EDA - Survey Monkey

Contents

The EDA has been mainly broken down into 3 parts based on the questions that has been asked. They are:

1.) Housing and neighbourhood section 2.) Retailers in use 3.) Fitness based questions 4.) General questions

(1.) Housing and Neighbourhood section

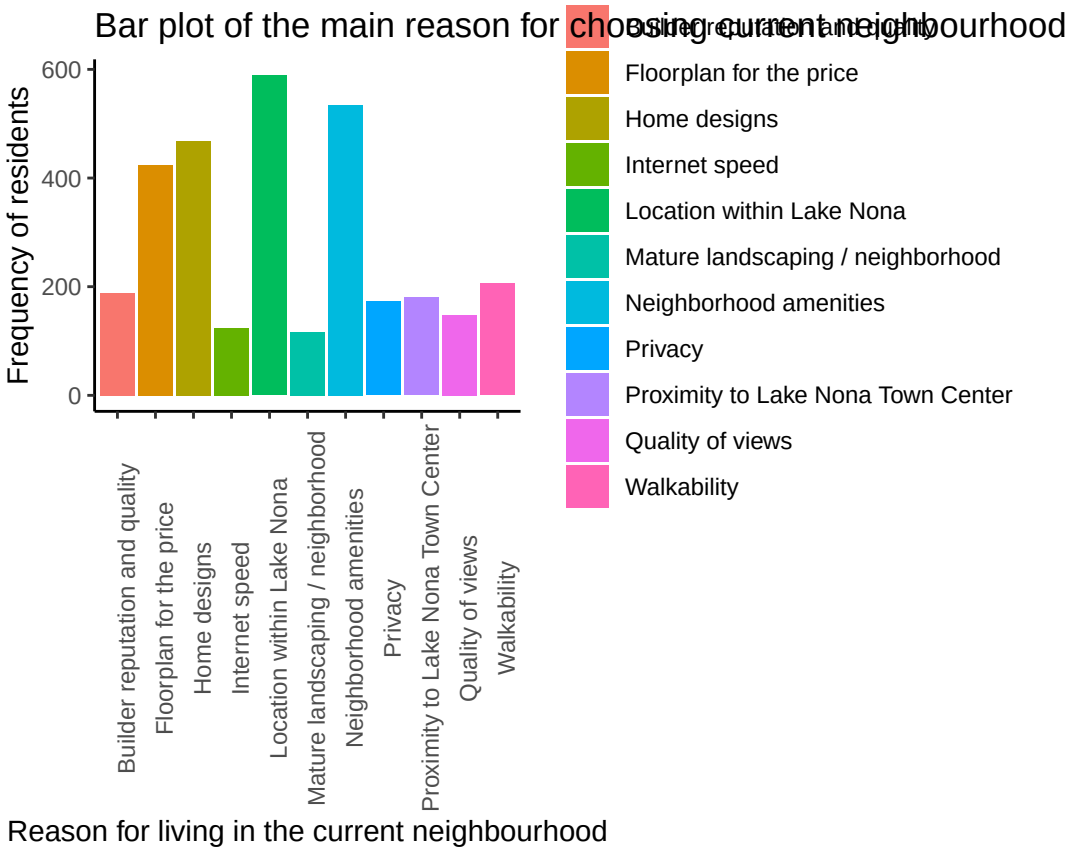
This sections has the EDA done for the housing and neighbourhood related questions of the survey

1.) The primary reason for living in this neighbourhood

The top 11 reasons for the reasons stated by the residents as primary reasons to live in Lake Nona si visualized here. This can be utiloized to directly improve the resident retention rate while attracting new residents to Lake Nona.

```
##------Reason for living in the current neighbourhood
nbhRsn <- data %>% select("ID","primary.reason.you.chose.to.live.in.neighborhood")
# sum(is.na(nbhRsn$primary.reason.you.chose.to.live.in.neighborhood))

#Getting the long form
nbhRsn_long <- nbhRsn %>%
  separate_rows(primary.reason.you.chose.to.live.in.neighborhood, sep = ",", convert = TRUE) %>%
  drop_na()

# sum(is.na(nbhRsn_long$primary.reason.you.chose.to.live.in.neighborhood))

#Dropping the rows with either NAs or empty string values ("")
nbhRsn_long <- nbhRsn_long[!(is.na(nbhRsn_long$primary.reason.you.chose.to.live.in.neighborhood) | nbhRs

#Getting the grouped by ID
#First convert to factor and then do this
nbhRsn_long$primary.reason.you.chose.to.live.in.neighborhood <- as.factor(nbhRsn_long$primary.reason.you

cnt_nbhRsn <- nbhRsn_long %>%
```

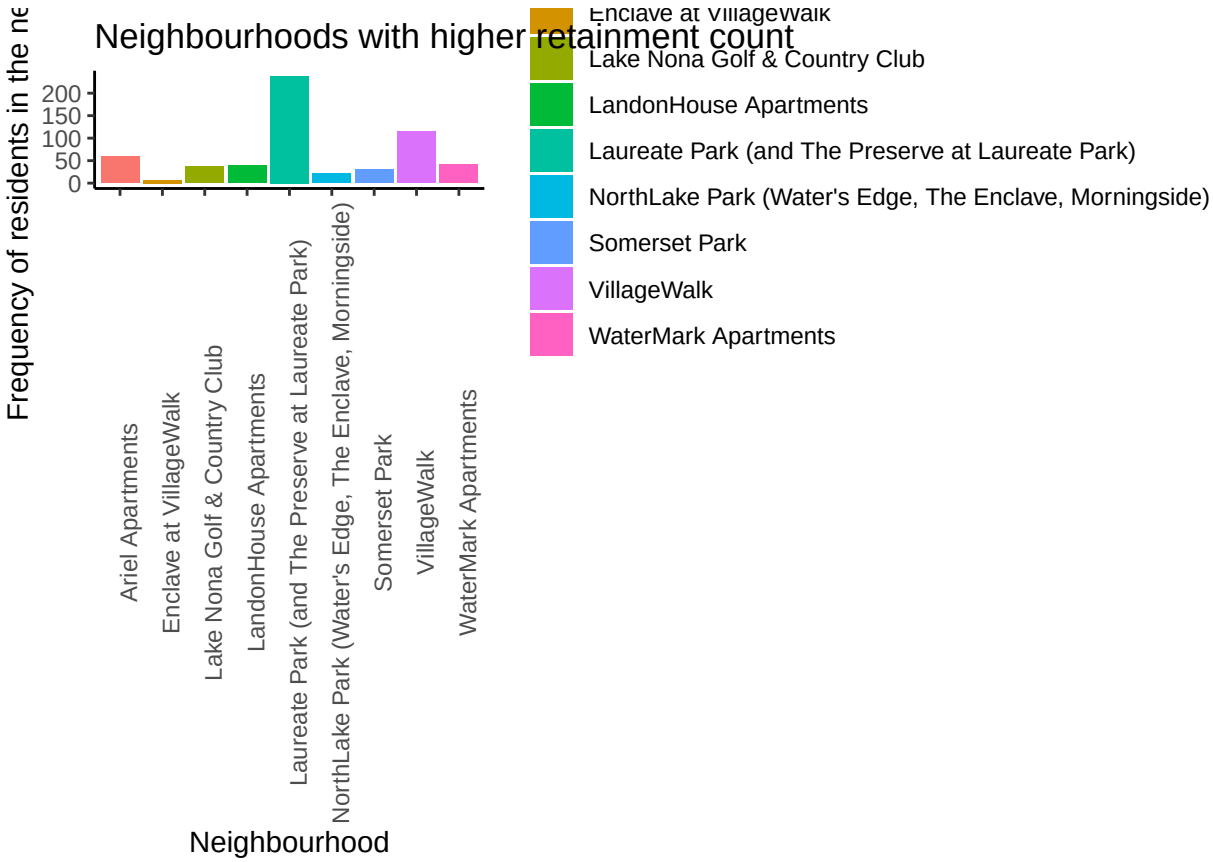
```
  dplyr::group_by(primary.reason.you.chose.to.live.in.neighborhood) %>%
  dplyr::summarise(count = n())

#Getting the most important reasons out of these
cnt_nbhRsn100 <- cnt_nbhRsn %>% filter(count>100)

#Plotting in a graph
ggplot(cnt_nbhRsn100,aes(x = primary.reason.you.chose.to.live.in.neighborhood,
           y = count,
           fill = primary.reason.you.chose.to.live.in.neighborhood)) +
  geom_bar(stat = "identity") +
  theme_classic() +
  labs(
    x = "Reason for living in the current neighbourhood",
    y = "Frequency of residents",
    title = paste("Bar plot of the main reason for choosing current neighbourhood"),
    fill = "Main reasons for choosing current neighbourhood"
  )+
  theme(axis.text.x = element_text(angle = 90))
```



1.1.) The location of the residents who have stated as the reason to live in Lake Nona

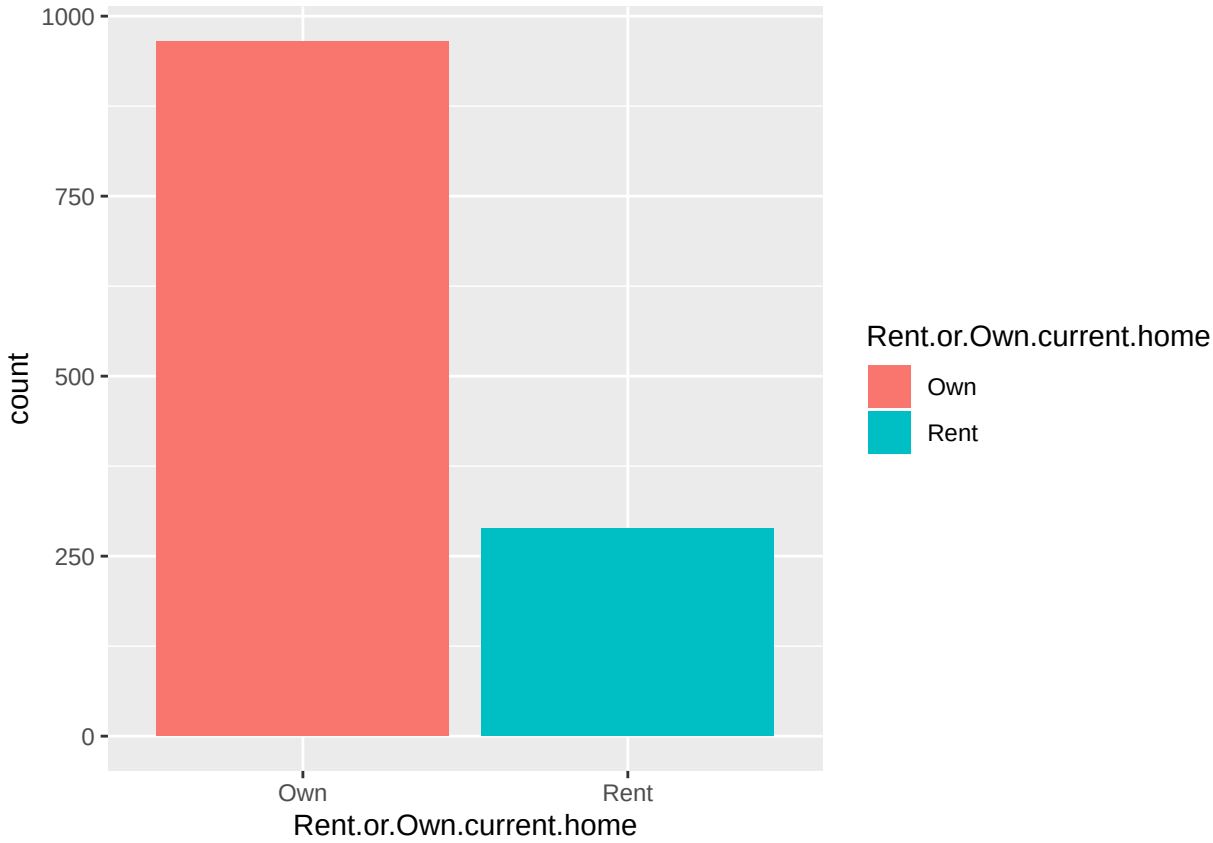
Out of the above, many have said that its the location within Lake Nona. Thus looking into the locations of those who said its this

```r
locdata <- data %>% select("ID",
                           "primary.reason.you.chose.to.live.in.neighborhood" ,
                           "Neighborhood.within.LakeNona")

#Getting the long form
locLong <- locdata %>%
  separate_rows(primary.reason.you.chose.to.live.in.neighborhood, sep = ",", convert = TRUE) %>%
  drop_na()

#Filtered to get those who said their location within lake nona is the reason to live in that neighbour
ftdLocLong <- locLong %>% filter(primary.reason.you.chose.to.live.in.neighborhood=="Location within Lak

#Now getting the count of each such neighbourhoods
nbh <- ftdLocLong %>%
  dplyr::group_by(Neighborhood.within.LakeNona) %>%
  dplyr::summarise(count = n())

#Plotting in a graph
ggplot(nbh,aes(x = Neighborhood.within.LakeNona,
               y = count,
               fill = Neighborhood.within.LakeNona)) +
  geom_bar(stat = "identity") +
  theme_classic() +
  labs(
    x = "Neighbourhood",
    y = "Frequency of residents in the neighbourhood",
    title = "Neighbourhoods with higher retainment count",
    fill = "Neighbourhoods influencing stay at Lake Nona"
  ) +
  theme(axis.text.x = element_text(angle = 90))
```



2.) The rent or own situation of the current house

The residents have either rented their current residence or bought them (own) directly. This occurence is displayed below.



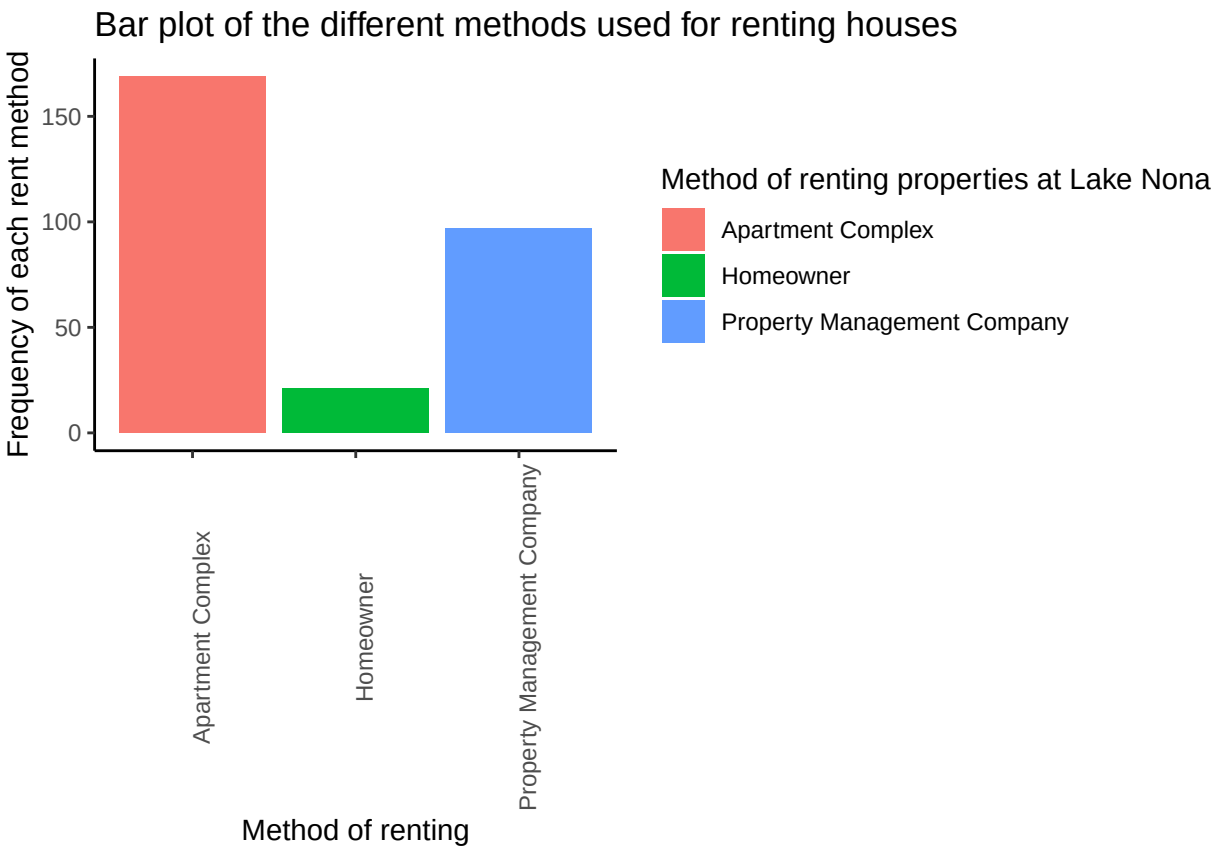
It can be seen that many have purchased their houses directly whereas a significant few tends to rent out their residence.

2.1.) The renting method used

Out of those who have rented out, they have the option to either gop through a property management company or otherwise. This behaviour is analysed below.

```
#Selecting the rented houses
rentd <- rentOwn %>% filter(Rent.or.Own.current.home=="Rent")

#Dropping the rows with either NAs or empty string values ("")
rentd <- rentd[!(is.na(rentd$Do.you.rent.your.home.through.a.property.management.company.or.through.the

#Getting the method through which each house was rented
rentMtd <- rentd %>%
  dplyr::group_by(Do.you.rent.your.home.through.a.property.management.company.or.through.the.homeowner.
  dplyr::summarise(count = n())

#Plotting in a graph
ggplot(rentMtd,aes(x = Do.you.rent.your.home.through.a.property.management.company.or.through.the.homeo
               y = count,
               fill = Do.you.rent.your.home.through.a.property.management.company.or.through.the.homeow
  geom_bar(stat = "identity") +
  theme_classic() +
  labs(
```

```
    x = "Method of renting",
    y = "Frequency of each rent method",
    title = paste("Bar plot of the different methods used for renting houses"),
    fill = "Method of renting properties at Lake Nona"
  ) +
  theme(axis.text.x = element_text(angle = 90))
```

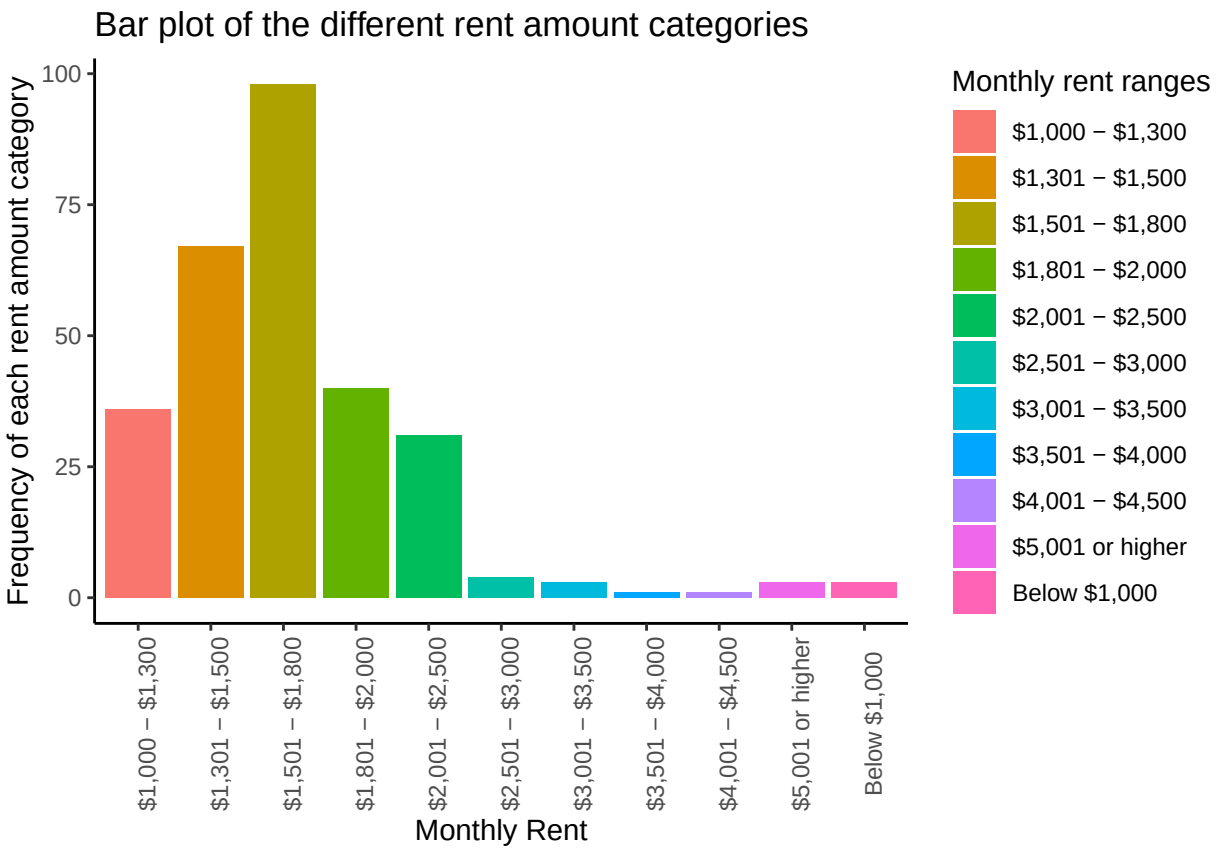


As displayed above, it can be seen that many have opted to go along the lines of using "Apartment Complexes" where as still a signifocant number has opted to use property management companies.

2.2.) The monthly rent analysis

```
rentAmt <- rentd %>%
  dplyr::group_by(Monthly.rent) %>%
  dplyr::summarise(count = n())

#Plotting in a graph
ggplot(rentAmt,aes(x = Monthly.rent,
                   y = count,
                   fill = Monthly.rent)) +
  geom_bar(stat = "identity") +
  theme_classic() +
  labs(
    x = "Monthly Rent",
```

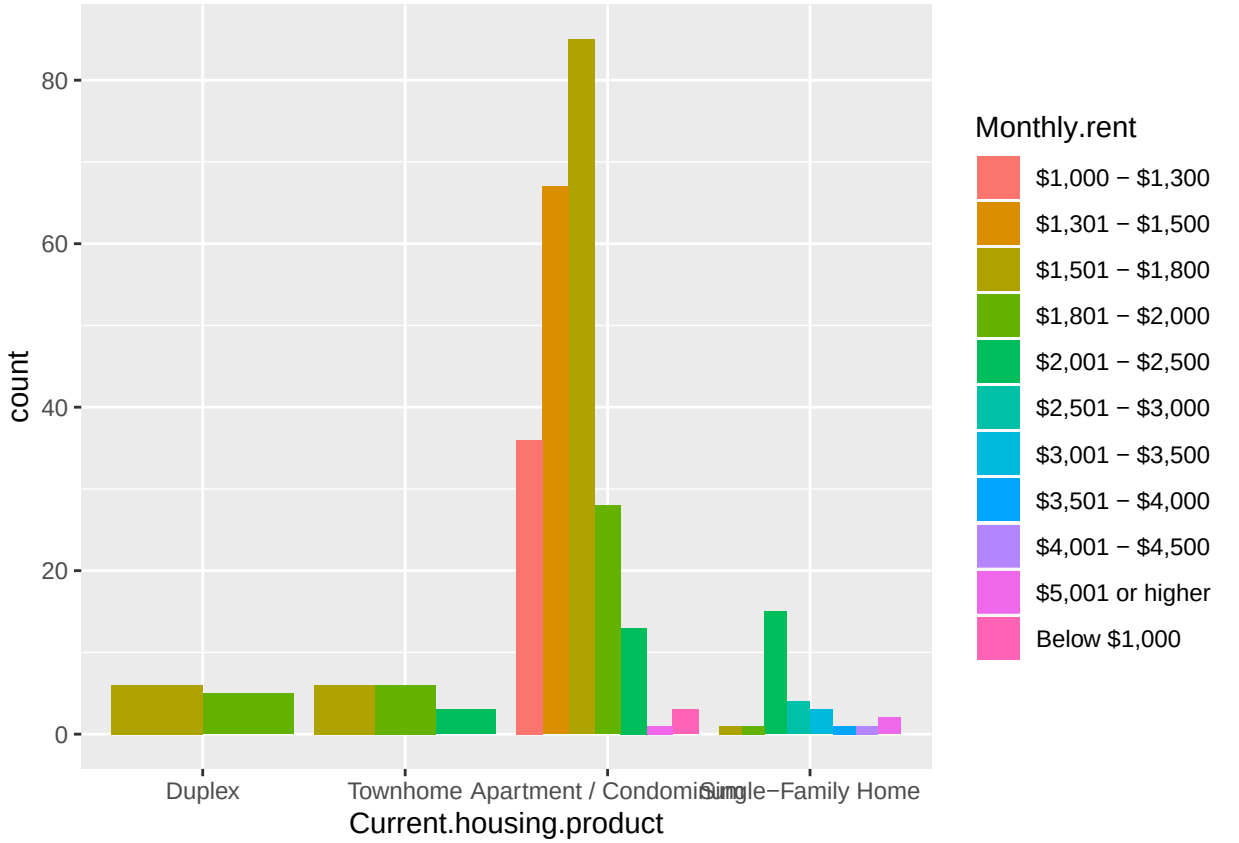
```
    y = "Frequency of each rent amount category",
    title = paste("Bar plot of the different rent amount categories"),
    fill = "Monthly rent ranges"
  ) +
  theme(axis.text.x = element_text(angle = 90))
```



The monthly rent that is preferred by most seems to be around $1501 - $1800.

In terms of this monthly rent analysis, it can be further detailed by looking at which produt type is rented at which monthly rental value each. This analysis is done below.

```
rentOnCategory <- rentd %>%
  select(ID,Current.housing.product,Monthly.rent)

#ggplot(rentOnCategory, aes(Monthly.rent, ..count..)) + geom_bar(aes(fill = Current.housing.product), p
ggplot(rentOnCategory, aes(Current.housing.product, ..count..)) + geom_bar(aes(fill = Monthly.rent), pos
```



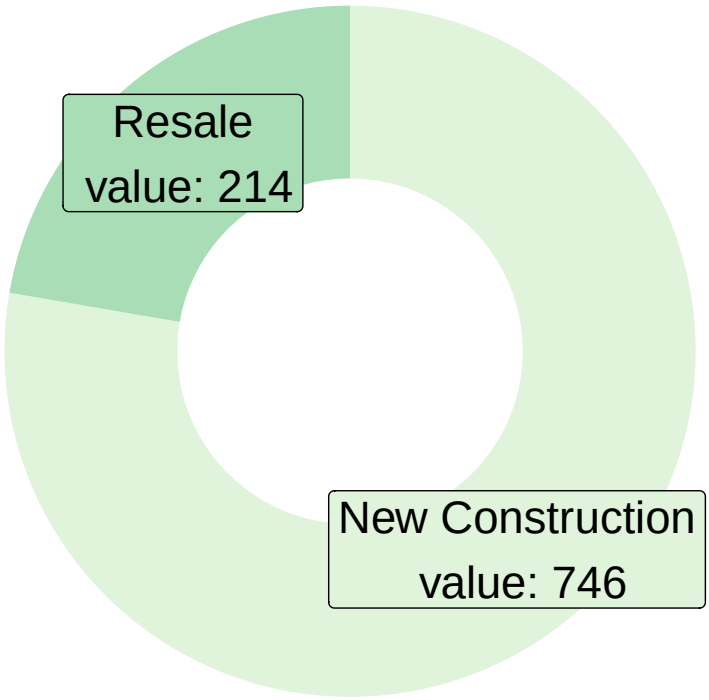
It can be seen that Apartmet/Condominiums and Family homes have a more resident interest and can be used to boost the revenue by marking a higher rent category as evident by the graph shown.

2.3.) Analysis on the residents who have purchased the residence

The residents have purchased the house as either a resale or as a new construction. This analysis is donr here in terms of those residents who stated that their current residence is "Own" purchase type

```
#Selecting the rented houses
prchsd <- rentOwn %>% filter(Rent.or.Own.current.home=="Own")

#Dropping the rows with either NAs or empty string values ("")
prchsd <- prchsd[!(is.na(prchsd$Purchase.method) | prchsd$Purchase.method==""), ]

#Getting the count of each purchase method
prchMtd <- prchsd %>%
  dplyr::group_by(Purchase.method) %>%
  dplyr::summarise(count = n())

#Visualization using a doughnut chart
# Compute percentages
prchMtd$fraction <- prchMtd$count / sum(prchMtd$count)

# Compute the cumulative percentages (top of each rectangle)
prchMtd$ymax <- cumsum(prchMtd$fraction)
```

```r
# Compute the bottom of each rectangle
prchMtd$ymin <- c(0, head(prchMtd$ymax, n=-1))

# Compute label position
prchMtd$labelPosition <- (prchMtd$ymax + prchMtd$ymin) / 2

# Compute a good label
prchMtd$label <- paste0(prchMtd$Purchase.method, "\n value: ", prchMtd$count)

# Make the plot
ggplot(prchMtd, aes(ymax=ymax, ymin=ymin, xmax=4, xmin=3, fill=Purchase.method)) +
  geom_rect() +
  geom_label( x=3.5, aes(y=labelPosition, label=label), size=6) +
  scale_fill_brewer(palette=4) +
  coord_polar(theta="y") +
  xlim(c(2, 4)) +
  theme_void() +
  theme(legend.position = "none")
```



As illustrated, the majority of the residents that have been purchased have been done so as a “New Construction” as opposed to a “Resale”

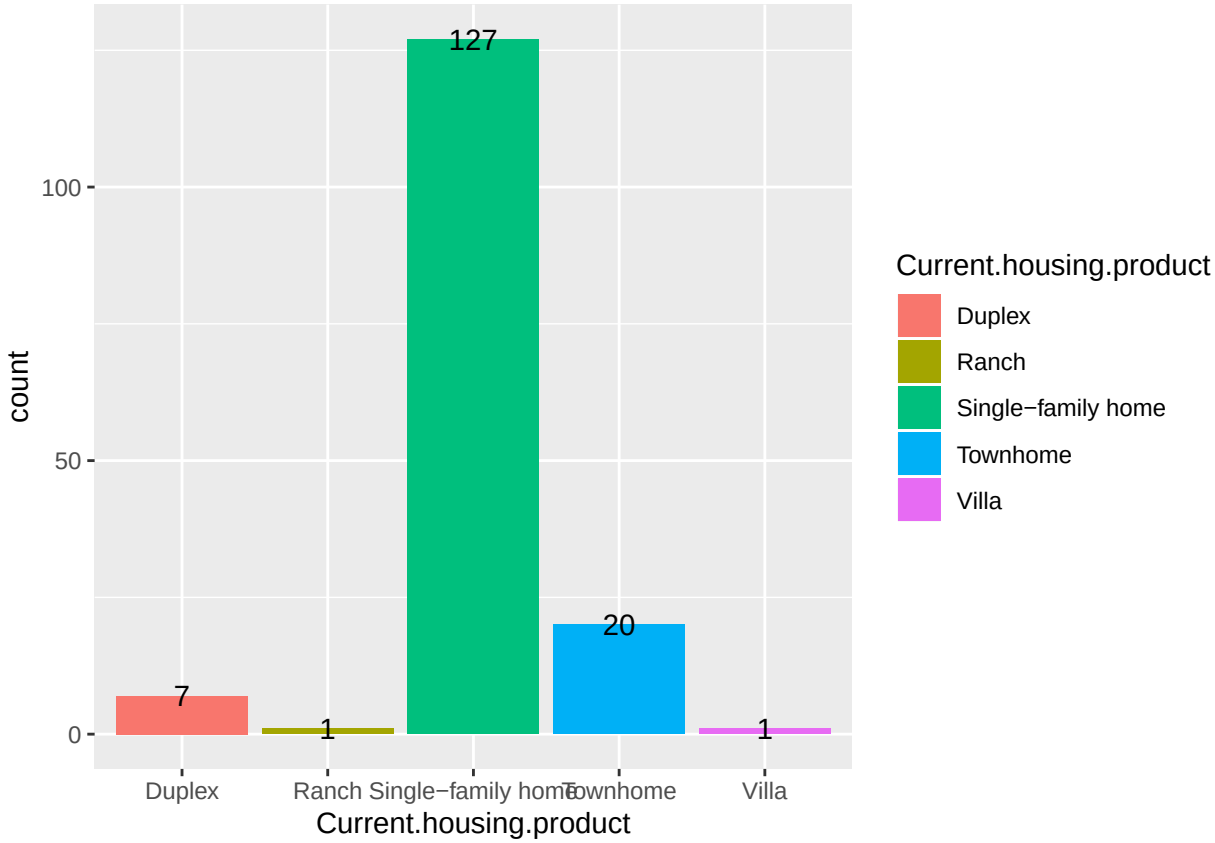
3.) Is this your first purchase in Lake Nona?

This analysis was considered as a means to identify the reasons that lead new comers (first time purchasers in Lake Nona) to consider Lake Nona and the respective features and impressions they hold towards liiving in Lake Nona.

```
FrstHmPrch <- data %>% select("ID",
                              "Rent.or.Own.current.home",
                              "Do.you.rent.your.home.through.a.property.management.company.or.through.t
                              "Monthly.rent",
                              "Purchase.method",
                              "X.Yes.No..Is.your.Lake.Nona.residence.your.first.home.purchase.")
```

3.1.) Is this your first purchase in Lake Nona?

```
#First home purchse analysis along with 1.)next move or 2.)How likely to recommend question 3.)Importan
#5.)Purchase price 6.)Previous residence 7.)Location expected to move next 8.)Neighbourhood within Lake

frstPurch <- data %>% select("ID",
                             "Current.housing.product",
                             "X.Yes.No..Is.your.Lake.Nona.residence.your.first.home.purchase.",
                             "Product.type.of.previous.home",
                             "Size.of.previous.home",
                             "Primary.reason.for.your.next.move",
                             "Best.represents.your.next.home.relative.to.your.current.home",
                             "Current.housing.product.1",
                             "Purchase.Price.of.Home",
                             "Previous.Residence",
                             "Location.expected.to.move.next",
                             "Builder.of.the.house",
                             "Cellular.Provider",
                             "Lake.Nona.community.lived.previously",
                             "Rent.or.Own.previous.home",
                             "Rent.or.Own.current.home")

#Filling missing values in the current housing product with current housing product 1 column
frstPurch <- setDT(frstPurch)[Current.housing.product=="", Current.housing.product:= Current.housing.pr

#-----------Analysing first time buyers
frstYes <- frstPurch %>% filter(X.Yes.No..Is.your.Lake.Nona.residence.your.first.home.purchase.=="Yes")
#Plotting the current housing product of first time purchasers
ggplot(frstYes, aes(Current.housing.product, ..count..)) + geom_bar(aes(fill = Current.housing.product)
  geom_text(aes(label=..count..),stat="count")
```

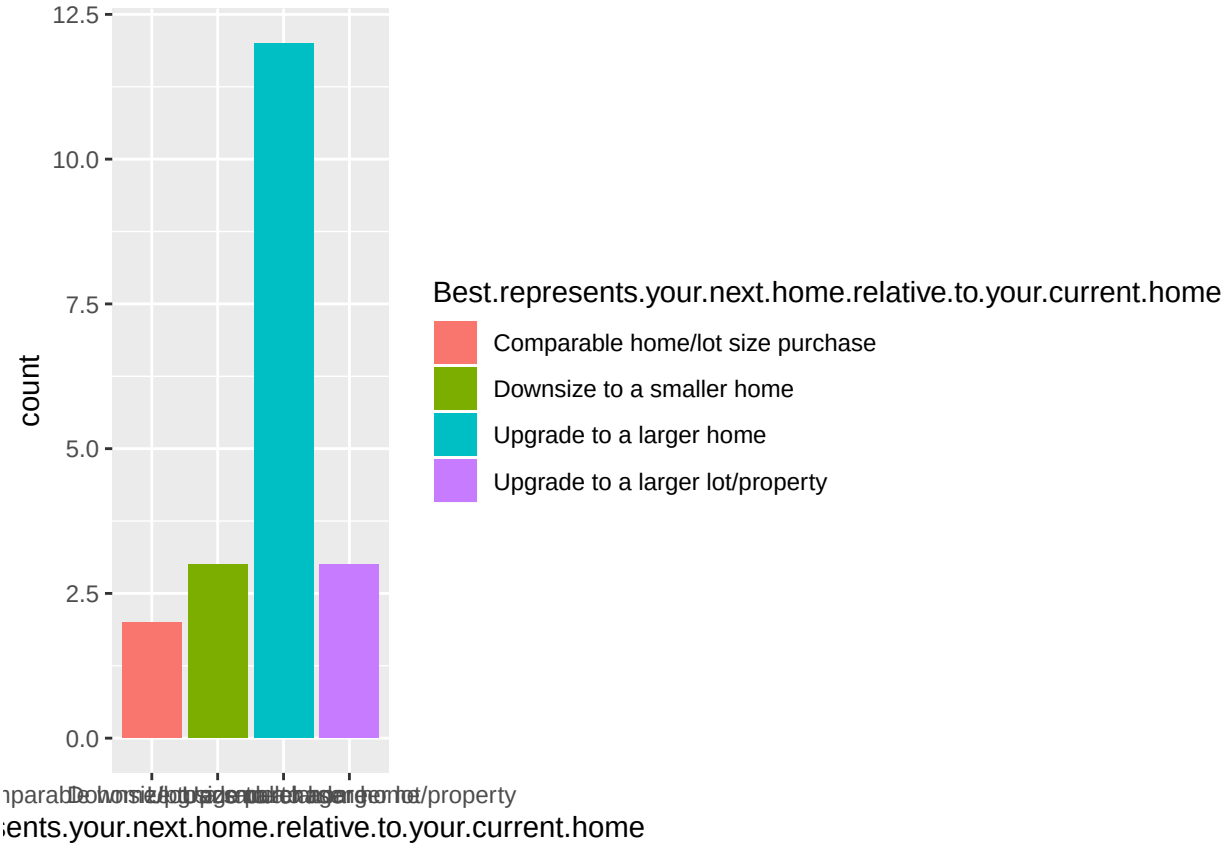


It can be seen that most of the first time purchasers have gone for Single-Family homes and the least purchased are Ranches and Villas.

Further, the analysis is taken forward along this viewpoint to see the movement of these first time purchasers. An important observation is that they tend to move either within Lake Nona or Outside of Lake Nona. These two seperate scenarios are looked into individually as follows.

3.1.a.) Movers within Lake Nona

```
#Looking into the turn over of residents who bought for the first time
frstYes_Move <- frstYes %>% select(Primary.reason.for.your.next.move,
                                   Best.represents.your.next.home.relative.to.your.current.home,
                                   Location.expected.to.move.next)


#Analysing those who are moving within Lake Nona for the reasons
frstYes_MoveWithin <- frstYes_Move %>% filter(Location.expected.to.move.next=="Another home in Lake Nona
ggplot(frstYes_MoveWithin, aes(Best.represents.your.next.home.relative.to.your.current.home, ..count..)
```



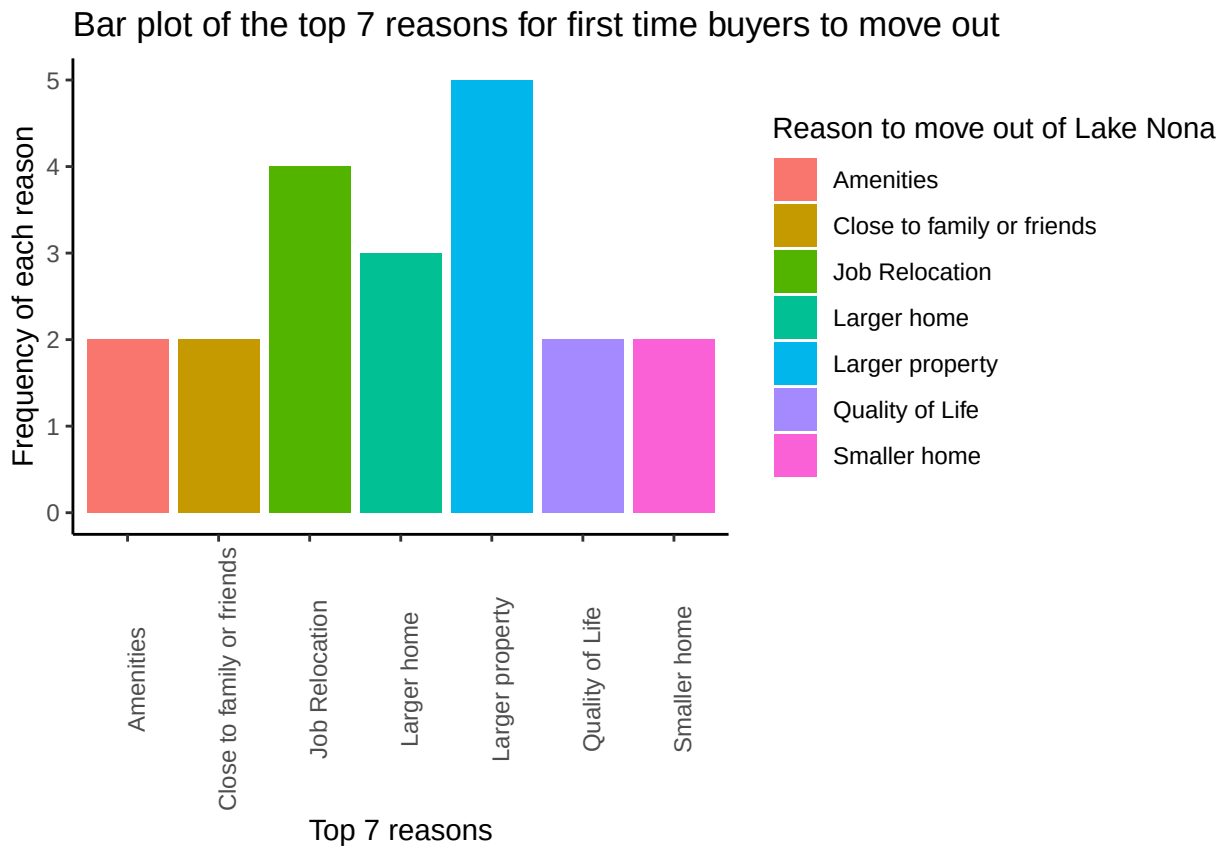
The insight expected out of the above plot is to see, the expectations of the movers that will purchase a residence within Lake Nona. This figure shows the features or the expected improvements of the residences of the homes as compared to that of the current homes. It can be seen that most of the movers within Lake Nona are expecting to upgrade to a larger home than what they currently have.

3.1.b.) Movers outside of Lake Nona

The analysis done below looks into the turnover of the first time purchasers and tries to map the potential reasons for them to move away from Lake Nona.

```
#Analysing the movers who are going to move out of Lake Nona
frstYes_MoveOut <- frstYes_Move %>% filter(Location.expected.to.move.next!="Another home in Lake Nona")
#Filling missing values in the entire frstYes_MoveOut dataset
frstYes_MoveOut2 <- frstYes_MoveOut[!(is.na(frstYes_MoveOut$Primary.reason.for.your.next.move) | frstYes
#Looking into the primary reasons for these movers outside of Lake Nona for their decision
frstYes_MoveOutReason <- frstYes_MoveOut2 %>%
  dplyr::group_by(Primary.reason.for.your.next.move) %>%
  dplyr::summarise(count=n()) %>%
  dplyr::filter(count>=2)
#Plotting the top 7 reasons for this movement
ggplot(frstYes_MoveOutReason,aes(x = Primary.reason.for.your.next.move,
                   y = count,
                   fill = Primary.reason.for.your.next.move)) +
  geom_bar(stat = "identity") +
  theme_classic() +
  labs(
```

```
    x = "Top 7 reasons",
    y = "Frequency of each reason",
    title = paste("Bar plot of the top 7 reasons for first time buyers to move out"),
    fill = "Reason to move out of Lake Nona"
  ) +
  theme(axis.text.x = element_text(angle = 90))
```




```
#Plotting as a faceted one
# ggplot(frstYes_MoveOut2, aes(Primary.reason.for.your.next.move, ..count..)) +
#   geom_bar(aes(fill = Best.represents.your.next.home.relative.to.your.current.home), position = "dodg
#   theme(axis.text.x = element_text(angle = 90))
```

As illustrated above, the first time buyers who are going to move out of Lake Nona is because they are looking for a larger property. Another significant reason seems to be the job relocation.

3.2.) Residents who have purchsed a home for the second time or more (None-first time purchasers)

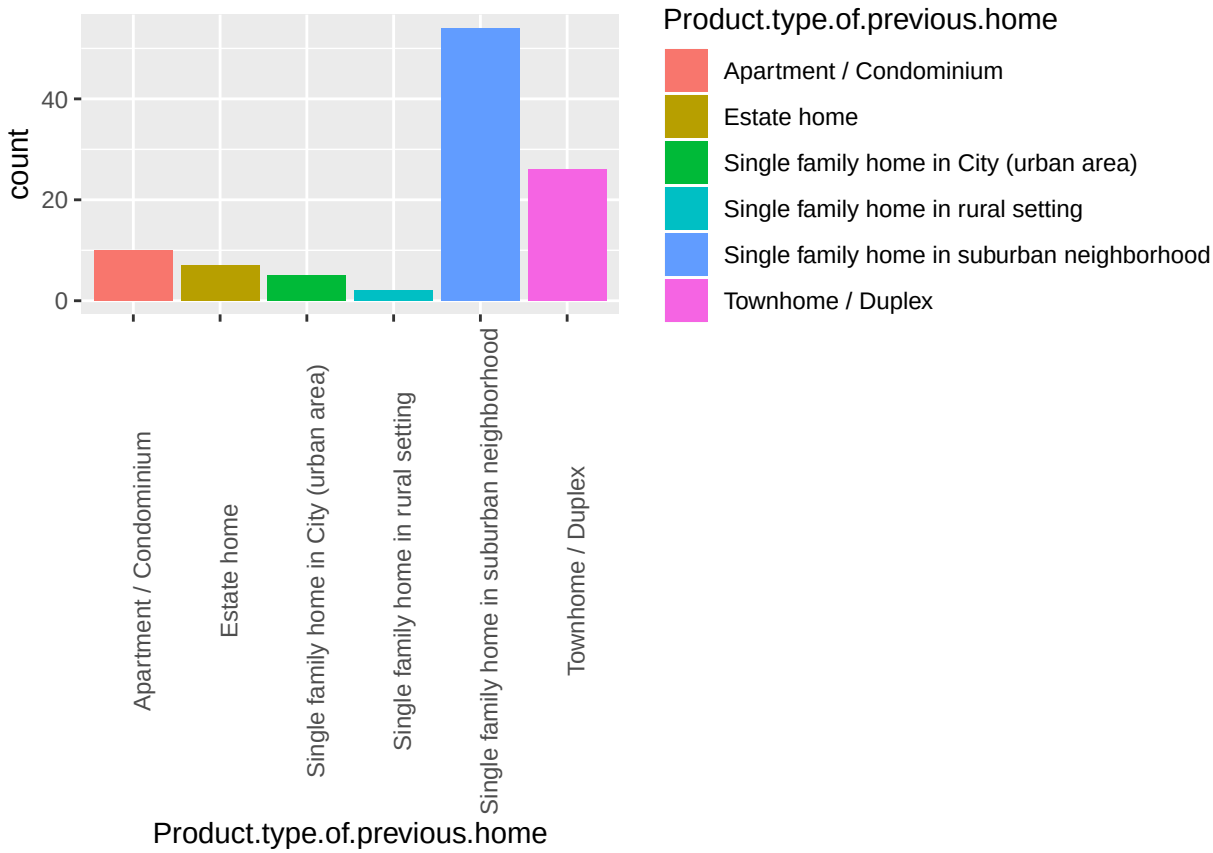
This analysis looks into identifying the change in the product type of the next residence in comparison to the current product, of those who have been purchasing residences in Lake Nona more than once.

3.2.a.) Change of the product type among second time buyers

```
#-----------Analysing second time (or more) purchasers
#Here we can see who were within Lake Nona in previous purchase and see the change in housing product t
frstNo <- frstPurch %>% filter(X.Yes.No..Is.your.Lake.Nona.residence.your.first.home.purchase.=="No") %
  select(-Current.housing.product.1) %>%
  select(Current.housing.product, Product.type.of.previous.home, Previous.Residence)
#Removing "" and NaNs
frstNo <- frstNo[!(is.na(frstNo$Previous.Residence) | frstNo$Previous.Residence==""), ]
#Seeing who was within Lake Nona
frstNo_Within <- frstNo %>%
  dplyr::filter(grepl("Within", Previous.Residence))
#Plotting previous product type
p1 <- ggplot(frstNo_Within, aes(Product.type.of.previous.home, ..count..)) +
  geom_bar(aes(fill = Product.type.of.previous.home), position = "dodge") +
  theme(axis.text.x = element_text(angle = 90))
#Plotting current product type
p2 <- ggplot(frstNo_Within, aes(Current.housing.product, ..count..)) +
  geom_bar(aes(fill = Current.housing.product), position = "dodge") +
  theme(axis.text.x = element_text(angle = 90))
#We can look at the two graphs and say how it has changed now. Which is more [popular and which has gon
```

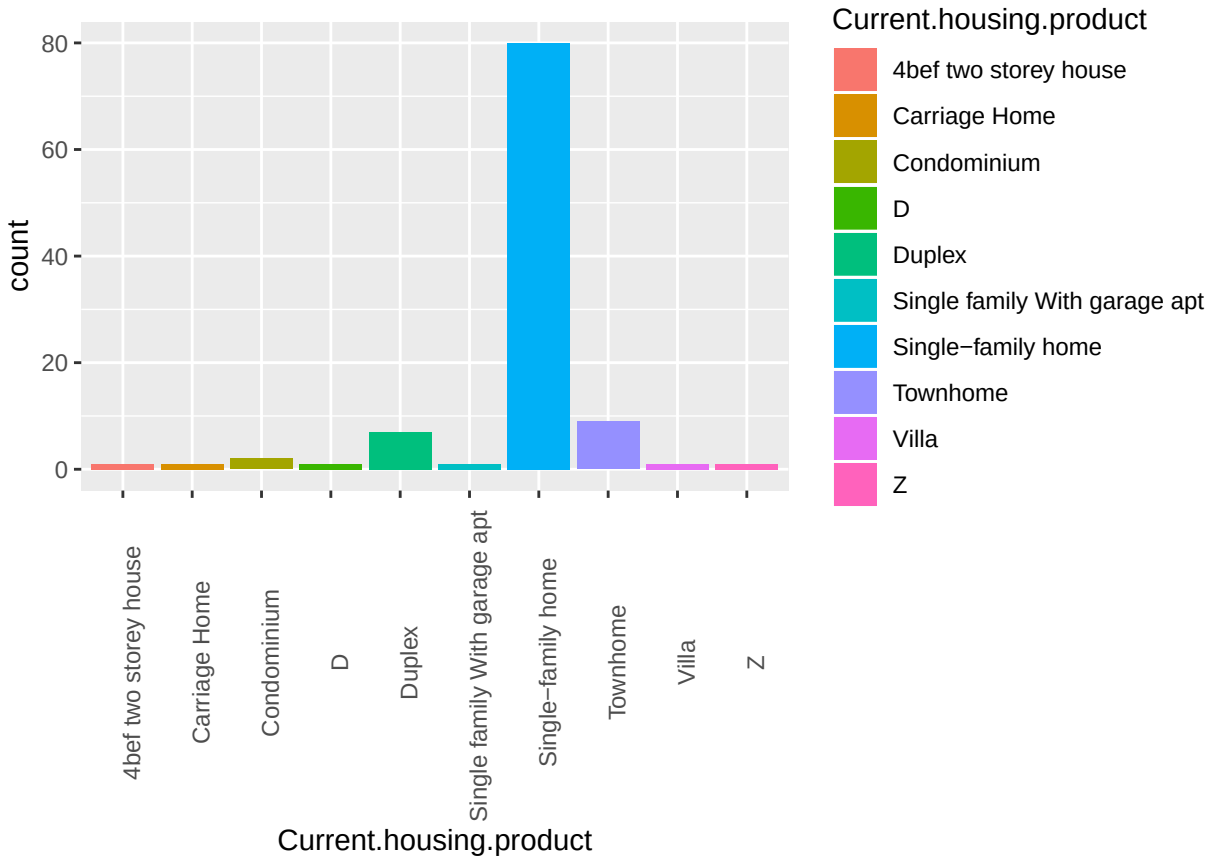
The previous product type is displayed below :

```
#previous product type
p1
```



The current product type is displayed below :

```
#current product type
p2
```



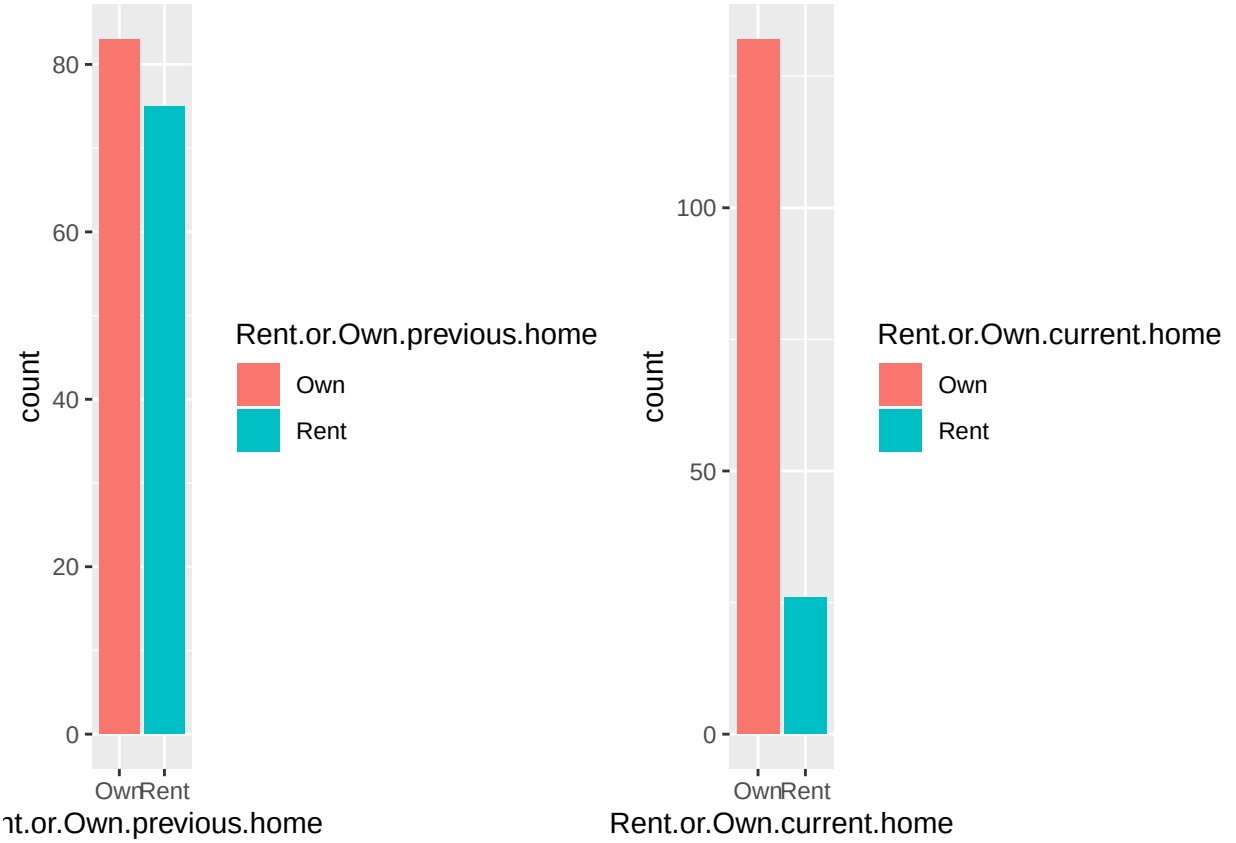
As illustrated by the above 2 graphs, it can be stated that, Single Family homes (both urban and suburban setting) are popular and is going to be the most preffered housing product among the next purchase of Lake Nona residents.

A noticable drop however can be seen in the preference towards Condominiums among the Lake Nona residents in their next purchase.

3.2.b.) Change of the Rent or Own condition among second time buyers

Here the change in the inclination towards renting or owning thge next residence among the current residents are considered.

```
#-----------Analysing the people who lived in Lake Nona and the respective rent/own situation - Current
frstPurch_2 <- frstPurch %>%
  select(ID, Previous.Residence, Rent.or.Own.previous.home,Rent.or.Own.current.home) %>%
  dplyr::filter(grepl("Within", Previous.Residence))

p2 <- ggplot(frstPurch_2, aes(Rent.or.Own.previous.home, ..count..)) + geom_bar(aes(fill = Rent.or.Own.p
p3 <- ggplot(frstPurch_2, aes(Rent.or.Own.current.home, ..count..)) + geom_bar(aes(fill = Rent.or.Own.cu

multiplot(p2,p3, cols = 2)
```



It can be seen that the number who owns houses in Lake Nona has increased

4.) Mode of transportation and time to work

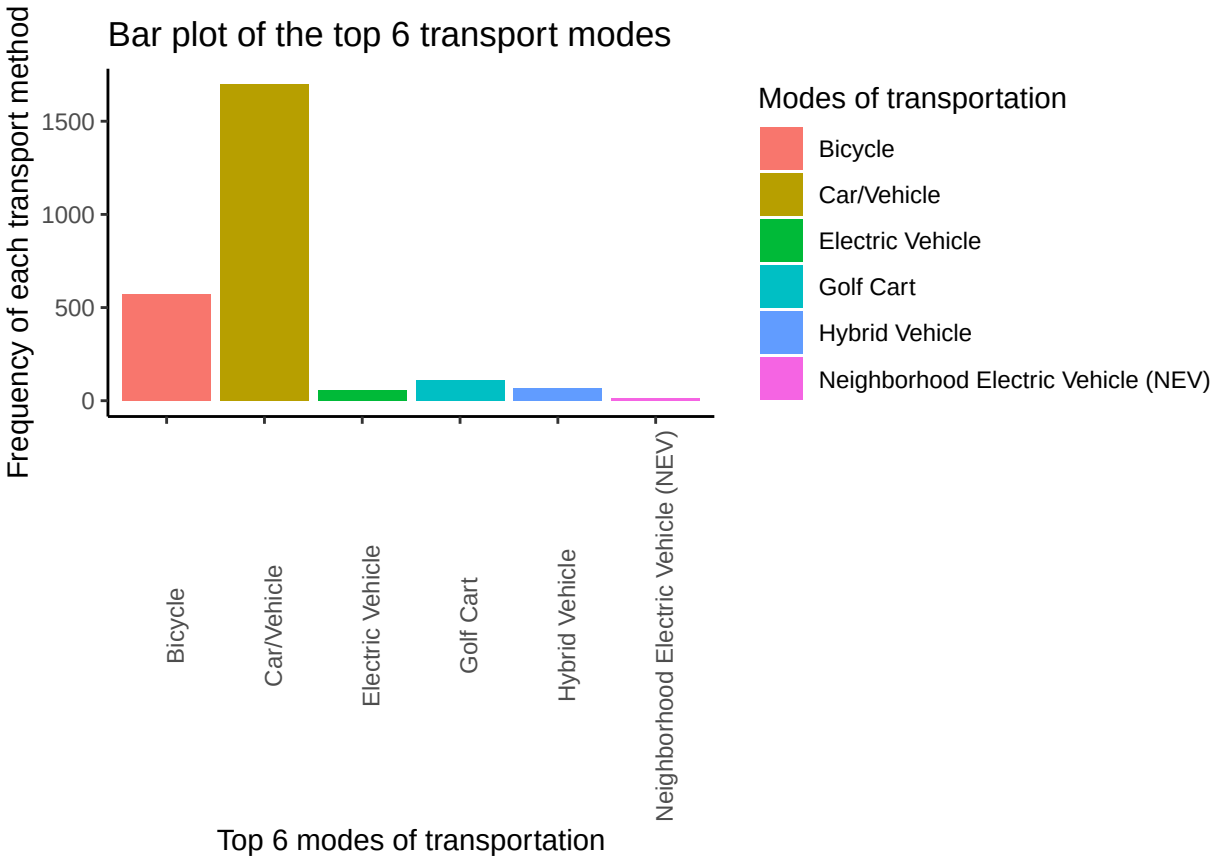
This analysis was done to identify the different work locations within and outside Lake Nona and see the time taken to commute to work through each mode of transportation to each work location of the residents.

```
#Mode of transportation used and time to work
tramode <- data %>% select("ID",
                           "Modes.of.Transportation" ,
                           "Neighborhood.within.LakeNona",
                           "Lake.Nona.working.place.1" ,
                           "Lake.Nona.working.place",
                           "Work.location.of.adults.in.the.household",
                           "Time.take.to.household.to.commute.to.work")

tramode2 <- tramode %>% select(-c(Lake.Nona.working.place,Lake.Nona.working.place.1)) %>%
  separate_rows("Work.location.of.adults.in.the.household", sep = ",", convert = TRUE)

tramode3 <- tramode2 %>%
  separate_rows("Modes.of.Transportation", sep = ",", convert = TRUE)

tramode4 <- tramode3 %>%
  separate_rows("Time.take.to.household.to.commute.to.work", sep = ",", convert = TRUE)

#Now getting rid of the rows that have not provided the work location
```

```
tramode4 <- tramode4[!(is.na(tramode4$Work.location.of.adults.in.the.household) | tramode4$Work.locatio
#Removing those who work at home as well
tramode4 <- tramode4 %>% filter((Work.location.of.adults.in.the.household!="Work at home") & (Work.loca

#Now getting rid of the rows that have not provided the time to go to work
tramode4 <- tramode4[!(is.na(tramode4$Time.take.to.household.to.commute.to.work) | tramode4$Time.take.t

#Count number of each work locations and getting the top 10 work locations of the residents
wrklcn <- tramode4 %>%
  dplyr::group_by(Work.location.of.adults.in.the.household) %>%
  dplyr::summarise(count = n()) %>%
  dplyr::filter(count>=25)
#Getting these top 10 work locations to a list
topWrkLcn_list <- wrklcn$Work.location.of.adults.in.the.household
#Filtering out the tramode4 dataframe based on these top 10 places of work locations
tramode5 <- tramode4 %>% filter(Work.location.of.adults.in.the.household %in% topWrkLcn_list)
```

Displayed below is the top 6 modes of transportation used by the Lake Nona residents as a whole

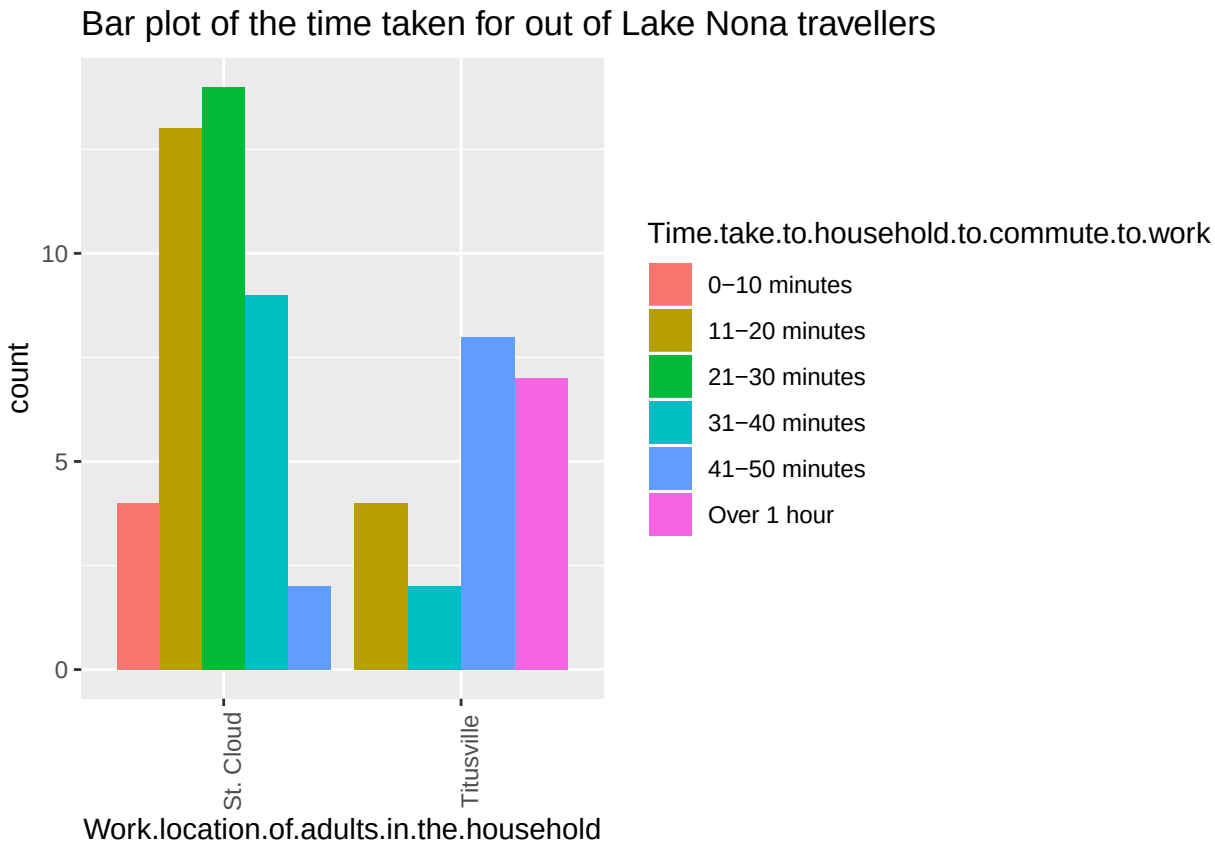
```
#Plotting the modes of transportaion for the top 6 popular methods
trvlMode <- tramode5 %>%
  dplyr::group_by(Modes.of.Transportation) %>%
  dplyr::summarise(count = n()) %>%
  dplyr::filter(count>=10)

ggplot(trvlMode,aes(x = Modes.of.Transportation,
                    y = count,
                    fill = Modes.of.Transportation)) +
  geom_bar(stat = "identity") +
  theme_classic() +
  labs(
    x = "Top 6 modes of transportation",
    y = "Frequency of each transport method",
    title = paste("Bar plot of the top 6 transport modes"),
    fill = "Modes of transportation"
  ) +
  theme(axis.text.x = element_text(angle = 90))
```



4.1.) Mode of transportation and time to work for the people who work Outside of Lake Nona

The below section looks at the residents that works outside of Lake Nona. The data suggests that St. Cloud and Tituville are the main work destinations traversed by those who are employed outside of Lake Nona.

```
#Looking at those who work OUTSIDE of Lake Nona
wrkInLN <- tramode[!(is.na(tramode$Lake.Nona.working.place) | tramode$Lake.Nona.working.place==""), ]
#Getting the work locations of these into a list and filtering these areas from tramode5 to get only th
wrkInLN <- wrkInLN %>%
  separate_rows("Work.location.of.adults.in.the.household", sep = ",", convert = TRUE)

wrkInLN_list <- wrkInLN$Work.location.of.adults.in.the.household

wrkOutLN <- subset(tramode5, !(tramode5$Work.location.of.adults.in.the.household %in% wrkInLN_list)) %>%
  filter(Time.take.to.household.to.commute.to.work != "Work from Home")


#Plotting the time taken for out of Lake Nona travellers (St. Cloud and Tituville) using any mode of tr
ggplot(wrkOutLN, aes(Work.location.of.adults.in.the.household, ..count..)) +
  geom_bar(aes(fill = Time.take.to.household.to.commute.to.work), position = "dodge") +
  labs(
    title = paste("Bar plot of the time taken for out of Lake Nona travellers")
  ) +
  theme(axis.text.x = element_text(angle = 90))
```

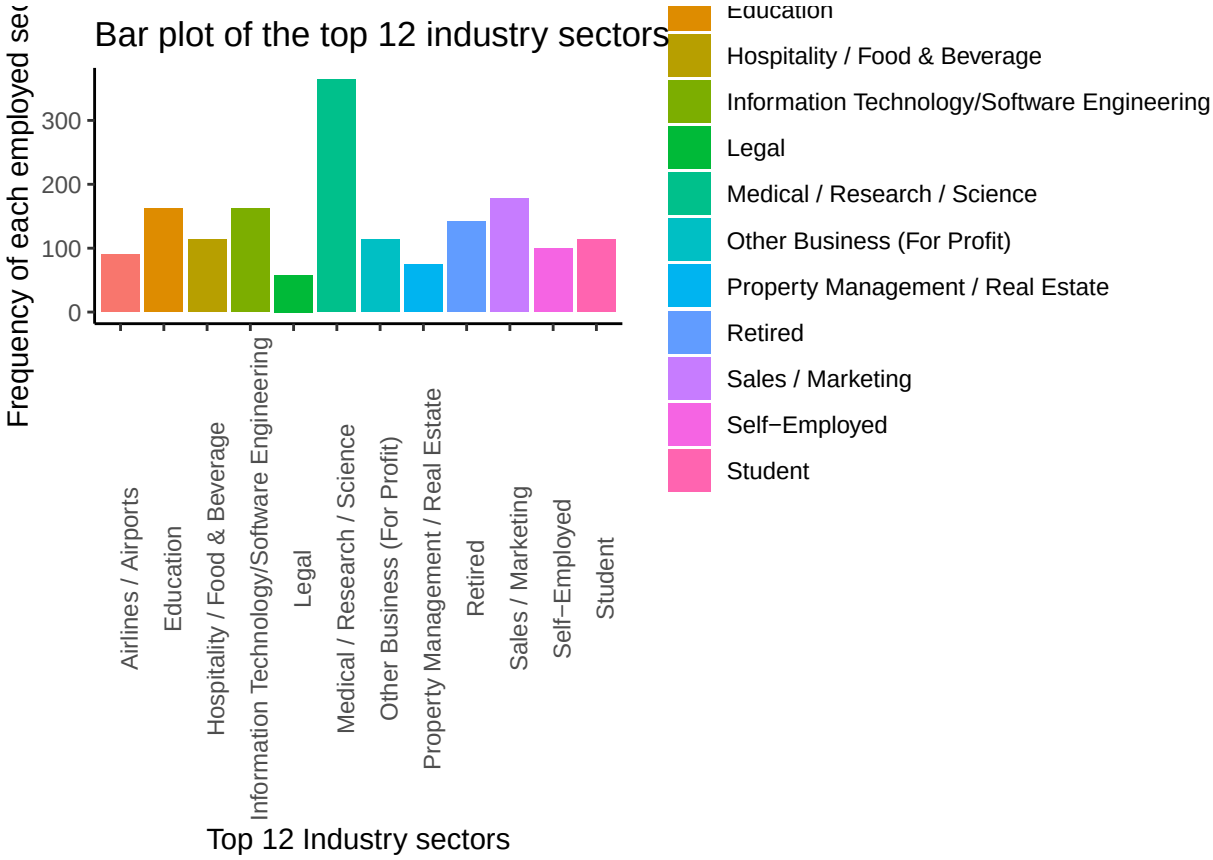


5.) Analysis of the industry sector

Given below is the analysis done in terms of the mostly employed industry sectors within Lake Nona. This is can be used to improve the respective corporations or companies related to these industry sectors and improve the employability and employee satisfaction level of these residents.

```
#Analysis on the Industry sector
indSec <- data %>% select("ID",
                          "Neighborhood.within.LakeNona",
                          "Lake.Nona.working.place.1",
                          "Work.location.of.adults.in.the.household",
                          "Time.take.to.household.to.commute.to.work")

indSec2 <- indSec  %>%
  separate_rows("Work.location.of.adults.in.the.household", sep = ",", convert = TRUE)

#Removing those who work at home as well
indSec3 <- indSec2 %>% filter((Work.location.of.adults.in.the.household!="Work at home") & (Work.locati

#Seperating the rows of industry sector
indSec4 <- indSec3 %>%
  separate_rows("Lake.Nona.working.place.1", sep = ",", convert = TRUE)

#Removing NA and "" values and getting the count of each industry sector
indSec4 <- indSec4[!(is.na(indSec4$Lake.Nona.working.place.1) | indSec4$Lake.Nona.working.place.1==""),
```

```r
indSec5 <- indSec4 %>%
  dplyr::group_by(Lake.Nona.working.place.1) %>%
  dplyr::summarise(count = n()) %>%
  dplyr::filter(count>=55)

#Plotting the top 12 industry sectors the residents work in
ggplot(indSec5,aes(x = Lake.Nona.working.place.1,
                   y = count,
                   fill = Lake.Nona.working.place.1)) +
  geom_bar(stat = "identity") +
  theme_classic() +
  labs(
    x = "Top 12 Industry sectors",
    y = "Frequency of each employed sector",
    title = paste("Bar plot of the top 12 industry sectors"),
    fill = "Industry Sectors"
  ) +
  theme(axis.text.x = element_text(angle = 90))
```

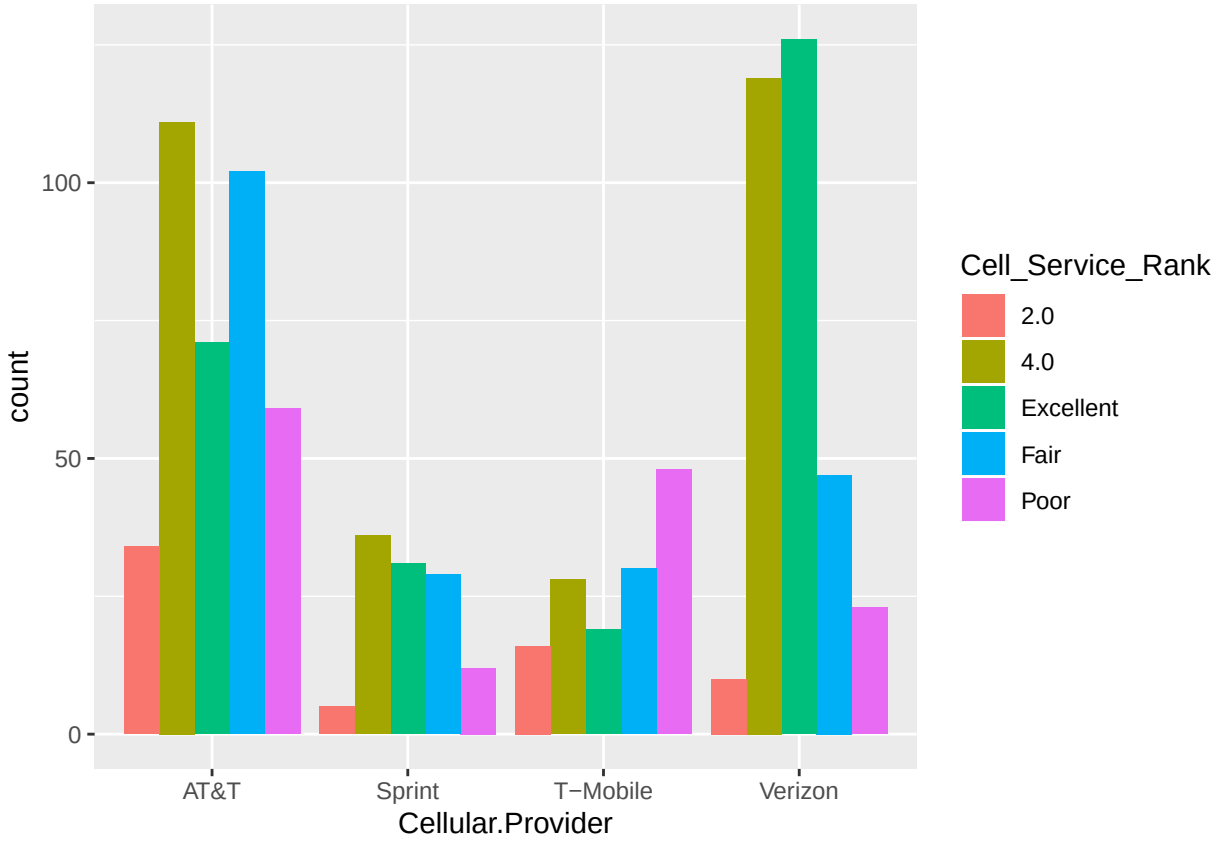


As per the above bar plot, it can be seen that the least employed in the field of Legal whereas most of the residents are employed in Medical/Research/Science doomain within Lake Nona. This can be used to better improve the employee satisfaction and market Lake Nona for potential residents to generate more retention within Lake Nona.

(2.) Retailers section

1.) Top cellular service providers and their rankings

```
##------------------------Cellular service provider-----------------------------------------#
cellServe <- data %>% select("ID",
                             "Cellular.Provider",
                             #"Internet.speed.satisfaction",
                             "X.Rank.question..Please.rank.the.quality.of.your.cellular.service.in.the.

#Renaming the lenghty column
colnames(cellServe)[colnames(cellServe) == 'X.Rank.question..Please.rank.the.quality.of.your.cellular.s

cellServe_Sep <- cellServe %>%
  separate_rows(Cell_Service_Rank, sep = ",", convert = TRUE) %>%
  drop_na()

#Dropping the rows with either NAs or empty string values ("")
cellServe_Sep <- cellServe_Sep[!(is.na(cellServe_Sep$Cell_Service_Rank) | cellServe_Sep$Cell_Service_Ra

#Extracting the observations that states the service to be either Poor,2,Fair,4 or Excellent
cellServiceRank_list <- c("Excellent", "Poor", "2.0", "4.0", "Fair")
cellServe_Sep2 <- cellServe_Sep %>% filter(Cell_Service_Rank %in% cellServiceRank_list)

#Getting the mostly used top 4 cellular providers into a list
Top_Cell_Providers <- cellServe_Sep2 %>%
  dplyr::group_by(Cellular.Provider) %>%
  dplyr::summarise(count = n()) %>%
  dplyr::filter(count>100)

Top_Cell_Providers_list <- Top_Cell_Providers$Cellular.Provider

#Getting the distribution of each rank for the top cell service providers
cellServe_Sep3 <- cellServe_Sep2 %>% filter(Cellular.Provider %in% Top_Cell_Providers_list)
ggplot(cellServe_Sep3, aes(Cellular.Provider, ..count..)) + geom_bar(aes(fill = Cell_Service_Rank), pos
```



From the above graph it can be seen that the most used cellular service providers are AT&T, Sprint, T-Mobile and Verizon.

Also as per the user preseferences and ranking, it can be seen that AT&T and Verizon have gotten the highest ranking as an "Excellent" service provider out of these 4 top used cellular service providers.

2.) Community features that lead to living in Lake Nona

Another important aspect that can lead to a valuable insight is the analysis done on the community features that have lead to a certain degree of motivation to live in Lake Nona. This analysis is done below.

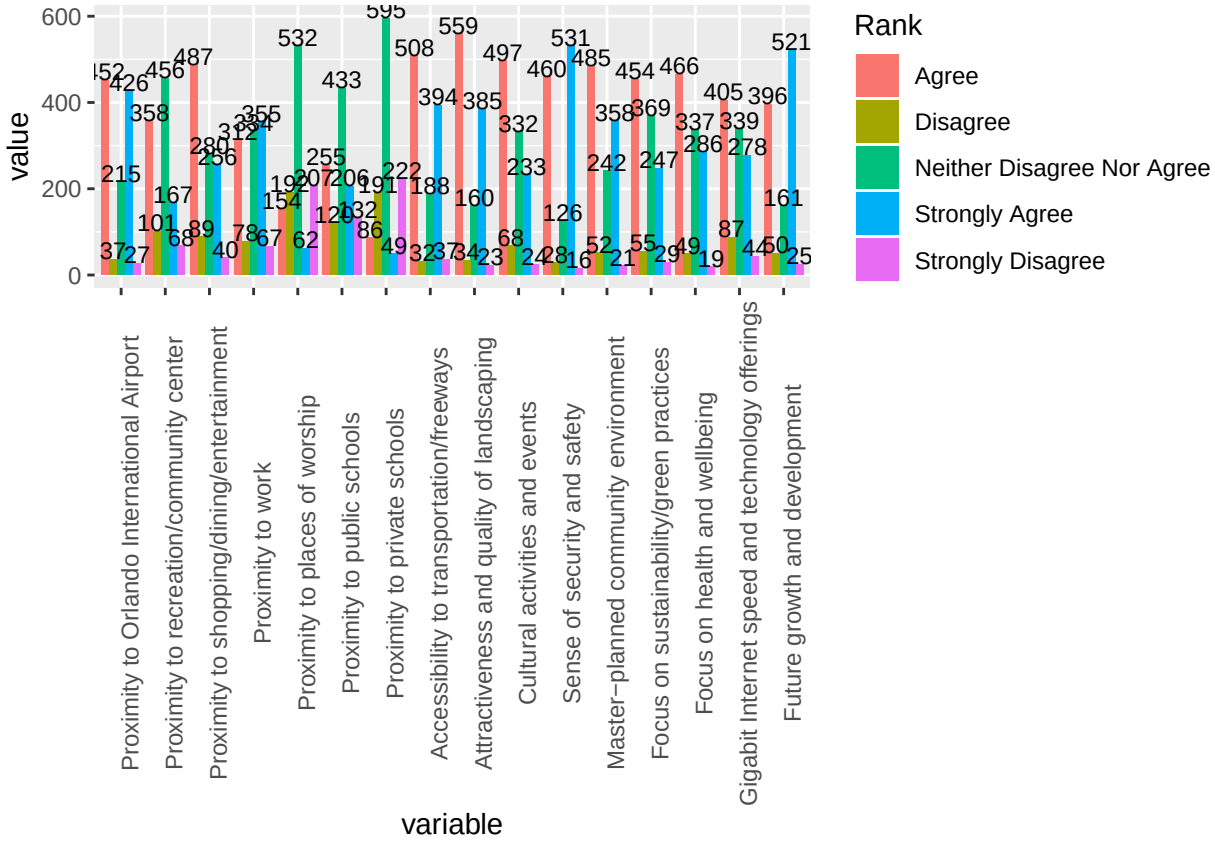
Notice that the interest here, was to identify the features that have motivated the second time buyers of residences in Lake Nona since they have been in Lake Nona for a considerable period of time as opposed to the first time buyers.

```
##------------------------Community features that lead to Live in Lake Nona----------------------------
headers <- read.csv("rank_data/community_features_liveInLakeNona.csv",
                    skip = 1,
                    header = F,
                    nrows = 1,
                    as.is = T)

comFeatures_df = read.csv("rank_data/community_features_liveInLakeNona.csv", skip = 2, header = F)
colnames(comFeatures_df) <-  headers
comFeatures_df$ID <- seq.int(nrow(comFeatures_df))

#SST_list <- colnames(df)
```

```r
comFeatures_listOrg <- colnames(comFeatures_df)[grep("Neither", colnames(comFeatures_df))]

#Getting the community features
comFeatures_list <- gsub(' - Neither Disagree Nor Agree', '', comFeatures_listOrg)

datalist = list()
vec <- 1:length(comFeatures_list)
for (i in seq_along(vec)) {

  var <- comFeatures_list[i]
  df1 <- comFeatures_df[ , grepl(var,names(comFeatures_df)) ]
  df1 <- mutate(df1, var = apply(df1[ , colnames(df1) ] , 1 , paste , collapse = "" )) %>% select(var)
  #df1 <- cbind(df1, ID = comFeatures_df$ID)
  names(df1)[names(df1) == 'var'] <- var

  datalist[[i]] <- df1
}

cleaned_ComFeaturesLakeNona <- do.call(cbind, datalist) %>% select(-c(`Proximity to community park(s)`))
cleaned_ComFeaturesLakeNona$ID <- seq.int(nrow(cleaned_ComFeaturesLakeNona))

#--------------------Joining these data with the second time buyers to see which comminity  features th
frstPurch <- data %>% dplyr::select("ID",
                              "Current.housing.product",
                              "X.Yes.No..Is.your.Lake.Nona.residence.your.first.home.purchase.",
                              "Product.type.of.previous.home",
                              "Size.of.previous.home",
                              "Primary.reason.for.your.next.move",
                              "Best.represents.your.next.home.relative.to.your.current.home",
                              "Current.housing.product.1",
                              "Purchase.Price.of.Home",
                              "Previous.Residence",
                              "Location.expected.to.move.next",
                              "Builder.of.the.house",
                              "Cellular.Provider",
                              "Lake.Nona.community.lived.previously",
                              "Rent.or.Own.previous.home",
                              "Rent.or.Own.current.home")

#Filling missing values in the current housing product with current housing product 1 column
frstPurch <- setDT(frstPurch)[Current.housing.product=="", Current.housing.product:= Current.housing.pr

#Joining frstPurch and cleaned_ComFeaturesLakeNona
frstPurch_CommFeat <- dplyr::left_join(cleaned_ComFeaturesLakeNona, frstPurch, by = c("ID"="ID"))

#Second time buyers
frstNo <- frstPurch_CommFeat %>% filter(X.Yes.No..Is.your.Lake.Nona.residence.your.first.home.purchase.=
  select(c(colnames(cleaned_ComFeaturesLakeNona), Location.expected.to.move.next)) %>% #extracting the
  select(ID, Location.expected.to.move.next, everything()) %>% drop_na() #rearrange the columns and dro

  # select(-Current.housing.product.1) %>%
  # select(Current.housing.product, Product.type.of.previous.home, Previous.Residence)
```

```r
#p3 <- ggplot(frstNo, aes(Rent.or.Own.previous.home, ..count..)) + geom_bar(aes(fill = Rent.or.Own.prev

##Getting the frequency count of each variable in each column in frstNo
#Removing the ID column to help the frequency table counting
freqDataComFeat <- cleaned_ComFeaturesLakeNona %>% select(-c(ID, `Active Home Owners Association (HOA)`

#Replacing the duplicate categorical variable in the "Gigabit Internet speed and technology offerings"
freqDataComFeat$`Gigabit Internet speed and technology offerings` <- str_replace_all(freqDataComFeat$`G

datalist2 = list()
i <- 1

#Removing the NA and Empty rows
for (colname in colnames(freqDataComFeat)){

  colname <- as.character(colname)
  df <- freqDataComFeat %>% select(colname)
  # df <- df[!(is.na(df$colname) | df$colname == ""), ]
  tbl = table(df)
  freqDFComFeat <- as.data.frame(tbl)

  #Renaming the 'df' column with the community feature
  names(freqDFComFeat)[names(freqDFComFeat) == 'df'] <- paste0("Rank - ", colname)
  names(freqDFComFeat)[names(freqDFComFeat) == 'Freq'] <- colname

  ##Removing the empty row
  #Assigning NA to the empty cell
  freqDFComFeat2 <- freqDFComFeat %>% mutate_all(na_if,"")
  #Removing the NA
  freqDFComFeat2 <- na.omit(freqDFComFeat2)

  #Appending to list
  datalist2[[i]] <- freqDFComFeat2

  i <- i+1

}

#Getting the full dataframe
freqDFComFeat3 = do.call(cbind, datalist2)

#Removing redundant "Community_Feature_Rank" columns
freqDFComFeat4 <- freqDFComFeat3 %>% select(colnames(freqDataComFeat))
freqDFComFeat4$Rank <- unique(freqDFComFeat3$`Rank - Proximity to Orlando International Airport`)
freqDFComFeat4 <- freqDFComFeat4 %>% select(Rank, everything())

#Converting the dataframe to long format for plotting purposes
long <- melt(freqDFComFeat4, id.vars = c("Rank"))
# long_fltd <- long %>% filter(Rank == 'Agree')
# long_fltd2 <- long %>% group_by(Rank, variable) %>% summarise(total_count = sum(value))

#Plotting the data
p2=ggplot(long, aes(fill=Rank, y=value, x=variable)) +
```

```
  geom_bar(position="dodge", stat="identity") +
  theme(axis.text.x = element_text(angle = 90)) +
  geom_text(aes(label=paste0(value)), colour="black", vjust = 0, position = position_dodge(0.9), size=3)
p2
```



As per the above bar chart it can be seen that out of all the community features, "Proximity to place of worship" and "Proximity to private schools" have been ranked as the least likey reasons to stay in Lake Nona. (This is out of the people who have voted for each feature during the survey)

On the other hand, many second time buyers have voted that the following features are more significant in terms of why they want to reside in Lake Nona:

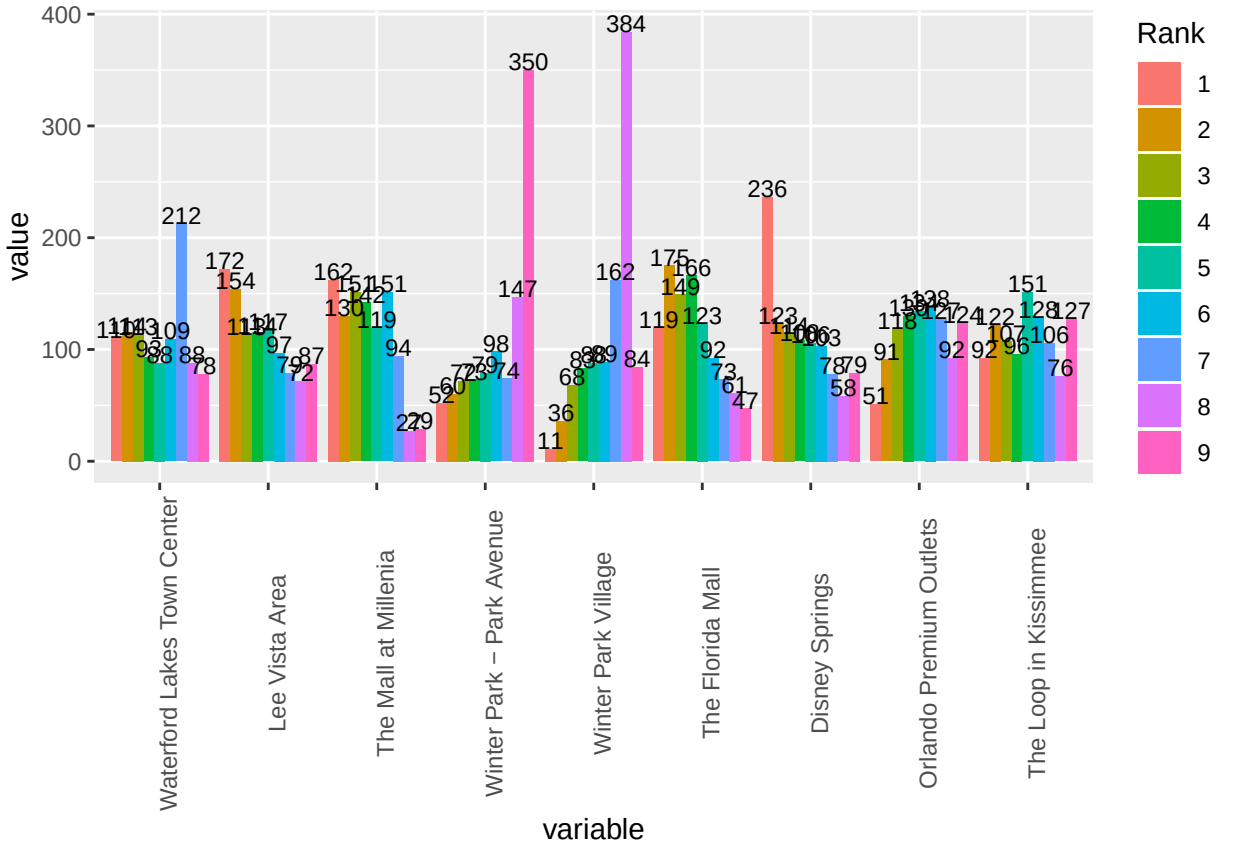
1.) Sense of security and safety 2.) Future groth and development 3.) Proximity to Orlando International Airport

3.) Most visited shopping destinations

```
##-----------------------MostVisitedShoppingDestinations-----------------------------------------------
headers <- read.csv("rank_data/MostVisitedShoppingDestinations.csv",
                    skip = 1,
                    header = F,
                    nrows = 1,
                    as.is = T)

comFeatures_df = read.csv("rank_data/MostVisitedShoppingDestinations.csv", skip = 3, header = F)
```

```r
colnames(comFeatures_df) <-  headers
comFeatures_df$ID <- seq.int(nrow(comFeatures_df))

#SST_list <- colnames(df)
comFeatures_listOrg <- colnames(comFeatures_df)[grep(" - 1", colnames(comFeatures_df))]

#Getting the community features
comFeatures_list <- gsub(' - 1', '', comFeatures_listOrg)

datalist = list()
vec <- 1:length(comFeatures_list)
for (i in seq_along(vec)) {

  var <- comFeatures_list[i]
  #df1 <- comFeatures_df[ , grepl(comFeatures_list[i],names(comFeatures_df)) ]
  df1 <- comFeatures_df[ , grepl(var,names(comFeatures_df)) ]
  df1[is.na(df1)]   <- " " #Converting NAs to ""
  df1 <- mutate(df1, var = apply(df1[ , colnames(df1) ] , 1 , paste , collapse = "" )) %>% select(var)
  df1 <- data.frame(lapply(df1, trimws), stringsAsFactors = TRUE)
  #df1 <- cbind(df1, ID = comFeatures_df$ID)
  names(df1)[names(df1) == 'var'] <- var

  datalist[[i]] <- df1
}

cleaned_MostVisitShop <- do.call(cbind, datalist)


##Removing the empty row
#Assigning NA to the empty cell
cleaned_MostVisitShop <- cleaned_MostVisitShop %>% mutate_all(na_if,"")
cleaned_MostVisitShop <- na.omit(cleaned_MostVisitShop)


##Manipulation for plotting
datalist4 = list()
i <- 1

#Removing the NA and Empty rows
for (colname in colnames(cleaned_MostVisitShop)){

  colname <- as.character(colname)
  df <- cleaned_MostVisitShop %>% select(colname)
  # df <- df[!(is.na(df$colname) | df$colname == ""), ]
  tbl = table(df)
  freqDFMostShopVisit <- as.data.frame(tbl)

  #Renaming the 'df' column with the community feature
  names(freqDFMostShopVisit)[names(freqDFMostShopVisit) == 'df'] <- paste0("Rank - ", colname)
  names(freqDFMostShopVisit)[names(freqDFMostShopVisit) == 'Freq'] <- colname

  ##Removing the empty row
  #Assigning NA to the empty cell
```

```
  freqDFMostShopVisit2 <- freqDFMostShopVisit %>% mutate_all(na_if,"")
  #Removing the NA
  freqDFMostShopVisit2 <- na.omit(freqDFMostShopVisit2)

  #Appending to list
  datalist4[[i]] <- freqDFMostShopVisit2

  i <- i+1

}

#Getting the full dataframe
freqDFMostShopVisit3 = do.call(cbind, datalist4)

#Removing redundant "Community_Feature_Rank" columns
freqDFMostShopVisit4 <- freqDFMostShopVisit3 %>% select(colnames(cleaned_MostVisitShop))
freqDFMostShopVisit4$Rank <- unique(freqDFMostShopVisit3$`Rank - Lee Vista Area`)
freqDFMostShopVisit4 <- freqDFMostShopVisit4 %>% select(Rank, everything())

#Converting the dataframe to long format for plotting purposes
long <- melt(freqDFMostShopVisit4, id.vars = c("Rank"))
# long_fltd <- long %>% filter(Rank == 'Agree')
# long_fltd2 <- long %>% group_by(Rank, variable) %>% summarise(total_count = sum(value))

#Plotting the data
p5=ggplot(long, aes(fill=Rank, y=value, x=variable)) +
  geom_bar(position="dodge", stat="identity") +
  theme(axis.text.x = element_text(angle = 90)) +
  geom_text(aes(label=paste0(value)), colour="black", vjust = 0, position = position_dodge(0.9), size=3
p5
```



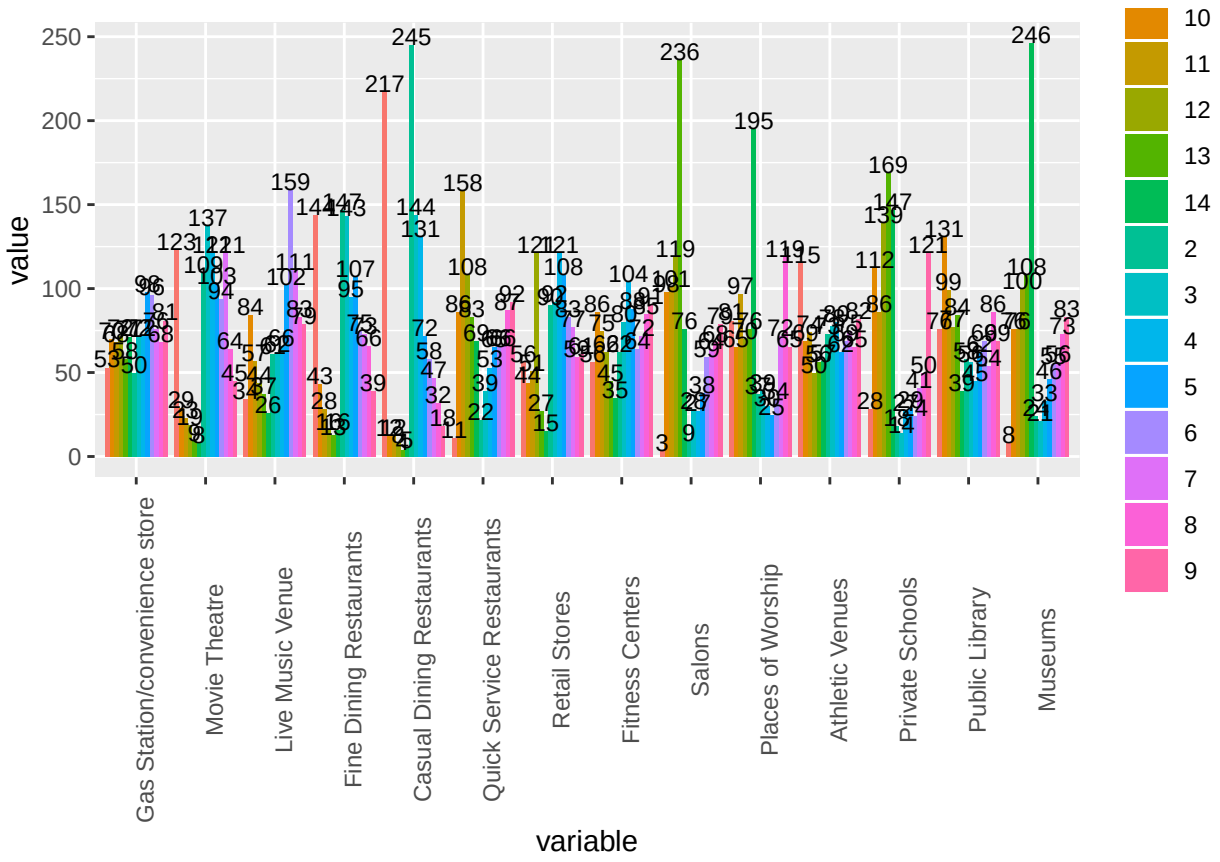
As per the above graph, the "Winter Park Village" and "Winter Park - Park Avenue" are ranked the least visited as per the residents. On the other hand, "Disney Springs" and "Lee Vista Area" are among the most visited shopping destinations as per the residents.

4.) Amenities in order of importance

```
##-------------------------Amenities in order of importance-----------------------------------------
headers <- read.csv("rank_data/AmenitiesInOrderOfImportance.csv",
                    skip = 1,
                    header = F,
                    nrows = 1,
                    as.is = T)

comFeatures_df = read.csv("rank_data/AmenitiesInOrderOfImportance.csv", skip = 3, header = F)
colnames(comFeatures_df) <-  headers
comFeatures_df$ID <- seq.int(nrow(comFeatures_df))

#SST_list <- colnames(df)
comFeatures_listOrg <- colnames(comFeatures_df)[grep("- 14", colnames(comFeatures_df))]

#Getting the community features
comFeatures_list <- gsub("- 14", '', comFeatures_listOrg)

datalist = list()
vec <- 1:length(comFeatures_list)
```

```r
for (i in seq_along(vec)) {

  var <- comFeatures_list[i]
  #df1 <- comFeatures_df[ , grepl(comFeatures_list[i],names(comFeatures_df)) ]
  df1 <- comFeatures_df[ , grepl(var,names(comFeatures_df)) ]
  df1[is.na(df1)]   <- " " #Converting NAs to ""
  df1 <- mutate(df1, var = apply(df1[ , colnames(df1) ] , 1 , paste , collapse = "" )) %>% select(var)
  df1 <- data.frame(lapply(df1, trimws), stringsAsFactors = TRUE)
  #df1 <- cbind(df1, ID = comFeatures_df$ID)
  names(df1)[names(df1) == 'var'] <- var

  datalist[[i]] <- df1
}

cleaned_AmenOrderImp <- do.call(cbind, datalist)

##Removing the empty row
#Assigning NA to the empty cell
cleaned_AmenOrderImp <- cleaned_AmenOrderImp %>% mutate_all(na_if,"")
cleaned_AmenOrderImp <- na.omit(cleaned_AmenOrderImp)

##Manipulation for plotting
datalist5 = list()
i <- 1

#Removing the NA and Empty rows
for (colname in colnames(cleaned_AmenOrderImp)){

  colname <- as.character(colname)
  df <- cleaned_AmenOrderImp %>% select(colname)
  # df <- df[!(is.na(df$colname) | df$colname == ""), ]
  tbl = table(df)
  freqDFAmenOrderImp <- as.data.frame(tbl)

  #Renaming the 'df' column with the community feature
  names(freqDFAmenOrderImp)[names(freqDFAmenOrderImp) == 'df'] <- paste0("Rank - ", colname)
  names(freqDFAmenOrderImp)[names(freqDFAmenOrderImp) == 'Freq'] <- colname

  ##Removing the empty row
  #Assigning NA to the empty cell
  freqDFAmenOrderImp2 <- freqDFAmenOrderImp %>% mutate_all(na_if,"")
  #Removing the NA
  freqDFAmenOrderImp2 <- na.omit(freqDFAmenOrderImp2)

  #Appending to list
  datalist5[[i]] <- freqDFAmenOrderImp2

  i <- i+1

}

#Getting the full dataframe
freqDFAmenOrderImp3 = do.call(cbind, datalist5)
```

```
#Removing redundant "Community_Feature_Rank" columns
freqDFAmenOrderImp4 <- freqDFAmenOrderImp3 %>% select(colnames(cleaned_AmenOrderImp))
freqDFAmenOrderImp4$Rank <- unique(freqDFAmenOrderImp3$`Rank - Movie Theatre `)
freqDFAmenOrderImp4 <- freqDFAmenOrderImp4 %>% select(Rank, everything())

#Converting the dataframe to long format for plotting purposes
long <- melt(freqDFAmenOrderImp4, id.vars = c("Rank"))
# long_fltd <- long %>% filter(Rank == 'Agree')
# long_fltd2 <- long %>% group_by(Rank, variable) %>% summarise(total_count = sum(value))

#Plotting the data
p6=ggplot(long, aes(fill=Rank, y=value, x=variable)) +
  geom_bar(position="dodge", stat="identity") +
  theme(axis.text.x = element_text(angle = 90)) +
  geom_text(aes(label=paste0(value)), colour="black", vjust = 0, position = position_dodge(0.9), size=3)
p6
```



5.) Community ammenities used within the last year

```
##------------------------Community Ammenities utilized within last 12 months----------------------------
# headers <- read.csv("rank_data/CommunityAmmenitiesUtilizedPast12.csv",
#                     skip = 1,
#                     header = F,
```

```
#                       nrows = 1,
#                       as.is = T)
#
# comFeatures_df = read.csv("rank_data/CommunityAmmenitiesUtilizedPast12.csv", skip = 3, header = F)
# colnames(comFeatures_df) <-  headers
```

(3.) Fitness related questions

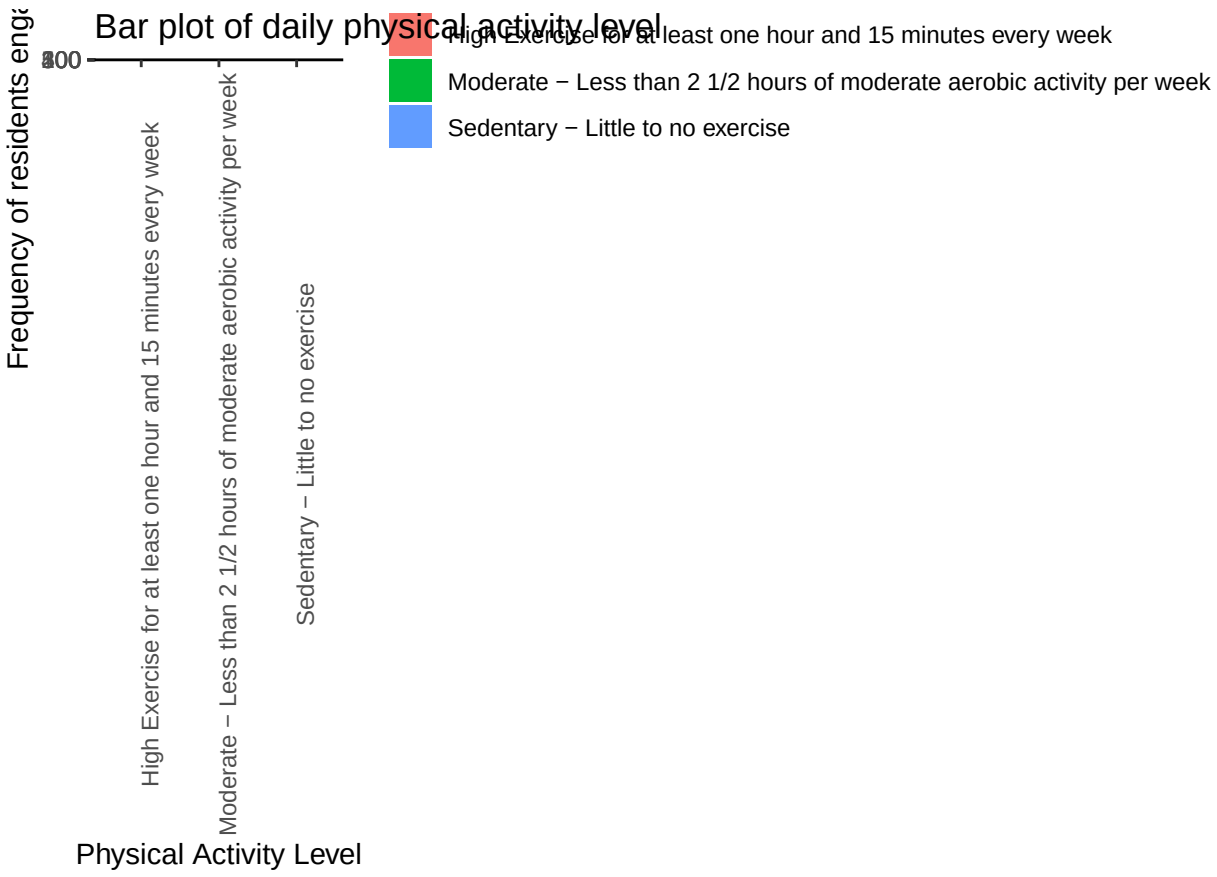
```
#Fitness data extraction
data_fitness <- data %>% select(X.Yes.No..Attended.any.of.the.LP.Fit.group.exercise.classes..Yoga.in.La
                                X.Multiple.answer.question..If.the.schedule.is.inhibiting.you.from.atte
                                X.Yes.No..If.you.were.not.already.aware..these.classes.are.now.complime
                                App.use.for.track.fitness,
                                Brand.of.wearable.fitness.tracker,
                                Daily.Physical.Activity.level,
                                Excercise.locations,
                                List.of.Gyms.Fitness.Facility.with.membership,
                                X.Yes.No..Is.your.Lake.Nona.residence.your.first.home.purchase.,
                                Neighborhood.within.LakeNona)

colnames(data_fitness) <- c("Attended_LP_FitGroup_Class",
                            "Better_Day_LP_FitGroup_Class",
                            "Attend_If_Complementary_Laureate Park",
                            "App_Used_Track_Fitness",
                            "Brand_Of_Fitness_Tracker",
                            "Daily_Physical_Activity_Level",
                            "Excercise_Location",
                            "Gyms_Where_A_Member",
                            "Is_LakeNona_First_Home_Purchase",
                            "Neighbourhood_Within_LakeNona")
```

In this section, the survey questions associated with the fitness of each resident will be analysed.

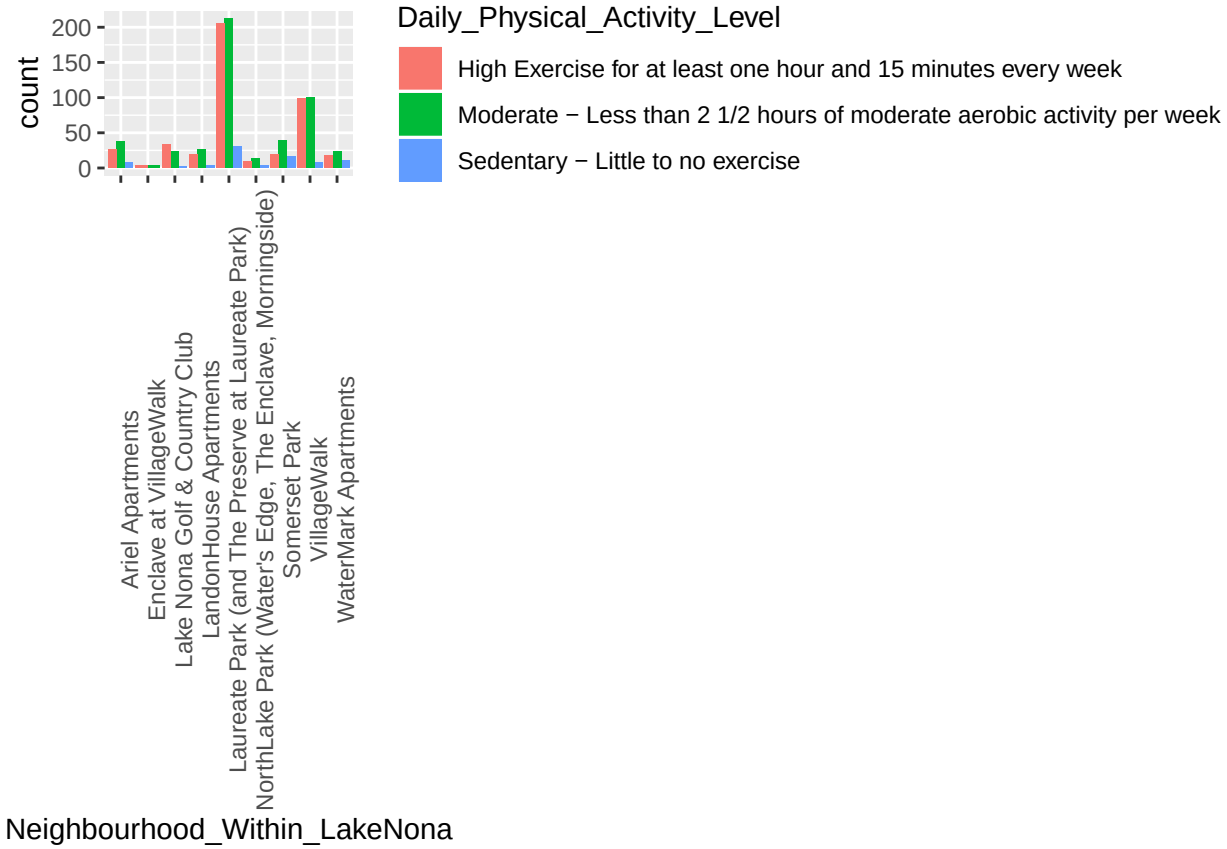
#1.) Daily physical activity level

```
#4. What is your daily physical activity level?

#We can use this as a start off analysis into the physical section - More like a general question
data_phyActLvl <- data_fitness %>% select(Daily_Physical_Activity_Level, Neighbourhood_Within_LakeNona)

##Removing the empty row
#Assigning NA to the empty cell
data_phyActLvl <- data_phyActLvl %>% mutate_all(na_if,"")
#Removing the NA
data_phyActLvl <- na.omit(data_phyActLvl)


#Getting the count of each activity level
cnt_data_phyActLvl <- data_phyActLvl %>%
  dplyr::group_by(Daily_Physical_Activity_Level) %>%
  dplyr::summarise(count = n())
```

```r
#Plotting in a graph
ggplot(cnt_data_phyActLvl,aes(x = Daily_Physical_Activity_Level,
                          y = count,
                          fill = Daily_Physical_Activity_Level)) +
  geom_bar(stat = "identity") +
  theme_classic() +
  labs(
    x = "Physical Activity Level",
    y = "Frequency of residents engaged in each activity",
    title = paste("Bar plot of daily physical activity level"),
    fill = "Physical Activity Level"
  )+
  theme(axis.text.x = element_text(angle = 90))
```




```r
#Combining with the neighbourhood in LakeNona
ggplot(data_phyActLvl, aes(Neighbourhood_Within_LakeNona, ..count..)) +
  geom_bar(aes(fill = Daily_Physical_Activity_Level), position = "dodge") +
  theme(axis.text.x = element_text(angle = 90))
```



2.) Where do you excercise/ Excercise Location

```
#5. Where do you exercise? Please select all that apply.

#Stemming from the above question 4, we can look into the places the people with higher physical activi
data_excPlace <- data_fitness %>% select(Daily_Physical_Activity_Level,
                                         Neighbourhood_Within_LakeNona,
                                         Excercise_Location)

#Getting the long form
data_excPlace_long <- data_excPlace %>%
  separate_rows(Excercise_Location, sep = ",", convert = TRUE) %>%
  drop_na()

#Filtering people who excercise in a high and moderate level
fltd_excPlace_long <- data_excPlace_long %>% filter(Daily_Physical_Activity_Level=="High Exercise for a
                                                    Daily_Physical_Activity_Level=="Moderate - Less th

##Removing the empty row
#Assigning NA to the empty cell
fltd_excPlace_long <- fltd_excPlace_long %>% mutate_all(na_if,"")
#Removing the NA
fltd_excPlace_long <- na.omit(fltd_excPlace_long)
```

```r
#Getting the dataframe which shows the count>5 for the excercise locations of each neighbourhood in Lak
top_ExcLocations_WithinEachNeighbourhood <- fltd_excPlace_long %>% group_by(Neighbourhood_Within_LakeNo
  dplyr::summarise(count=n()) %>%
  group_by(Neighbourhood_Within_LakeNona) %>%
  filter(count >=5 )

topExc_Locations_list <- top_ExcLocations_WithinEachNeighbourhood$Excercise_Location

#Filtering the fltd_excPlace_long based on this top excercise location list
fltd_top_ExcLocations_WithinEachNeighbourhood <- fltd_excPlace_long %>%
  filter(Excercise_Location %in% topExc_Locations_list)

#Plot of the excercise location of high excercise people in each Neighbourhood in Lake Nona
ggplot(fltd_top_ExcLocations_WithinEachNeighbourhood, aes(Neighbourhood_Within_LakeNona, ..count..)) +
  geom_bar(aes(fill = Excercise_Location), position = "dodge") +
  theme(axis.text.x = element_text(angle = 90))
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood’s fitness facility' in
## 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood’s fitness facility' in
## 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood’s fitness facility' in
## 'mbcsToSbcs': dot substituted for <99>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood’s fitness facility' in
## 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood’s fitness facility' in
## 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood’s fitness facility' in
## 'mbcsToSbcs': dot substituted for <99>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood’s fitness facility' in
## 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood’s fitness facility' in
## 'mbcsToSbcs': dot substituted for <80>
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood's fitness facility' in
## 'mbcsToSbcs': dot substituted for <99>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood's fitness facility' in
## 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood's fitness facility' in
## 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood's fitness facility' in
## 'mbcsToSbcs': dot substituted for <99>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood's fitness facility' in
## 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood's fitness facility' in
## 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood's fitness facility' in
## 'mbcsToSbcs': dot substituted for <99>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <93>
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <e2>
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood’s fitness facility' in
## 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood’s fitness facility' in
## 'mbcsToSbcs': dot substituted for <80>
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood’s fitness facility' in
## 'mbcsToSbcs': dot substituted for <99>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood’s fitness facility' in
## 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood’s fitness facility' in
## 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood’s fitness facility' in
## 'mbcsToSbcs': dot substituted for <99>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood’s fitness facility' in
## 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood’s fitness facility' in
## 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood’s fitness facility' in
## 'mbcsToSbcs': dot substituted for <99>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood’s fitness facility' in
## 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood’s fitness facility' in
## 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood’s fitness facility' in
## 'mbcsToSbcs': dot substituted for <99>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood’s fitness facility' in
## 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood’s fitness facility' in
## 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood’s fitness facility' in
## 'mbcsToSbcs': dot substituted for <99>
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <e2>
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <80>
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood’s fitness facility' in
## 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood’s fitness facility' in
## 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood’s fitness facility' in
## 'mbcsToSbcs': dot substituted for <99>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood’s fitness facility' in
## 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood’s fitness facility' in
## 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood’s fitness facility' in
## 'mbcsToSbcs': dot substituted for <99>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood’s fitness facility' in
## 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood’s fitness facility' in
## 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood’s fitness facility' in
## 'mbcsToSbcs': dot substituted for <99>
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood’s fitness facility' in
## 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood’s fitness facility' in
## 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood’s fitness facility' in
## 'mbcsToSbcs': dot substituted for <99>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood’s fitness facility' in
## 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood’s fitness facility' in
## 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood’s fitness facility' in
## 'mbcsToSbcs': dot substituted for <99>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood’s fitness facility' in
## 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood’s fitness facility' in
## 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood’s fitness facility' in
## 'mbcsToSbcs': dot substituted for <99>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood’s fitness facility' in
## 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood’s fitness facility' in
## 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood’s fitness facility' in
## 'mbcsToSbcs': dot substituted for <99>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood’s fitness facility' in
## 'mbcsToSbcs': dot substituted for <e2>
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood's fitness facility' in
## 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood's fitness facility' in
## 'mbcsToSbcs': dot substituted for <99>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood's fitness facility' in
## 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood's fitness facility' in
## 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood's fitness facility' in
## 'mbcsToSbcs': dot substituted for <99>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood's fitness facility' in
## 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood's fitness facility' in
## 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood's fitness facility' in
## 'mbcsToSbcs': dot substituted for <99>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood's fitness facility' in
## 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood's fitness facility' in
## 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood's fitness facility' in
## 'mbcsToSbcs': dot substituted for <99>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood's fitness facility' in
## 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood's fitness facility' in
## 'mbcsToSbcs': dot substituted for <80>
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood's fitness facility' in
## 'mbcsToSbcs': dot substituted for <99>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood's fitness facility' in
## 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood's fitness facility' in
## 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood's fitness facility' in
## 'mbcsToSbcs': dot substituted for <99>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood's fitness facility' in
## 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood's fitness facility' in
## 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood's fitness facility' in
## 'mbcsToSbcs': dot substituted for <99>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood's fitness facility' in
## 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood's fitness facility' in
## 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'At my neighborhood's fitness facility' in
## 'mbcsToSbcs': dot substituted for <99>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <93>
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <e2>
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <80>
```

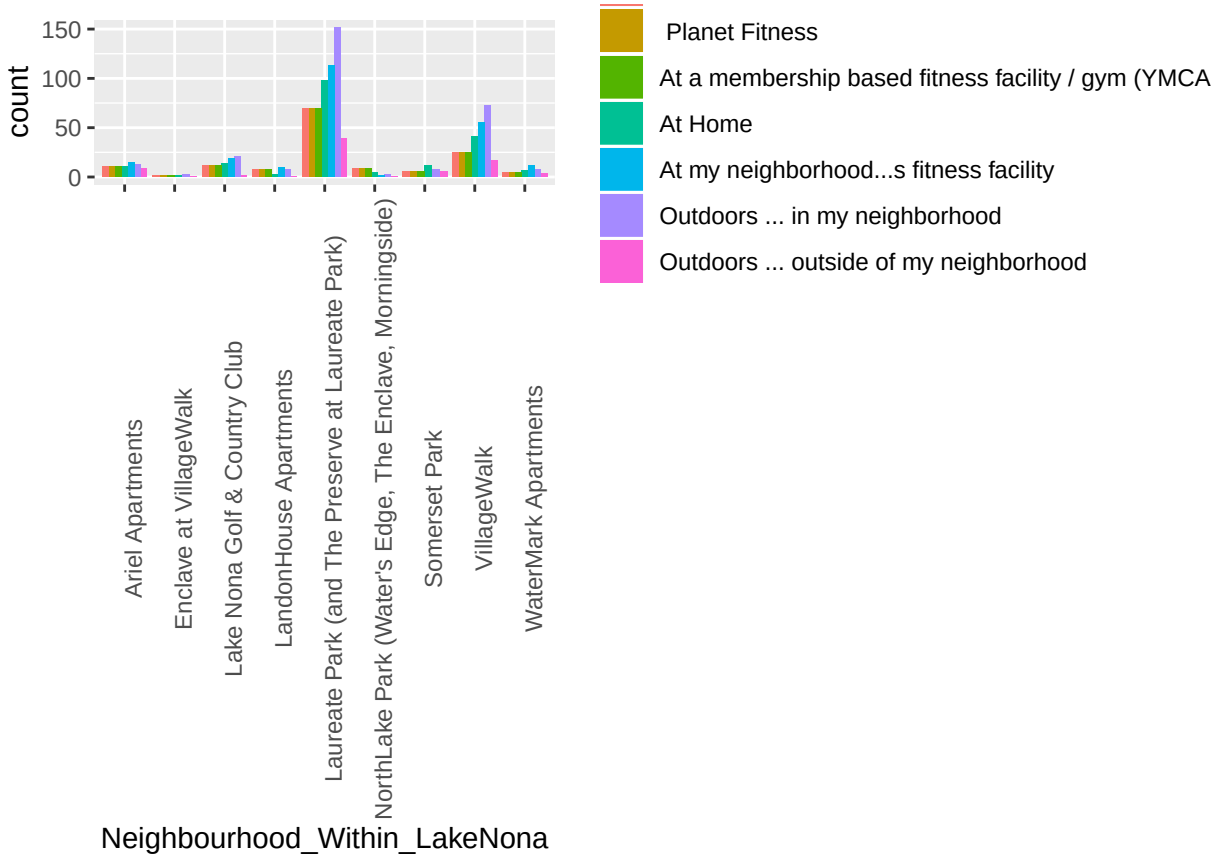
```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Outdoors - in my neighborhood' in 'mbcsToSbcs': dot
## substituted for <93>
```

```
## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label),
## x$x, x$y, : conversion failure on 'Outdoors - in my neighborhood' in
## 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label),
## x$x, x$y, : conversion failure on 'Outdoors - in my neighborhood' in
## 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label),
## x$x, x$y, : conversion failure on 'Outdoors - in my neighborhood' in
## 'mbcsToSbcs': dot substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <e2>
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <80>
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <93>
```

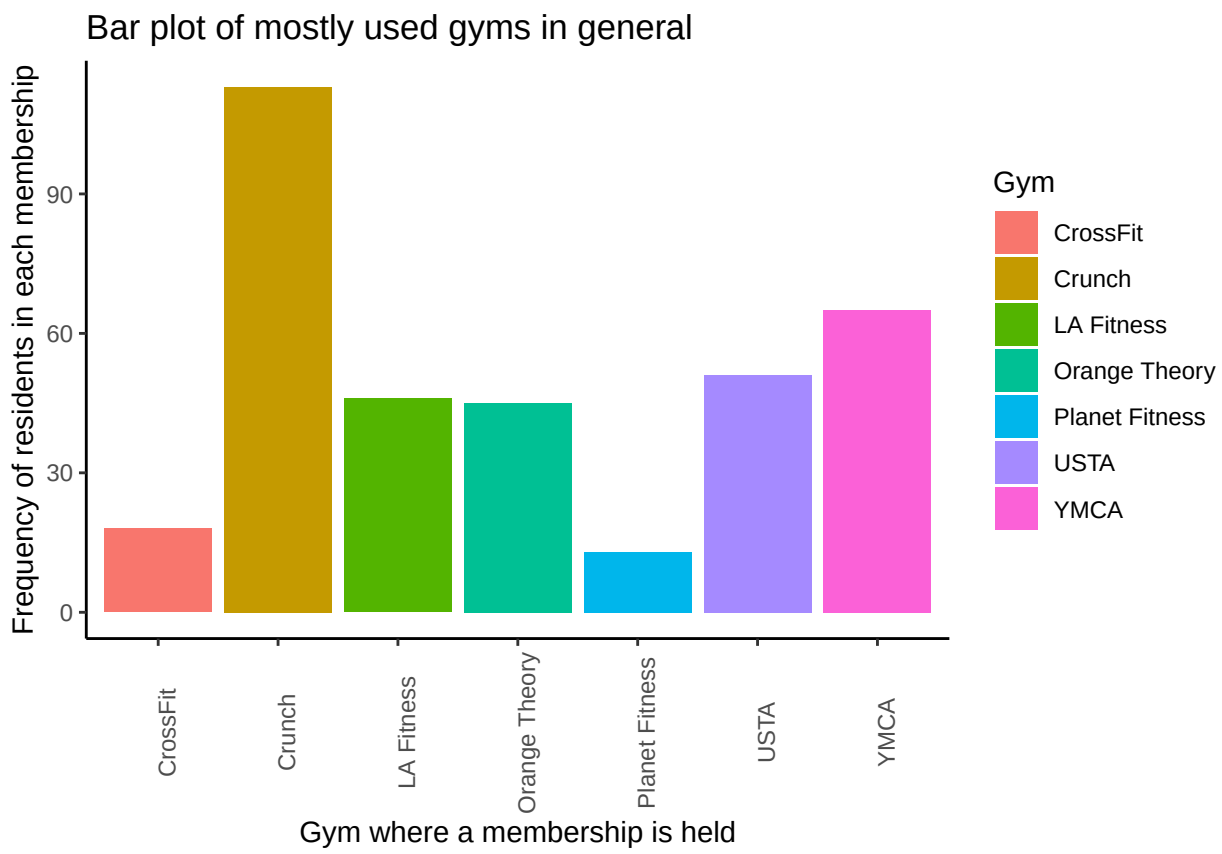
```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <93>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x,
## x$y, : conversion failure on 'Outdoors - outside of my neighborhood' in
## 'mbcsToSbcs': dot substituted for <93>
```



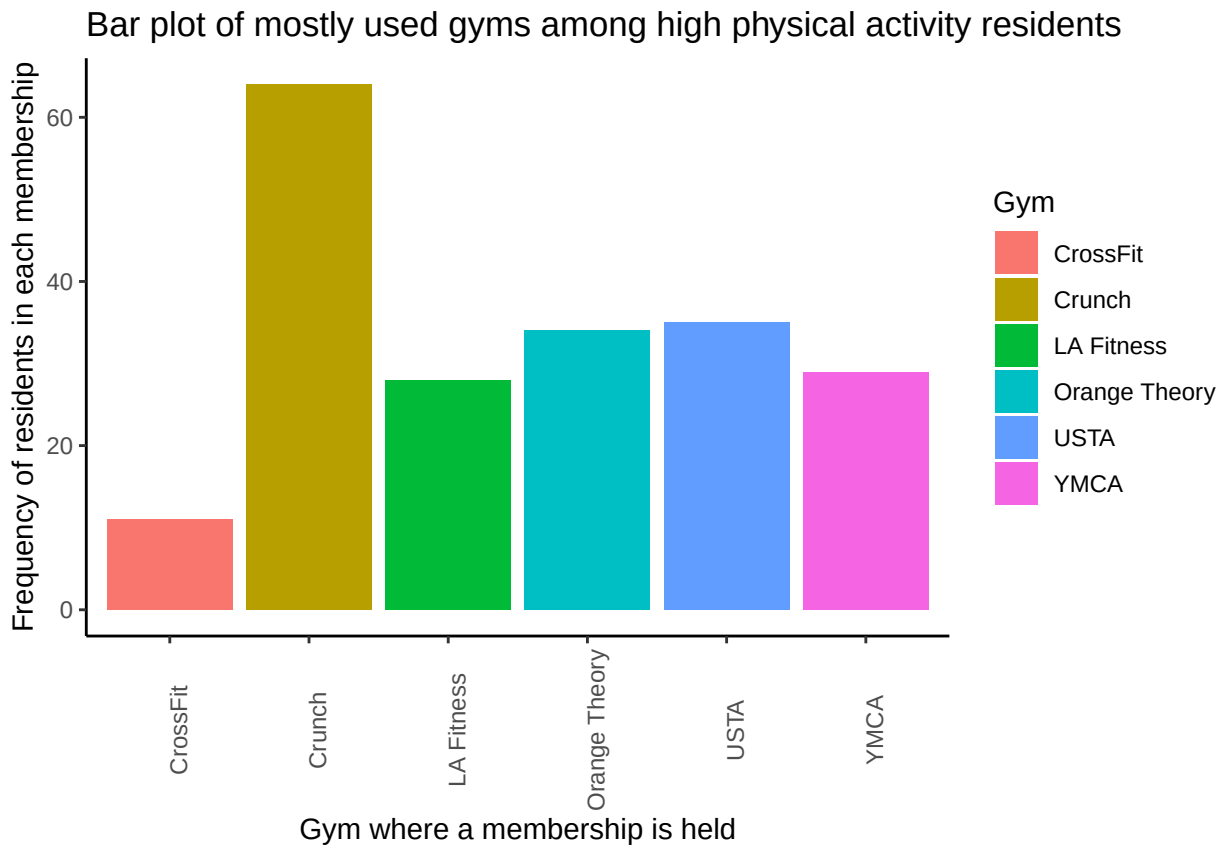
3.) Which gyms are you a member of

```
#7. Which fitness facility or gym do you have a membership to? Please select all that apply.

#Here we can see first - In general what is the gyms mostly used by the residents
#Secondly can see - for those with higher physical activity level - What gym/gyms do they use more ofte

data_gym <- data_fitness %>% select(Daily_Physical_Activity_Level,
                                    Neighbourhood_Within_LakeNona,
                                    Gyms_Where_A_Member)

#Getting the long form
data_gym_long <- data_gym %>%
  separate_rows(Gyms_Where_A_Member, sep = ",", convert = TRUE) %>%
  drop_na()

##Removing the empty row
#Assigning NA to the empty cell
data_gym_long <- data_gym_long %>% mutate_all(na_if,"")
#Removing the NA
data_gym_long <- na.omit(data_gym_long)

#Replacing the duplicate categorical variable in the "Gyms_Where_A_Member" column
```

```r
# data_gym_long$Gyms_Where_A_Member <- str_replace_all(data_gym_long$Gyms_Where_A_Member, "Crunchy", "C
data_gym_long$Gyms_Where_A_Member <- as.character(data_gym_long$Gyms_Where_A_Member)
data_gym_long$Gyms_Where_A_Member[grepl("Crunch", data_gym_long$Gyms_Where_A_Member, ignore.case=TRUE)]

#First showing the gyms that are mostly used
mostUsedGyms_general <- data_gym_long %>%
  group_by(Gyms_Where_A_Member) %>%
  dplyr::summarise(count=n()) %>%
  filter(Gyms_Where_A_Member != "I do not have a membership to a fitness facility") %>%
  filter(count>=13)

#Plotting in a graph
ggplot(mostUsedGyms_general,aes(x = Gyms_Where_A_Member,
                                y = count,
                                fill = Gyms_Where_A_Member)) +
  geom_bar(stat = "identity") +
  theme_classic() +
  labs(
    x = "Gym where a membership is held",
    y = "Frequency of residents in each membership",
    title = paste("Bar plot of mostly used gyms in general"),
    fill = "Gym"
  )+
  theme(axis.text.x = element_text(angle = 90))
```



```r
#Filtering out the gyms used by those who have a hogh physical activity level
data_gym_long_highActivity <- data_gym_long %>% filter(Daily_Physical_Activity_Level=="High Exercise fo
                                                       Daily_Physical_Activity_Level=="Moderate - Less

#Getting the most used gyms among these
mostUsedGyms_highActivity <- data_gym_long_highActivity %>%
  group_by(Gyms_Where_A_Member) %>%
  dplyr::summarise(count=n()) %>%
  filter(Gyms_Where_A_Member != "I do not have a membership to a fitness facility") %>%
  filter(count>=11)

#Plotting in a graph
ggplot(mostUsedGyms_highActivity,aes(x = Gyms_Where_A_Member,
                                     y = count,
                                     fill = Gyms_Where_A_Member)) +
  geom_bar(stat = "identity") +
  theme_classic() +
  labs(
    x = "Gym where a membership is held",
    y = "Frequency of residents in each membership",
    title = paste("Bar plot of mostly used gyms among high physical activity residents"),
    fill = "Gym"
  )+
  theme(axis.text.x = element_text(angle = 90))
```



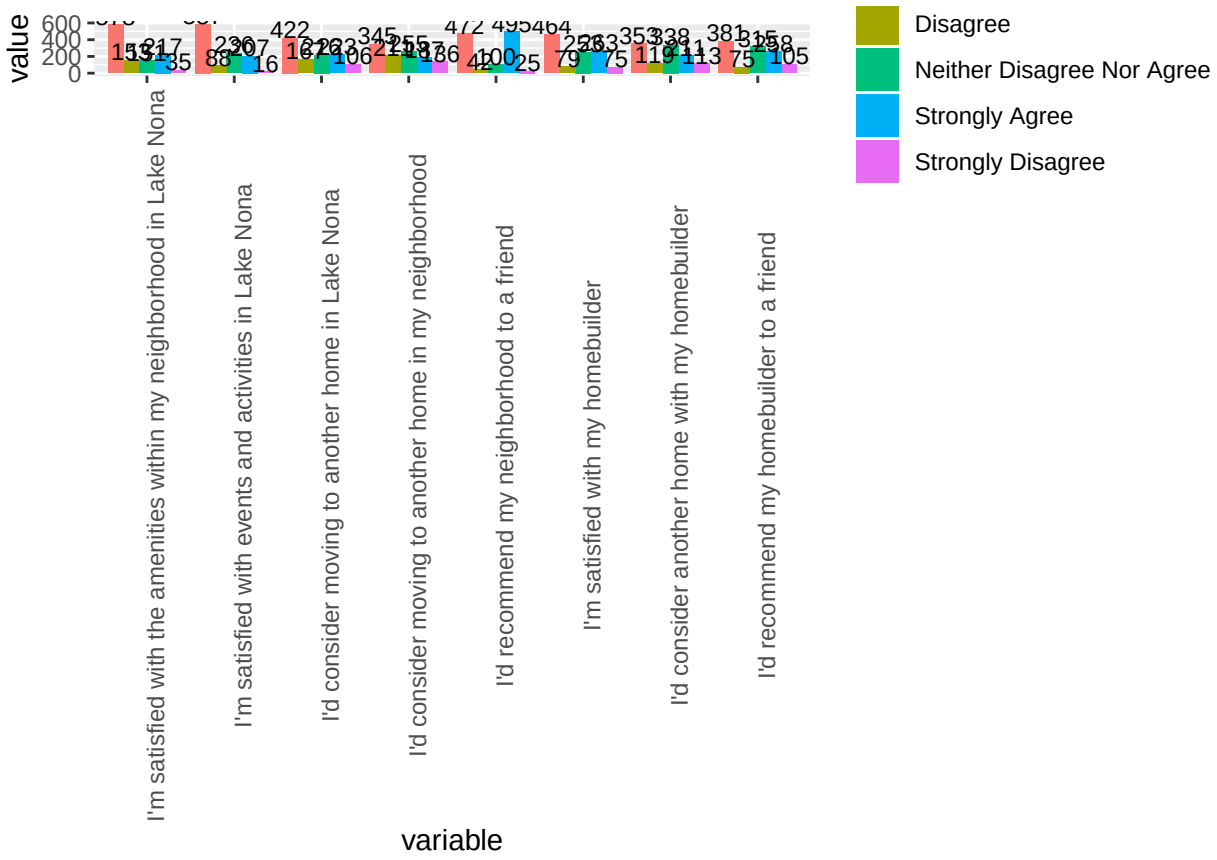
(4.) General Questions

1.) Satisfaction level with different features within Lake Nona

The survey has also inquired regarding the satisfaction levels each resident held in terms of various random components and aspects within Lake Nona. The analysis on the responses given by the residents for these questions are as follows:

```
##-----------------------Satisfaction level in Lake Nona-------------------------------------------------
headers <- read.csv("rank_data/LakeNonaSatisfaction.csv",
                    skip = 1,
                    header = F,
                    nrows = 1,
                    as.is = T)

comFeatures_df = read.csv("rank_data/LakeNonaSatisfaction.csv", skip = 3, header = F)
colnames(comFeatures_df) <-  headers
comFeatures_df$ID <- seq.int(nrow(comFeatures_df))

#SST_list <- colnames(df)
comFeatures_listOrg <- colnames(comFeatures_df)[grep("Neither", colnames(comFeatures_df))]

#Getting the community features
comFeatures_list <- gsub(' - Neither Disagree Nor Agree', '', comFeatures_listOrg)

datalist = list()
vec <- 1:length(comFeatures_list)
for (i in seq_along(vec)) {

  var <- comFeatures_list[i]
  df1 <- comFeatures_df[ , grepl(var,names(comFeatures_df)) ]
  df1 <- mutate(df1, var = apply(df1[ , colnames(df1) ] , 1 , paste , collapse = "" )) %>% select(var)
  #df1 <- cbind(df1, ID = comFeatures_df$ID)
  names(df1)[names(df1) == 'var'] <- var

  datalist[[i]] <- df1
}

SatisfactionLevelLakeNona <- do.call(cbind, datalist)

##Removing the empty row
#Assigning NA to the empty cell
SatisfactionLevelLakeNona <- SatisfactionLevelLakeNona %>% mutate_all(na_if,"") %>%
  select(-`I'm satisfied with the recreational amenities in Lake Nona (parks, playgrounds, etc)`)
#Removing the NA
SatisfactionLevelLakeNona <- na.omit(SatisfactionLevelLakeNona)

##Manipulation for plotting
datalist3 = list()
i <- 1

#Removing the NA and Empty rows
for (colname in colnames(SatisfactionLevelLakeNona)){
```

```
  colname <- as.character(colname)
  df <- SatisfactionLevelLakeNona %>% select(colname)
  # df <- df[!(is.na(df$colname) | df$colname == ""), ]
  tbl = table(df)
  freqDFSatLevel <- as.data.frame(tbl)

  #Renaming the 'df' column with the community feature
  names(freqDFSatLevel)[names(freqDFSatLevel) == 'df'] <- paste0("Rank - ", colname)
  names(freqDFSatLevel)[names(freqDFSatLevel) == 'Freq'] <- colname

  ##Removing the empty row
  #Assigning NA to the empty cell
  freqDFSatLevel2 <- freqDFSatLevel %>% mutate_all(na_if,"")
  #Removing the NA
  freqDFSatLevel2 <- na.omit(freqDFSatLevel2)

  #Appending to list
  datalist3[[i]] <- freqDFSatLevel2

  i <- i+1

}

#Getting the full dataframe
freqDFSatLevel3 = do.call(cbind, datalist3)

#Removing redundant "Community_Feature_Rank" columns
freqDFSatLevel4 <- freqDFSatLevel3 %>% select(colnames(SatisfactionLevelLakeNona))
freqDFSatLevel4$Rank <- unique(freqDFSatLevel3$`Rank - I'm satisfied with my homebuilder`)
freqDFSatLevel4 <- freqDFSatLevel4 %>% select(Rank, everything())

#Converting the dataframe to long format for plotting purposes
long <- melt(freqDFSatLevel4, id.vars = c("Rank"))
# long_fltd <- long %>% filter(Rank == 'Agree')
# long_fltd2 <- long %>% group_by(Rank, variable) %>% summarise(total_count = sum(value))

#Plotting the data
p4=ggplot(long, aes(fill=Rank, y=value, x=variable)) +
  geom_bar(position="dodge", stat="identity") +
  theme(axis.text.x = element_text(angle = 90)) +
  geom_text(aes(label=paste0(value)), colour="black", vjust = 0, position = position_dodge(0.9), size=3)
p4
```



The above graph shows the satisfaction levels in each aspet inquired regarding Lake Nona. As per the graph, many have shown positive remarks in terms of their satisfaction level regarding the amenities and events in Lake Nona.