode package as a new dataframe

> zipcodes <- data(zipcode) #saved as zipcode

>

>

> merged <- merge(medianincome, zipcode, by="zip")

> str(merged)

'data.frame': 32634 obs. of 8 variables:

$ zip : chr "01001" "01002" "01003" "01005" ...

$ median : chr "56663" "49853" "28462" "75423" ...

$ mean : chr "66688" "75063" "35121" "82442" ...

$ population: chr "16445" "28069" "8491" "4798" ...

$ city : chr "Agawam" "Amherst" "Amherst" "Barre" ...

$ state : chr "MA" "MA" "MA" "MA" ...

$ latitude : num 42.1 42.4 42.4 42.4 42.3 ...

$ longitude : num -72.6 -72.5 -72.6 -72.1 -72.4 ...

>

> #Remove HW and Al

> merged <- merged[!grepl("HI", merged$state),]

> merged <- merged[!grepl("AK", merged$state),]

>

>

>

> #Change medianincome and population to numeric

> merged$median <- as.numeric(merged$median)

> merged$population <- as.numeric(merged$population)

> str(merged)

'data.frame': 32321 obs. of 8 variables:

$ zip : chr "01001" "01002" "01003" "01005" ...

$ median : num 56663 49853 28462 75423 79076 ...

$ mean : chr "66688" "75063" "35121" "82442" ...

$ population: num 16445 28069 8491 4798 12962 ...

$ city : chr "Agawam" "Amherst" "Amherst" "Barre" ...

$ state : chr "MA" "MA" "MA" "MA" ...

$ latitude : num 42.1 42.4 42.4 42.4 42.3 ...

$ longitude : num -72.6 -72.5 -72.6 -72.1 -72.4 ...

>

>

> ########################################################STEP 2 Show the Income and Population per State##################################################################

>

> income <- tapply(merged$median, merged$state, mean) # calc mean of median by state

> str(income)

num [1:49(1d)] 40550 36961 48132 62629 56303 ...

- attr(\*, "dimnames")=List of 1

..$ : chr [1:49] "AL" "AR" "AZ" "CA" ...

>

> state <- rownames(income) # place rownames from income into state variable

>

> medianIncome <- data.frame(state, income) # create a df with state variable & income variable

> str(medianIncome)

'data.frame': 49 obs. of 2 variables:

$ state : Factor w/ 49 levels "AL","AR","AZ",..: 1 2 3 4 5 6 7 8 9 10 ...

$ income: num [1:49(1d)] 40550 36961 48132 62629 56303 ...

..- attr(\*, "dimnames")=List of 1

.. ..$ : chr "AL" "AR" "AZ" "CA" ...

> head(medianIncome)

state income

AL AL 40549.90

AR AR 36960.95

AZ AZ 48132.07

CA CA 62628.72

CO CO 56303.02

CT CT 78520.16

>

> pop <- tapply(merged$population, merged$state, sum ) # sum up population for each state

> str(pop)

num [1:49(1d)] 4770242 2936699 6360679 36927999 4979279 ...

- attr(\*, "dimnames")=List of 1

..$ : chr [1:49] "AL" "AR" "AZ" "CA" ...

> head(pop)

AL AR AZ CA CO CT

4770242 2936699 6360679 36927999 4979279 3548308

> state <- rownames(pop)

> statePop <- data.frame(state, pop)

>

> dfSimple <- merge(medianIncome, statePop, by="state") # create new df by merging df's medianIncome, stateIncome

> str(dfSimple)

'data.frame': 49 obs. of 3 variables:

$ state : Factor w/ 49 levels "AL","AR","AZ",..: 1 2 3 4 5 6 7 8 9 10 ...

$ income: num [1:49(1d)] 40550 36961 48132 62629 56303 ...

..- attr(\*, "dimnames")=List of 1

.. ..$ : chr "AL" "AR" "AZ" "CA" ...

$ pop : num [1:49(1d)] 4770242 2936699 6360679 36927999 4979279 ...

..- attr(\*, "dimnames")=List of 1

.. ..$ : chr "AL" "AR" "AZ" "CA" ...

> head(dfSimple)

state income pop

1 AL 40549.90 4770242

2 AR 36960.95 2936699

3 AZ 48132.07 6360679

4 CA 62628.72 36927999

5 CO 56303.02 4979279

6 CT 78520.16 3548308

>

>

> #Match statenames per state abbreviations

> str(state.abb)

chr [1:50] "AL" "AK" "AZ" "AR" "CA" "CO" "CT" "DE" "FL" "GA" "HI" "ID" "IL" "IN" "IA" "KS" "KY" "LA" "ME" "MD" ...

> head(state.abb)

[1] "AL" "AK" "AZ" "AR" "CA" "CO"

> match(dfSimple$state,state.abb) # the relative position of state.abb in dfSimple$state

[1] 1 4 3 5 6 7 NA 8 9 10 15 12 13 14 16 17 18 21 20 19 22 23 25 24 26 33 34 27 29 30 31 28 32 35 36 37 38 39 40 41 42 43 44 46 45 47 49 48 50

> dfSimple$state

[1] AL AR AZ CA CO CT DC DE FL GA IA ID IL IN KS KY LA MA MD ME MI MN MO MS MT NC ND NE NH NJ NM NV NY OH OK OR PA RI SC SD TN TX UT VA VT WA WI WV WY

49 Levels: AL AR AZ CA CO CT DC DE FL GA IA ID IL IN KS KY LA MA MD ME MI MN MO MS MT NC ND NE NH NJ NM NV NY OH OK OR PA RI SC SD TN TX UT VA VT WA WI ... WY

> dfSimple$stateName <- state.name[match(dfSimple$state,state.abb)]

> str(dfSimple)

'data.frame': 49 obs. of 4 variables:

$ state : Factor w/ 49 levels "AL","AR","AZ",..: 1 2 3 4 5 6 7 8 9 10 ...

$ income : num [1:49(1d)] 40550 36961 48132 62629 56303 ...

..- attr(\*, "dimnames")=List of 1

.. ..$ : chr "AL" "AR" "AZ" "CA" ...

$ pop : num [1:49(1d)] 4770242 2936699 6360679 36927999 4979279 ...

..- attr(\*, "dimnames")=List of 1

.. ..$ : chr "AL" "AR" "AZ" "CA" ...

$ stateName: chr "Alabama" "Arkansas" "Arizona" "California" ...

> head(dfSimple)

state income pop stateName

1 AL 40549.90 4770242 Alabama

2 AR 36960.95 2936699 Arkansas

3 AZ 48132.07 6360679 Arizona

4 CA 62628.72 36927999 California

5 CO 56303.02 4979279 Colorado

6 CT 78520.16 3548308 Connecticut

>

>

> #State names to lowercase

> dfSimple$stateName <- tolower(dfSimple$stateName)

> head(dfSimple)

state income pop stateName

1 AL 40549.90 4770242 alabama

2 AR 36960.95 2936699 arkansas

3 AZ 48132.07 6360679 arizona

4 CA 62628.72 36927999 california

5 CO 56303.02 4979279 colorado

6 CT 78520.16 3548308 connecticut

>

>

> #Show the US map with avg median income

> us <- map\_data("state") # performed above, not adding anything new

>

> mapIncome <- ggplot(dfSimple, aes(map\_id = stateName))

> mapIncome <- mapIncome + geom\_map(map = us, aes(fill = dfSimple$income))

> mapIncome <- mapIncome + expand\_limits(x = us$long, y = us$lat)

> mapIncome <- mapIncome + coord\_map()

> mapIncome <- mapIncome + ggtitle("average median Income of the U.S")

> mapIncome

>

> #Show the US map with color representing state pop

> us <- map\_data("state") # performed above, not adding anything new

>

> mapPop <- ggplot(dfSimple, aes(map\_id = stateName))

> mapPop <- mapPop + geom\_map(map = us, aes(fill = dfSimple$pop))

> mapPop <- mapPop + expand\_limits(x = us$long, y = us$lat)

> mapPop <- mapPop + coord\_map()

> mapPop <- mapPop + ggtitle("Population of the US")

> mapPop

>

>

>

> ##############################################STEP 3 Show the Income per Zip Code##################################################################

>

> head(merged)

zip median mean population city state latitude longitude

1 01001 56663 66688 16445 Agawam MA 42.07061 -72.62029

2 01002 49853 75063 28069 Amherst MA 42.37765 -72.50323

3 01003 28462 35121 8491 Amherst MA 42.36956 -72.63599

4 01005 75423 82442 4798 Barre MA 42.41209 -72.10443

5 01007 79076 85802 12962 Belchertown MA 42.27842 -72.41100

6 01008 63980 78391 1244 Blandford MA 42.17431 -72.94828

> merged$stateName <- state.name[match(merged$state,state.abb)]

> merged$stateName <- tolower(merged$stateName)

> head(merged)

zip median mean population city state latitude longitude stateName

1 01001 56663 66688 16445 Agawam MA 42.07061 -72.62029 massachusetts

2 01002 49853 75063 28069 Amherst MA 42.37765 -72.50323 massachusetts

3 01003 28462 35121 8491 Amherst MA 42.36956 -72.63599 massachusetts

4 01005 75423 82442 4798 Barre MA 42.41209 -72.10443 massachusetts

5 01007 79076 85802 12962 Belchertown MA 42.27842 -72.41100 massachusetts

6 01008 63980 78391 1244 Blandford MA 42.17431 -72.94828 massachusetts

>

>

>

> mapZip <- ggplot(merged, aes(map\_id = stateName))

> mapZip <- mapZip + geom\_map(map=us, fill="black", color="white")

> mapZip <- mapZip + expand\_limits(x =us$long, y = us$lat)

> mapZip <- mapZip + geom\_point(data = merged, aes(x = merged$longitude, y = merged$latitude, color=merged$median))

> mapZip <- mapZip + coord\_map() + ggtitle("Income per zip code")

> mapZip

>

>

> ##############################################STEP 4 Show Zipcode density##################################################################

>

> # remove al & HI from dfNew$state - Did this above, but doing it again here

> merged$state<-merged[merged$state!="AK" & merged$state!="HI",]

> str(merged)

'data.frame': 32321 obs. of 9 variables:

$ zip : chr "01001" "01002" "01003" "01005" ...

$ median : num 56663 49853 28462 75423 79076 ...

$ mean : chr "66688" "75063" "35121" "82442" ...

$ population: num 16445 28069 8491 4798 12962 ...

$ city : chr "Agawam" "Amherst" "Amherst" "Barre" ...

$ state :'data.frame': 32321 obs. of 9 variables:

..$ zip : chr "01001" "01002" "01003" "01005" ...

..$ median : num 56663 49853 28462 75423 79076 ...

..$ mean : chr "66688" "75063" "35121" "82442" ...

..$ population: num 16445 28069 8491 4798 12962 ...

..$ city : chr "Agawam" "Amherst" "Amherst" "Barre" ...

..$ state : chr "MA" "MA" "MA" "MA" ...

..$ latitude : num 42.1 42.4 42.4 42.4 42.3 ...

..$ longitude : num -72.6 -72.5 -72.6 -72.1 -72.4 ...

..$ stateName : chr "massachusetts" "massachusetts" "massachusetts" "massachusetts" ...

$ latitude : num 42.1 42.4 42.4 42.4 42.3 ...

$ longitude : num -72.6 -72.5 -72.6 -72.1 -72.4 ...

$ stateName : chr "massachusetts" "massachusetts" "massachusetts" "massachusetts" ...

>

>

> mapZip <- ggplot(merged, aes(map\_id = stateName))

> mapZip <- mapZip + geom\_map(map=us, fill="black", color="white")

> mapZip <- mapZip + expand\_limits(x =us$long, y = us$lat)

> mapZip <- mapZip + geom\_point(data = merged,aes(x = merged$longitude, y = merged$latitude, color=merged$median))

> mapZip <- mapZip + coord\_map() + ggtitle("Income per zip code")

> mapD <- mapZip + geom\_density\_2d(data = merged, aes(x = merged$longitude, y = merged$latitude))

> mapD

>

>

> ##############################################STEP 5 Zoom in to the region around NYC##################################################################

>

> latlon <- geocode("NYC, ny")

Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=NYC,%20ny&sensor=false

> mapZipZoomed <- mapZip + geom\_point(aes(x = latlon$lon, y = latlon$lat), color="darkred", size = 3)

> mapZipZoomed <- mapZipZoomed + xlim(latlon$lon-10, latlon$lon+10) + ylim(latlon$lat-10,latlon$lat+10) + coord\_map()

> mapZipZoomed

Warning message:

Removed 20925 rows containing missing values (geom\_point).

>

>

>

> mapDZoomed <- mapD + geom\_point(aes(x = latlon$lon, y = latlon$lat), color="darkred", size = 3)

> mapDZoomed <- mapDZoomed + xlim(latlon$lon-10, latlon$lon+10) + ylim(latlon$lat-10,latlon$lat+10) + coord\_map()

> mapDZoomed

Warning messages:

1: Removed 20925 rows containing non-finite values (stat\_density2d).

2: Removed 20925 rows containing missing values (geom\_point).