APS Odd Semester 2024 Coding Problem Set-2

library(ggplot2)  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

# Load the house price dataset  
hData = read.csv('Data/houseprices.csv', header = TRUE, stringsAsFactors = FALSE, na.strings = c("", "NA", "Not Available", "not available"))  
str(hData)

## 'data.frame': 225 obs. of 8 variables:  
## $ locality : chr "BTM Layout" "BTM Layout" "BTM Layout" "BTM Layout" ...  
## $ area : int 565 1837 1280 2220 1113 1332 1815 1400 3006 1600 ...  
## $ rent : int 20060 97434 54448 117000 34388 36394 112000 41266 129000 92849 ...  
## $ price\_per\_sqft: int 6195 9254 7422 9234 5391 4767 10744 5143 7485 10125 ...  
## $ facing : chr "North-West" "East" "East" "North" ...  
## $ BHK : int 1 3 2 3 2 2 3 2 4 3 ...  
## $ bathrooms : int 1 3 2 3 2 2 2 2 5 2 ...  
## $ parking : chr "Bike" "Bike and Car" "Car" "Bike and Car" ...

# Create a New Data column House\_Value = area\*price\_per\_sqft which can we used to make buy, rent decision   
hData = hData %>% mutate(House\_Value = area\*price\_per\_sqft)  
str(hData)

## 'data.frame': 225 obs. of 9 variables:  
## $ locality : chr "BTM Layout" "BTM Layout" "BTM Layout" "BTM Layout" ...  
## $ area : int 565 1837 1280 2220 1113 1332 1815 1400 3006 1600 ...  
## $ rent : int 20060 97434 54448 117000 34388 36394 112000 41266 129000 92849 ...  
## $ price\_per\_sqft: int 6195 9254 7422 9234 5391 4767 10744 5143 7485 10125 ...  
## $ facing : chr "North-West" "East" "East" "North" ...  
## $ BHK : int 1 3 2 3 2 2 3 2 4 3 ...  
## $ bathrooms : int 1 3 2 3 2 2 2 2 5 2 ...  
## $ parking : chr "Bike" "Bike and Car" "Car" "Bike and Car" ...  
## $ House\_Value : int 3500175 16999598 9500160 20499480 6000183 6349644 19500360 7200200 22499910 16200000 ...

# create a New Data Column Decision either to buy or rent the house based on the standard house Price-to-Rent Ratio  
hData = hData %>% mutate(Decision = ifelse((House\_Value / (rent \* 12)) < 15, "Rent", "Buy"))  
str(hData)

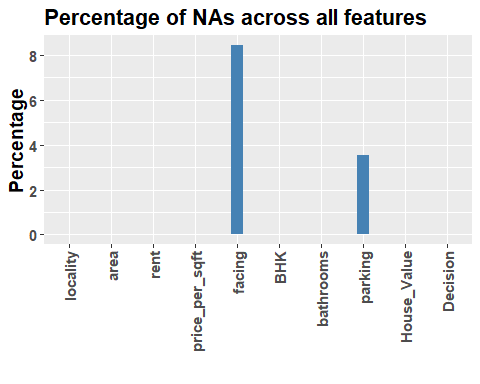
## 'data.frame': 225 obs. of 10 variables:  
## $ locality : chr "BTM Layout" "BTM Layout" "BTM Layout" "BTM Layout" ...  
## $ area : int 565 1837 1280 2220 1113 1332 1815 1400 3006 1600 ...  
## $ rent : int 20060 97434 54448 117000 34388 36394 112000 41266 129000 92849 ...  
## $ price\_per\_sqft: int 6195 9254 7422 9234 5391 4767 10744 5143 7485 10125 ...  
## $ facing : chr "North-West" "East" "East" "North" ...  
## $ BHK : int 1 3 2 3 2 2 3 2 4 3 ...  
## $ bathrooms : int 1 3 2 3 2 2 2 2 5 2 ...  
## $ parking : chr "Bike" "Bike and Car" "Car" "Bike and Car" ...  
## $ House\_Value : int 3500175 16999598 9500160 20499480 6000183 6349644 19500360 7200200 22499910 16200000 ...  
## $ Decision : chr "Rent" "Rent" "Rent" "Rent" ...

# Convert 'locality', 'facing' and 'parking' columns to factors  
categorical\_cols = c('locality', 'facing', 'parking', 'Decision')  
hData[categorical\_cols] = lapply(hData[categorical\_cols], as.factor)  
str(hData)

## 'data.frame': 225 obs. of 10 variables:  
## $ locality : Factor w/ 9 levels "Attibele","BTM Layout",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ area : int 565 1837 1280 2220 1113 1332 1815 1400 3006 1600 ...  
## $ rent : int 20060 97434 54448 117000 34388 36394 112000 41266 129000 92849 ...  
## $ price\_per\_sqft: int 6195 9254 7422 9234 5391 4767 10744 5143 7485 10125 ...  
## $ facing : Factor w/ 7 levels "East","North",..: 4 1 1 2 1 7 3 6 1 5 ...  
## $ BHK : int 1 3 2 3 2 2 3 2 4 3 ...  
## $ bathrooms : int 1 3 2 3 2 2 2 2 5 2 ...  
## $ parking : Factor w/ 3 levels "Bike","Bike and Car",..: 1 2 3 2 2 2 3 2 2 2 ...  
## $ House\_Value : int 3500175 16999598 9500160 20499480 6000183 6349644 19500360 7200200 22499910 16200000 ...  
## $ Decision : Factor w/ 1 level "Rent": 1 1 1 1 1 1 1 1 1 1 ...

# Continuous columns  
continuous\_cols = setdiff(colnames(hData), categorical\_cols)

# Plot percentage of NAs in each column of the data frame  
hData\_NA = setNames(stack(sapply(hData, function(x){(sum(is.na(x))/length(x))\*100}))[2:1], c('Feature','Value'))  
p = ggplot(data = hData\_NA, aes(x = Feature, y = Value)) +  
 geom\_bar(stat = 'identity', fill = 'steelblue', width = 0.3) +  
 theme(text = element\_text(size = 14, face = 'bold'),  
 axis.text.x = element\_text(angle = 90, hjust = 1, vjust = 0.5)) +  
 xlab('') + ylab('Percentage') +  
 ggtitle('Percentage of NAs across all features')  
p



categorical\_cols1 = c('facing', 'parking')  
# Add NA as a factor level for categorical columns facing and parking only  
hData[categorical\_cols1] = lapply(hData[categorical\_cols1], addNA)  
str(hData)

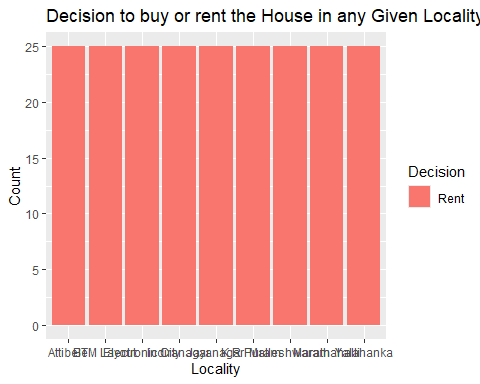
## 'data.frame': 225 obs. of 10 variables:  
## $ locality : Factor w/ 9 levels "Attibele","BTM Layout",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ area : int 565 1837 1280 2220 1113 1332 1815 1400 3006 1600 ...  
## $ rent : int 20060 97434 54448 117000 34388 36394 112000 41266 129000 92849 ...  
## $ price\_per\_sqft: int 6195 9254 7422 9234 5391 4767 10744 5143 7485 10125 ...  
## $ facing : Factor w/ 8 levels "East","North",..: 4 1 1 2 1 7 3 6 1 5 ...  
## $ BHK : int 1 3 2 3 2 2 3 2 4 3 ...  
## $ bathrooms : int 1 3 2 3 2 2 2 2 5 2 ...  
## $ parking : Factor w/ 4 levels "Bike","Bike and Car",..: 1 2 3 2 2 2 3 2 2 2 ...  
## $ House\_Value : int 3500175 16999598 9500160 20499480 6000183 6349644 19500360 7200200 22499910 16200000 ...  
## $ Decision : Factor w/ 1 level "Rent": 1 1 1 1 1 1 1 1 1 1 ...

#see what all locations are present  
distinct\_values = unique(hData$locality)  
print(distinct\_values)

## [1] BTM Layout Attibele K R Puram Marathahalli   
## [5] Indiranagar Electronic City Yalahanka Malleshwaram   
## [9] Jayanagar   
## 9 Levels: Attibele BTM Layout Electronic City Indiranagar ... Yalahanka

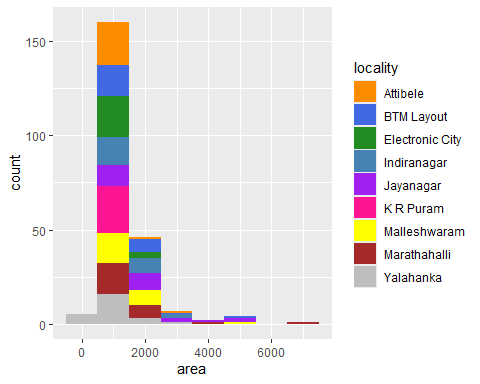
pcustom\_colors = c(  
 "Attibele" = "darkorange", # Dark Orange  
 "BTM Layout" = "royalblue", # Royal Blue  
 "Electronic City" = "forestgreen", # Forest Green  
 "Indiranagar" = "steelblue", # Gold  
 "Jayanagar" = "purple", # Rebecca Purple  
 "K R Puram " = "deeppink", # Deep Pink  
 "Malleshwaram" = "yellow", # Medium Sea Green  
 "Marathahalli" = "brown", # Saddle Brown  
 "Yalahanka" = "gray" # Firebrick  
)

p = ggplot(data = hData) +  
 geom\_bar(aes(x = locality, fill = Decision)) +  
 labs(x = "Locality", y = "Count", title = "Decision to buy or rent the House in any Given Locality")  
p

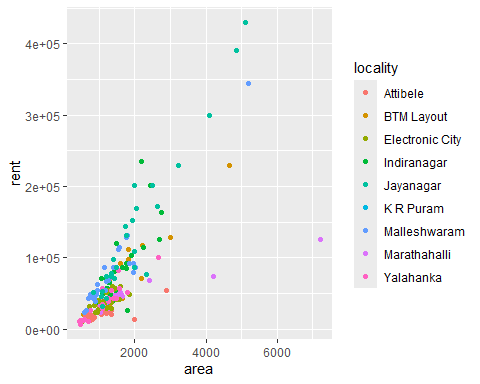


# From Here I conclude that price\_per\_sqft Is not so Important Based the Standard Rules

# Area based Houses in different loacalities in Banglaore  
p = ggplot(data = hData) +  
 geom\_histogram(aes(x = area, fill = locality), binwidth = 1000) +  
 scale\_fill\_manual(values = pcustom\_colors)   
p



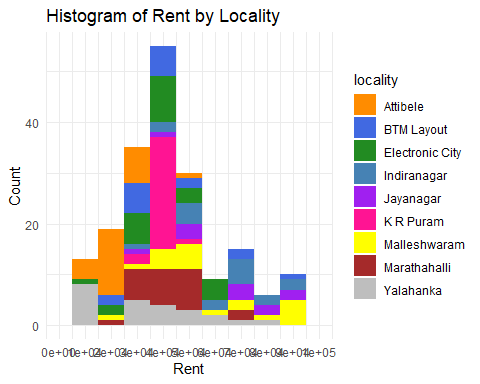
p = ggplot(data = hData) + geom\_point(aes(x = area, y = rent, colour = locality))   
#scale\_y\_continuous(breaks = seq(10000, max(hData$rent, na.rm = TRUE), by = 10000))  
p



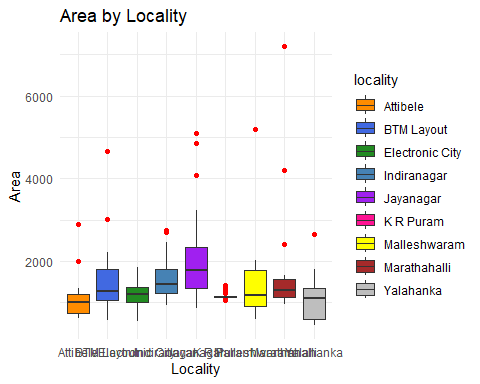
p = ggplot(data = hData) +  
 geom\_histogram(aes(x = rent, fill = locality), binwidth = 10000) +  
 scale\_fill\_manual(values = pcustom\_colors) +  
 scale\_x\_continuous(limits = c(0, 100000), breaks = seq(0, 100000, by = 10000)) +  
 labs(x = "Rent", y = "Count", title = "Histogram of Rent by Locality") +  
 theme\_minimal()  
p

## Warning: Removed 30 rows containing non-finite outside the scale range  
## (`stat\_bin()`).

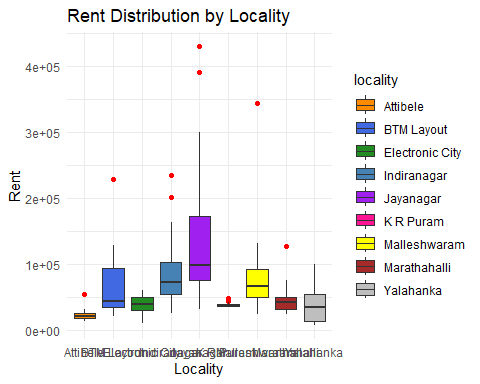
## Warning: Removed 18 rows containing missing values or values outside the scale range  
## (`geom\_bar()`).



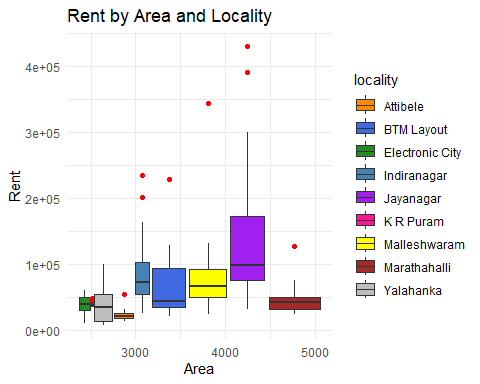
# Boxplot area in all Localities  
p\_area\_facing = ggplot(data = hData) +  
 geom\_boxplot(aes(x = locality, y = area, fill = locality), outlier.colour = "red") +   
 scale\_fill\_manual(values = pcustom\_colors) +  
 labs(x = "Locality", y = "Area", title = "Area by Locality") +  
 theme\_minimal()  
  
p\_area\_facing



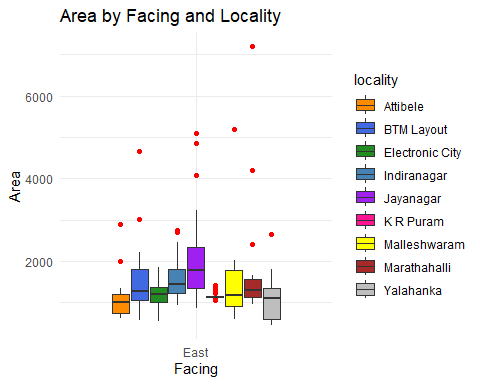
# box plot rent by locality  
p\_area\_rent = ggplot(data = hData) +  
 geom\_boxplot(aes(x = locality, y = rent, fill = locality), outlier.colour = "red") +  
 scale\_fill\_manual(values = pcustom\_colors) +  
 labs(x = "Locality", y = "Rent", title = "Rent Distribution by Locality") +  
 theme\_minimal()  
  
# Display the plot  
p\_area\_rent



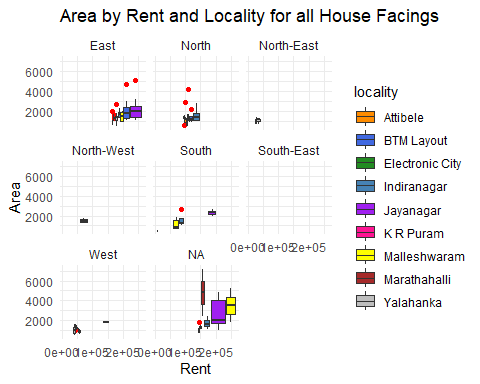
# Boxplot area vs rent in all Localities  
p\_area\_facing = ggplot(data = hData) +  
 geom\_boxplot(aes(x = area, y = rent, fill = locality), outlier.colour = "red") +   
 scale\_fill\_manual(values = pcustom\_colors) +  
 labs(x = "Area", y = "Rent", title = "Rent by Area and Locality") +  
 theme\_minimal()  
p\_area\_facing



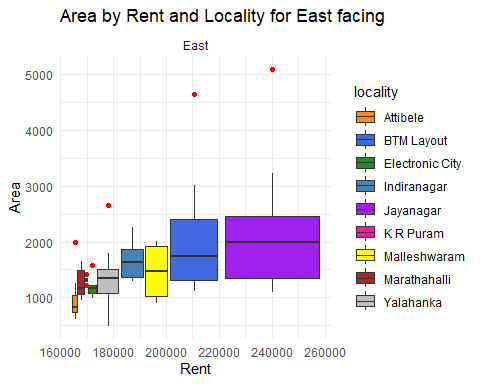
# Create the plot  
p\_area\_facing = ggplot(data = hData) +  
 geom\_boxplot(aes(x = (facing = 'East'), y = area, fill = locality), outlier.colour = "red") +   
 scale\_fill\_manual(values = pcustom\_colors) +  
 labs(x = "Facing", y = "Area", title = "Area by Facing and Locality") +  
 theme\_minimal()  
  
# Display the plot  
p\_area\_facing



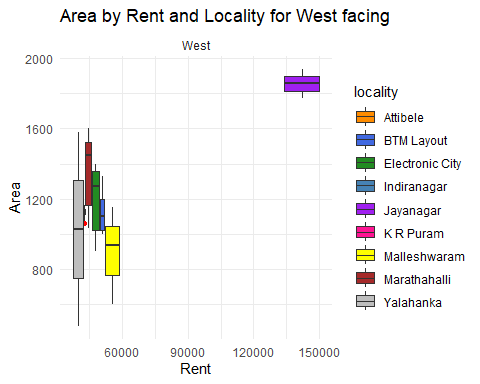
p\_area\_rent = ggplot(data = hData) +  
 geom\_boxplot(aes(x = rent, y = area, fill = locality), outlier.colour = "red") +   
 scale\_fill\_manual(values = pcustom\_colors) +  
 labs(x = "Rent", y = "Area", title = "Area by Rent and Locality for all House Facings") +  
 facet\_wrap(~ facing) + # Add this line to create separate plots for each facing  
 theme\_minimal()  
  
# Display the plot  
p\_area\_rent



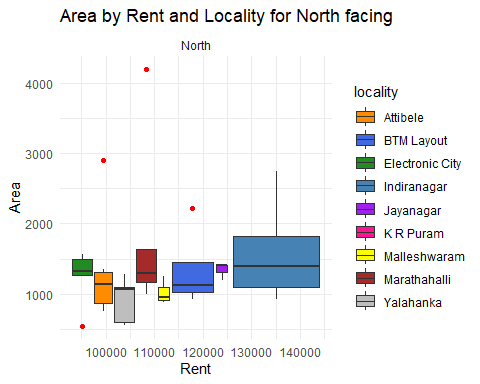
hData\_EF = hData %>% filter(facing == 'East')  
  
p\_area\_rent = ggplot(data = hData\_EF) +  
 geom\_boxplot(aes(x = rent, y = area, fill = locality), outlier.colour = "red") +   
 scale\_fill\_manual(values = pcustom\_colors) +  
 labs(x = "Rent", y = "Area", title = "Area by Rent and Locality for East facing") +  
 facet\_wrap(~ facing) + # Add this line to create separate plots for each facing  
 theme\_minimal()  
  
# Display the plot  
p\_area\_rent



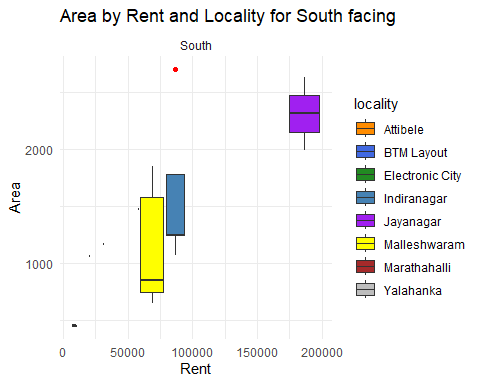
hData\_WF = hData %>% filter(facing == 'West')  
  
p\_area\_rent = ggplot(data = hData\_WF) +  
 geom\_boxplot(aes(x = rent, y = area, fill = locality), outlier.colour = "red") +   
 scale\_fill\_manual(values = pcustom\_colors) +  
 labs(x = "Rent", y = "Area", title = "Area by Rent and Locality for West facing") +  
 facet\_wrap(~ facing) + # Add this line to create separate plots for west facing  
 theme\_minimal()  
  
# Display the plot  
p\_area\_rent



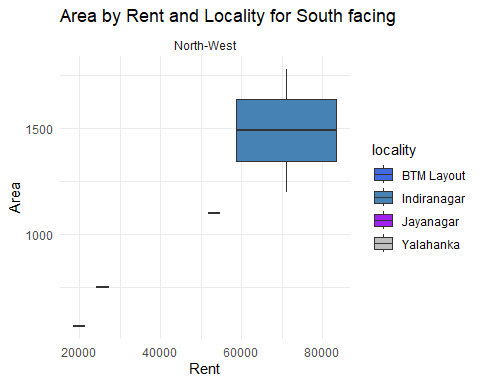
hData\_NF = hData %>% filter(facing == 'North')  
  
p\_area\_rent = ggplot(data = hData\_NF) +  
 geom\_boxplot(aes(x = rent, y = area, fill = locality), outlier.colour = "red") +   
 scale\_fill\_manual(values = pcustom\_colors) +  
 labs(x = "Rent", y = "Area", title = "Area by Rent and Locality for North facing") +  
 facet\_wrap(~ facing) + # Add this line to create separate plots for west facing  
 theme\_minimal()  
  
# Display the plot  
p\_area\_rent



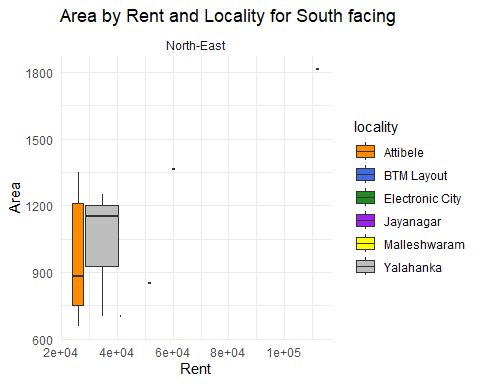
hData\_SF = hData %>% filter(facing == 'South')  
  
p\_area\_rent = ggplot(data = hData\_SF) +  
 geom\_boxplot(aes(x = rent, y = area, fill = locality), outlier.colour = "red") +   
 scale\_fill\_manual(values = pcustom\_colors) +  
 labs(x = "Rent", y = "Area", title = "Area by Rent and Locality for South facing") +  
 facet\_wrap(~ facing) + # Add this line to create separate plots for west facing  
 theme\_minimal()  
  
# Display the plot  
p\_area\_rent



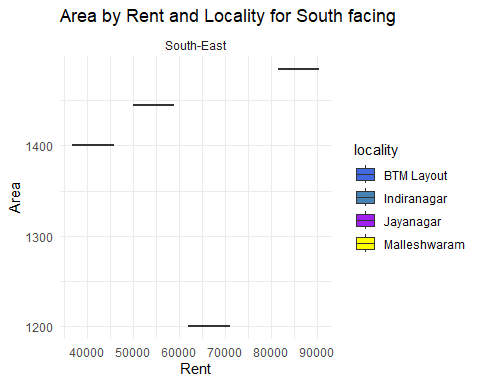
hData\_NWF = hData %>% filter(facing == 'North-West')  
  
p\_area\_rent = ggplot(data = hData\_NWF) +  
 geom\_boxplot(aes(x = rent, y = area, fill = locality), outlier.colour = "red") +   
 scale\_fill\_manual(values = pcustom\_colors) +  
 labs(x = "Rent", y = "Area", title = "Area by Rent and Locality for South facing") +  
 facet\_wrap(~ facing) + # Add this line to create separate plots for west facing  
 theme\_minimal()  
  
# Display the plot  
p\_area\_rent



hData\_NEF = hData %>% filter(facing == 'North-East')  
  
p\_area\_rent = ggplot(data = hData\_NEF) +  
 geom\_boxplot(aes(x = rent, y = area, fill = locality), outlier.colour = "red") +   
 scale\_fill\_manual(values = pcustom\_colors) +  
 labs(x = "Rent", y = "Area", title = "Area by Rent and Locality for South facing") +  
 facet\_wrap(~ facing) + # Add this line to create separate plots for west facing  
 theme\_minimal()  
  
# Display the plot  
p\_area\_rent



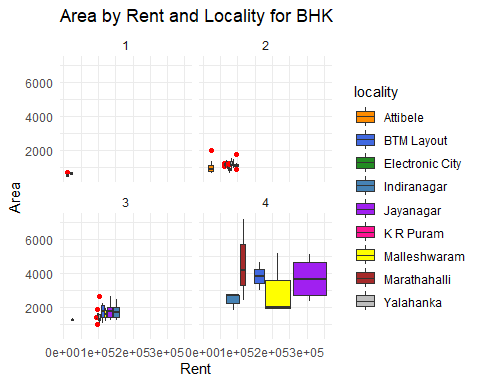
hData\_SEF = hData %>% filter(facing == 'South-East')  
  
p\_area\_rent = ggplot(data = hData\_SEF) +  
 geom\_boxplot(aes(x = rent, y = area, fill = locality), outlier.colour = "red") +   
 scale\_fill\_manual(values = pcustom\_colors) +  
 labs(x = "Rent", y = "Area", title = "Area by Rent and Locality for South facing") +  
 facet\_wrap(~ facing) + # Add this line to create separate plots for west facing  
 theme\_minimal()  
  
# Display the plot  
p\_area\_rent



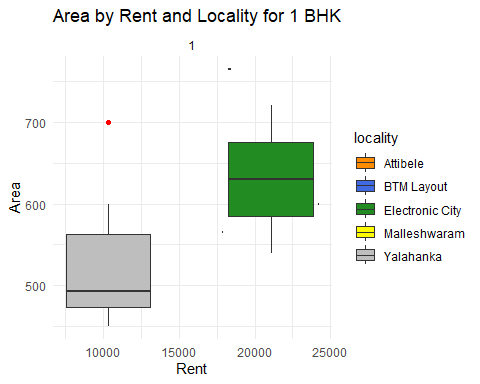
distinct\_values = unique(hData$facing)  
print(distinct\_values)

## [1] North-West East North West North-East South-East South   
## [8] <NA>   
## Levels: East North North-East North-West South South-East West <NA>

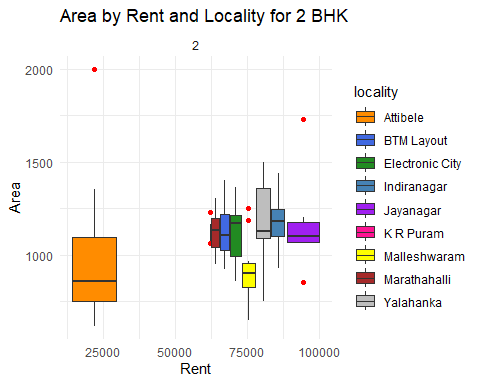
p\_area\_rent = ggplot(data = hData) +  
 geom\_boxplot(aes(x = rent, y = area, fill = locality), outlier.colour = "red") +   
 scale\_fill\_manual(values = pcustom\_colors) +  
 labs(x = "Rent", y = "Area", title = "Area by Rent and Locality for BHK") +  
 facet\_wrap(~ BHK) + # Add this line to create separate plots for each facing  
 theme\_minimal()  
  
# Display the plot  
p\_area\_rent



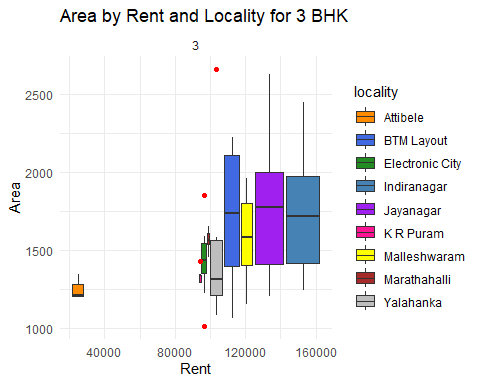
hData\_1BHK = hData %>% filter(BHK == 1)  
  
p\_area\_rent = ggplot(data = hData\_1BHK) +  
 geom\_boxplot(aes(x = rent, y = area, fill = locality), outlier.colour = "red") +   
 scale\_fill\_manual(values = pcustom\_colors) +  
 labs(x = "Rent", y = "Area", title = "Area by Rent and Locality for 1 BHK") +  
 facet\_wrap(~ BHK) + # Assuming you have a column 'facing'  
 theme\_minimal()  
  
# Display the plot  
p\_area\_rent



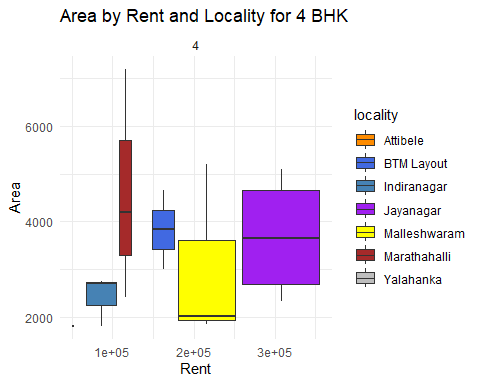
hData\_2BHK = hData %>% filter(BHK == 2)  
  
# Create the plot for 2 BHK  
p\_area\_rent = ggplot(data = hData\_2BHK) +  
 geom\_boxplot(aes(x = rent, y = area, fill = locality), outlier.colour = "red") +   
 scale\_fill\_manual(values = pcustom\_colors) +  
 labs(x = "Rent", y = "Area", title = "Area by Rent and Locality for 2 BHK") +  
 facet\_wrap(~ BHK) + # Assuming you have a column 'facing'  
 theme\_minimal()  
  
# Display the plot  
p\_area\_rent



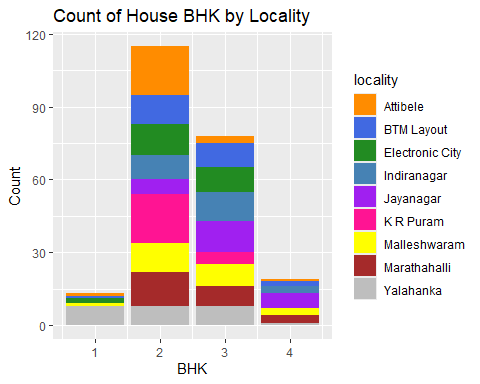
hData\_3BHK = hData %>% filter(BHK == 3)  
  
# Create the plot for 2 BHK  
p\_area\_rent = ggplot(data = hData\_3BHK) +  
 geom\_boxplot(aes(x = rent, y = area, fill = locality), outlier.colour = "red") +   
 scale\_fill\_manual(values = pcustom\_colors) +  
 labs(x = "Rent", y = "Area", title = "Area by Rent and Locality for 3 BHK") +  
 facet\_wrap(~ BHK) + # Assuming you have a column 'facing'  
 theme\_minimal()  
  
# Display the plot  
p\_area\_rent



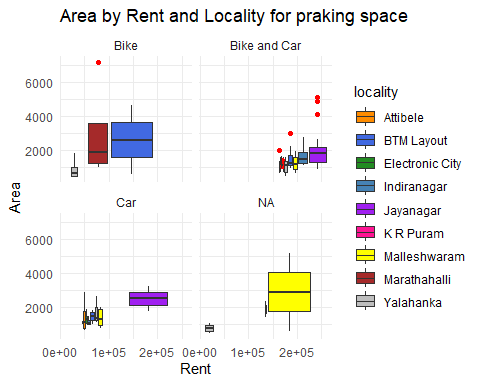
hData\_4BHK = hData %>% filter(BHK == 4)  
  
# Create the plot for 2 BHK  
p\_area\_rent = ggplot(data = hData\_4BHK) +  
 geom\_boxplot(aes(x = rent, y = area, fill = locality), outlier.colour = "red") +   
 scale\_fill\_manual(values = pcustom\_colors) +  
 labs(x = "Rent", y = "Area", title = "Area by Rent and Locality for 4 BHK") +  
 facet\_wrap(~ BHK) + # Assuming you have a column 'facing'  
 theme\_minimal()  
  
# Display the plot  
p\_area\_rent



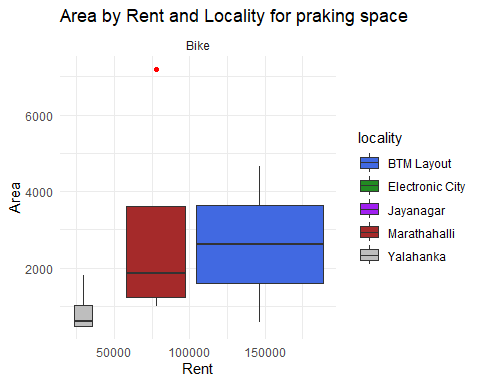
p = ggplot(data = hData) +  
 geom\_bar(aes(x = BHK, fill = locality)) + scale\_fill\_manual(values = pcustom\_colors) +  
 labs(x = "BHK", y = "Count", title = "Count of House BHK by Locality")  
p



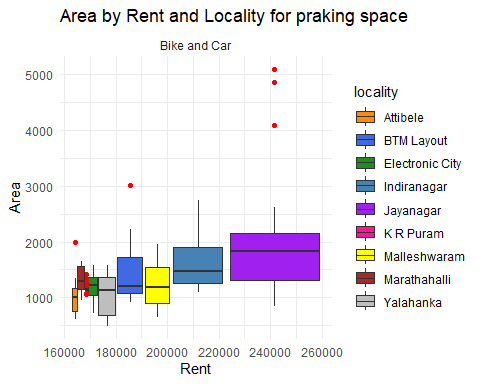
p\_area\_rent = ggplot(data = hData) +  
 geom\_boxplot(aes(x = rent, y = area, fill = locality), outlier.colour = "red") +   
 scale\_fill\_manual(values = pcustom\_colors) +  
 labs(x = "Rent", y = "Area", title = "Area by Rent and Locality for praking space") +  
 facet\_wrap(~parking) + # Add this line to create separate plots for each facing  
 theme\_minimal()  
  
# Display the plot  
p\_area\_rent



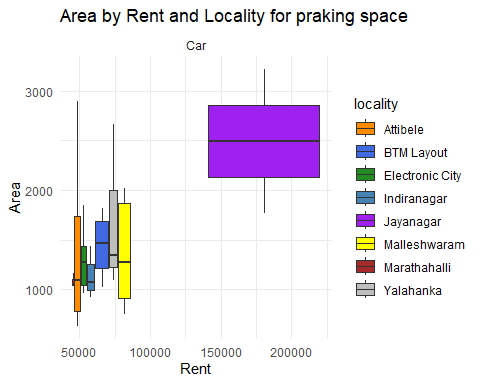
hData\_Bike = hData %>% filter(parking == 'Bike')  
  
p\_area\_rent = ggplot(data = hData\_Bike) +  
 geom\_boxplot(aes(x = rent, y = area, fill = locality), outlier.colour = "red") +   
 scale\_fill\_manual(values = pcustom\_colors) +  
 labs(x = "Rent", y = "Area", title = "Area by Rent and Locality for praking space") +  
 facet\_wrap(~parking) + # Add this line to create separate plots for each facing  
 theme\_minimal()  
  
# Display the plot  
p\_area\_rent



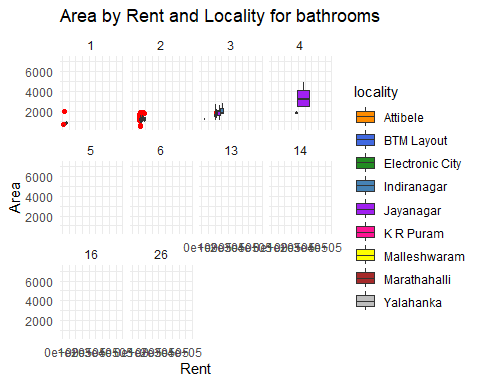
hData\_Bikecar = hData %>% filter(parking == 'Bike and Car')  
  
p\_area\_rent = ggplot(data = hData\_Bikecar) +  
 geom\_boxplot(aes(x = rent, y = area, fill = locality), outlier.colour = "red") +   
 scale\_fill\_manual(values = pcustom\_colors) +  
 labs(x = "Rent", y = "Area", title = "Area by Rent and Locality for praking space") +  
 facet\_wrap(~parking) + # Add this line to create separate plots for each facing  
 theme\_minimal()  
  
# Display the plot  
p\_area\_rent



hData\_car = hData %>% filter(parking == 'Car')  
  
p\_area\_rent = ggplot(data = hData\_car) +  
 geom\_boxplot(aes(x = rent, y = area, fill = locality), outlier.colour = "red") +   
 scale\_fill\_manual(values = pcustom\_colors) +  
 labs(x = "Rent", y = "Area", title = "Area by Rent and Locality for praking space") +  
 facet\_wrap(~parking) + # Add this line to create separate plots for each facing  
 theme\_minimal()  
  
# Display the plot  
p\_area\_rent



p\_area\_rent = ggplot(data = hData) +  
 geom\_boxplot(aes(x = rent, y = area, fill = locality), outlier.colour = "red") +   
 scale\_fill\_manual(values = pcustom\_colors) +  
 labs(x = "Rent", y = "Area", title = "Area by Rent and Locality for bathrooms") +  
 facet\_wrap(~bathrooms) + # Add this line to create separate plots for each facing  
 theme\_minimal()  
  
# Display the plot  
p\_area\_rent



SUMMARY

#One line of conculsion is that it is better to be in a rented house in the bangalore city given according to this dataset and it is seen that the prices vary a liitle from area to area and most of the houses fall in the range of 2000 sqrt and in the price range below 1,00,000 Rs and only in the area Jayanagar we can see that the price of the houses are above 1,00,000 and we have visiualsed the house and its renting facilites.