Final-report

Ashwin Ayyasamy Elamurugu

Introduction:

We have been given three data files, namely, Edmunds, IRS, and LA Cell Towers, that have just one common variable among them (Zip Code) and give out totally different and seemingly disconnected information. While Edmunds data deals with consumers' interest in cars and the dealerships to which the information was passed to, IRS and LA Cell Towers deal with tax returns filed and information about cell towers, both based on zip codes, respectively.

The IRS data shows a lot of potential for analysis tests to be performed on it. Considering the abundance of numeric variables in it, I can be choosing any two of them to perform a t-test, to find out if they are related to one another. It is highly important to perform a t-test as it helps us figure out if the variables that look disconnected are actually similar, and vice-versa. It would also save our time and space, in future, as we can avoid working on variables that we find to be the same

Using the numeric variables, in IRS, that tell us about the number of returns filed as 'single' and the number of dependents, a linear regression can be performed to figure out the effect that one variable might possibly have on the other, and if there is a confoudning variable that might be impacting both the variables. While t-test tells us if any two variables are in some way related, linear regression gives us the actual possible relationship. In this case, it is essential to create a linear model as it would give out the possible change that the latter varible might undergo when the former varies. Moreover, it might be a subject of interest to many to know the extent to which the number of returns filed as 'single' variable could explain the variation in the number of dependents variable.

A custom function 'zipit' that takes a vector of inputs and outputs a table that gives the user almost all of the essential information related to each zip code, using all the three given data files. While it is always useful to create functions that make it easy for the user to get the required information at once and without much time consumption, this function, in particular, might be highly useful for further research involving zip codes.

Edmunds Data:

Edmunds Data, originally taken out from Edmunds.com, tells us about a consumer's interest in a car and the dealership to which the information about the consumer has been passed so as to make it feasible for a transaction to occur. Edmunds.com terms each such consumer's interest as 'lead' and gives each one an id (can be seen as lead_id in the data file). So, each lead_id is an observation and the variables give out information regarding the date on which the lead was submitted, the specifications of the car model like its price (as quoted by the dealership and also as suggested by the manufacturer), its model year, and more, and the details about the dealership to which the particular lead was passed to. The loaction information of each dealership, like its place and zip code, make up the details involving the deealership.

Cleaning Edmunds:

1. Retained just the required varibales like lead_id, lead_date, model_year, make, model, msrp, dealer_dma, and dealer_zip, by subsetting.

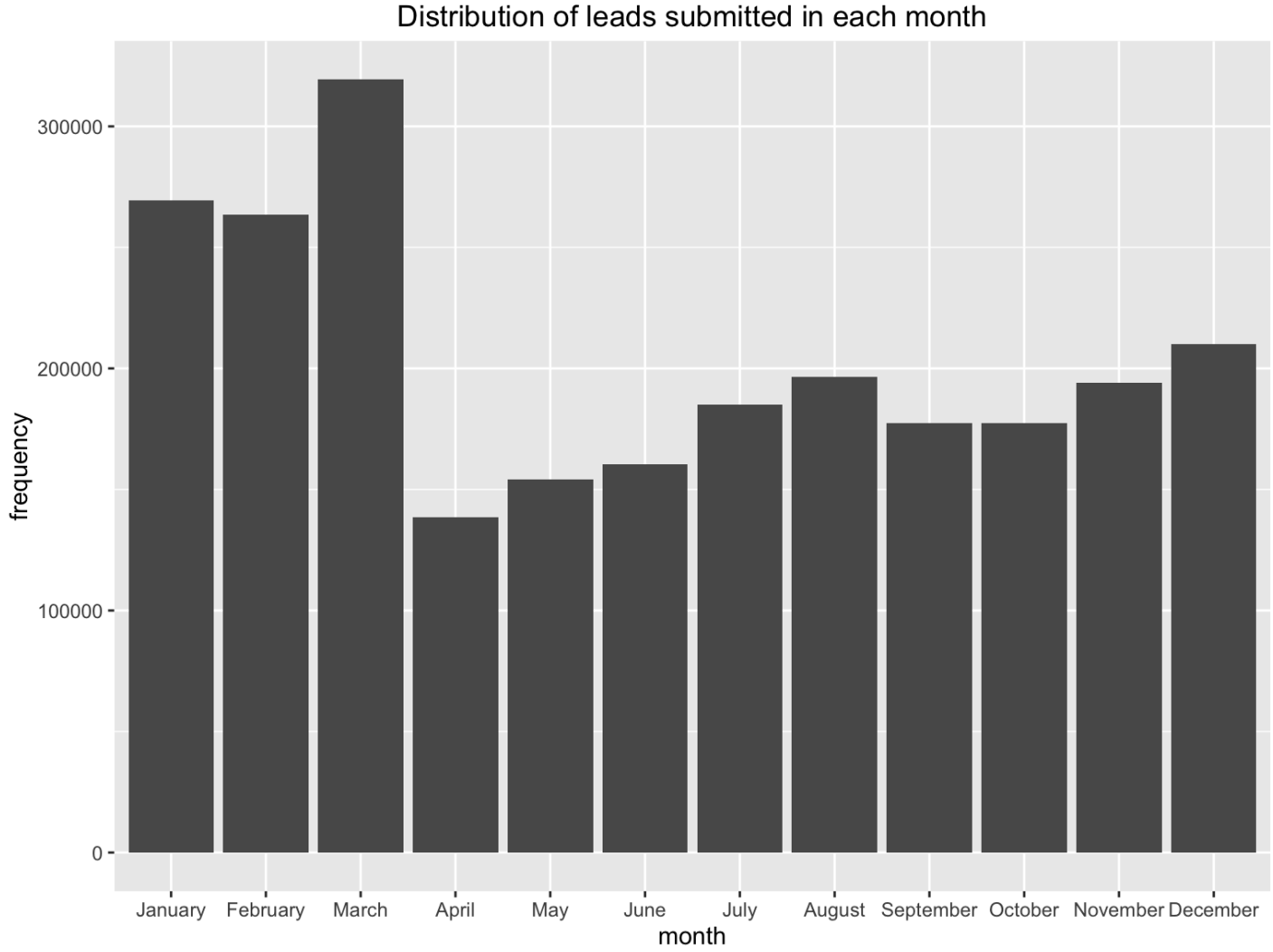
2. Replaced missing values with N/A by subsetting, and ensured that variables are in their recommeneded classes using 'type_convert'.

Dimensions of original Edmunds data: 2445924 observations and 24 variables

Dimensions of clean Edmunds data : 2445924 observations and 8 variables

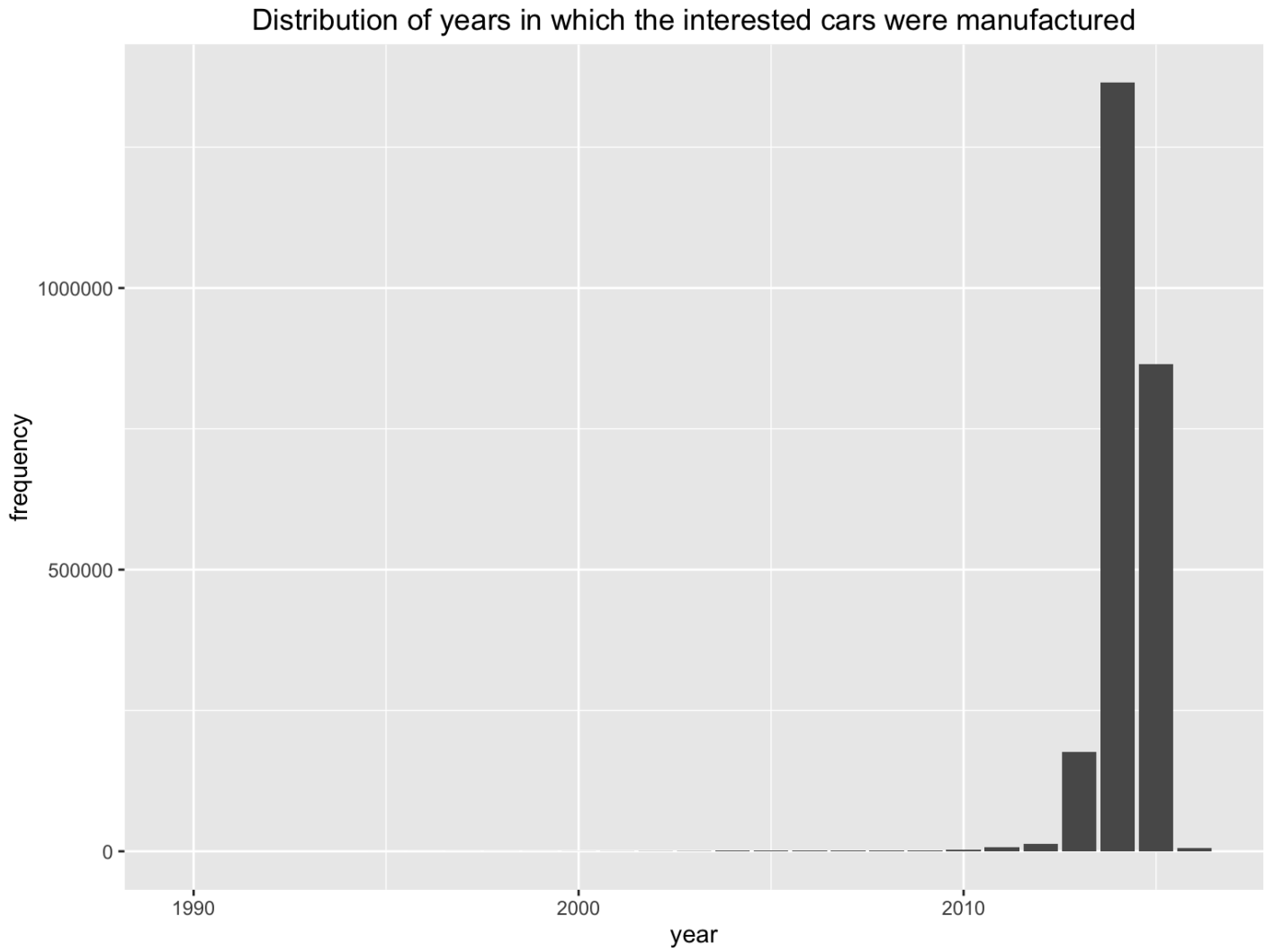
Summarizing Edmunds:

(1) Distribution of leads based on months:



The graph looks slightly right-skewed with the first three months getting the most number of leads. The spread seems to be centered around May while March gets the maximum number of leads.

(2) Distribution of model_year:



The graph, centered around 2014, looks heavily left-skewed with most of the cars being produced in the last two years. The maximum number of interested cars were manufactured in the year 2014.

(3) Tables of most and least popular manufacturers:

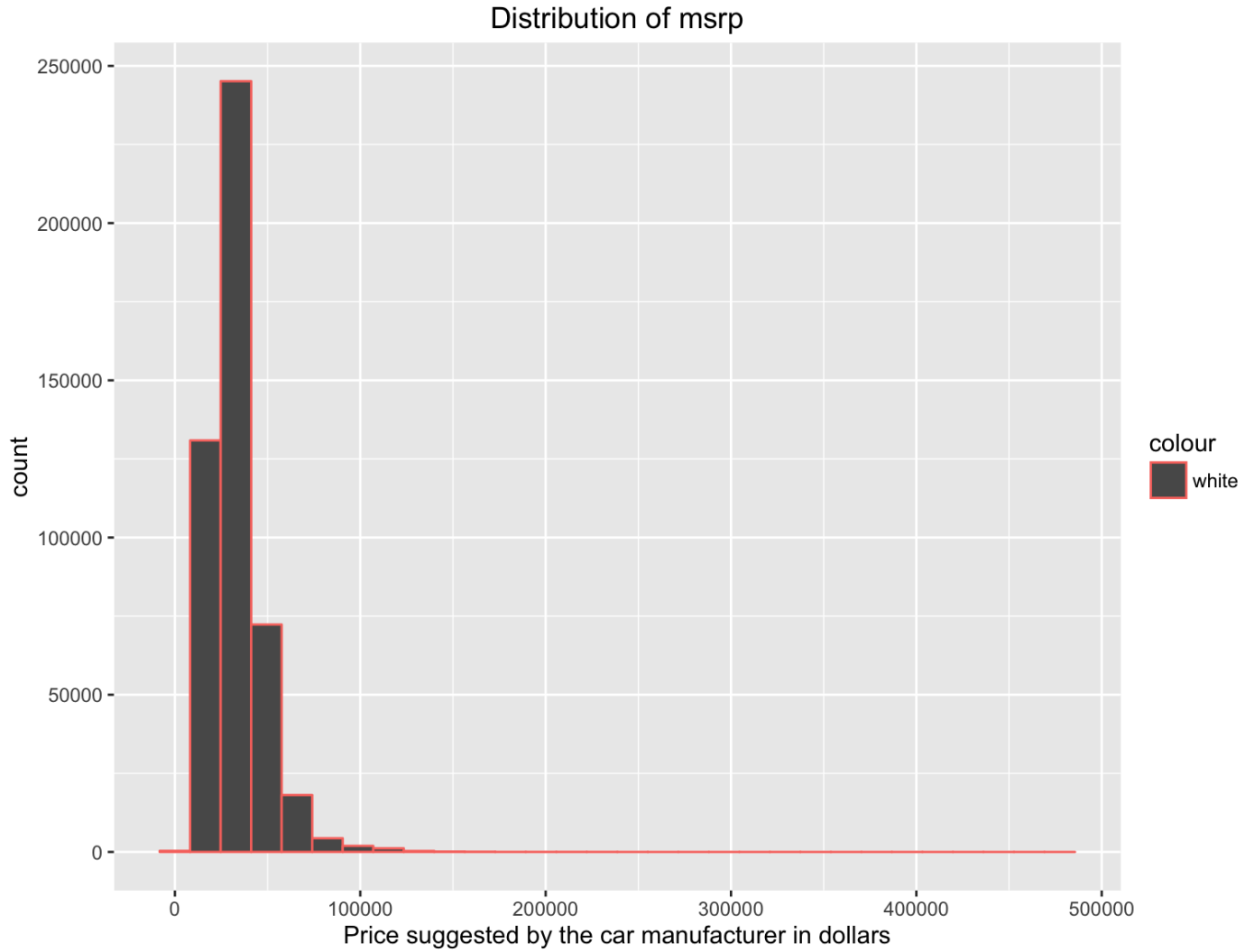
make	most_popular_manufacturers
honda	358908
toyota	353630
ford	144697
nissan	118091
subaru	106417
hyundai	81812

make	least_popular_manufacturers
geo	3
Isuzu	3

make	least_popular_manufacturers
Lotus	3
oldsmobile	3
isuzu	1
Oldsmobile	1

Honda was the most popular manufacturer while ‘Oldsmobile’ and ‘isuzu’ were the least popular.

(4) Distribution of msrp :



Summary of msrp:

mean_price	median	standard_deviation	IQR
33307.12	29910	14224.3	14649

With the distribution of msrp being heavily right-skewed, the typical value is given by the median and not the mean. So, typical value of msrp is 29910 dollars, and the spread varies by 14649 dollars (given by Inter Quartile Range).

IRS data:

This data, obtained from IRS tax returns in year 2014, tells us all about the different kinds of income and tax returns filed, based on different categories like filed as single, filed as married, number of dependents, tax exemptions, and more, from different Zip codes. So, each Zip code is an observation in this data file and the most important variables include number of tax returns filed, those filed a single, married, or head of household, number of dependents, gross income, and annual income. The original data contains number of variables and number of observations. The clean data contains number of variables and number of observations.

Cleaning IRS:

1. Retained required variables like N1, MARS1, MARS2, MARS4, NUMDEP, AOO1OO, AND A02650, by subsetting.

2. Gave retained variables more descriptive names, using names(), replaced missing values with NA, and ensured that variables are in their recommeneded classes.

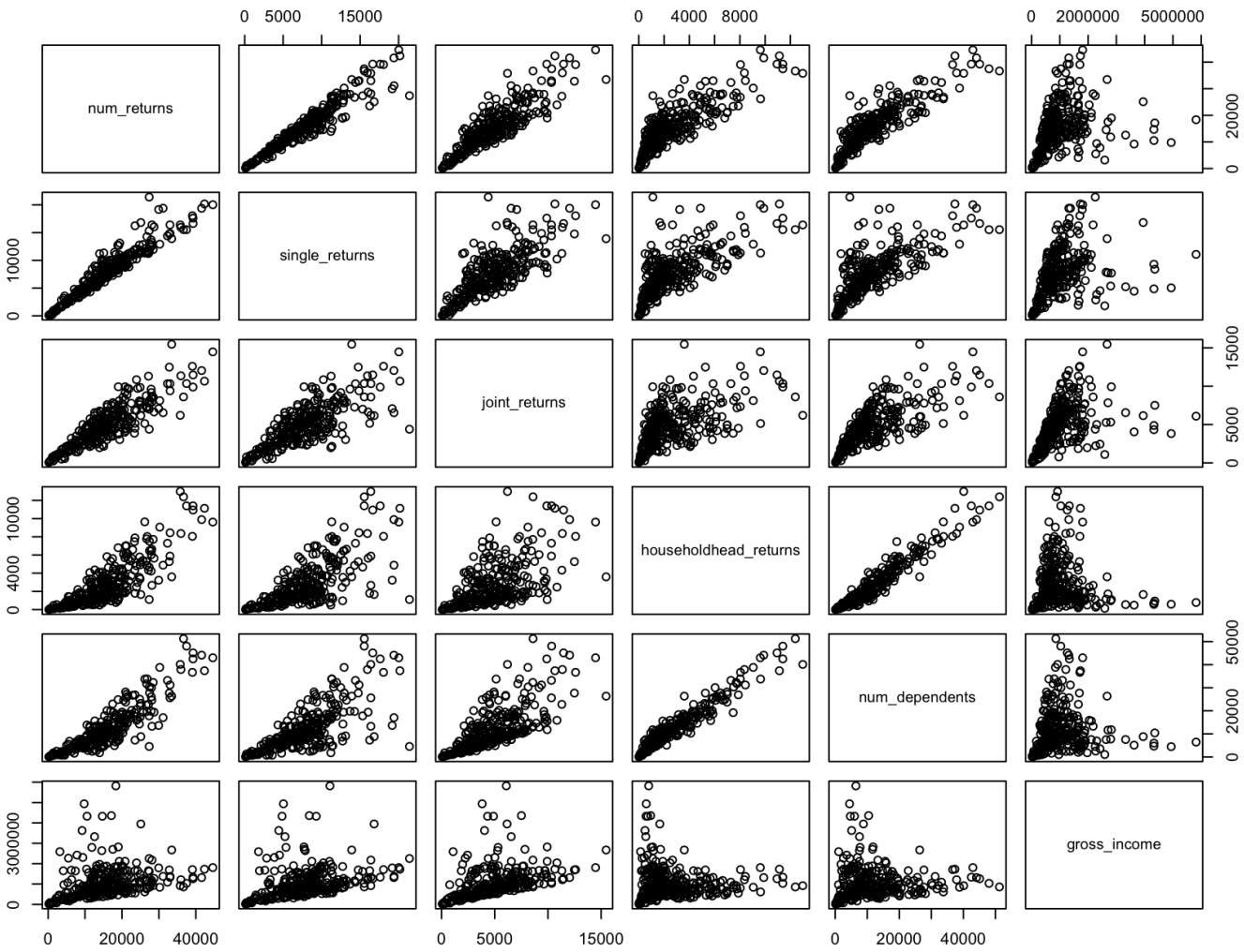
Dimensions of original IRS data : 288 observations and 111 variables

Dimensions of clean IRS data: 288 observations and 8 variables

Summarizing IRS:

The variables N1, MARS1, MARS2, MARS4, NUMDEP and A00100, which were renamed to num_returns, single_returns, joint_returns, household_returns, num_dependents, and gross_income, tell us about the number of returns filed, number of returns with filing status being “single”, “married”, and “head of household”, number of dependents, and the gross income of residents, in each zip code.

The correlation of these variables if given by :



Looking at the plot, it can be concluded that the variables are positively correlated to one another. For example, an increase in num_returns would mostly result in an increase in the other variables in the plot and vice versa. The same applies to all the other variables taken into consideration.

Analyzing IRS:

1. t-test on gross_income and annual_income:

I chose the adjusted gross income and the annual income variables to perform a t-test as I want to know if there is a need to retain both the variables. While the variables seem similar to one another, I would like to figure out if they are the same or if there are any confounding variables acting on them, by performing a t-test. By looking at their difference in means, with a 95% confidence interval, I hope to know if the variables are essentially the same and if one of them can be taken out so that the data set becomes easier to analayze for further research. Using a t-test is appropriate in this case because I am just hoping to know if the variables are the same and not to figure out the effect that one variable has on the other (we use linear regression for such a case).

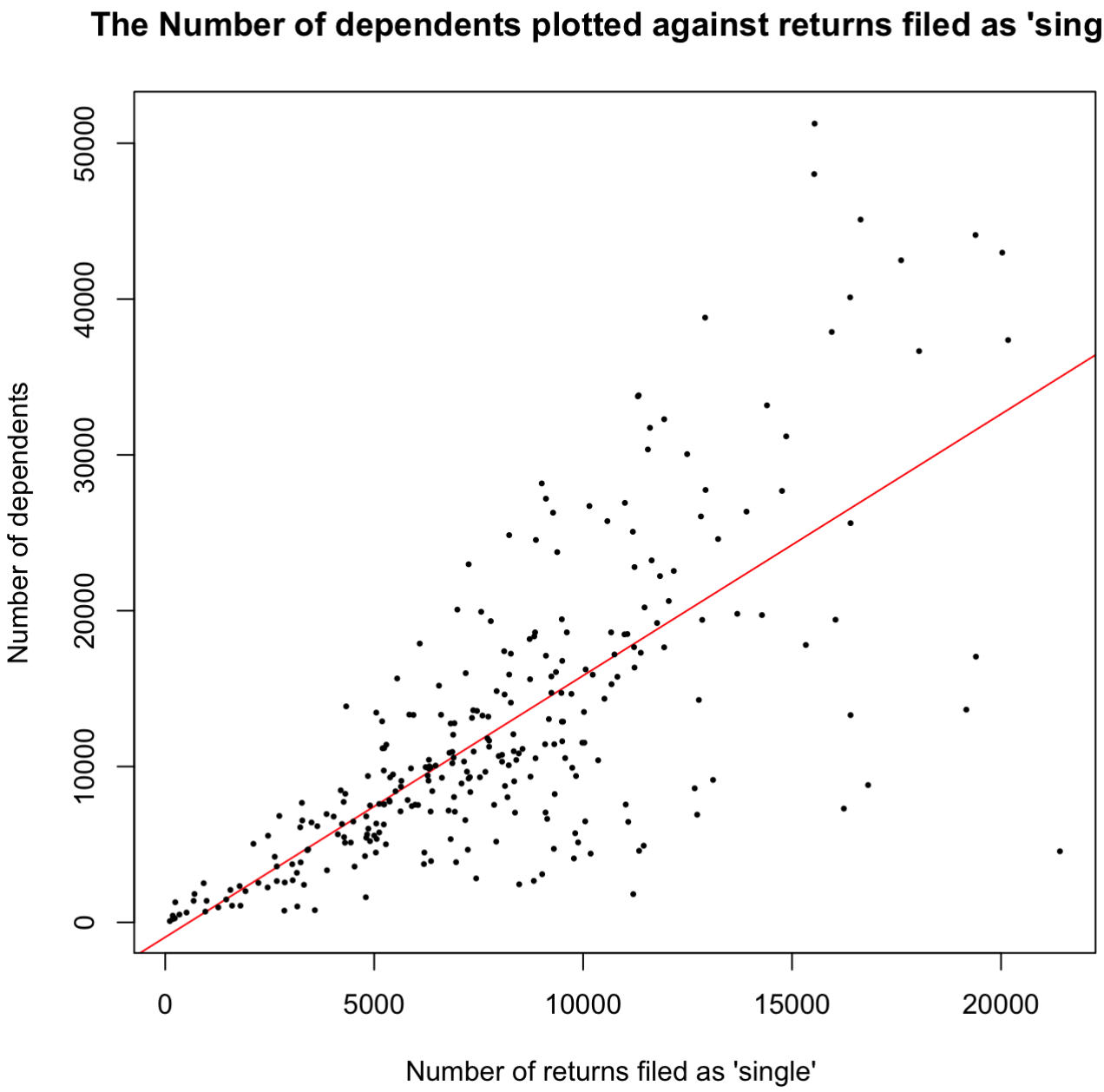
t.test results:

```
##
##  Welch Two Sample t-test
##
## data:  irs$gross_income and irs$annual_income
## t = -0.25547, df = 573.79, p-value = 0.7985
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -149364.0  114981.1
## sample estimates:
## mean of x mean of y
##   1065761   1082953
```

t-test Conclusion : We have obtained a negative t-statistic which suggests that the sample mean of gross_income is less than the sample mean of annual_income. But then, this statistic matters only when the obtained p-value is less than the mentioned alpha value (alpha = 1 - (confidence interval/100)). With the alpha being 0.05 and the obtained p-value being 0.7985, it can be conluded that the difference in means is 'not statistically signifacnt' ,i.e., the null hypothesis, that the true difference in means is equal to 0, is true! Thus, with the null hypothesis turning out to be true, I now know that the gross_income and annual_income varibales are "actually" same. I can confidently remove the annual_income variable from the data set and save space and time while conducting further analysis or research.

2. Linear regression on num_dependents and single_returns:

Creating a linear model that relates the number of dependents and the number of return filed as ‘single’ would help us clearly discern the effect that the former variable has on the latter:



```
##
## Call:
## lm(formula = num_dependents ~ single_returns)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -30433.2  -2372.4   -217.6   2191.9  26122.2
##
## Coefficients:
##                  Estimate Std. Error t value            Pr(>|t|)
## (Intercept)    -953.06755  837.58388  -1.138               0.256
## single_returns    1.67895    0.09425  17.814 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6558 on 286 degrees of freedom
## Multiple R-squared:  0.526,  Adjusted R-squared:  0.5243
## F-statistic: 317.3 on 1 and 286 DF,  p-value: < 0.00000000000000022
```

Looking at the graph, with a linear trend and a positive slope, it can be concluded that the 'number of returns filed as single' and 'number of dependents' variables are positively correlated. The slope coefficient tells us that for every one unit increase in the number of 'single' returns filed, the number of dependents increases by 1.679 times, and the R-squared value signifies that about 52.6 percent of the variation in the number of dependents may be explained by the other variable. While we cannot really state that the variables are strongly correlelated (in reference to R-squared value), it is safe to conclude that they are reasonably related to one another. Also, the presence of a lurking variable is highly doubted as the explonatory and the dependent variables exhibit a linear trendThe results are surprising as we would generally expect a decrease in the number of dependents when the number of returns filed as 'single' increases. It would be highly interesting to subject these variables to further research to discern the reason behind the unexpected results.

LA Cell Towers Data:

This data, sourced from HSIP Freedom Land_Mobile_private, gives out information regarding the cell towers located in various zip codes in Los Angeles. So, each cell tower (given by an OBJECTID) is an observation and the variables give out information about the location details of each tower such as its address, post_id, latitude and longitude, the kind of communication (like Land/Mobile) for which each tower is used, and the link from which the information was obtained.

Cleaning LA Cell Towers:

1. Retained variables like OBJECTID, city, state, ZIP, longitude, and latitude, and dropped the rest using their column numbers.

2. Ensured that retained variables are in their recommeneded classes. Note: Missing values were replaced with NA while reading the data in.

Dimensions of original data: 9248 observations and 22 variables

Dimensions of clean data: 9248 observations and 6 variables

Summarizing LA Cell Towers:

1. Tables showing the Zip Codes with most and least number of cell towers respectively:

ZIP	mostcelltowers
91042	398
90275	250
90012	248
91311	198
90045	160
91759	155

ZIP	leastcelltowers
93527	1
93545	1
93549	1
94513	1
95607	1
95670	1

While zipcode 91042 has the most number of cell towers, there seem to be quite a few zipcodes that have just one cell tower in them.

2. Names of the cities that are located in the zipcode with the most number of cell towers:

```
##  [1] "Tujunga"           "Los Angeles"       "Sunland"
##  [4] "Montrose"          "La Crescenta"      "La Cresenta"
##  [7] "Mount Lukens"      "Sylmar"            "La Canada"
## [10] "Lacrescenta"       "Burbank"           "Highway Highlands"
## [13] "Mt Lukens"         "Valencia"          "Pasadena"
## [16] "7920 Sunset Blvd"  "Montrous"          "Tijunga"
```

Custom Function:

```
x <- c()
zipit <- function(x){
   #Using if/else to notify the user about no input.
   if(missing(x)) {
     print("Error in Zip Codes input")
   }
  else{
     # Looking for 'x' (by filtering) in each data set and then summarizing the required
 information using          summarize()
    ntowers <-  la_towers %>%
    filter( ZIP %in% x) %>%
    group_by(ZIP) %>%
    summarize(num_cell_towers = n()) %>%
    rename(zip = ZIP) #Renaming to make it easier while joining data sets

nreturns <-  irs %>%
    filter(zip %in% x) %>%
    group_by(zip) %>%
    mutate(combined_returns = single_returns + joint_returns + householdhead_returns) %
>%
    #Creating a new variable that outputs the required information
    summarize(total_filed_tax_returns = combined_returns)


nleads <- edmunds %>%
     filter(dealer_zip %in% x) %>%
    group_by(dealer_zip) %>%
    summarize(num_car_leads = n()) %>%
    rename(zip = dealer_zip)

  return(full_join (
            full_join(nleads, ntowers, by= "zip"),
                                nreturns, by = "zip"))
#Joining all the three obtained data sets and returning it to user
  }
}
```

Examples that show that ‘zipit’ works:

1. zipit()

```
## [1] "Error in Zip Codes input"
```

2. zipit(c(90001, 90095, 64055))

```
## # A tibble: 3 × 4
##     zip num_car_leads num_cell_towers total_filed_tax_returns
##   <dbl>         <int>           <int>                   <dbl>
## 1 64055           180              NA                      NA
## 2 90001           736              14                   20920
## 3 90095            NA              14                      NA
```

3. zipit(7890)

```
## # A tibble: 0 × 4
## # ... with 4 variables: zip <dbl>, num_car_leads <int>,
## #   num_cell_towers <int>, total_filed_tax_returns <lgl>
```

Thus, the function displays an error when there is no input, outputs a table with the required information if the zipcodes match with those in the data files, and just gives out an empty table if the zipcodes do not match.

Joining Data Files:

1. Appending the information in 'irs-la-zip.xls' to the data in 'edmunds.dta' and naming it 'edm_irs' :

2. Combining the data in 'edm_irs' and 'LA Cell towers' by the Zip code that has the most number of cell towers :

Dimensions of edm_irs1 :

```
## [1]  0 16
```

Thus, there are zero observations left after semi-joining.

Map that shows the location of towers: