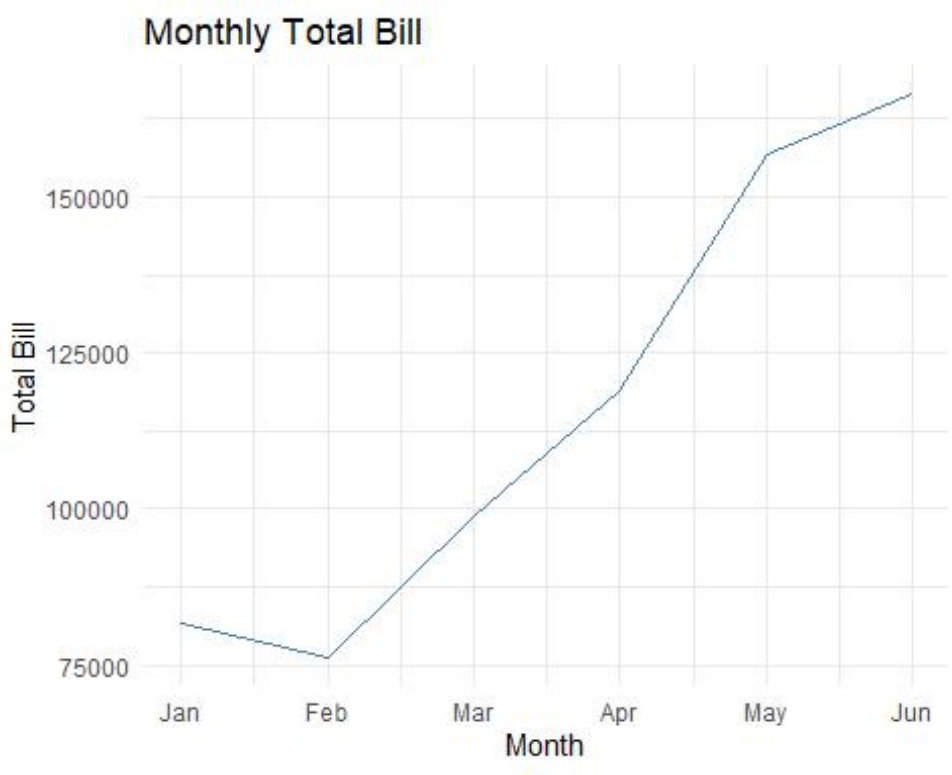
Q1.

a.

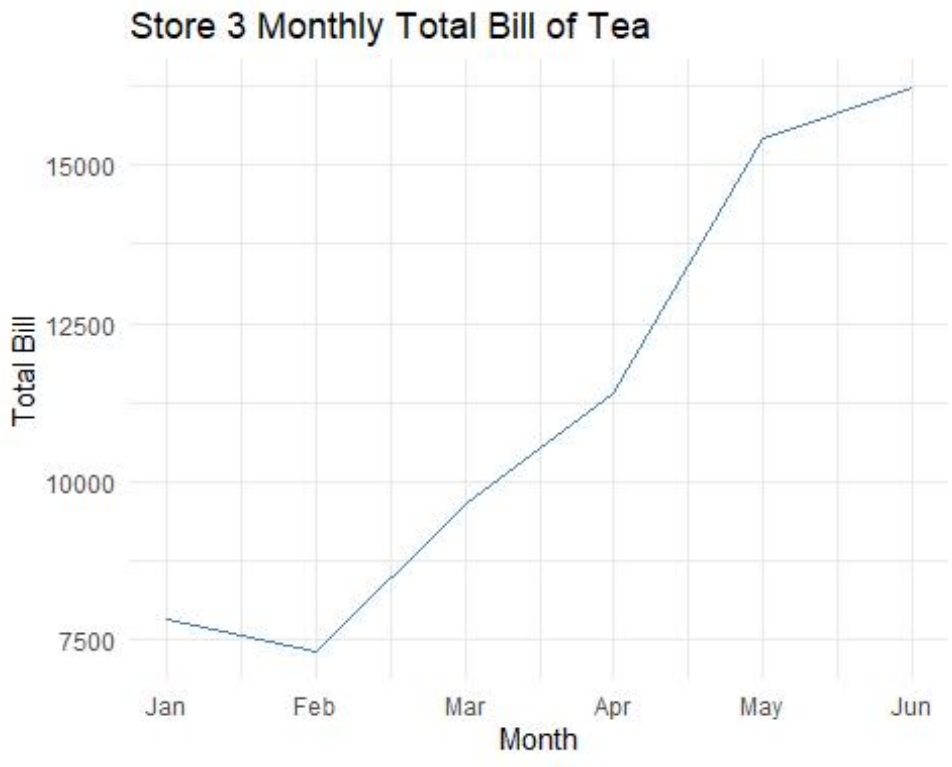
Monthly total bill of all three stores

```
df <- read.csv("Coffee_Shop_Data_cleaned.csv")

library(dplyr)

##
## 载入程辑包：'dplyr'

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

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

monthly_total_bill <- df %>%
  group_by(Month) %>%
  summarise(Total_Bill = sum(Total_Bill))

library(ggplot2)
ggplot(monthly_total_bill, aes(x = Month, y = Total_Bill)) +
  geom_line(color = "steelblue") +
  labs(x = "Month", y = "Total Bill", title = "Monthly Total Bill") +
  scale_x_continuous(breaks = 1:12, labels = c("Jan", "Feb", "Mar", "Apr
", "May", "Jun",
                                               "Jul", "Aug", "Sep", "Oct", "
Nov", "Dec")) +
  theme_minimal()
```



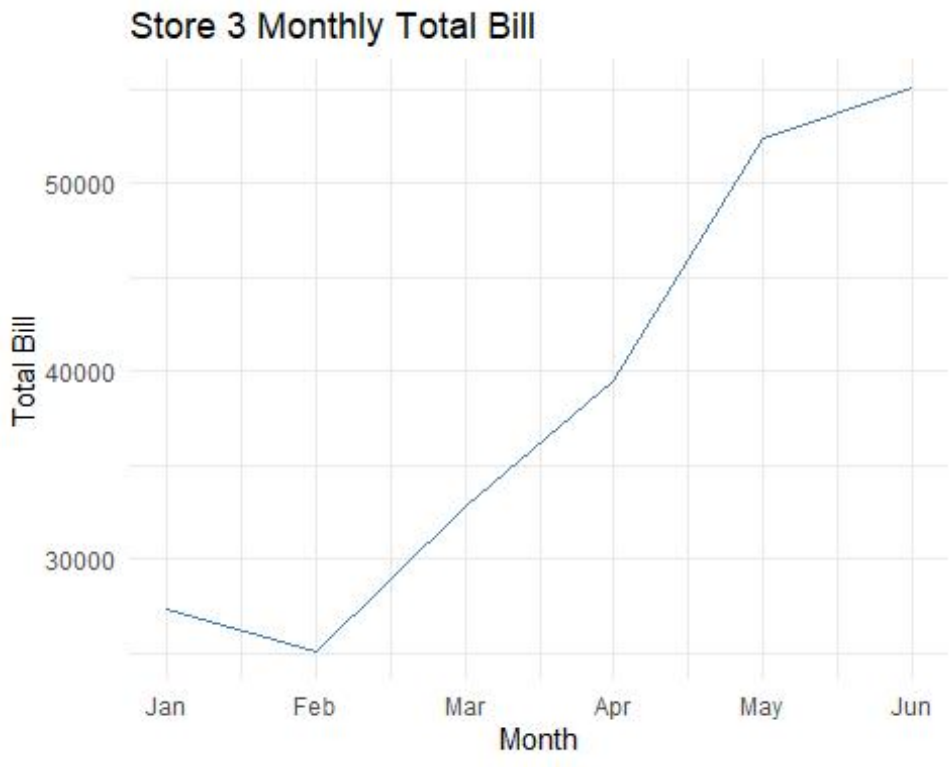
Monthly total bills of Tea of store 3

```
Tea_3_df <- subset(df, product_category == "Tea" & store_id == 3)

library(dplyr)
monthly_total_bill <- Tea_3_df %>%
  group_by(Month) %>%
  summarise(Total_Bill = sum(Total_Bill))

library(ggplot2)
ggplot(monthly_total_bill, aes(x = Month, y = Total_Bill)) +
  geom_line(color = "steelblue") +
  labs(x = "Month", y = "Total Bill", title = "Store 3 Monthly Total Bil
l of Tea") +
  scale_x_continuous(breaks = 1:12, labels = c("Jan", "Feb", "Mar", "Apr
", "May", "Jun",
                                               "Jul", "Aug", "Sep", "Oct", "
Nov", "Dec")) +
  theme_minimal()
```



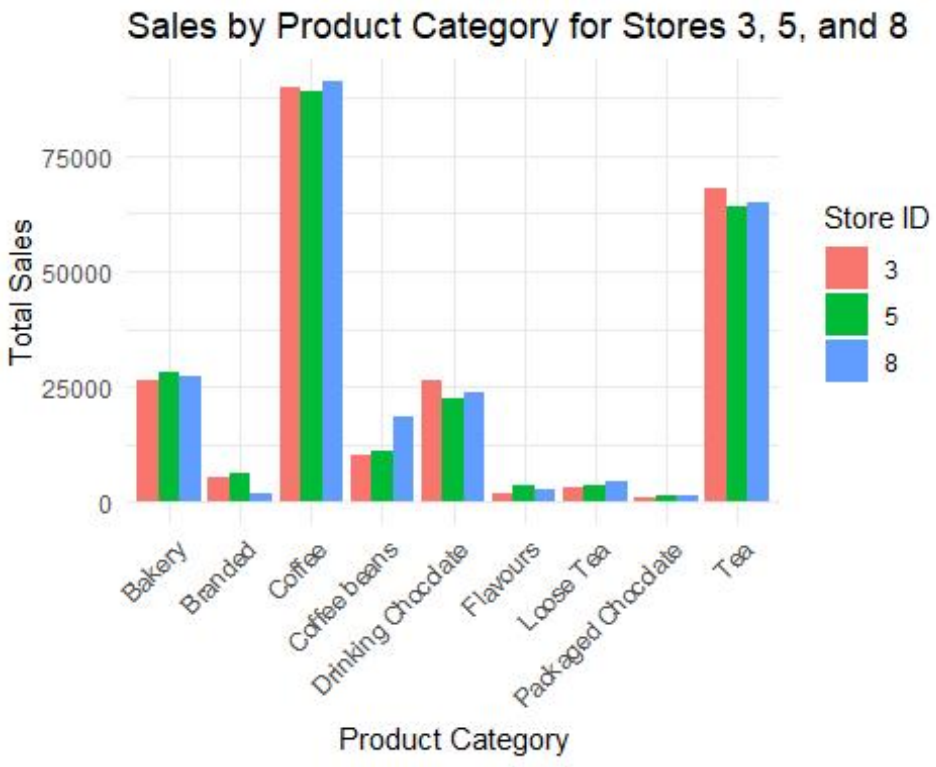
Monthly total bills of of store 3

```
store_3_df <- subset(df, store_id == 3)

library(dplyr)
monthly_total_bill <- store_3_df %>%
  group_by(Month) %>%
  summarise(Total_Bill = sum(Total_Bill))

library(ggplot2)
ggplot(monthly_total_bill, aes(x = Month, y = Total_Bill)) +
  geom_line(color = "steelblue") +
  labs(x = "Month", y = "Total Bill", title = "Store 3 Monthly Total Bil
l") +
  scale_x_continuous(breaks = 1:12, labels = c("Jan", "Feb", "Mar", "Apr
", "May", "Jun",
                                               "Jul", "Aug", "Sep", "Oct", "
Nov", "Dec")) +
  theme_minimal()
```



b.

```
library(dplyr)
sales_by_category <- df %>%
  group_by(store_id, product_category) %>%
  summarise(Total_Sales = sum(Total_Bill))

## `summarise()` has grouped output by 'store_id'. You can override usin
g the
## `.groups` argument.

library(ggplot2)
ggplot(sales_by_category, aes(x = product_category, y = Total_Sales, fil
l = factor(store_id))) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(x = "Product Category", y = "Total Sales", title = "Sales by Prod
uct Category for Stores 3, 5, and 8") +
  scale_fill_discrete(name = "Store ID") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

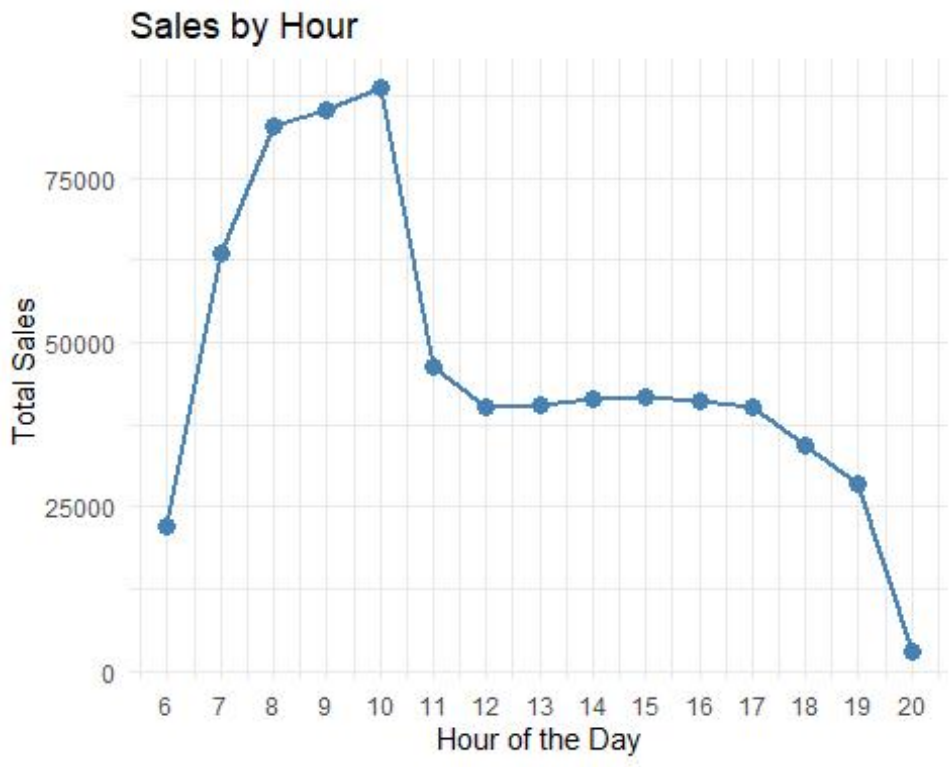


c.

line chart

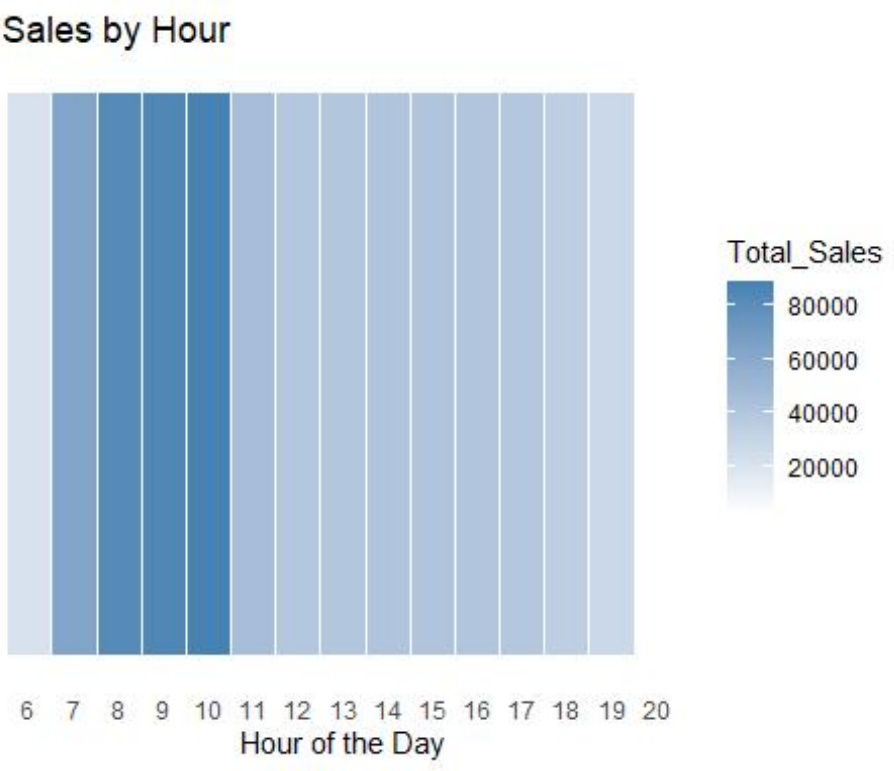
```
library(dplyr)
sales_by_hour <- df %>%
  group_by(Hour) %>%
  summarise(Total_Sales = sum(Total_Bill))

library(ggplot2)
ggplot(sales_by_hour, aes(x = Hour, y = Total_Sales)) +
  geom_line(color = "steelblue", size = 1) +
  geom_point(color = "steelblue", size = 3) +
  labs(x = "Hour of the Day", y = "Total Sales", title = "Sales by Hour")
 +
  scale_x_continuous(breaks = seq(0, 23, by = 1)) +
  theme_minimal()
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.
4.0.
## ℹ Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning
 was
## generated.
```



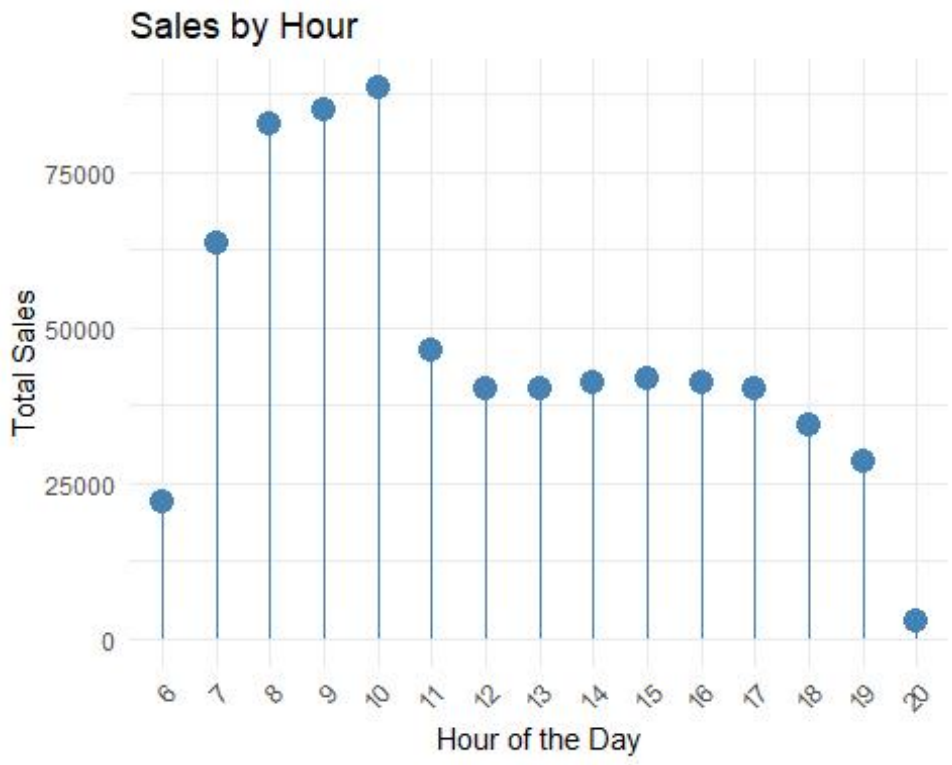
heat map

```
library(ggplot2)
ggplot(sales_by_hour, aes(x = factor(Hour), y = 1, fill = Total_Sales))
+
  geom_tile(color = "white", size = 0.5) +
  scale_fill_gradient(low = "white", high = "steelblue") +
  labs(x = "Hour of the Day", y = "", title = "Sales by Hour") +
  theme_minimal() +
  theme(axis.text.y = element_blank(),
        axis.ticks.y = element_blank(),
        panel.grid = element_blank())
```



lollipop chart

```
library(ggplot2)
ggplot(sales_by_hour, aes(x = factor(Hour), y = Total_Sales)) +
  geom_point(color = "steelblue", size = 4) +
  geom_segment(aes(x = factor(Hour), xend = factor(Hour), y = 0, yend =
Total_Sales), color = "steelblue") +
  labs(x = "Hour of the Day", y = "Total Sales", title = "Sales by Hour")
 +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Q2.

```
# install.packages("randomForest")
library(randomForest)

## randomForest 4.7-1.1

## Type rfNews() to see new features/changes/bug fixes.

##
## 载入程辑包：'randomForest'

## The following object is masked from 'package:ggplot2':
##
##     margin

## The following object is masked from 'package:dplyr':
##
##     combine

library(dplyr)

tea_sales <- df %>%
  filter(product_category == "Tea") %>%
  group_by(store_id, Day.of.Week, Hour) %>%
  summarise(Total_Bill = sum(Total_Bill))
```

```
## `summarise()` has grouped output by 'store_id', 'Day.of.Week'. You ca
n override
## using the `.groups` argument.

set.seed(123)
train_index <- sample(nrow(tea_sales), 0.7 * nrow(tea_sales))
train_data <- tea_sales[train_index, ]
test_data <- tea_sales[-train_index, ]

rf_model <- randomForest(Total_Bill ~ store_id + Day.of.Week + Hour,
                         data = train_data,
                         ntree = 500,
                         mtry = 2,
                         importance = TRUE)

# Make predictions on the test data
predictions <- predict(rf_model, newdata = test_data)

# Calculate the mean squared error (MSE)
mse <- mean((test_data$Total_Bill - predictions)^2)
print(paste0("Mean Squared Error (MSE): ", mse))

## [1] "Mean Squared Error (MSE): 8019.75246870488"

# Calculate the root mean squared error (RMSE)
rmse <- sqrt(mse)
print(paste0("Root Mean Squared Error (RMSE): ", rmse))

## [1] "Root Mean Squared Error (RMSE): 89.5530706827236"

# Calculate the R-squared (R^2)
r_squared <- 1 - (sum((test_data$Total_Bill - predictions)^2) / sum((tes
t_data$Total_Bill - mean(test_data$Total_Bill))^2))
print(paste0("R-squared (R^2): ", r_squared))

## [1] "R-squared (R^2): 0.905851146727362"

importance_scores <- importance(rf_model)
print(importance_scores)

##              %IncMSE IncNodePurity
## store_id     61.68627     2651828.2
## Day.of.Week -11.06519      714362.8
## Hour        102.86598    15624045.0

new_data <- data.frame(
  store_id = c(3, 5, 8),
  Day.of.Week = rep("Wednesday", 3),
  Hour = rep(11, 3)
)
```

```r
predictions <- predict(rf_model, newdata = new_data)

results <- data.frame(new_data, Total_Bill = predictions)
print(results)
```

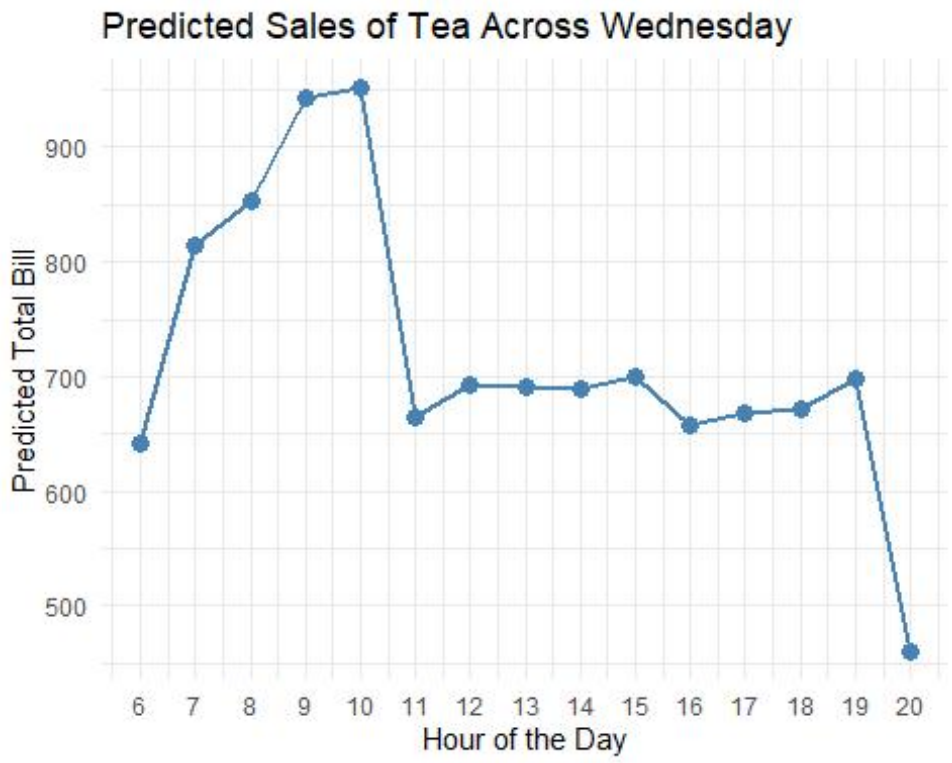
```
##   store_id Day.of.Week Hour Total_Bill
## 1        3   Wednesday   11   657.3200
## 2        5   Wednesday   11   530.0715
## 3        8   Wednesday   11   547.5709
```

Extra Credit

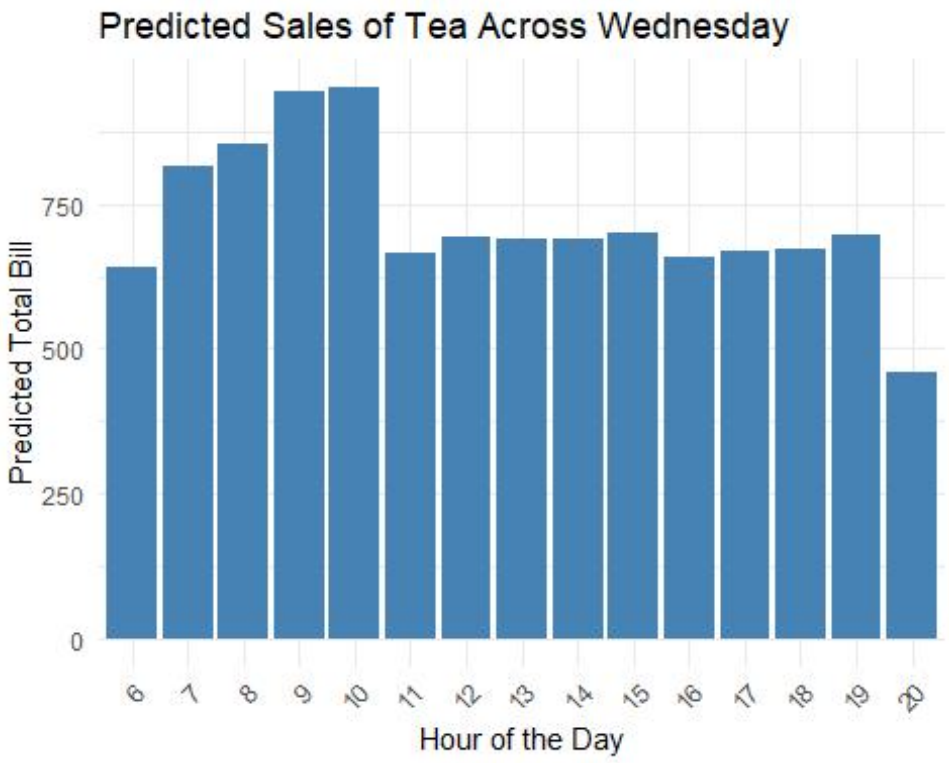
```r
new_data <- data.frame(
  store_id = rep(3, 15),
  Day.of.Week = rep(2,15),
  Name.Day = ("Wednesday"),
  product_catery = ("Tea"),
  Hour = seq(6, 20)
)

predictions <- predict(rf_model, newdata = new_data)
results <- data.frame(new_data, Total_Bill = predictions)
results
```

```
##    store_id Day.of.Week  Name.Day product_catery Hour Total_Bill
## 1         3           2 Wednesday           Tea    6   641.1136
## 2         3           2 Wednesday           Tea    7   814.7872
## 3         3           2 Wednesday           Tea    8   853.3714
## 4         3           2 Wednesday           Tea    9   943.6609
## 5         3           2 Wednesday           Tea   10   951.7408
## 6         3           2 Wednesday           Tea   11   664.2715
## 7         3           2 Wednesday           Tea   12   693.6467
## 8         3           2 Wednesday           Tea   13   690.5155
## 9         3           2 Wednesday           Tea   14   689.8835
## 10        3           2 Wednesday           Tea   15   700.4262
## 11        3           2 Wednesday           Tea   16   658.0419
## 12        3           2 Wednesday           Tea   17   667.8849
## 13        3           2 Wednesday           Tea   18   672.1328
## 14        3           2 Wednesday           Tea   19   697.2537
## 15        3           2 Wednesday           Tea   20   460.2027
```

```r
library(ggplot2)
# Line plot
ggplot(results, aes(x = Hour, y = Total_Bill)) +
  geom_line(color = "steelblue", size = 1) +
  geom_point(color = "steelblue", size = 3) +
  labs(x = "Hour of the Day", y = "Predicted Total Bill", title = "Predi
cted Sales of Tea Across Wednesday") +
  scale_x_continuous(breaks = seq(0, 23, by = 1)) +
  theme_minimal()
```




```
# Bar plot
ggplot(results, aes(x = factor(Hour), y = Total_Bill)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  labs(x = "Hour of the Day", y = "Predicted Total Bill", title = "Predi
cted Sales of Tea Across Wednesday") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```


Predicted Sales of Tea Across Wednesday
Predicted Total Bill
750
500
250
0
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
Hour of the Day