Link: [**https://thuphuongkatenguyen.shinyapps.io/RShiny\_Version2/**](https://thuphuongkatenguyen.shinyapps.io/RShiny_Version2/)

**Assignment summary**

Module 1 - 3: Review the data set and begin an initial analysis → done

Module 3: Assigned to groups and share initial analysis → done

**Module 5: Create initial RShiny visualization → We are working on this one**

Module 4: Create initial Tableau visualization → Upcoming

Module 6: Submit final dashboard → Upcoming

**Dataset Overview**

* Original dataset: 180,627 observations, 60 variables
* After cleaning: 63,326 observations, 18 variables
  + **Only Condominium: 63,134 observations, 17 variables**
    - These variables include the city or town of the parcel (CITY), land use description (LU\_DES), living area square footage of the property (LIVING\_AREA), the year the property was built (YR\_BUILT), overall condition of the parcel (OVERALL\_COND), total number of bedrooms (BED\_RMS), total number of full baths (FULL\_BTH), total rooms (TT\_RMS), bedroom condition (BDRM\_COND), bathroom style (BTHRM\_STYLE1), kitchen style (KITCHEN\_STYLE1), AC type (AC\_TYPE), number of parking spaces (NUM\_PARKING), property view (PROP\_VIEW), **and the total assessed value of the property (TOTAL\_VALUE)**
* If you want to analyze more using R (use the CVS file or the R file with the original variable names like above; the Excel file has been cleaned with the more meaningful variable names will be use later for our Tableau assignment) → Please use the same cleaned dataset so we can have everything the same with each other.

**Project Objective**

* We will FOCUS only the condominium so it is easy to compare the property value for the same land type (It makes more sense to compare condos with condos rather than compare condos with industrial land…). In addition, the condominium has the most samples (without missing value) in the dataset so I think we should choose it (Also I think the professor likes our focus as he commented in our last assignment)
* **IMPORTANT:** **It is important to make our works consistent and link with each other; therefore, if there is no problem or concern on the cleaned dataset I uploaded (both in excel and R), we should use that for RShiny and Tableau (we might need the Excel file). So our data will be the same and make more sense while we work individually and add them together as a group. To clarify again, our new dataset has only CONDOMINIUM in it and that will be our focus.**
* Target Audiences: Property Investors, Home Buyers, etc.
* Questions we want to answer:

**Topic: Smart Investment in Boston Housing Market**

1. Which city/town has the highest property value?

→ Idea: Make a viz to display Top 5 city/town with highest property value (could be a bar chart…; use TOTAL\_VALUE by CITY)

→ More advanced: if we can, we should make check option buttons (top 5, top 10, top 15, etc.)

1. Potential cities to invest in?

→ Idea: Make a line chart to display the changing of price (TOTAL VALUE) for the condos for cities in a decade (ONLY LOOK AT YR\_BUILT after 2010 until 2022. The city/town that has the rapid increase in price will be the potential cities to invest in)

1. Which year has the most condos built? Which year built has the highest property value?

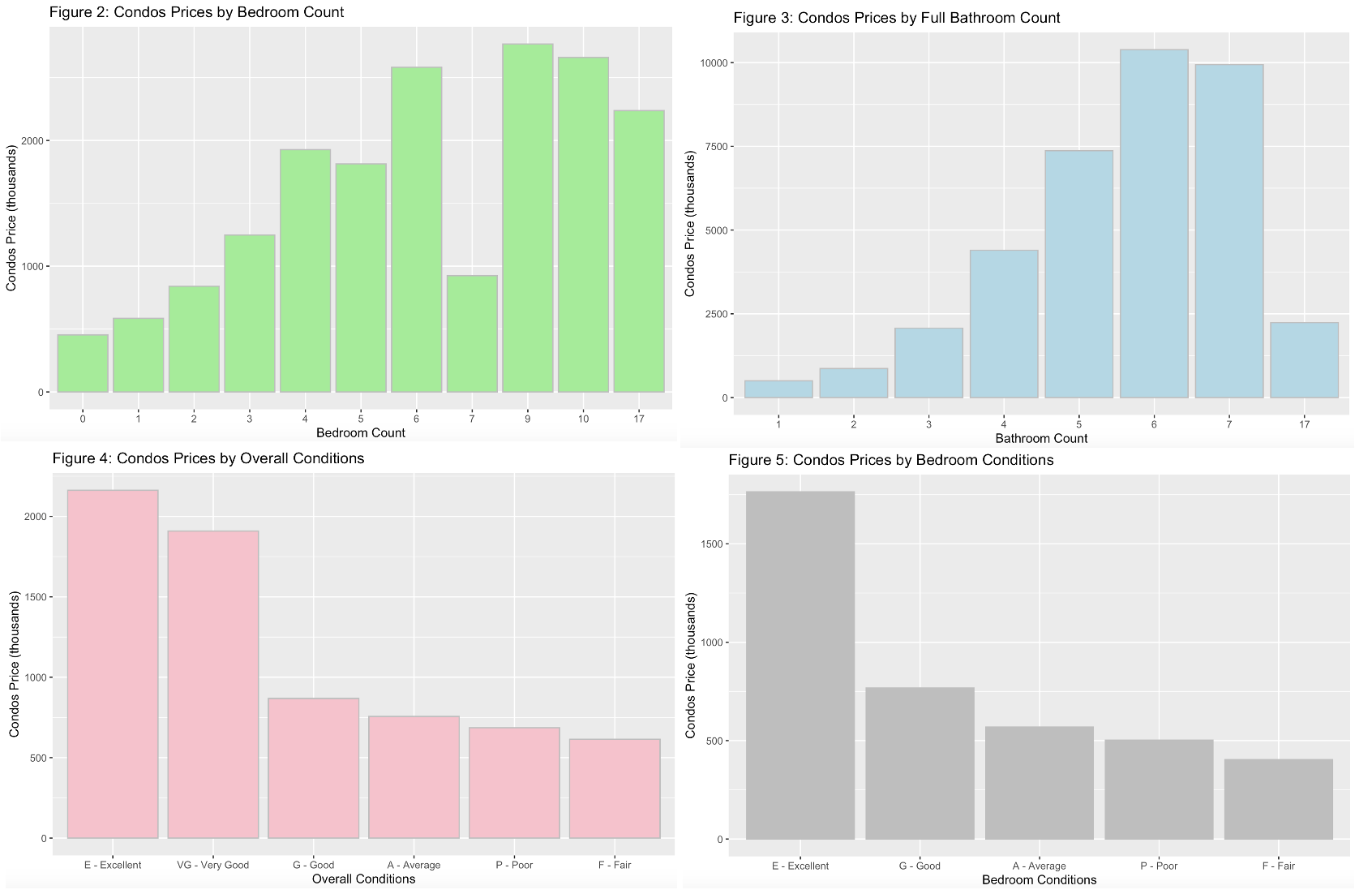
→ Idea: Make a viz to display property value (TOTAL\_VALUE by YR\_BUILT). I suggest we subset/ group the year into decades since there are many years in the dataset.

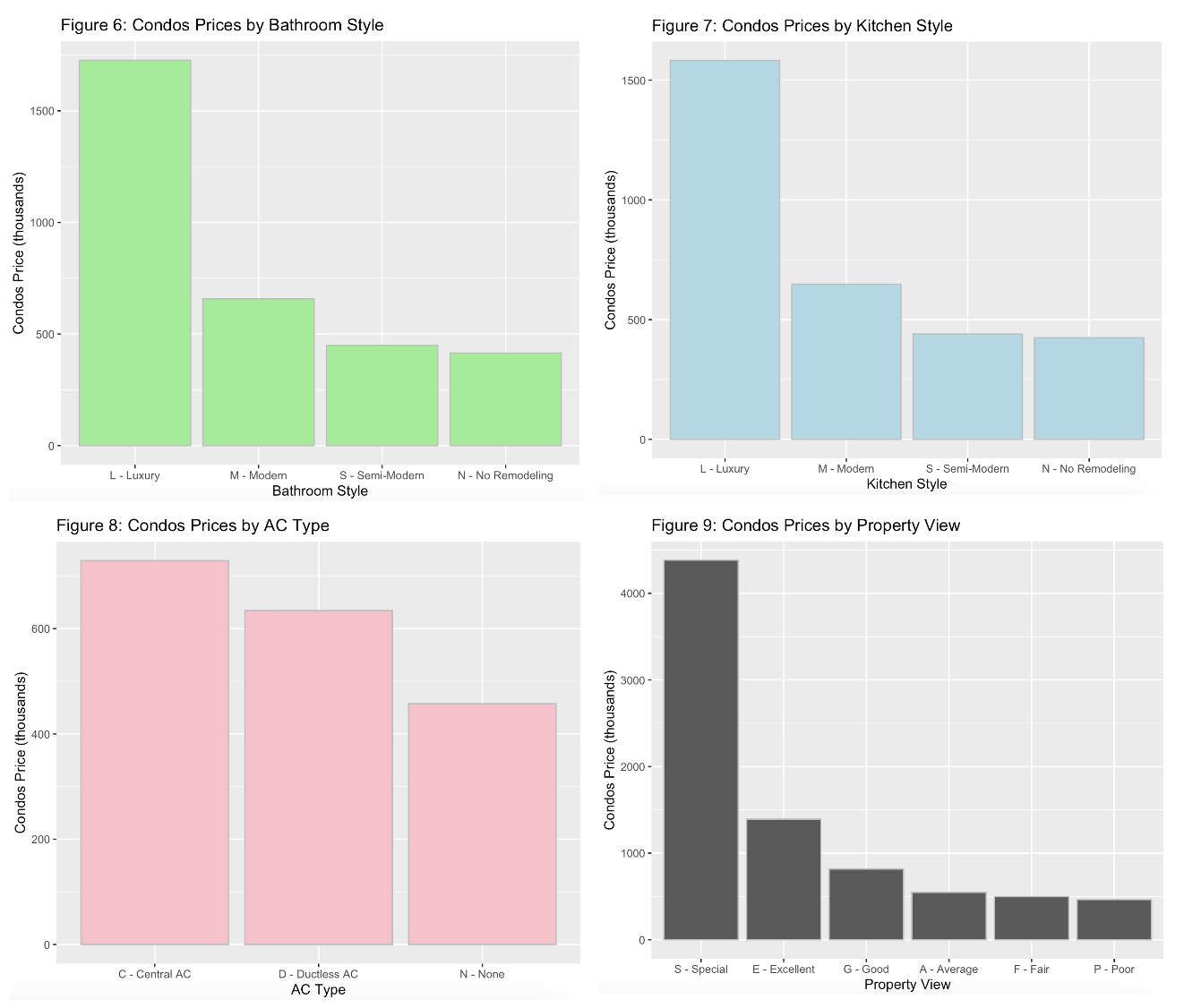
1. Which factors affect the total property value/ the condo price? Bedroom Count/ Bathroom Count/ Kitchen Style/ Bathroom Style/ AC type / Bedroom Conditions/ Overall Conditions, Property Views, etc.

→ Ideas: Make an interactive viz to change for each category to display the Condos price (TOTAL\_VALUE).

→ More advanced: if we have time and do more research, we can explain more to our audience about why there is a decrease in property value when the bedroom count and bathroom count go higher than 7 (Figure 2, figure 3). Why do condos with poor and fair conditions have similar total value? (If we can, we can explain more about that. I think it would be interesting to dig more into it but I don’t think we have enough time to cover everything in this RShiny Assignment, maybe in our Tableau…)

Examples are the following viz:





**Design Principles** (likes Gestalt principles, contrast, alignment, repetition, and proximity, etc.):

→ We can apply it when we design our viz with meaningful color, highlighting the important ones….. to tell our story (Again, our story will be which factors affect the price/ total property value of the condos → OUR response variable is TOTAL\_VALUE variable)

**RShiny Dashboard Brainstorm:**

1. We should try to play around with small pieces first as what the professor said in class. Maybe try to build some foundation for each question. (The base codes)
2. Then we will add more features into the dashboard
3. There would be around 4-5 tabs in the RShiny
   1. Summary
   2. Tell Story by charts
   3. Tell Story by charts
   4. Dashboard

→ I suggest three people (person1, person2, person3) will work on the base codes and some easy questions first. Then one person (person4, person5) will build from the base codes of the three members and come up with new ideas to make it clean, clear, ethical, consistent, good-looking with meaningful color (don’t overuse the color), and be able to tell our story. The last member (person6) with strong writing skills will finish the report by answering the questions, formatting the report, and tell our story using the dashboard, etc. (with the support of person 4 and person 5).

IT'S ONLY A SUGGESTION. IF ANYONE HAS A BETTER IDEA TO DIVIDE WORKS, PLEASE LET US KNOW. THKS!

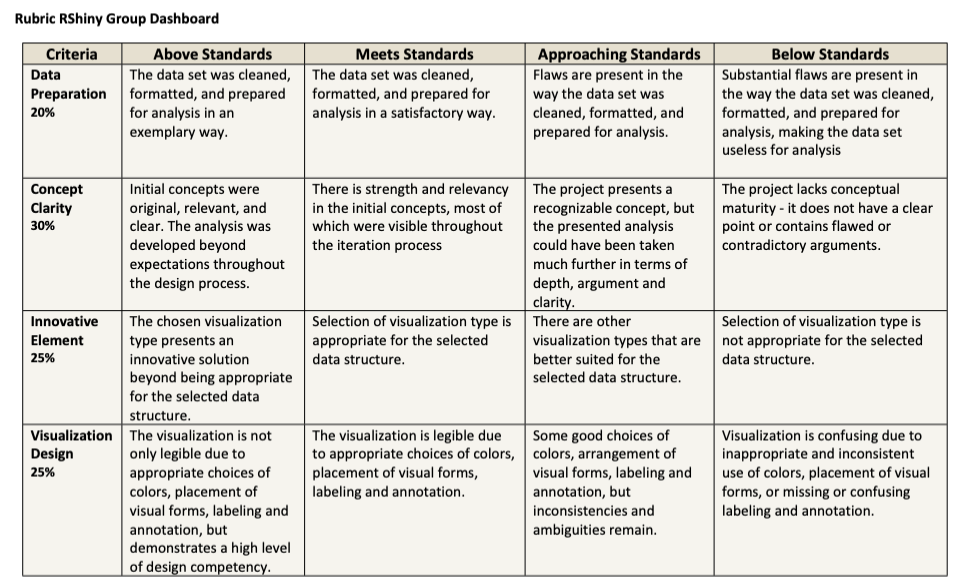
IF NOT, WE CAN FOLLOW IT BY PUTTING OUR NAMES IN THE LAST PAGE AND FOLLOW THE DEADLINES.

Deadlines for each group:

**Group 1: Foundation coding team** (person1, person2, person3): Friday night or Saturday afternoon Dec 2 (ASAP because the dashboard creators need more time)

**Group 2: Dashboard creators** (person4, person5): try to get it done before Thursday Dec 6 if possible or Dec 7

**Group 3: Report (it could be short)** (person6): due Dec 8



Write your name to choose the group you want to be in:

Person 1: Xuejia Li

Person 2:

Person 3:Shinuo Xiang

Person 4: Kate

Person 5:YI XU,pengxiang wu

Person 6: Taige Zhu

**Foundation Code:**

install.packages(c("shiny", "shinydashboard", "ggplot2"))

library(shiny)

library(shinydashboard)

library(ggplot2)

data <- read.csv("/Users/cigarlll/Downloads/Cleaned Dataset.csv")

summary(data)

ui <- dashboardPage(

dashboardHeader(title = "Property Dashboard"),

dashboardSidebar(

selectInput("city", "Select City:", choices = unique(data$CITY)),

selectInput("land\_use", "Select Land Use:", choices = unique(data$LU\_DES)),

selectInput("plot\_type", "Select Plot Type:", choices = c("Bar Chart", "Scatter Plot"))

),

dashboardBody(

plotOutput("plot")

)

)

server <- function(input, output) {

output$plot <- renderPlot({

filtered\_data <- data[data$CITY == input$city & data$LU\_DES == input$land\_use, ]

if (input$plot\_type == "Bar Chart") {

ggplot(filtered\_data, aes(x = BED\_RMS, fill = factor(BDRM\_COND))) +

geom\_bar(position = "dodge") +

labs(title = "Bedroom Condition Distribution by Number of Bedrooms")

} else if (input$plot\_type == "Scatter Plot") {

ggplot(filtered\_data, aes(x = LIVING\_AREA, y = TOTAL\_VALUE)) +

geom\_point() +

labs(title = "Living Area vs Total Value")

}

})

}

shinyApp(ui, server)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

3RD PART:

library(shiny)

library(ggplot2)

library(dplyr)

# Use your own path here

data <- read.csv("~/R files/Myclass\_Rproject/Datasets/Cleaned Dataset (1).csv")

# Preprocess the data

data$Decade <- cut(as.numeric(data$YR\_BUILT), breaks = seq(1900, 2031, by = 10), right = FALSE, labels = seq(1900, 2021, by = 10))

decade\_data <- data %>%

group\_by(Decade) %>%

summarize(TotalProperties = n(), TotalValue = sum(TOTAL\_VALUE, na.rm = TRUE))

annual\_growth\_data <- data %>%

filter(YR\_BUILT >= 2010 & YR\_BUILT <= 2022) %>%

group\_by(YR\_BUILT, CITY) %>%

summarize(TotalValue = sum(TOTAL\_VALUE, na.rm = TRUE)) %>%

arrange(CITY, YR\_BUILT) %>%

group\_by(CITY) %>%

mutate(GrowthRate = (TotalValue / lag(TotalValue) - 1) \* 100)

# User interface

ui <- fluidPage(

titlePanel("Property Dashboard"),

sidebarLayout(

sidebarPanel(

selectInput("city", "Select City:", choices = unique(data$CITY)),

selectInput("plot\_type", "Select Plot Type:", choices = c("Bar Chart", "Scatter Plot", "Growth Rate Plot", "Decade Summary", "Factor Impact")),

selectInput("factor", "Select Factor:", choices = c("BED\_RMS", "FULL\_BTH", "KITCHEN\_STYLE1", "BTHRM\_STYLE1", "AC\_TYPE", "BDRM\_COND", "OVERALL\_COND", "PROP\_VIEW"))

),

mainPanel(

plotOutput("plot")

)

)

)

# Server logic

server <- function(input, output) {

output$plot <- renderPlot({

filtered\_data <- data[data$CITY == input$city, ]

if (input$plot\_type == "Bar Chart") {

ggplot(filtered\_data, aes(x = BED\_RMS, fill = factor(BDRM\_COND))) +

geom\_bar(position = "dodge") +

labs(title = "Bedroom Condition Distribution by Number of Bedrooms")

} else if (input$plot\_type == "Scatter Plot") {

ggplot(filtered\_data, aes(x = LIVING\_AREA, y = TOTAL\_VALUE)) +

geom\_point() +

labs(title = "Living Area vs Total Value")

} else if (input$plot\_type == "Growth Rate Plot") {

filtered\_growth\_data <- annual\_growth\_data[annual\_growth\_data$CITY == input$city, ]

ggplot(filtered\_growth\_data, aes(x = YR\_BUILT, y = GrowthRate)) +

geom\_line() +

labs(title = "Annual Growth Rate of Property Value", y = "Growth Rate (%)", x = "Year")

} else if (input$plot\_type == "Decade Summary") {

ggplot(decade\_data, aes(x = Decade)) +

geom\_bar(aes(y = TotalProperties), stat = "identity", fill = "steelblue") +

geom\_line(aes(y = TotalValue / 1e6, group = 1), color = "red", size = 1) +

scale\_y\_continuous(sec.axis = sec\_axis(~ . \* 1e6, name = "Total Value (Millions)")) +

labs(title = "Properties Count and Total Value by Decade", y = "Properties Count", x = "Decade")

} else if (input$plot\_type == "Factor Impact") {

ggplot(filtered\_data, aes\_string(x = input$factor, y = "TOTAL\_VALUE")) +

geom\_boxplot() +

labs(title = paste("Impact of", input$factor, "on Property Value"), y = "Total Value", x = input$factor) +

theme(axis.text.x = element\_text(angle = 45, hjust = 1))

}

})

}

# Run the application

shinyApp(ui, server)

##########################################################################

Dashboard

library(shiny)

library(plyr)

library(ggplot2)

library(dplyr)

library(readr)

# Load and prepare data

data <- read\_csv("Cleaned Dataset.csv")

summary(data)

data$Decade <- cut(as.numeric(data$YR\_BUILT), breaks = seq(1791, 2031, by = 10), right = FALSE, labels = seq(1791, 2021, by = 10))

decade\_data <- data %>%

group\_by(Decade) %>%

summarize(TotalProperties = n(), AverageValue = mean(TOTAL\_VALUE, na.rm = TRUE))

growth\_rate <- decade\_data %>%

arrange(Decade) %>%

mutate(GrowthRate = round(((AverageValue / lag(AverageValue) - 1)), 2)) # Annual Growth Rate

data <- data %>%

left\_join(growth\_rate %>% select(Decade, GrowthRate), by = "Decade")

# Ui

ui <- fluidPage(

headerPanel("Boston Property Analysis"),

sidebarLayout(

sidebarPanel(

sliderInput("builtYearRange", "Built Year Range:", value = c(1990, 2021), min = 1791, max = 2021

),

hr(),

checkboxGroupInput(inputId = "city",

label = "Choose city?",

choices = unique(data$CITY)

),

hr(),

radioButtons(inputId = "topThreeCity",

label = "Show top 3 City with Highest Average Condos Prices:",

choices = list("Default" = "default",

"Show Top 3 Cities/ Towns" = "topThree")

),

hr(),

hr(),

selectInput(inputId = "factors"

, label = "Select a Factor (displayed in third tab):"

, choices = c("Bedroom Count" = "BED\_RMS",

"Full Bathroom Count" = "FULL\_BTH",

"Kitchen Style" = "KITCHEN\_STYLE1",

"Bathroom Style" = "BTHRM\_STYLE1",

"AC Type" = "AC\_TYPE",

"Bedroom Conditions" = "BDRM\_COND",

"Overall Conditions" = "OVERALL\_COND",

"Property View" = "PROP\_VIEW")

),

hr(),

hr()

),

mainPanel(

tabsetPanel(

tabPanel("Summary" , textOutput("textDisplay")),

tabPanel("Property Value by Year Built", plotOutput("propertyValueGraph")),

tabPanel("Property Value by Factors" , plotOutput("factorsGraph")),

tabPanel("Growth Rate over Year Built", plotOutput("growthRateGraph")),

tabPanel("Take Away" , textOutput("takeAway"))

)

)

)

)

# Server

server <- function(input, output) {

passData <- reactive({

filteredData <- data[data$YR\_BUILT >= as.numeric(input$builtYearRange[1]) & data$YR\_BUILT <= as.numeric(input$builtYearRange[2]), ]

if(class(input$city)=="character"){

filteredData <- filteredData[filteredData$CITY %in% input$city, ]

}

filteredData

})

# Line Graph: Property Value by Built Year

output$propertyValueGraph <- renderPlot({

graphData <- aggregate(data = passData(),

cbind(LIVING\_AREA, TOTAL\_VALUE\_in\_thousands, BED\_RMS, FULL\_BTH)~CITY+YR\_BUILT

, FUN = mean)

if(input$topThreeCity == "default"){

theGraph <- ggplot(graphData, aes(x = YR\_BUILT, y = TOTAL\_VALUE\_in\_thousands, group = CITY, colour = CITY)) +

geom\_line() +

ylab("Condos Price (in thousands)") +

xlab("Year Built")

}

if(input$topThreeCity == "topThree"){

# Identify the top three cities based on TOTAL\_VALUE\_in\_thousands

top\_three\_cities <- graphData %>%

group\_by(CITY) %>%

summarize(TotalValue = mean(TOTAL\_VALUE\_in\_thousands)) %>%

top\_n(3, TotalValue) %>%

pull(CITY)

final\_points <- graphData %>%

filter(CITY %in% top\_three\_cities) %>%

group\_by(CITY) %>%

slice(n())

theGraph <- ggplot(graphData, aes(x = YR\_BUILT, y = TOTAL\_VALUE\_in\_thousands, group = CITY, colour = CITY)) +

geom\_line() +

geom\_text(data = final\_points, aes(label = CITY), hjust = 0.75, vjust = 0.5, size = 2.5, color = "black") +

scale\_color\_manual(values = ifelse(graphData$CITY %in% top\_three\_cities, "orange", "darkgrey")) +

ylab("Condos Price (in thousands)") +

xlab("Year Built")

}

print(theGraph)

})

# Bar Charts: Property Value by Potential Factors

output$factorsGraph <- renderPlot({

if (input$factors == "BED\_RMS") {

MeanValues\_BRC <- data %>%

group\_by(BED\_RMS) %>%

summarise(MeanValue\_BRC = mean(TOTAL\_VALUE\_in\_thousands))

theGraph <- ggplot(MeanValues\_BRC, aes(x = factor(BED\_RMS), y = MeanValue\_BRC)) +

geom\_bar(stat = "identity", color = "lightgrey",

fill = c("lightblue","lightblue","lightblue","lightblue","lightblue",

"lightblue","lightblue","lightgrey","lightgrey", "lightgrey","lightgrey")) +

labs(x = "Bedroom Count", y = "Condos Price (thousands)") +

labs(title = "\n Condos Price INCREASES along with the Bedroom Count, EXCEPT for 7+ Bedrooms:\n

since the number of samples for 7+ bedrooms Condos is limited (can cause bias).\n") +

theme\_minimal() +

theme(plot.title = element\_text(face = "bold", size = 14))

}

if (input$factors == "FULL\_BTH") {

MeanValues\_FBTHC <- data %>%

group\_by(FULL\_BTH) %>%

summarise(MeanValue\_FBTHC = mean(TOTAL\_VALUE\_in\_thousands))

theGraph <- ggplot(MeanValues\_FBTHC, aes(x = factor(FULL\_BTH), y = MeanValue\_FBTHC)) +

geom\_bar(stat = "identity", color = "lightgrey",

fill = c("lightblue","lightblue","lightblue","lightblue",

"lightblue","lightblue", "lightgrey","lightgrey")) +

labs(x = "Full Bath Count", y = "Condos Price (thousands)") +

labs(title = "\n Condos Price INCREASES along with the Full Bathroom Count, EXCEPT for 7+ Full Bathrooms:\n

since the number of samples for 7+ full bathrooms Condos is limited (can cause bias).\n") +

theme\_minimal() +

theme(plot.title = element\_text(face = "bold", size = 14))

}

if (input$factors == "KITCHEN\_STYLE1") {

MeanValues\_KS <- data %>%

group\_by(KITCHEN\_STYLE1) %>%

summarise(MeanValue\_KS = mean(TOTAL\_VALUE\_in\_thousands))

theGraph <- ggplot(MeanValues\_KS, aes(x = reorder(factor(KITCHEN\_STYLE1), MeanValue\_KS), y = MeanValue\_KS)) +

geom\_bar(stat = "identity", color = "lightgrey",

fill = c("lightblue", "lightgrey", "lightgrey","lightgrey")) +

labs(x = "", y = "Condos Price (thousands)") +

labs(title = "\n Kitchen Style adds more value to the Condos:\n

1. Luxury Kitchen Style has highest Average Condos Price.\n

2. Semi-modern and No-remodeling do NOT differ much in price.\n

3. Therefore, should NOT buy Condos with No-remodeling Kicthen Style.\n

4. SHOULD buy the Semi-modern or Modern and remodel to Luxury one.\n") +

theme\_minimal() +

theme(plot.title = element\_text(size = 14))

theGraph <- theGraph + coord\_flip()

}

if (input$factors == "BTHRM\_STYLE1") {

MeanValues\_BS <- data %>%

group\_by(BTHRM\_STYLE1) %>%

summarise(MeanValue\_BS = mean(TOTAL\_VALUE\_in\_thousands))

theGraph <- ggplot(MeanValues\_BS, aes(x = reorder(factor(BTHRM\_STYLE1), MeanValue\_BS), y = MeanValue\_BS)) +

geom\_bar(stat = "identity", color = "lightgrey",

fill = c("lightblue", "lightgrey", "lightgrey","lightgrey")) +

labs(x = "", y = "Condos Price (thousands)") +

labs(title = "\n Bathroom Style adds more value to the Condos:\n

1. Luxury Bathroom Style has significant highest Average Condos Price.\n

2. Semi-modern and No-remodeling are NOT much difference in prices.\n

3. Therefore, should NOT buy Condos with No-remodeling Bathroom Style.\n

4. SHOULD buy the Semi-modern or Modern and remodel to Luxury one.\n") +

theme\_minimal() +

theme(plot.title = element\_text(size = 14))

theGraph <- theGraph + coord\_flip()

}

if (input$factors == "AC\_TYPE") {

MeanValues\_AC <- data %>%

group\_by(AC\_TYPE) %>%

summarise(MeanValue\_AC = mean(TOTAL\_VALUE\_in\_thousands))

theGraph <- ggplot(MeanValues\_AC, aes(x = reorder(factor(AC\_TYPE), MeanValue\_AC), y = MeanValue\_AC)) +

geom\_bar(stat = "identity", color = "lightgrey",

fill = c("lightblue", "lightgrey", "lightgrey")) +

labs(x = "", y = "Condos Price (thousands)") +

labs(title = "\n AC Type adds more value to the Condos:\n

1. Central A/C Type has highest Average Condos Price.\n

2. Following is Ductless A/C.\n") +

theme\_minimal() +

theme(plot.title = element\_text(size = 14))

theGraph <- theGraph + coord\_flip()

}

if (input$factors == "BDRM\_COND") {

MeanValues\_BEDCOND <- data %>%

group\_by(BDRM\_COND) %>%

summarise(MeanValue\_BEDCOND = mean(TOTAL\_VALUE\_in\_thousands))

theGraph <- ggplot(MeanValues\_BEDCOND, aes(x = reorder(factor(BDRM\_COND), MeanValue\_BEDCOND), y = MeanValue\_BEDCOND)) +

geom\_bar(stat = "identity", color = "lightgrey",

fill = c("lightgrey", "lightblue", "lightgrey", "lightgrey", "lightgrey")) +

labs(x = "", y = "Condos Price (thousands)") +

labs(title = "\n Bedroom Condition adds more value to the Condos:\n

1. Excellent Bedroom Condition has significant highest Average Condos Price.\n

2. Average, Fair, and Poor Conditions are nearly approximate in prices.\n

3. Buyer/ Investor: Fair Bedroom Conditions is an affordable option to buy (then remodel to Excellent one).\n

4. Seller: remodel to Excellent Bedroom Condition to add value to property value.") +

theme\_minimal() +

theme(plot.title = element\_text(size = 14))

theGraph <- theGraph + coord\_flip()

}

if (input$factors == "OVERALL\_COND") {

MeanValues\_OC <- data %>%

group\_by(OVERALL\_COND) %>%

summarise(MeanValue\_OC = mean(TOTAL\_VALUE\_in\_thousands))

theGraph <- ggplot(MeanValues\_OC, aes(x = reorder(factor(OVERALL\_COND), MeanValue\_OC), y = MeanValue\_OC)) +

geom\_bar(stat = "identity", color = "lightgrey",

fill = c("lightgrey", "lightblue", "lightgrey", "lightgrey", "lightgrey", "lightblue")) +

labs(x = "", y = "Condos Price (thousands)") +

labs(title = "\n Overall Condition adds more value to the Condos:\n

1. Excellent and Very Good Overall Conditions have highest Average Condos Prices.\n

2. Good, Average, Poor, Fair Conditions are nearly approximate in prices.\n

3. Buyer/ Investor: SHOULD buy Condos with Fair or Good then 'remodel' to Excellent or Very Good.\n

4. Seller: SHOULD remodel to Excellent or Very Good Overall Conditions.") +

theme\_minimal() +

theme(plot.title = element\_text(size = 14))

theGraph <- theGraph + coord\_flip()

}

if (input$factors == "PROP\_VIEW") {

MeanValues\_VIEW <- data %>%

group\_by(PROP\_VIEW) %>%

summarise(MeanValue\_VIEW = mean(TOTAL\_VALUE\_in\_thousands))

theGraph <- ggplot(MeanValues\_VIEW, aes(x = reorder(factor(PROP\_VIEW), MeanValue\_VIEW), y = MeanValue\_VIEW)) +

geom\_bar(stat = "identity", color = "lightgrey",

fill = c("lightgrey", "lightblue", "lightgrey", "lightgrey", "lightgrey", "lightblue")) +

labs(x = "", y = "Condos Price (thousands)") +

labs(title = "\n Property View adds more value to the Condos:\n

1. Special and Excellent Views have significant higher Average Condos Prices than other views.\n

2. Average, Fair, and Poor Views are nearly approximate in prices.\n") +

theme\_minimal() +

theme(plot.title = element\_text(size = 14))

theGraph <- theGraph + coord\_flip()

}

print(theGraph)

})

# Line Charts: Annual Growth Rate

output$growthRateGraph <- renderPlot({

graphData <- aggregate(data = passData(),

cbind(GrowthRate)~CITY+YR\_BUILT

, FUN = mean)

theGraph <- ggplot(data = graphData, aes(x = YR\_BUILT, y = GrowthRate, group = CITY, colour = CITY)) +

geom\_line() +

ylab("Annual Growth Rate") +

xlab("Year Built")

print(theGraph)

})

# Summary

output$textDisplay <- renderText({

paste(

"We are group 2, including Xuejia Li, Thu Phuong Nguyen, Shinuo Xiang, Taige Zhu, Pengxiang Wu, and Yi Xu.

Our Boston Property Dataset contains 63,134 properties in the Greater Boston Area.

The dataset mainly focuses on 'Condominium' with the purpose of analyzing and understanding

which factors affect the total Property Value of the Condominium of 15 cities/towns in the Greater Boston Area.

Our target audiences are property investors, home buyers/ sellers, etc."

)

})

# Take Away

output$takeAway <- renderText({

paste("(1) Top 3 with highest Average Property Value (built after 1990): Boston, Charlestown, South Boston/Mattapan;

(2) Big picture of the Boston housing market: optimistic for Condos built in last decade (annual growth %: > 10%), except 2021;

(3) Factors could impact Condos Price: Bedroom Count, Kitchen Style, Bathroom Style, AC Type, Bedroom Cond.,

Overall Cond., especially Full Bathroom Count and Special Property View.")

})

}

# Run the application

shinyApp(ui, server)