Question 1

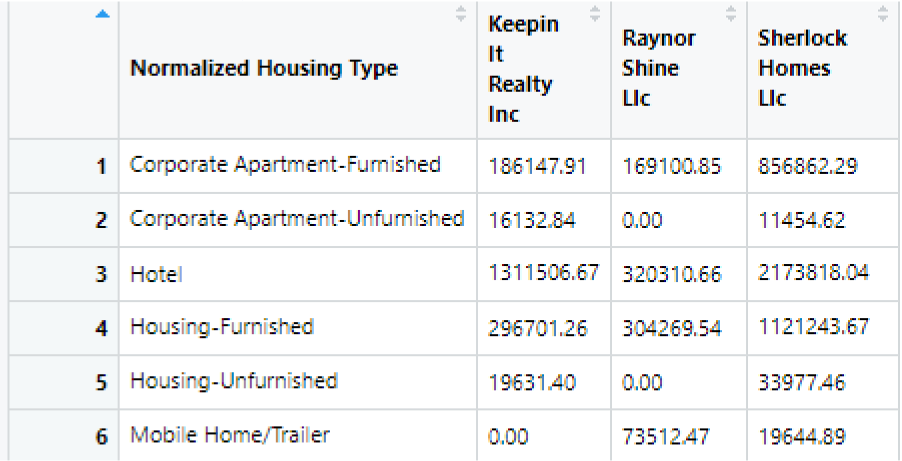
```
> # Read in the data from the ACME Corp Spreadsheet
> library('readxl')
> file <-"F:/Assignments/DPA/Assigmnents_3/ACME_Corp.xlsx"
> df <- read_excel(file, sheet = "Sheet1")
> df <- as.data.frame(df)
> # 2 points
> # 1. The three vendors each use a different definition of housing type. However, ACME's
official types
> # are listed on Sheet2 of the Excel sheet.
> # Create a new column called 'Normalized Housing Type' based on the standardized mappin
g.
> sheet2 <- read_excel(file, sheet = "Sheet2")
> df$`Normalized Housing Type` <- sheet2$`Clean Value`[match(df$`Housing Type (Condo, Hot
el, Apartment, Single Family Home)`, sheet2$`Lookup Value`)]
> head(df)
                Vendor Current Adjuster Claim Number Policyholder Last Name
1 Keepin It Realty Inc  Kristina Burkey      273132N                Chapman
2 Keepin It Realty Inc  Kristina Burkey   2015144364               Castillo
3 Keepin It Realty Inc    Laurie Stover      275813N                 Picard
4 Keepin It Realty Inc    Vanessa Vyles   2015147135                 Jansen
5 Keepin It Realty Inc    Cynthia Poppe   2015148130                  Black
6 Keepin It Realty Inc   Jennie Prewitt   2015149053                Guevara
  Policyholder City Policy holder State
1          Milltown                  MT
2     Grand Prairie                  TX
3           Lincoln                  NE
4       San Antonio                  TX
5    Willingborough                  NJ
6        Sacramento                  CA
  Housing Type (Condo, Hotel, Apartment, Single Family Home) Move-in/Check-In Date
1                              Single Family Home-Furnished            2015-01-20
2                                       Apartment-Furnished            2015-01-20
3                                       Apartment-Furnished            2015-02-02
4                             Corporate Apt/Condo-Furnished            2015-02-12
5                                       Apartment-Furnished            2015-03-09
6                                       Apartment-Furnished            2015-03-25
  Move-out/Check-Out Date Occupancy Status # of\r\nBedrooms # of\r\nBaths # Days
1              2015-04-28        Moved Out                4             2     99
2              2015-06-19        Moved Out                2             1    151
3              2016-04-01         Occupied                2             2    425
4              2015-03-11        Moved Out                2             2     28
5              2015-06-08        Moved Out                2             2     92
6              2015-11-16        Moved Out                2             2    237
  Daily Housing Rate Daily Admin Fee Total Housing Spend Total Admin Spend
1           61.51313        7.927273             6089.80             784.8
2           76.03311        9.602649            11481.00            1450.0
3           65.16179        7.905882            27693.76            3360.0
4          118.29964        6.214286             3312.39             174.0
5          113.65478        9.456522            10456.24             870.0
6           69.70228        7.827848            16519.44            1855.2
        Normalized Housing Type
1             Housing-Furnished
2 Corporate Apartment-Furnished
3 Corporate Apartment-Furnished
4 Corporate Apartment-Furnished
5 Corporate Apartment-Furnished
6 Corporate Apartment-Furnished
```

Question 2

```
> houseSpendPolicyState<-sort(tapply(df$`Total Housing Spend`, INDEX = df$`Policy holder
State`, FUN = sum), decreasing = TRUE)
> percentageHSpolicy<- houseSpendPolicyState/sum(houseSpendPolicyState)
> SpendPolicyState_df<- data.frame(houseSpendPolicyState,percentageHSpolicy)
> head(SpendPolicyState_df)
   houseSpendPolicyState percentageHSpolicy
CA             1748342.6         0.25285841
TX             1234807.7         0.17858714
GA              406756.2         0.05882812
NC              280265.5         0.04053409
MD              251564.0         0.03638307
VA              231620.7         0.03349872
```

Question 3

```
> # Load the library
> library(reshape2)
> new_df<- data.frame(df$Vendor,df$`Normalized Housing Type`, df$`Total Housing Spend`)
> names(new_df)[1]<- "Vendor"
> names(new_df)[2]<- "Normalized Housing Type"
> names(new_df)[3]<- "Total Housing Spend"
> # Cast the library into wide format
> table_df <- dcast(new_df, `Normalized Housing Type` ~ Vendor, fun.aggregate = sum, valu
e.var = "Total Housing Spend")
> view(table_df)
```



	Normalized Housing Type	Keepin It Realty Inc	Raynor Shine Llc	Sherlock Homes Llc
1	Corporate Apartment-Furnished	186147.91	169100.85	856862.29
2	Corporate Apartment-Unfurnished	16132.84	0.00	11454.62
3	Hotel	1311506.67	320310.66	2173818.04
4	Housing-Furnished	296701.26	304269.54	1121243.67
5	Housing-Unfurnished	19631.40	0.00	33977.46
6	Mobile Home/Trailer	0.00	73512.47	19644.89

Question 4

```
> # 4. Obtain top 20 most frequent Policy holder City and Policy holder State combos
> combos<- paste(df$`Policyholder City`, df$`Policy holder State`, sep = ', ')
> top20<-head(sort(table(combos), decreasing = TRUE), 20)
> print(top20)
combos
       Houston, TX     San Antonio, TX   Indianapolis, IN     Fort Worth, TX 
                33                  21                 15                 14 
       Phoenix, AZ         Atlanta, GA Virginia Beach, VA      Bremerton, WA 
                14                  12                 12                 11 
        Dallas, TX          Tucson, AZ        Pearland, TX        Raleigh, NC 
                11                  11                 10                  9 
       Fontana, CA         Hampton, GA   Jurupa Valley, CA      Las Vegas, NV 
                 8                   8                  8                  8 
   Los Angeles, CA        Townsend, DE       Charlotte, NC           Cobb, CA 
                 8                   8                  7                  7 
```

Question 5

```
> # 5. Write a function obtains the lat lon for a given city and state
> # Note: You'll propsefully need to do some research on how to obtain this.
> # There are a few ways of doing this.
> key <- 'AIzaSyBf1Md3BLean7Ox_ldHdQwWogCyRY3UhzE'
> register_google(key = key)
> cityStateLatLon <- function(cityStat){
+   return(geocode(cityStat))
+ }
```

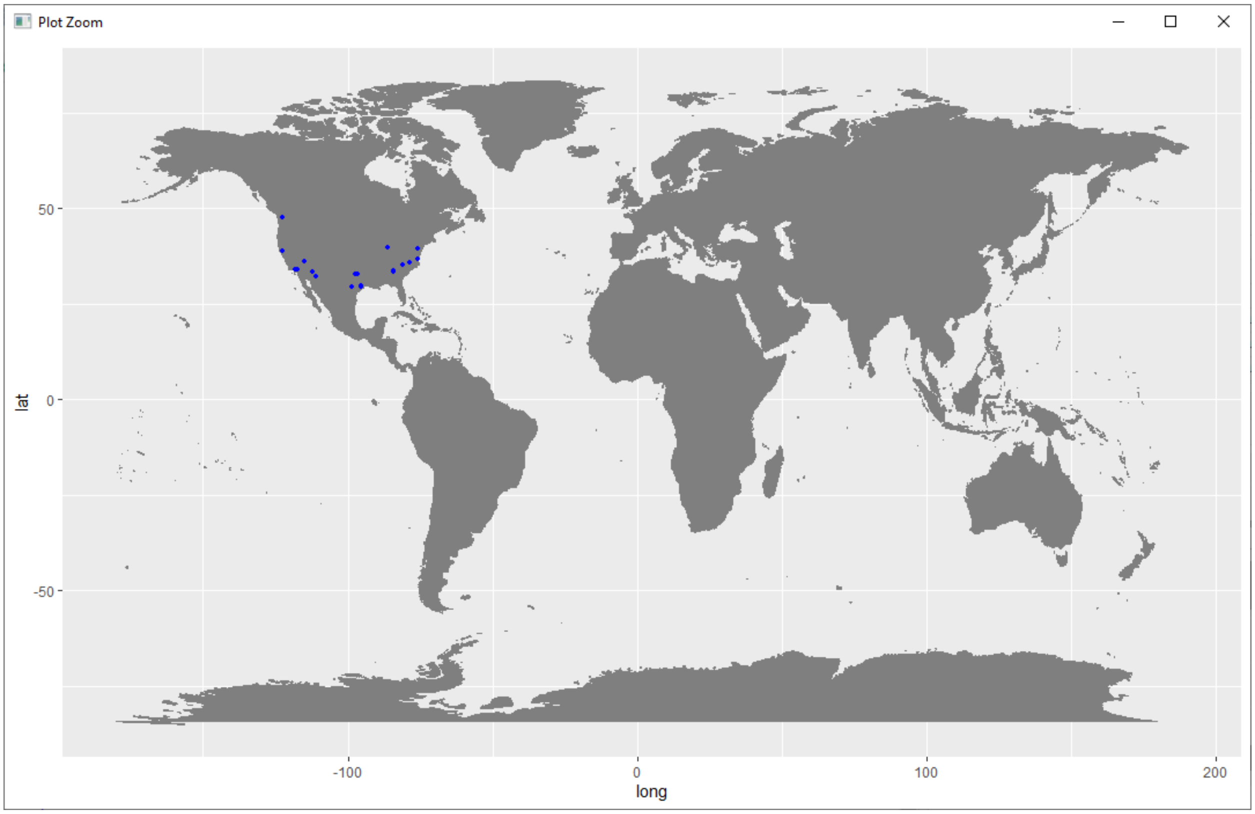
Question 6

```
> citystatescombos<-names(top20)
> cityStateLatLon_df <- NULL
> for (i in citystatescombos){
+   cityStateLatLon_df <-rbind(cityStateLatLon_df,data.frame(cityStateLatLon(i)))
+ }
> cityStateLatLon_df<-cbind(data.frame(citystatescombos),cityStateLatLon_df)
> cityStateLatLon_df
```

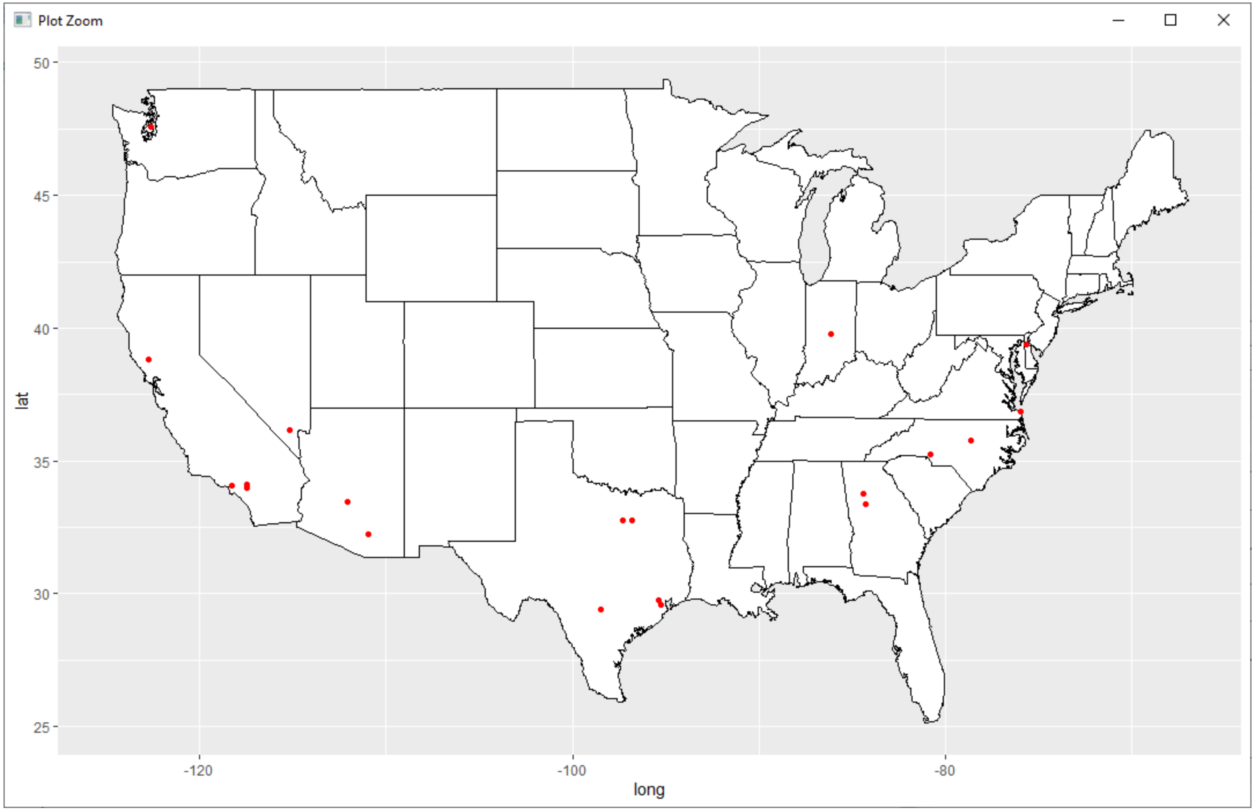
```
     citystatescombos        lon      lat
1         Houston, TX  -95.36980 29.76043
2     San Antonio, TX  -98.49363 29.42412
3    Indianapolis, IN  -86.15807 39.76840
4      Fort Worth, TX  -97.33077 32.75549
5         Phoenix, AZ -112.07404 33.44838
6         Atlanta, GA  -84.38798 33.74900
7  Virginia Beach, VA  -75.97798 36.85293
8       Bremerton, WA -122.62698 47.56501
9          Dallas, TX  -96.79699 32.77666
10         Tucson, AZ -110.97471 32.22261
11       Pearland, TX  -95.28605 29.56357
12        Raleigh, NC  -78.63818 35.77959
13        Fontana, CA -117.43505 34.09223
14        Hampton, GA  -84.28298 33.38706
15  Jurupa Valley, CA -117.48548 33.99720
16      Las Vegas, NV -115.13983 36.16994
17    Los Angeles, CA -118.24368 34.05223
18       Townsend, DE  -75.69160 39.39511
19      Charlotte, NC  -80.84313 35.22709
20           Cobb, CA -122.72096 38.83346
```

Question 7

```
library(maptools)
library(maps)
library(ggplot2)
#On World Map
> mp <- NULL
> mapWorld <- borders("world", colour="gray50", fill="gray50") # create a layer of border
s
> mp <- ggplot() +   mapWorld
> #Now Layer the cities on top
> mp <- mp+ geom_point(aes(x=cityStateLatLon_df$lon,y=cityStateLatLon_df$lat) ,color="blu
e", size=1)
> mp
```



```
> #On USA Map
> m = map_data('state')
> ggplot()+geom_polygon( data=m, aes(x=long, y=lat,group=group),colour="black", fill="whi
te" )+geom_point(data=cityStateLatLon_df,aes(x=cityStateLatLon_df$lon ,y=cityStateLatLon_
df$lat),colour="red",)
```



Question 8

```
> # 4 points
> # 8. There are some misspellings and other issues
> # with the "Current Adjuster" field. Leverage the text
> # analysis tools and levenstein distance to clean up
> # the names properly. Put them into a new column called
> # "Current Adjuster Cleaned"
> # Hint: you must deal with issues of case, whitespace,
> # ,name misspellings and common name differences (ie Dave vs David).
> # You will be graded on how well you complete this.
> library(stringdist)
> allUpper <- toupper(df$`Current Adjuster`)
> unvalidname<-(!grepl("^[a-zA-Z]",allUpper))
> sum(unvalidname)#check no of invalid names thoes contain alphanumeric
[1] 0
> allUniques<-unique(allUpper)
> worddistance<-NULL
> worddistance<-stringdistmatrix(allUniques, allUniques, method = 'lv', useNames = "strin
gs") #similar word distance
> worddistance<-subset(melt(worddistance), value>0 & value<5)
> orderedwords <- worddistance[order(worddistance$value, decreasing = FALSE),]

> orderedwords
                  Var1              Var2 value
746        IRA DOBBINS      IRA  DOBBINS     1
4787  SUSAN CHAMBERLIN SUSAN CHAMBERLAIN     1
6141      IRA  DOBBINS       IRA DOBBINS     1
6945 SUSAN CHAMBERLAIN  SUSAN CHAMBERLIN     1
2928    JOSHUA HURLEY       JOSH HURLEY     2
5999      JOSH HURLEY     JOSHUA HURLEY     2
3696    RONALD CROWDER       RON CROWDER     3
7016       RON CROWDER    RONALD CROWDER     3
3085       LYNN HARVEY   LYNNETTE HARVEY     4
4869      TERESA SMITH       TRACY SMITH     4
5077   LYNNETTE HARVEY       LYNN HARVEY     4
6778       TRACY SMITH      TERESA SMITH     4
```

```
# Var1              Var2 value
# 4787  SUSAN CHAMBERLIN SUSAN CHAMBERLAIN     1
# 6141      IRA  DOBBINS       IRA DOBBINS     1
# 5999       JOSH HURLEY     JOSHUA HURLEY     2
# 7016       RON CROWDER    RONALD CROWDER     3
# 3085       LYNN HARVEY   LYNNETTE HARVEY     4
> realNames<- sapply(df$`Current Adjuster`, function(name) switch(name,
+                                                      'SUSAN CHAMBERLIN' = 'SUSAN CHAMBE
RLAIN',
+                                                      'IRA  DOBBINS' = 'IRA DOBBINS',
+                                                      'JOSH HURLEY' = 'JOSHUA HURLEY',
+                                                      'RON CROWDER' = 'RONALD CROWDER',
+                                                      'LYNN HARVEY' = 'LYNNETTE HARVEY',
name))
> df[,"Current Adjuster Cleaned"] <-realNames
> head(df)
                Vendor Current Adjuster Claim Number Policyholder Last Name
1 Keepin It Realty Inc  Kristina Burkey      273132N                Chapman
2 Keepin It Realty Inc  Kristina Burkey   2015144364               Castillo
3 Keepin It Realty Inc    Laurie Stover      275813N                 Picard
4 Keepin It Realty Inc    Vanessa Vyles   2015147135                 Jansen
5 Keepin It Realty Inc    Cynthia Poppe   2015148130                  Black
6 Keepin It Realty Inc   Jennie Prewitt   2015149053                Guevara
  Policyholder City Policy holder State
1          Milltown                  MT
2     Grand Prairie                  TX
3           Lincoln                  NE
4       San Antonio                  TX
5    Willingborough                  NJ
6        Sacramento                  CA
  Housing Type (Condo, Hotel, Apartment, Single Family Home) Move-in/Check-In Date
1                               Single Family Home-Furnished            2015-01-20
2                                        Apartment-Furnished            2015-01-20
3                                        Apartment-Furnished            2015-02-02
4                              Corporate Apt/Condo-Furnished            2015-02-12
5                                        Apartment-Furnished            2015-03-09
6                                        Apartment-Furnished            2015-03-25
  Move-out/Check-Out Date Occupancy Status # of\r\nBedrooms # of\r\nBaths # Days
1              2015-04-28        Moved Out               4            2     99
2              2015-06-19        Moved Out               2            1    151
3              2016-04-01         Occupied               2            2    425
4              2015-03-11        Moved Out               2            2     28
5              2015-06-08        Moved Out               2            2     92
6              2015-11-16        Moved Out               2            2    237
  Daily Housing Rate Daily Admin Fee Total Housing Spend Total Admin Spend
1           61.51313        7.927273             6089.80             784.8
2           76.03311        9.602649            11481.00            1450.0
3           65.16179        7.905882            27693.76            3360.0
4          118.29964        6.214286             3312.39             174.0
5          113.65478        9.456522            10456.24             870.0
6           69.70228        7.827848            16519.44            1855.2
        Normalized Housing Type Current Adjuster Cleaned
1             Housing-Furnished          Kristina Burkey
2 Corporate Apartment-Furnished          Kristina Burkey
3 Corporate Apartment-Furnished            Laurie Stover
4 Corporate Apartment-Furnished            Vanessa Vyles
5 Corporate Apartment-Furnished            Cynthia Poppe
6 Corporate Apartment-Furnished           Jennie Prewitt
```

Question 9

```
> library(dplyr)
> n = 3
> state = "CA"
> date = '2015-03'
> reportParameter <-function(n, state, date){
+   temp_df<-NULL
+   temp_df<-df[which(df$`Policy holder State` == state & substr(df$`Move-in/Check-In Dat
e`,1,7) == date), ]
+   report_df<- temp_df %>% group_by(temp_df$`Current Adjuster Cleaned`, temp_df$`Occupan
cy Status`) %>% count()
+   names(report_df)[1] <- "Adjuster"
+   names(report_df)[2] <- "Occupancy"
+   report_df<-data.frame(dcast(report_df, Adjuster ~ Occupancy, fun.aggregate = sum, val
ue.var = 'n'))
+   report_df$Total <- report_df$Checked.Out + report_df$Moved.Out
+   report_df <- report_df[order(report_df$Total, decreasing = TRUE),]
+   return(print(head(report_df,n),row.names = FALSE))
+
+ }
> reportParameter(n,state,date)

       Adjuster Checked.Out Moved.Out Total
 Jennie Prewitt           2         1     3
 Larry Callahan           3         0     3
   Brett Munsey           2         0     2
```