Data Wrangling for Coworking Space Market Feasibility Project

What data do we need?

We are attempting to put together a demographic sample of citizens of our market areas, South Coast Santa Barbara and pertinent Ventura County areas, in order to gauge if demographics and information about the area show evidence of a strong likelihood /potential for locals to want to join our coworking space as members. However, glancing at the data dictionary for the PUMS data provided by the US Government, we see that there are 285 variables entered about these folks - which I doubt we will even come close to needing all of this information.

2016 Global Coworking Survey - What statistics are we searching for?

As a result of this, instead we will look to another important piece of this puzzle - the 2016 Global Coworking Survey Demographic. Please note that this is data extrapolated only for the United States, which is more helpful to us compared to global information.

Distance to home

Starting on this page at slide 47, we see a list of the top ten reasons why folks choose a coworking space. Most of these things are factors that we have to take control of in terms of our environment, but there is one that translates toward a demographic market feasibility: A close distance to home.

Continuing on to slide 57, we see the breakdown of how folks get to their coworking spaces:

- Driving: 51%
- Cycling: 22%
- Walking: 19%
- Public Transit: 8%

In addition, we see that in an area with the market size we're looking at, 81% of respondents will arrive to their coworking space within 20 minutes of departing their house.

Age of Member

Moving on to slide 49, we see the age breakdown of participants of the survey which is clearly important:

- Age 18-29: 28%
- Age 30-39: 39%
- Age 40-49: 19%
- Age 50-59: 13%
- Age 60+: 1%

Job Status

On slide 51 we can see that job status could play a role in who the makeup of members are. According to survey results (as referenced on slide 76), as an owner of a coworking space you want to prioritize the acceptance of freelancers and entrepreneurs as members. The reasoning for this is that many folks join these spaces for networking and collaboration - people want to use the space to find contracts, job opportunities, or to see if there are any folks around who want to collaborate on a project with them (I must admit - sounds pretty exciting!). We see the following:

- Employee: 51%
- Entrepreneur: 12%
- Freelancer: 32%
- Other: 5%

Moving over to slide 55, we further see the following breakdown of Job type. Please note that there were separate categories for Project Management (4%) and Consulting (15%) however those do not correlate to any jobs found in the US Census Occupation variable. As a result, I have moved Project Management's share of the overall total in halves to IT and Business Development. I have spread Consulting around evenly to IT, PR, Design, Research, Business Development, and Higher Management.

- IT (Software Engineer, Web Developer): 31.5%
- PR, Marketing, Sales, Advertising: 7.5%
- Design: 9.5%
- Research: 6.5%
- Writing: 5%

- Business Development (includes entrepreneurs): 7.5%
- Education: 5%
- Higher Management: 7.5%
- Art: 3%
- Other: 16%

Gender

As reported by coworking spaces, the average share of female members sits at 38%; as reported by the individual member respondents, 41% of them were female. Let's sit at a happy medium of 40%.

Relationship Status

In the US, relationship status of survey respondents were as follows. However, the US Census does not distinguish between Single and In a Relationship so I have combined these categories into a single "Unmarried" category:

- In a relationship, unmarried: 41%
- Married: 47%
- Separated, Divorced, Widowed: 7%
- Other or NA: 5%

In addition to marital status, The 2017 Global Coworking Survey contains global information regarding number of children of respondents:

- 0 children: 64%
- 1 child: 11%
- 2 children: 18%
- 3 children: 7%

Health Insurance of members

In the US, the following breakdown of health insurance per respondent was as follows. Please note that the US Census only differentiates between "Health Insurance" and "No Health Insurance" and as a result I have rolled "Other" and NA into "No Health Insurance":

- Health Insurance: 87%
- No Health Insurance: 13%

Last place of work - before starting coworking

In the US, the following breakdown illustrates where respondents worked before deciding to go to a coworking space. However, the only information we can glean about this category from the US Census is whether or not somebody worked in a Home Office or not. As a result, I will boil down "Traditional Office", "Coffee Shop", "Small Shared Office Community", and "No Fixed Location" into "Not Home Office":

- Home Office: 44%
- Not Home Office: 56%

Highest Level of School Education

As seen in The 2017 Global Coworking Survey, we also have information on education levels of survey respondents on slide 5:

- Doctoral or higher: 4%
- Master: 41%
- Bachelor: 41%
- High School: 10%
- No Education: 1%
- NA: 3%

Relative Income

The survey contains information regarding how the respondent gauges their own level of income vs. their cost of living. I really wish this would be represented better, as straight income data, but we'll see what we can do with this later on. As seen on slide 60, respondents rated their income compared to cost of living as:

- Very High: 5%
- Rather High: 34%
- Somewhere in the Middle: 45%
- Rather Low: 9%
- Very Low: 1%
- NA: 6%

What data do we need?

In summary, we need to be on the lookout for the following in our PUMS Data:

- Does the person Drive, Bike, Walk, or Transit as their commute?
- How long would that person's mode of transportation take to get to our coworking space?
- What is the age of the person?
- What is the person's job type?
- What field does the person work in?
- What is the person's gender?
- What is the person's marital status?
- How many children does the person have?
- What type of health insurance does the person have, if any?
- What is the person's current workspace?
- What is that person's highest level of education?
- What is this person's relative income?

Checking out the PUMS Data Dictionary we can see what columns correlate to our desired variables.

- Mode of Transportation: "JWTR"
- Travel Time to Work: We need to create this variable using Google Maps API using "JWAP" and "PUMA"
- Age: "AGEP"
- Job Type: "COW"
- Occupation: "OCCP"
- Gender: "SEX"
- Marital Status: "MAR"
- Number of Children: Unfortunately this data is not available.
- Health Insurance Status: "HICOV"
- Current Workspace: "JWTR" contains Home Office information
- Education information: "SCHL"
- Income Information: "PINCP" adjusted by "ADJINC"

How to get this data

Before you start, make sure you have the following packages installed. Feel free to copy and paste the following into your console:

```
install.packages("tidycensus")
install.packages("tidyverse")
install.packages("dplyr")
install.packages("data.table")
install.packages("bit64")
install.packages("gmapsdistance")
```

We can get a sample of data from our market areas by going to the PUMS Data Website and following these steps:

- In the "topic name" search bar, type "PUMS" and hit search. Wait for your search to load.
- Click "2016 ACS 1-year Public Use Microdata Samples (PUMS) - CSV format". You should be directed to a new page.
- Select "California Population Records". Your browser should begin downloading a CSV file. It should be very large. Save this to the folder "Starting Point Data Files" in the repository for this R Project. It is recommended that you delete this file once you have run the script below due to its very large size.
- Navigate to the list of PUMA codes on the US Census website and search for the PUMA codes you're looking for. In my case, I need Santa Barbara South Coast (8303), Ventura City (11104), Oxnard/Port Hueneme (11103), and Moorpark area for good measure (11106).
- Run the R script below in order to get only the market areas you want from this file. This script will also save your filtered result in a file that will take much less time to read upon subsequent analyses, as long as you have begun this activity from the designated R Project instead of just this file alone.

```
Census <- read.csv("Starting Point Data Files/ss16pca.csv")
MarketCensus <- Census[Census$PUMA == 8303 | Census$PUMA == 11104 | Census$PUMA ==
    11103 | Census$PUMA == 11106, ]

MarketCensus <- subset(MarketCensus, select = c(JWTR, JWAP, AGEP, COW, OCCP, SEX,
    MAR, HICOV, SCHL, PINCP, ADJINC, PUMA))

write.csv(MarketCensus, file = "Starting Point Data Files/PUMS Data SB-Ventura County Market Areas.csv",
    row.names = FALSE)
```

I now recommend that you delete the "ss16pca.csv" file from the directory if you plan on sharing your results online or in a repository, due to

its size. If you need to reference your information on your market area, you can now access via the file “Starting Point Data Files/PUMS Data SB-Ventura County Market Areas.csv”.

Cleaning the Data

Now that we have all the data we could possibly want, it’s time for us to clean it up and make it usable. Since we know that we will have to create our own training data based off of the results of the 2016 Global Coworking Survey, we know we basically have to transform our data to look like the results of that survey since that’s the kind of data we’ll be constructing anyway. We’ll start with the names:

```
names(MarketCensus) <- c("TransportationMode", "ArrivalTime", "Age", "ClassOfWorker",
    "Occupation", "Sex", "MaritalStatus", "HealthInsurance", "EducationLevel", "TotalIncome",
    "IncomeAdjuster", "PUMA")
```

Work From Home Office

Let’s make a category within our data set to encode for if this person works in a home office or not. If this person works from home, it will be denoted as a 1. Otherwise it is a 0.

```
MarketCensus$HomeOffice <- ifelse(MarketCensus$TransportationMode == 11, 1, 0)
MarketCensus$HomeOffice[is.na(MarketCensus$HomeOffice)] <- 0
```

Mode of Transportation

We immediately see that this is an integer variable that needs to be translated to a categorical variable, where the categories should be “Driving”, “Biking”, “Walking”, and “Public Transit”. We see that this survey encodes for many things, so we will break it down as follows:

- “Driving” will code for “1”, “7” and “8” which stands for “Car, truck, or van”, “Taxicab”, and “Motorcycle”.
- “Biking” will code for “9” which stands for “Bicycle”.
- “Walking” will code for “10” which stands for “Walked”.
- “Public Transit” will code for “2”, “3”, “4”, “5” which stands for “Bus or trolley bus”, “Streetcar or trolley car”, “Subway or elevated”, and “Railroad”.
- The rest will be NA.

```
MarketCensus$TransportationMode <- as.factor(MarketCensus$TransportationMode)
levels(MarketCensus$TransportationMode) <- c("Driving", "Public Transit", "Public Transit",
    "Driving", "Driving", "Biking", "Walking", "Home Office", NA)
```

Age of Respondent

We Need to translate this into 5 factor levels in order for our sample to match our survey data. Also we will eliminate all samples with age less than 18 years old.

```
MarketCensus <- subset(MarketCensus, subset = MarketCensus$Age >= 18)
MarketCensus$AgeBracket <- NULL
for (i in 1:nrow(MarketCensus)) if (MarketCensus$Age[i] >= 18 & MarketCensus$Age[i] <=
    29) {
    MarketCensus$AgeBracket[i] <- "18-29"
} else if (MarketCensus$Age[i] >= 30 & MarketCensus$Age[i] <= 39) {
    MarketCensus$AgeBracket[i] <- "30-39"
} else if (MarketCensus$Age[i] >= 40 & MarketCensus$Age[i] <= 49) {
    MarketCensus$AgeBracket[i] <- "40-49"
} else if (MarketCensus$Age[i] >= 50 & MarketCensus$Age[i] <= 59) {
    MarketCensus$AgeBracket[i] <- "50-59"
} else {
    MarketCensus$AgeBracket[i] <- "60+"
}
MarketCensus$AgeBracket <- as.factor(MarketCensus$AgeBracket)
```

Job Type

We will classify the Job Type of the worker as follows:

- “Employee” will map to 1, 2, 3, 4, 5; or “Employee of Private Not for-profit”, “Employee of Private Non-Profit”, “Local Government Employee”, “State Government Employee”, and “Federal Government Employee”
- “Entrepreneur” will map to 7; or “Self-Employed in own incorporated business”

- “Freelancer” will map to 6; “Self-Employed in own not incorporated business”
- “Other” will map to 8; or “Working without pay in family business”
- We will drop data with COW status 9; or “Unemployed and last worked 5 years ago or more”

```
MarketCensus$ClassOfWorker <- as.factor(MarketCensus$ClassOfWorker)
levels(MarketCensus$ClassOfWorker) <- c("Employee", "Employee", "Employee", "Employee",
    "Employee", "Freelancer", "Entrepreneur", "Other", NA)
```

We have data relating to the person’s job as well. This data is too voluminous to discuss here in detail, but it breaks down as so:

- IT (Software Engineer, Web Developer, includes half Consulting): 1006-1200, 1240
- Consulting: NA
- PR, Marketing, Sales, Advertising: 40, 50, 60, 735, 2825, 4800, 4840-4930, 4965
- Design: 2630
- Project Management: NA
- Research: 1005, 735, 1220, 1600-1965
- Writing: 2840-2850
- Business Development (includes entrepreneurs, half consulting, and Project Management category): 510-726, 740
- Education: 2025, 2050, 2200-2550
- Higher Management: 10-20, 100-430
- Art: 500, 2600, 2700-2810, 2830, 2860-2920, 4000
- Other: 800-950, 1300-1550, 2000-2016, 2040, 2060-2160, 3000-3540, 3600-3955, 4010-4760, 4810-4830, 4940-4950, 5000-9830

```
MarketCensus$OccupationType <- NULL
for (i in 1:nrow(MarketCensus)) {
    if (is.na(MarketCensus$Occupation[i])) {
        MarketCensus$OccupationType[i] <- NA
    } else if ((MarketCensus$Occupation[i] >= 1006 & MarketCensus$Occupation[i] <=
        1200) | MarketCensus$Occupation[i] == 1240) {
        MarketCensus$OccupationType[i] <- "IT"
    } else if ((MarketCensus$Occupation[i] >= 40 & MarketCensus$Occupation[i] <= 60) |
        MarketCensus$Occupation[i] == 735 | MarketCensus$Occupation[i] == 2825 |
        MarketCensus$Occupation[i] == 4800 | (MarketCensus$Occupation[i] >= 4840 &
        MarketCensus$Occupation[i] <= 4930) | MarketCensus$Occupation[i] == 4965) {
        MarketCensus$OccupationType[i] <- "PR, Marketing, Sales, Advertising"
    } else if (MarketCensus$Occupation[i] == 2630) {
        MarketCensus$OccupationType[i] <- "Design"
    } else if (MarketCensus$Occupation[i] == 1005 | MarketCensus$Occupation[i] == 735 |
        MarketCensus$Occupation[i] == 1220 | (MarketCensus$Occupation[i] >= 1600 &
        MarketCensus$Occupation[i] <= 1965)) {
        MarketCensus$OccupationType[i] <- "Research"
    } else if (MarketCensus$Occupation[i] >= 2840 & MarketCensus$Occupation[i] <= 2850) {
        MarketCensus$OccupationType[i] <- "Writing"
    } else if ((MarketCensus$Occupation[i] >= 510 & MarketCensus$Occupation[i] <= 726) |
        MarketCensus$Occupation[i] == 740) {
        MarketCensus$OccupationType[i] <- "Business Development"
    } else if (MarketCensus$Occupation[i] == 2025 | MarketCensus$Occupation[i] == 2050 |
        (MarketCensus$Occupation[i] >= 2200 & MarketCensus$Occupation[i] <= 2550)) {
        MarketCensus$OccupationType[i] <- "Education"
    } else if ((MarketCensus$Occupation[i] >= 10 & MarketCensus$Occupation[i] <= 20) |
        (MarketCensus$Occupation[i] >= 100 & MarketCensus$Occupation[i] <= 430)) {
        MarketCensus$OccupationType[i] <- "Higher Management"
    } else if (MarketCensus$Occupation[i] == 500 | MarketCensus$Occupation[i] == 2600 |
        (MarketCensus$Occupation[i] >= 2700 & MarketCensus$Occupation[i] <= 2810) |
        MarketCensus$Occupation[i] == 2830 | (MarketCensus$Occupation[i] >= 2860 &
        MarketCensus$Occupation[i] <= 2920) | MarketCensus$Occupation[i] == 4000) {
        MarketCensus$OccupationType[i] <- "Art"
    } else {
        MarketCensus$OccupationType[i] <- "Other"
    }
}

MarketCensus$OccupationType[MarketCensus$ClassOfWorker == "Entrepreneur"] <- "Business Development"
MarketCensus$OccupationType <- as.factor(MarketCensus$OccupationType)
```

Gender

For Gender, Male will encode to “1” which is “male” and Female will encode to “2” which is “female”.

```
MarketCensus$Sex <- as.factor(MarketCensus$Sex)
levels(MarketCensus$Sex) <- c("Male", "Female")
```

Relationship Status

We will encode Relationship Status as the following:

- Not Married will encode to “5” which represents “Never Married”
- Married will encode to “1” which represents “Married”
- Separated, Divorced, Widowed will encode to “2”, “3”, and “4” which represents “Widowed”, “Divorced”, and “Separated”.
- Other or NA will encode to NA.

```
MarketCensus$MaritalStatus <- as.factor(MarketCensus$MaritalStatus)
levels(MarketCensus$MaritalStatus) <- c("Married", "Separated, Divorced, Widowed",
    "Separated, Divorced, Widowed", "Separated, Divorced, Widowed", "Not Married")
```

Health Insurance Status

Health Insurance will be encoded as below:

- “Health Insurance” will encode to “1” which represents “With health insurance coverage”.
- “No Health Insurance” will encode to “2” which represents “No Health Insurance Coverage”.

```
MarketCensus$HealthInsurance <- as.factor(MarketCensus$HealthInsurance)
levels(MarketCensus$HealthInsurance) <- c("Health Insurance", "No Health Insurance")
```

Last Type of Workspace

The only distinctions that the US Census makes in terms of your workspace is whether or not you work in a home office. This information is extracted from Mode Of Transportation variable, remember that we have already created this column called HomeOffice earlier on in the script.

```
MarketCensus$HomeOffice <- as.factor(MarketCensus$HomeOffice)
levels(MarketCensus$HomeOffice) <- c("Not Home Office", "Home Office")
```

Education Level

- Doctoral or higher: 24
- Master: 22
- Bachelor: 21, 23
- High School: 16-20
- No Education: 01-15
- NA: NA

```
MarketCensus$EducationLevel[MarketCensus$EducationLevel == 21 | MarketCensus$EducationLevel ==
    23] <- 21
MarketCensus$EducationLevel[MarketCensus$EducationLevel >= 16 & MarketCensus$EducationLevel <=
    20] <- 16
MarketCensus$EducationLevel[MarketCensus$EducationLevel >= 1 & MarketCensus$EducationLevel <=
    15] <- 1
MarketCensus$EducationLevel <- as.factor(MarketCensus$EducationLevel)
levels(MarketCensus$EducationLevel) <- c("No Education", "High School", "Bachelor",
    "Master", "Doctoral or Higher")
```

Relative Yearly Income Level

Since the Coworking Survey doesn’t include any actual numbers to describe the relative income distribution of their respondents, I will in this case translate the income of the US Census respondents into classes using information from (Investopedia) [https://www.investopedia.com/financial-edge/0912/which-income-class-are-you.aspx].

- “Very High” will map to Upper Class which is $350,000+
- “Rather High” will map to Upper Middle Class which ranges between $113,001 - $349,999
- “Somewhere in the Middle” will map to the Middle Middle Class, ranging between $47,178 - $113,000
- “Rather Low” will map to the Lower Middle Class, ranging between $18,871 - $47,177
- “Very Low” will map to the poverty level, or $0 - $18,870

- NA: NA

```
MarketCensus$TotalIncome <- MarketCensus$TotalIncome * 1.007588
MarketCensus$RelativeIncome[MarketCensus$TotalIncome <= 18870] <- "Very Low"
MarketCensus$RelativeIncome[MarketCensus$TotalIncome <= 47177 & MarketCensus$TotalIncome >=
    18871] <- "Rather Low"
MarketCensus$RelativeIncome[MarketCensus$TotalIncome <= 113000 & MarketCensus$TotalIncome >=
    47178] <- "Somewhere In The Middle"
MarketCensus$RelativeIncome[MarketCensus$TotalIncome <= 349999 & MarketCensus$TotalIncome >=
    113001] <- "Rather High"
MarketCensus$RelativeIncome[MarketCensus$TotalIncome >= 350000] <- "Very High"
MarketCensus$RelativeIncome <- as.factor(MarketCensus$RelativeIncome)
```

Travel Time to Work

In order to do this, we need to know where our Census folks live which is something we don't know; however I suspect we can simulate this information. If we could randomly assign folks to the various census tracts of Santa Barbara and Ventura Counties, we could then use the centroid latitude and longitude of those census tracts, run it through Google Maps API, and ascertain approximately how long it would take folks from all over the region to get to the Coworking Space. Unfortunately, this does scramble our data into non-useful high-income and low-income areas that may not be indicative of our region and thus doesn't allow us to target marketing areas effectively for our advertising campaign. However, if we were to obtain mean income of all census tracts, we may be able to assign folks into a census tract more intelligently based on how their income compares to the mean census tract information. So the steps are:

- Obtain Population Estimates for all pertinent census tracts and determine how many people in our 1% sample will be in each census tract, namely n_c where c denotes census tract.
- Obtain Average Income for all pertinent census tracts.
- Assign each census respondent to the census tract with the mean income closest to theirs. Increment the count of that census tract population, and if the census tract with the closest mean income has it's total population already full, assign to the next closest mean income census tract.
- Run this person's location through Google Maps API according to their mode of transportation to determine transportation time.

Assumptions:

- Almost all folks living within Carpinteria will be walking or biking to Coworking Space.
- Folks who live within proximity to Amtrak are more likely to take Amtrak which is very close to Coworking Space

---

Assigning Samples to Census Tracts

```
# Looks like SB County is FIPS Code 083 and Ventura County is 111

# Now if we want to get population and average income by Census Tract in Santa
# Barbara and Ventura Counties
CensusPopCA <- get_acs(geography = "tract", variables = c("B01003_001", "B19013_001E"),
    state = "CA")
```

```
## Please note: `get_acs()` now defaults to a year or endyear of 2016.
```

```
CensusPopCA <- subset(CensusPopCA, select = -moe)
CensusPopCA <- CensusPopCA %>% spread(variable, estimate)
names(CensusPopCA) <- c("GEOID", "NAME", "Population", "AverageIncome")

CensusPopCA$County <- substr(CensusPopCA$GEOID, 3, 5)
CensusPopCA$SampleCount <- 0

CensusPop <- subset(CensusPopCA, subset = CensusPopCA$County == "083" | CensusPopCA$County ==
    "111")

# Now we must filter out ONLY the market areas we want to look at. We wish to
# eliminate Santa Maria area, inland Ventura county, and any farmland areas where
# agriculture are the biggest industries. Also, these areas are simply too far
# away and not as easily accessible so it's too difficult to target folks from
# these areas anyway. We will stick to South Coast SB, Ventura City, Oxnard, Port
# Hueneme, and Moorpark for good measure.

PUMATractDecoder <- fread("https://www2.census.gov/geo/docs/maps-data/data/rel/2010_Census_Tract_to_2010_PUM
A.txt",
    colClasses = c("character", "character", "character", "character"))
PUMATractDecoderCA <- PUMATractDecoder[PUMATractDecoder$STATEFP == "06"]
```

```
PUMATractDecoder <- PUMATractDecoder[PUMATractDecoder$STATEFP == "06" & (PUMATractDecoder$PUMA5CE ==
    "08303" | PUMATractDecoder$PUMA5CE == "11104" | PUMATractDecoder$PUMA5CE == "11103" |
    PUMATractDecoder$PUMA5CE == "11106"), ]


PUMATractDecoder$GEOID <- paste(PUMATractDecoder$STATEFP, PUMATractDecoder$COUNTYFP,
    PUMATractDecoder$TRACTCE, sep = "")

CensusPop <- merge(x = CensusPop, y = PUMATractDecoder, by = "GEOID")

MarketCensusSB <- subset(MarketCensus, subset = MarketCensus$PUMA == 8303)
MarketCensus11103 <- subset(MarketCensus, subset = MarketCensus$PUMA == 11103)
MarketCensus11104 <- subset(MarketCensus, subset = MarketCensus$PUMA == 11104)
MarketCensus11106 <- subset(MarketCensus, subset = MarketCensus$PUMA == 11106)

CensusPop$ID <- c(1:nrow(CensusPop))
CensusSampleCounterSB <- subset(CensusPop, subset = PUMA5CE == "08303")
CensusSampleCounter11103 <- subset(CensusPop, subset = PUMA5CE == "11103")
CensusSampleCounter11104 <- subset(CensusPop, subset = PUMA5CE == "11104")
CensusSampleCounter11106 <- subset(CensusPop, subset = PUMA5CE == "11106")

PopTotalSB <- sum(CensusSampleCounterSB$Population)
SampleTotalSB <- nrow(MarketCensusSB)
CensusSampleCounterSB$SampleSize <- ceiling((CensusSampleCounterSB$Population/PopTotalSB) *
    SampleTotalSB)

PopTotal11103 <- sum(CensusSampleCounter11103$Population)
SampleTotal11103 <- nrow(MarketCensus11103)
CensusSampleCounter11103$SampleSize <- ceiling((CensusSampleCounter11103$Population/PopTotal11103) *
    SampleTotal11103)

PopTotal11104 <- sum(CensusSampleCounter11104$Population)
SampleTotal11104 <- nrow(MarketCensus11104)
CensusSampleCounter11104$SampleSize <- ceiling((CensusSampleCounter11104$Population/PopTotal11104) *
    SampleTotal11104)

PopTotal11106 <- sum(CensusSampleCounter11106$Population)
SampleTotal11106 <- nrow(MarketCensus11106)
CensusSampleCounter11106$SampleSize <- ceiling((CensusSampleCounter11106$Population/PopTotal11106) *
    SampleTotal11106)

for (i in 1:nrow(MarketCensusSB)) {
    if (is.na(MarketCensusSB$TotalIncome[i])) {
        MarketCensusSB$Location[i] <- NA
    } else {
        MarketCensusSB$Location[i] <- CensusSampleCounterSB$ID[which.min(abs(MarketCensusSB$TotalIncome[i] -

            CensusSampleCounterSB$AverageIncome))]
        CensusSampleCounterSB$SampleCount[CensusSampleCounterSB$ID == MarketCensusSB$Location[i]] <- CensusS
ampleCounterSB$SampleCount[CensusSampleCounterSB$ID ==
            MarketCensusSB$Location[i]] + 1
        if (CensusSampleCounterSB$SampleCount[CensusSampleCounterSB$ID == MarketCensusSB$Location[i]] >=
            CensusSampleCounterSB$SampleSize[CensusSampleCounterSB$ID == MarketCensusSB$Location[i]]) {
            CensusSampleCounterSB <- subset(CensusSampleCounterSB, subset = CensusSampleCounterSB$ID !=
                MarketCensusSB$Location[i])
        }
    }
}

for (i in 1:nrow(MarketCensus11103)) {
    if (is.na(MarketCensus11103$TotalIncome[i])) {
        MarketCensus11103$Location[i] <- NA
    } else {
        MarketCensus11103$Location[i] <- CensusSampleCounter11103$ID[which.min(abs(MarketCensus11103$TotalIn
come[i] -
            CensusSampleCounter11103$AverageIncome))]
        CensusSampleCounter11103$SampleCount[CensusSampleCounter11103$ID == MarketCensus11103$Location[i]] <
- CensusSampleCounter11103$SampleCount[CensusSampleCounter11103$ID ==
            MarketCensus11103$Location[i]] + 1
        if (CensusSampleCounter11103$SampleCount[CensusSampleCounter11103$ID == MarketCensus11103$Location[i
]] >=
            CensusSampleCounter11103$SampleSize[CensusSampleCounter11103$ID == MarketCensus11103$Location[i]
```

```
]) {
            CensusSampleCounter11103 <- subset(CensusSampleCounter11103, subset = CensusSampleCounter11103$I
D !=
                MarketCensus11103$Location[i])
        }
    }
}

for (i in 1:nrow(MarketCensus11104)) {
    if (is.na(MarketCensus11104$TotalIncome[i])) {
        MarketCensus11104$Location[i] <- NA
    } else {
        MarketCensus11104$Location[i] <- CensusSampleCounter11104$ID[which.min(abs(MarketCensus11104$TotalIn
come[i] -
            CensusSampleCounter11104$AverageIncome))]
        CensusSampleCounter11104$SampleCount[CensusSampleCounter11104$ID == MarketCensus11104$Location[i]] <
- CensusSampleCounter11104$SampleCount[CensusSampleCounter11104$ID ==
            MarketCensus11104$Location[i]] + 1
        if (CensusSampleCounter11104$SampleCount[CensusSampleCounter11104$ID == MarketCensus11104$Location[i
]] >=
            CensusSampleCounter11104$SampleSize[CensusSampleCounter11104$ID == MarketCensus11104$Location[i]
]) {
            CensusSampleCounter11104 <- subset(CensusSampleCounter11104, subset = CensusSampleCounter11104$I
D !=
                MarketCensus11104$Location[i])
        }
    }
}

for (i in 1:nrow(MarketCensus11106)) {
    if (is.na(MarketCensus11106$TotalIncome[i])) {
        MarketCensus11106$Location[i] <- NA
    } else {
        MarketCensus11106$Location[i] <- CensusSampleCounter11106$ID[which.min(abs(MarketCensus11106$TotalIn
come[i] -
            CensusSampleCounter11106$AverageIncome))]
        CensusSampleCounter11106$SampleCount[CensusSampleCounter11106$ID == MarketCensus11106$Location[i]] <
- CensusSampleCounter11106$SampleCount[CensusSampleCounter11106$ID ==
            MarketCensus11106$Location[i]] + 1
        if (CensusSampleCounter11106$SampleCount[CensusSampleCounter11106$ID == MarketCensus11106$Location[i
]] >=
            CensusSampleCounter11106$SampleSize[CensusSampleCounter11106$ID == MarketCensus11106$Location[i]
]) {
            CensusSampleCounter11106 <- subset(CensusSampleCounter11106, subset = CensusSampleCounter11106$I
D !=
                MarketCensus11106$Location[i])
        }
    }
}

MarketCensus <- rbind(MarketCensusSB, MarketCensus11103, MarketCensus11104, MarketCensus11106)
for (i in 1:nrow(MarketCensus)) {
    MarketCensus$Tract[i] <- CensusPop$NAME[MarketCensus$Location[i] == CensusPop$ID]
}
```

Obtaining Drive Time Data

```
# In order to get the latitude and longitude of the census tracts

CensusLoc <- fread("https://www2.census.gov/geo/docs/maps-data/data/gazetteer/2017_Gazetteer/2017_gaz_tracts
_06.txt")
CensusLoc$GEOID <- as.character(CensusLoc$GEOID)
CensusLoc$GEOID <- paste("0", CensusLoc$GEOID, sep = "")

# Now we have all the CA census tracts geographic location. Let's try and only
# select the census tracts we need for our two counties. We know that CA code is
# 06, SB county code is 083, and Ventura county code is 111. GEOID is structured
# as STATE+COUNTY+TRACT (number of digits SS-CCC-TTTTTT). Please note that
# fread() above got rid of the leading 0 of the state ID. Also note that all of
# the GEOIDs we need are included in the Data Frame CensusPop. Let's try and join
# up the census population data and the census locations, knowing that all the
# GEOIDs we want are in CensusPop. Merge will only show the matches so it should
```

```
# work well.

CensusPop <- merge(x = CensusPop, y = CensusLoc, by = "GEOID")

# Note that Google Maps API requires LAT-LONG format.
CensusPop$GoogleInput <- paste(CensusPop$INTPTLAT, CensusPop$INTPTLONG, sep = "+")

# Drop unnecessary columns
CensusPop <- subset(CensusPop, select = -c(USPS, ALAND, AWATER))

# We can now get the travel time to Carpinteria per census tract using Google
# Maps API for car, public transport, and bike. Please note that since there is a
# limit of 2,500 calls to Google Maps API per day, you may have problems with
# this in larger market areas. Make sure you have install.packages(gmapsdistance)
# and you have registered for a Google Maps API Key at
# https://developers.google.com/maps/documentation/distance-matrix/get-api-key#key.
# Then make sure you run the command set.api.key('YOUR KEY HERE')


tomorrow <- as.character(Sys.Date() + 1)
DriveTime <- gmapsdistance(origin = CensusPop$GoogleInput, destination = "410+Palm+Ave,+Carpinteria,+CA+9301
3",
    mode = "driving", arr_date = tomorrow, arr_time = "17:00:00")

BikeTime <- gmapsdistance(origin = CensusPop$GoogleInput, destination = "410+Palm+Ave,+Carpinteria,+CA+93013
",
    mode = "bicycling", key = "AIzaSyAT1wxOfPoPZowF2lPliMGA884ArKVQ7XU", arr_date = tomorrow,
    arr_time = "17:00:00")

TransitTime <- gmapsdistance(origin = CensusPop$GoogleInput, destination = "410+Palm+Ave,+Carpinteria,+CA+93
013",
    mode = "transit", key = "AIzaSyAT1wxOfPoPZowF2lPliMGA884ArKVQ7XU", arr_date = tomorrow,
    arr_time = "17:00:00")

WalkingTime <- gmapsdistance(origin = CensusPop$GoogleInput, destination = "410+Palm+Ave,+Carpinteria,+CA+93
013",
    mode = "walking", key = "AIzaSyAT1wxOfPoPZowF2lPliMGA884ArKVQ7XU", arr_date = tomorrow,
    arr_time = "17:00:00")

# DriveTime is a list with three data frames (Time, Distance, Status). Let's get
# Time over to the CensusPop Data Frame as DrivingTime variable.

DriveTime$Time$or <- as.character(DriveTime$Time$or)
names(DriveTime$Time) <- c("GoogleInput", "DrivingTime")
CensusPop <- merge(x = CensusPop, y = DriveTime$Time, by = "GoogleInput")

BikeTime$Time$or <- as.character(BikeTime$Time$or)
names(BikeTime$Time) <- c("GoogleInput", "BikingTime")
CensusPop <- merge(x = CensusPop, y = BikeTime$Time, by = "GoogleInput")

TransitTime$Time$or <- as.character(TransitTime$Time$or)
names(TransitTime$Time) <- c("GoogleInput", "TransitTime")
CensusPop <- merge(x = CensusPop, y = TransitTime$Time, by = "GoogleInput")

WalkingTime$Time$or <- as.character(WalkingTime$Time$or)
names(WalkingTime$Time) <- c("GoogleInput", "WalkingTime")
CensusPop <- merge(x = CensusPop, y = WalkingTime$Time, by = "GoogleInput")

# Write it to an excel file to analyze or reference later
write.table(CensusPop, file = "SB-Ventura Census Tract Populations.txt", row.names = FALSE)

for (i in 1:nrow(MarketCensus)) {
    MarketCensus$DriveTime[i] <- CensusPop$DrivingTime[MarketCensus$Location[i] ==
        CensusPop$ID]
    MarketCensus$BikeTime[i] <- CensusPop$BikingTime[MarketCensus$Location[i] ==
        CensusPop$ID]
    MarketCensus$TransitTime[i] <- CensusPop$TransitTime[MarketCensus$Location[i] ==
        CensusPop$ID]
    MarketCensus$WalkTime[i] <- CensusPop$WalkingTime[MarketCensus$Location[i] ==
        CensusPop$ID]
}

MarketCensus$TransportationMode[is.na(MarketCensus$TransportationMode)] <- "Driving"
```

```
MarketCensus$TransportationMode[is.na(MarketCensus$TransportationMode)] <- "Driving"

MarketCensus$CommuteTime <- NULL
for (i in 1:nrow(MarketCensus)) {
    if (is.na(MarketCensus$TransportationMode[i])) {
        MarketCensus$CommuteTime[i] <- NA
    } else if (MarketCensus$TransportationMode[i] == "Driving") {
        MarketCensus$CommuteTime[i] <- MarketCensus$DriveTime[i]
    } else if (MarketCensus$TransportationMode[i] == "Public Transit") {
        MarketCensus$CommuteTime[i] <- MarketCensus$TransitTime[i]
    } else if (MarketCensus$TransportationMode[i] == "Biking") {
        MarketCensus$CommuteTime[i] <- MarketCensus$BikeTime[i]
    } else if (MarketCensus$TransportationMode[i] == "Walking") {
        MarketCensus$CommuteTime[i] <- MarketCensus$WalkTime[i]
    } else {
        MarketCensus$CommuteTime[i] <- MarketCensus$DriveTime
    }
}
```

Time to make some assumptions about travel. I will be running off of the following assumptions:

- Anybody living in the same Census Tract as the Coworking Space will be able to easily walk or bike to the Coworking Space. I will change 80% of TransportationMode to Walking or Biking.
- Anybody living within the same Census Tract as the Santa Barbara Amtrak Station can take that train to the Coworking Space within 13 minutes. I will change about 25% of those entries to Public Transit and post a CommuteTime of “Yes”.
- Anybody living within the same Census Tract as the Ventura Amtrak station can take that train to the Coworking Space within 20 minutes. I will change about 25% of those entries to Public Transit and post a CommuteTime of “Yes”
- My model assumes anybody who is walking, biking, or taking public transit to work currently will continue to do so to the Coworking Space, however that is not the case. Say, for example, somebody is walking to their current job because it is a 15 minute walk away, however it would take them an hour to walk to the coworking space or a 15 minute drive would elect to drive to the coworking space.
- I will take 80% of entries who are walking, biking, or public transit whose CommuteTime is greater than 30 minutes and convert their method to Driving.
- I will leave the remaining 20% as is to represent folks who may not have access to a car or else enjoy the fact they can walk or bike to work.
- Anybody who is employed and has an NA for TransportationMode will be placed in the Driving category.
- Anybody listed as HomeOffice will have their commute represented by DriveTime.

```
MarketCensus$CommuteTime <- MarketCensus$CommuteTime/60
WalkBikeTrans <- MarketCensus[(MarketCensus$TransportationMode == "Walking" | MarketCensus$TransportationMod
e ==
    "Biking" | MarketCensus$TransportationMode == "Public Transit") & MarketCensus$CommuteTime >
    30, ]
WalkBikeTrans$Random <- c(1:nrow(WalkBikeTrans))
WalkBikeTrans$Random <- sample.split(WalkBikeTrans$Random, SplitRatio = 0.8)
WalkBikeTrans$TransportationMode[WalkBikeTrans$Random == TRUE] <- "Driving"
WalkBikeTrans$CommuteTime[WalkBikeTrans$Random == TRUE] <- WalkBikeTrans$DriveTime/60
```

```
## Warning in WalkBikeTrans$CommuteTime[WalkBikeTrans$Random == TRUE] <-
## WalkBikeTrans$DriveTime/60: number of items to replace is not a multiple of
## replacement length
```

```
WalkBikeTrans <- subset(WalkBikeTrans, select = -Random)
MarketCensus <- subset(MarketCensus, subset = (MarketCensus$TransportationMode !=
    "Walking" & MarketCensus$TransportationMode != "Biking" & MarketCensus$TransportationMode !=
    "Public Transit") | MarketCensus$CommuteTime <= 30)
MarketCensus <- rbind(MarketCensus, WalkBikeTrans)

MarketCensus$CommuteTime[MarketCensus$CommuteTime <= 20] <- 1
MarketCensus$CommuteTime[MarketCensus$CommuteTime > 20] <- 21
MarketCensus$CommuteTime <- as.factor(MarketCensus$CommuteTime)
levels(MarketCensus$CommuteTime) <- c("Yes", "No")

TractCarp <- MarketCensus[MarketCensus$Location == 26, ]
TractCarp$Random <- c(1:nrow(TractCarp))
TractCarp$Random <- sample.split(TractCarp$Random, SplitRatio = 0.8)
TractCarp$TransportationMode[TractCarp$Random == TRUE] <- "Walking"
TractCarp <- subset(TractCarp, select = -Random)
MarketCensus <- subset(MarketCensus, subset = MarketCensus$Location != 26)
MarketCensus <- rbind(MarketCensus, TractCarp)

TractSBAmtrak <- MarketCensus[MarketCensus$Location == 19 | MarketCensus$Location ==
    14 | MarketCensus$Location == 15, ]
TractSBAmtrak$Random <- c(1:nrow(TractSBAmtrak))
TractSBAmtrak$Random <- sample.split(TractSBAmtrak$Random, SplitRatio = 0.25)
TractSBAmtrak$TransportationMode[TractSBAmtrak$Random == TRUE] <- "Public Transit"
TractSBAmtrak$CommuteTime[TractSBAmtrak$Random == TRUE] <- "Yes"
TractSBAmtrak <- subset(TractSBAmtrak, select = -Random)
MarketCensus <- subset(MarketCensus, subset = (MarketCensus$Location != 19 & MarketCensus$Location !=
    14 & MarketCensus$Location != 15))
MarketCensus <- rbind(MarketCensus, TractSBAmtrak)

TractVentAmtrak <- MarketCensus[MarketCensus$Location == 66, ]
TractVentAmtrak$Random <- c(1:nrow(TractVentAmtrak))
TractVentAmtrak$Random <- sample.split(TractVentAmtrak$Random, SplitRatio = 0.25)
TractVentAmtrak$TransportationMode[TractVentAmtrak$Random == TRUE] <- "Public Transit"
TractVentAmtrak$CommuteTime[TractVentAmtrak$Random == TRUE] <- "Yes"
TractVentAmtrak <- subset(TractVentAmtrak, select = -Random)
MarketCensus <- subset(MarketCensus, subset = MarketCensus$Location != 66)
MarketCensus <- rbind(MarketCensus, TractVentAmtrak)

MarketCensus <- subset(MarketCensus, select = -c(Location, Tract, DriveTime, BikeTime,
    TransitTime, WalkTime, ArrivalTime, Age, Occupation, TotalIncome, IncomeAdjuster,
    PUMA))
```

Conclusion to Data Wrangling for Market Area

See below the general structure of our data set:

```
head(MarketCensus)
```

```
##      TransportationMode ClassOfWorker    Sex MaritalStatus
## 569             Driving      Employee   Male   Not Married
## 806             Driving          <NA>   Male       Married
## 807             Driving          <NA> Female       Married
## 1603            Driving      Employee Female   Not Married
## 1921            Driving          <NA> Female   Not Married
## 1942        Home Office      Employee   Male       Married
##       HealthInsurance EducationLevel      HomeOffice AgeBracket
## 569  Health Insurance    High School Not Home Office      40-49
## 806  Health Insurance       Bachelor Not Home Office        60+
## 807  Health Insurance       Bachelor Not Home Office        60+
## 1603 Health Insurance         Master Not Home Office      18-29
## 1921 Health Insurance    High School Not Home Office      18-29
## 1942 Health Insurance       Bachelor     Home Office      40-49
##         OccupationType          RelativeIncome CommuteTime
## 569              Other                Very Low          No
## 806               <NA>             Rather High          No
## 807               <NA>             Rather High          No
## 1603         Education Somewhere In The Middle          No
## 1921              <NA>                Very Low          No
## 1942 Higher Management Somewhere In The Middle         Yes
```

```
tail(MarketCensus)
```

```
##      TransportationMode ClassOfWorker  Sex MaritalStatus HealthInsurance
## NA       Public Transit          <NA> <NA>          <NA>            <NA>
## NA.1     Public Transit          <NA> <NA>          <NA>            <NA>
## NA.2               <NA>          <NA> <NA>          <NA>            <NA>
## NA.3               <NA>          <NA> <NA>          <NA>            <NA>
## NA.4               <NA>          <NA> <NA>          <NA>            <NA>
## NA.5               <NA>          <NA> <NA>          <NA>            <NA>
##      EducationLevel HomeOffice AgeBracket OccupationType RelativeIncome
## NA             <NA>       <NA>       <NA>           <NA>           <NA>
## NA.1           <NA>       <NA>       <NA>           <NA>           <NA>
## NA.2           <NA>       <NA>       <NA>           <NA>           <NA>
## NA.3           <NA>       <NA>       <NA>           <NA>           <NA>
## NA.4           <NA>       <NA>       <NA>           <NA>           <NA>
## NA.5           <NA>       <NA>       <NA>           <NA>           <NA>
##      CommuteTime
## NA           Yes
## NA.1         Yes
## NA.2        <NA>
## NA.3        <NA>
## NA.4        <NA>
## NA.5        <NA>
```

```
write.table(x = MarketCensus, file = "Springboard Deliverables/Market Census Final Data.txt",
    row.names = FALSE)
```

Creating Training Data

In order to run a successful scoring system, we must create a training set of data. In order to do this, I will create a data set of 100 observations using the breakdown of the 2017 Global Coworking Survey and each assign them a “1” in a new category called “CoworkingMember”. This indicates that these people are indeed customers of coworking spaces. I will then take the total population of California and divide that by the number of coworking space members in the state of California. Multiplying that number by 100, we should get the total number of observations of non-coworking individuals in our training data set. That number will be big enough that we shouldn’t see problems with bias in sample size, so we will extract at random that number of samples from our huge file of 1% of the California population. These will have assignments of “0” in the “CoworkingMember” column. Our training data should then be complete, with the same ratio of coworking members to non-coworking members that we see in real life.

Alternatively, should we want to make a less biased sample size, we could create more than 100 observations of CoworkingMember set to 1 and then attach that to the full 300,000 count sample of the population of California.

Creating Coworker Member Profiles

Mode of Transportation and Distance to Coworking Space

Mode of Transportation should be pretty easy to simulate. We will make 51 drivers, 22 cyclists, 19 walkers, and 8 Public Transportation Users.

```
TransportationMode <- c(rep("Driving", times = 51 * 2), rep("Public Transit", times = 8 *
    2), rep("Biking", times = 22 * 2), rep("Walking", times = 19 * 2))


CommuteTime <- c(rep("Yes", times = 81 * 2), rep("No", times = 19 * 2))
```

Age of Member

```
AgeBracket <- c(rep("18-29", times = 28 * 2), rep("30-39", times = 39 * 2), rep("40-49",
    times = 19 * 2), rep("50-59", times = 13 * 2), "60+", "60+")
```

Job Status

```
ClassOfWorker <- c(rep("Employee", times = 51 * 2), rep("Freelancer", times = 32 *
    2), rep("Entrepreneur", times = 12 * 2), rep("Other", times = 5 * 2))

OccupationType <- c(rep("Art", times = 3 * 2), rep("Business Development", times = 8 *
    2), rep("Design", times = 10 * 2), rep("Education", times = 5 * 2), rep("Higher Management",
    times = 7 * 2), rep("IT", times = 31 * 2), rep("Other", times = 16 * 2), rep("PR, Marketing, Sales, Adv
ertising",
    times = 8 * 2), rep("Research", times = 7 * 2), rep("Writing", times = 5 * 2))
```

Gender

```
Sex <- c(rep("Male", times = 60 * 2), rep("Female", times = 40 * 2))
```

Relationship Status

```
MaritalStatus <- c(rep("Married", times = 47 * 2), rep("Separated, Divorced, Widowed",
    times = 7 * 2), rep("Not Married", times = 41 * 2), rep(NA, times = 5 * 2))
```

Health Insurance

```
HealthInsurance <- c(rep("Health Insurance", times = 87 * 2), rep("No Health Insurance",
    times = 13 * 2))
```

Last Place of Work

```
HomeOffice <- c(rep("Not Home Office", times = 56 * 2), rep("Home Office", time = 44 *
    2))
```

Highest Level of School Education

```
EducationLevel <- c("No Education", "No Education", rep("High School", times = 10 *
    2), rep("Bachelor", times = 41 * 2), rep("Master", times = 41 * 2), rep("Doctoral or Higher",
    times = 4 * 2), rep(NA, times = 3 * 2))
```

Relative Income

```
RelativeIncome <- c(rep("Rather High", times = 34 * 2), rep("Rather Low", times = 9 *
    2), rep("Somewhere In The Middle", times = 45 * 2), rep("Very High", times = 5 *
    2), "Very Low", "Very Low", rep(NA, times = 6 * 2))
```

Coworker Profiles

```
TrainCensus <- data.frame(TransportationMode)

TrainCensus <- TrainCensus[sample(nrow(TrainCensus)), ]
TrainCensus <- data.frame(TrainCensus, CommuteTime)
names(TrainCensus) <- c("TransportationMode", "CommuteTime")

TrainCensus <- TrainCensus[sample(nrow(TrainCensus)), ]
TrainCensus <- data.frame(TrainCensus, AgeBracket)

TrainCensus <- TrainCensus[sample(nrow(TrainCensus)), ]
TrainCensus <- data.frame(TrainCensus, ClassOfWorker)

TrainCensus <- TrainCensus[sample(nrow(TrainCensus)), ]
TrainCensus <- data.frame(TrainCensus, OccupationType)

TrainCensus <- TrainCensus[sample(nrow(TrainCensus)), ]
TrainCensus <- data.frame(TrainCensus, Sex)

TrainCensus <- TrainCensus[sample(nrow(TrainCensus)), ]
TrainCensus <- data.frame(TrainCensus, MaritalStatus)

TrainCensus <- TrainCensus[sample(nrow(TrainCensus)), ]
TrainCensus <- data.frame(TrainCensus, HealthInsurance)

TrainCensus <- TrainCensus[sample(nrow(TrainCensus)), ]
TrainCensus <- data.frame(TrainCensus, HomeOffice)

TrainCensus <- TrainCensus[sample(nrow(TrainCensus)), ]
TrainCensus <- data.frame(TrainCensus, EducationLevel)

TrainCensus <- TrainCensus[sample(nrow(TrainCensus)), ]
TrainCensus <- data.frame(TrainCensus, RelativeIncome)


TrainCensus$CoworkMember <- 1
```

Creating Non-Member Profiles

We will now create non-member profiles from a complete sample of the Los Angeles population.

```
Census$PUMA <- as.character(Census$PUMA)
Census <- subset(Census, subset = substr(Census$PUMA, 1, 2) == "37")

Census <- subset(Census, select = c(JWTR, JWAP, AGEP, COW, OCCP, SEX, MAR, HICOV,
    SCHL, PINCP, ADJINC, PUMA))


names(Census) <- c("TransportationMode", "ArrivalTime", "Age", "ClassOfWorker", "Occupation",
    "Sex", "MaritalStatus", "HealthInsurance", "EducationLevel", "TotalIncome", "IncomeAdjuster",
    "PUMA")

Census$HomeOffice <- ifelse(Census$TransportationMode == 11, 1, 0)
Census$HomeOffice[is.na(Census$HomeOffice)] <- 0

Census$TransportationMode <- as.factor(Census$TransportationMode)
levels(Census$TransportationMode) <- c("Driving", "Public Transit", "Public Transit",
    "Public Transit", "Public Transit", NA, "Driving", "Driving", "Biking", "Walking",
    "Home Office", NA)

Census <- subset(Census, subset = Census$Age >= 18)

Census$Age[Census$Age >= 18 & Census$Age <= 29] <- 18
Census$Age[Census$Age >= 30 & Census$Age <= 39] <- 30
Census$Age[Census$Age >= 40 & Census$Age <= 49] <- 40
Census$Age[Census$Age >= 50 & Census$Age <= 59] <- 50
Census$Age[Census$Age >= 60] <- 60

Census$Age <- as.factor(Census$Age)
levels(Census$Age) <- c("18-29", "30-39", "40-49", "50-59", "60+")
Census$AgeBracket <- Census$Age
```

```
Census$ClassOfWorker <- as.factor(Census$ClassOfWorker)
levels(Census$ClassOfWorker) <- c("Employee", "Employee", "Employee", "Employee",
    "Employee", "Freelancer", "Entrepreneur", "Other", NA)

Census$Occupation[is.na(Census$Occupation)] <- 0
Census$Occupation[(Census$Occupation >= 1006 & Census$Occupation <= 1200) | Census$Occupation ==
    1240] <- 1006
Census$Occupation[(Census$Occupation >= 40 & Census$Occupation <= 60) | Census$Occupation ==
    735 | Census$Occupation == 2825 | Census$Occupation == 4800 | (Census$Occupation >=
    4840 & Census$Occupation <= 4930) | Census$Occupation == 4965] <- 2825
Census$Occupation[Census$Occupation == 1005 | Census$Occupation == 735 | Census$Occupation ==
    1220 | (Census$Occupation >= 1600 & Census$Occupation <= 1965)] <- 1005
Census$Occupation[Census$Occupation >= 2840 & Census$Occupation <= 2850] <- 2840
Census$Occupation[(Census$Occupation >= 510 & Census$Occupation <= 726) | Census$Occupation ==
    740] <- 510
Census$Occupation[Census$Occupation == 2025 | Census$Occupation == 2050 | (Census$Occupation >=
    2200 & Census$Occupation <= 2550)] <- 2025
Census$Occupation[(Census$Occupation >= 10 & Census$Occupation <= 20) | (Census$Occupation >=
    100 & Census$Occupation <= 430)] <- 10
Census$Occupation[Census$Occupation == 500 | Census$Occupation == 2600 | (Census$Occupation >=
    2700 & Census$Occupation <= 2810) | Census$Occupation == 2830 | (Census$Occupation >=
    2860 & Census$Occupation <= 2920) | Census$Occupation == 4000] <- 500
Census$Occupation[Census$Occupation != 500 & Census$Occupation != 10 & Census$Occupation !=
    2025 & Census$Occupation != 510 & Census$Occupation != 2840 & Census$Occupation !=
    1005 & Census$Occupation != 2630 & Census$Occupation != 2825 & Census$Occupation !=
    1006 & Census$Occupation != 0] <- 800

Census$Occupation <- as.factor(Census$Occupation)
levels(Census$Occupation) <- c(NA, "Higher Management", "Art", "Business Development",
    "Other", "Research", "IT", "Education", "Design", "PR, Marketing, Sales, Advertising",
    "Writing")



Census$Occupation[Census$ClassOfWorker == "Entrepreneur"] <- "Business Development"

Census$OccupationType <- Census$Occupation

Census$Sex <- as.factor(Census$Sex)
levels(Census$Sex) <- c("Male", "Female")

Census$MaritalStatus <- as.factor(Census$MaritalStatus)
levels(Census$MaritalStatus) <- c("Married", "Separated, Divorced, Widowed", "Separated, Divorced, Widowed",

    "Separated, Divorced, Widowed", "Not Married")

Census$HealthInsurance <- as.factor(Census$HealthInsurance)
levels(Census$HealthInsurance) <- c("Health Insurance", "No Health Insurance")

Census$HomeOffice <- as.factor(Census$HomeOffice)
levels(Census$HomeOffice) <- c("Not Home Office", "Home Office")

Census$EducationLevel[Census$EducationLevel == 21 | Census$EducationLevel == 23] <- 21
Census$EducationLevel[Census$EducationLevel >= 16 & Census$EducationLevel <= 20] <- 16
Census$EducationLevel[Census$EducationLevel >= 1 & Census$EducationLevel <= 15] <- 1
Census$EducationLevel <- as.factor(Census$EducationLevel)
levels(Census$EducationLevel) <- c("No Education", "High School", "Bachelor", "Master",
    "Doctoral or Higher")

Census$TotalIncome <- Census$TotalIncome * 1.007588
Census$RelativeIncome[Census$TotalIncome <= 18870] <- "Very Low"
Census$RelativeIncome[Census$TotalIncome <= 47177 & Census$TotalIncome >= 18871] <- "Rather Low"
Census$RelativeIncome[Census$TotalIncome <= 113000 & Census$TotalIncome >= 47178] <- "Somewhere In The Middl
e"
Census$RelativeIncome[Census$TotalIncome <= 349999 & Census$TotalIncome >= 113001] <- "Rather High"
Census$RelativeIncome[Census$TotalIncome >= 350000] <- "Very High"
Census$RelativeIncome <- as.factor(Census$RelativeIncome)

Census$CoworkMember <- 0

Census <- subset(Census, select = -c(ArrivalTime, Age, Occupation, IncomeAdjuster,
    PUMA))
```

Commute Time to nearest Coworking Space

```
CensusPopCA <- subset(CensusPopCA, subset = CensusPopCA$County == "037" & !is.na(CensusPopCA$AverageIncome))
CensusPopCA$ID <- c(1:nrow(CensusPopCA))
CensusPopCA2 <- CensusPopCA
PopTotalCA <- sum(CensusPopCA$Population)
SampleTotalCA <- nrow(Census)
CensusPopCA2$SampleSize <- ceiling((CensusPopCA2$Population/PopTotalCA) * SampleTotalCA)


for (i in 1:nrow(Census)) {
    if (is.na(Census$TotalIncome[i])) {
        Census$Location[i] <- NA
    } else {
        Census$Location[i] <- CensusPopCA2$ID[which.min(abs(Census$TotalIncome[i] -
            CensusPopCA2$AverageIncome))]
        CensusPopCA2$SampleCount[CensusPopCA2$ID == Census$Location[i]] <- CensusPopCA2$SampleCount[CensusPo
pCA2$ID ==
            Census$Location[i]] + 1
        if (CensusPopCA2$SampleCount[CensusPopCA2$ID == Census$Location[i]] >= CensusPopCA2$SampleSize[Censu
sPopCA2$ID ==
            Census$Location[i]]) {
            CensusPopCA2 <- subset(CensusPopCA2, subset = CensusPopCA2$ID != Census$Location[i])
        }
    }
}

CensusPopCA <- merge(x = CensusPopCA, y = CensusLoc, by = "GEOID")
```

We must now extract the list of all Coworking Spaces in LA County in order to determine if our citizens are within a 20 minute drive of a coworking space. We will do this by using coworker.com API to request every listed coworking space within a 150 Kilometer radius of the center point of Los Angeles County. Since there are over 2,000 census tracts, we cannot run a Google Maps search on all coworking spaces for each census tract, so we will run an algorithm that will determine the closest coworking space by Euclidean distance to each Census Tract. Then we will run the Google Maps API only 2,307 times instead of 2,307 * 130 times in order to fit within our daily quota of Google Maps calls per day. Please note that I will not be actually running this in the RMarkdown document, only displaying it, so as to reduce the number of calls to my API partner, coworker.com.

```
x1 <- GET(url = "https://www.coworker.com/api/nearbyspaces/format/json?lat=34.332393&lon=-118.170937&rad=150
")
Coworking <- as.data.frame(fromJSON(content(x1, as = "text")))
```

```
Coworking <- as.data.frame(read.table(file = "Starting Point Data Files/CoworkingLA",
    header = TRUE))

for (i in 1:nrow(CensusPopCA)) {
    ID <- NULL
    Distance <- NULL
    for (p in 1:nrow(Coworking)) {
        y <- Coworking$cs_id[p]
        x <- sqrt((CensusPopCA$INTPTLAT[i] - Coworking$latitude[p])^2 + (CensusPopCA$INTPTLONG[i] -
            Coworking$longitude[p])^2)
        ID <- c(ID, y)
        Distance <- c(Distance, x)
    }
    dfclose <- data.frame(ID, Distance)
    CensusPopCA$Closest[i] <- dfclose$ID[which.min(dfclose$Distance)]
}

CensusPopCA$GoogleInput <- paste(CensusPopCA$INTPTLAT, CensusPopCA$INTPTLONG, sep = "+")
Coworking <- subset(Coworking, select = c(cs_id, latitude, longitude))
CensusPopCA <- merge(x = CensusPopCA, y = Coworking, by.x = "Closest", by.y = "cs_id")
CensusPopCA$SpaceInput <- paste(CensusPopCA$latitude, CensusPopCA$longitude, sep = "+")
```

```
set.api.key("AIzaSyDr7XpI24NrDVnAZIgDkj32zhm3q0Z8Xcg")
TrainDriveTime <- gmapsdistance(origin = CensusPopCA$GoogleInput, destination = CensusPopCA$SpaceInput,
    combinations = "pairwise", mode = "driving", arr_date = tomorrow, arr_time = "17:00:00")

TrainDriveTime$Time$or <- as.character(TrainDriveTime$Time$or)
TrainDriveTime$Time <- subset(TrainDriveTime$Time, select = -de)
names(TrainDriveTime$Time) <- c("GoogleInput", "DrivingTime")
CensusPopCA <- merge(x = CensusPopCA, y = TrainDriveTime$Time, by = "GoogleInput")
CensusPopCA$DrivingTime <- CensusPopCA$DrivingTime/60
write.table(x = CensusPopCA, file = "Starting Point Data Files/CensusPopCA")
```

```
CensusPopCA <- read.table(file = "Starting Point Data Files/CensusPopCA", header = TRUE)
CensusPopMerge <- subset(CensusPopCA, select = c(ID, DrivingTime))
Census <- merge(x = Census, y = CensusPopMerge, by.x = "Location", by.y = "ID")

colnames(Census)[colnames(Census) == "DrivingTime"] <- "CommuteTime"
Census$CommuteTime[Census$CommuteTime <= 20] <- 1
Census$CommuteTime[Census$CommuteTime > 20] <- 21
Census$CommuteTime <- as.factor(Census$CommuteTime)
levels(Census$CommuteTime) <- c("Yes", "No")

Census <- subset(Census, select = -c(Location, TotalIncome))
TrainCensus <- rbind(TrainCensus, Census)
```

Running a Linear Regression on Training Data

```
model <- lm(CoworkMember ~ TransportationMode + AgeBracket + ClassOfWorker + OccupationType +
    Sex + MaritalStatus + HealthInsurance + HomeOffice + EducationLevel + RelativeIncome,
    data = TrainCensus)
```