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" ...

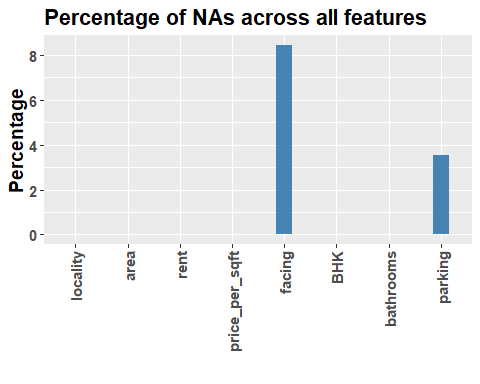
# Convert 'locality', 'facing' and 'parking' columns to factors  
categorical\_cols = c('locality', 'facing', 'parking')  
hData[categorical\_cols] = lapply(hData[categorical\_cols], as.factor)  
str(hData)

## 'data.frame': 225 obs. of 8 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 ...

categorical\_cols1 = c('facing', 'parking')

# 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



# Add NA as a factor level for categorical columns   
hData[categorical\_cols1] = lapply(hData[categorical\_cols1], addNA)  
str(hData)

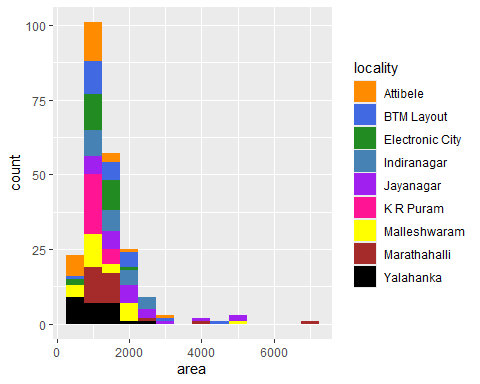
## 'data.frame': 225 obs. of 8 variables:  
## $ locality : Factor w/ 9 levels "Attibele","BTM Layout",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ area : int 565 1837 1280 2220 1113 1332 1815 1400 3006 1600 ...  
## $ rent : int 20060 97434 54448 117000 34388 36394 112000 41266 129000 92849 ...  
## $ price\_per\_sqft: 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 ...

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

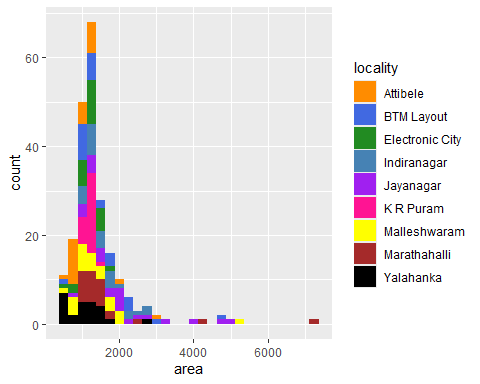
## [1] BTM Layout Attibele K R Puram Marathahalli   
## [5] Indiranagar Electronic City Yalahanka Malleshwaram   
## [9] Jayanagar   
## 9 Levels: Attibele BTM Layout Electronic City Indiranagar ... Yalahanka

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" = "black" # Firebrick  
)

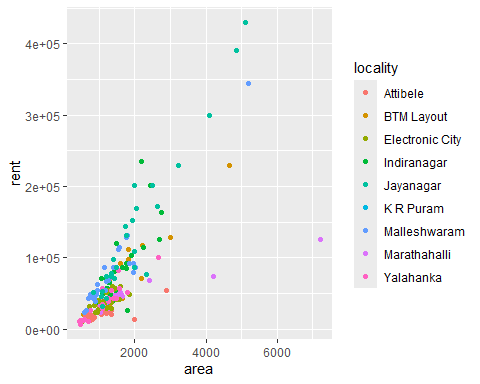
# Area based Houses in different loacalities in Banglaore  
p = ggplot(data = hData) +  
 geom\_histogram(aes(x = area, fill = locality), binwidth = 500) +  
 scale\_fill\_manual(values = pcustom\_colors)   
p



# In different binwidth  
p = ggplot(data = hData) +  
 geom\_histogram(aes(x = area, fill = locality), binwidth = 250) +  
 scale\_fill\_manual(values = pcustom\_colors)  
p



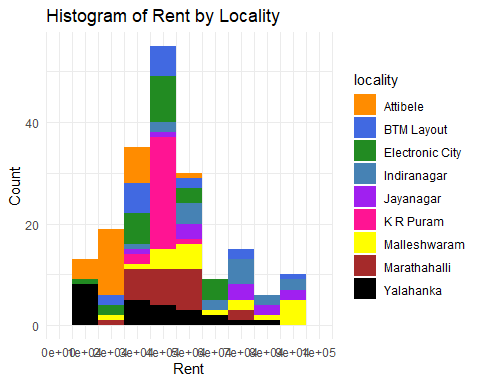
p = ggplot(data = hData) + geom\_point(aes(x = area, y = rent, colour = locality))   
#scale\_y\_continuous(breaks = seq(10000, max(hData$rent, na.rm = TRUE), by = 10000))  
p



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)) + # Set limits and breaks  
 labs(x = "Rent", y = "Count", title = "Histogram of Rent by Locality") +  
 theme\_minimal()  
 # Set limits and breaks  
   
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()`).

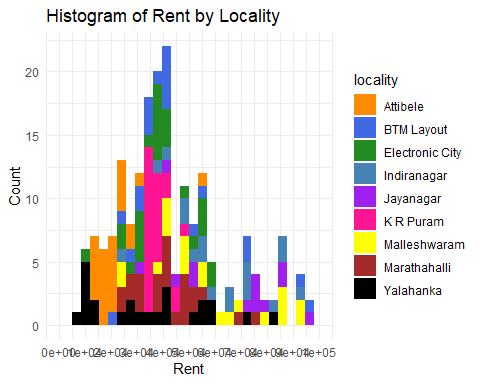


p = ggplot(data = hData) +  
 geom\_histogram(aes(x = rent, fill = locality)) +  
 scale\_fill\_manual(values = pcustom\_colors) +  
 scale\_x\_continuous(limits = c(0, 100000), breaks = seq(0, 100000, by = 10000)) + # Set limits and breaks  
 labs(x = "Rent", y = "Count", title = "Histogram of Rent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

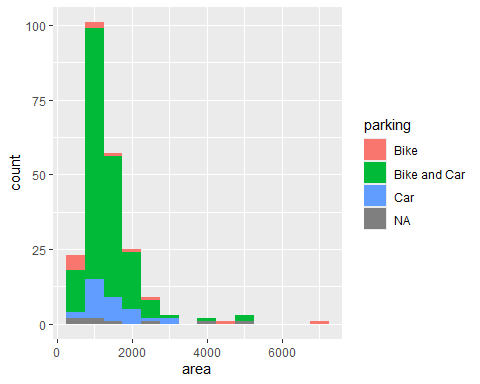
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

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

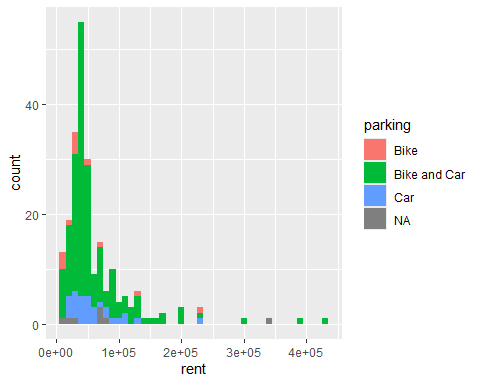
## Warning: Removed 18 rows containing missing values or values outside the scale range  
## (`geom\_bar()`).



p = ggplot(data = hData) +  
 geom\_histogram(aes(x = area, fill = parking), binwidth = 500) # +  
 #scale\_fill\_manual(values = pcustom\_colors)  
p

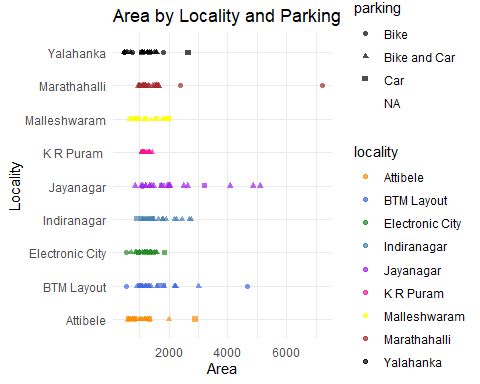


p = ggplot(data = hData) +  
 geom\_histogram(aes(x = rent, fill = parking), binwidth = 10000)  
p



p <- ggplot(data = hData) +  
 geom\_point(aes(x = area, y = locality, colour = locality, shape = parking), alpha = 0.7) +  
 scale\_color\_manual(values = pcustom\_colors) + # Use your custom colors  
 labs(x = "Area", y = "Locality", title = "Area by Locality and Parking") +  
 theme\_minimal()  
  
print(p)

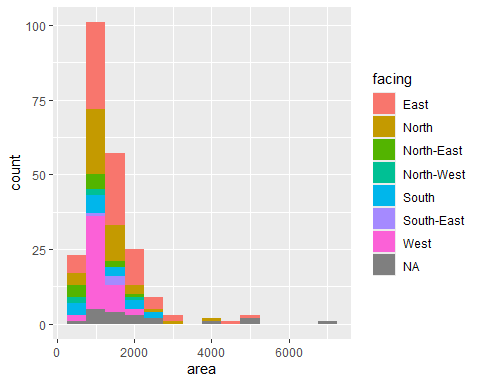
## Warning: Removed 8 rows containing missing values or values outside the scale range  
## (`geom\_point()`).



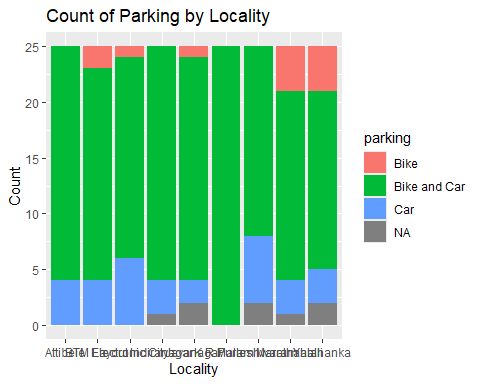
distinct\_values = unique(hData$parking)  
print(distinct\_values)

## [1] Bike Bike and Car Car <NA>   
## Levels: Bike Bike and Car Car <NA>

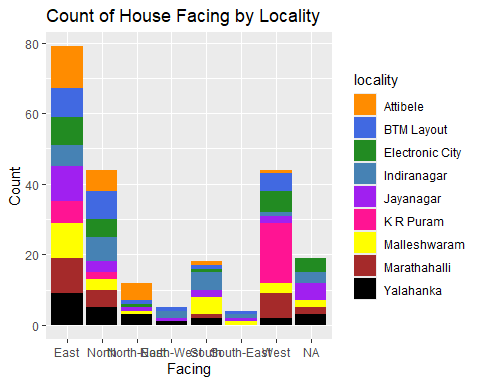
p = ggplot(data = hData) +  
 geom\_histogram(aes(x = area, fill = facing), binwidth = 500)   
 #scale\_fill\_manual(values = pcustom\_colors)   
p



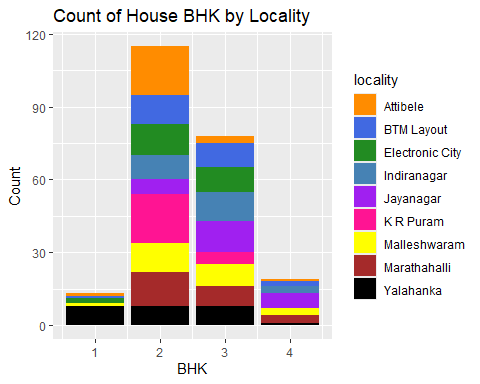
p = ggplot(data = hData) +  
 geom\_bar(aes(x = locality, fill = parking)) +  
 labs(x = "Locality", y = "Count", title = "Count of Parking by Locality")  
p



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



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 = ggplot(data = hData) +  
 geom\_point(aes(x = rent, y = bathrooms), alpha = 1, colour = 'blue') +  
 labs(x = "Number of Bedrooms (BHK)", y = "Number of Bathrooms", title = "Relationship between BHK and Bathrooms")  
p

