Mall Customers Analytics

W. Kerney

2024-05

Project Introduction

This project will focus on analyzing customer behavior for a shopping mall, including funnel optimization, user segmentation, cohort analyses, and time series analyses. We'll also perform A/B testing and statistical analyses to derive insights.

Step 1: Setting Up Your Environment

Installing the necessary R packages and reading libraries in.

```
install.packages("tidyverse")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.4'
## (as 'lib' is unspecified)
```

```
install.packages("data.table")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.4'
## (as 'lib' is unspecified)
```

```
install.packages("lubridate")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.4'
## (as 'lib' is unspecified)
```

```
install.packages("ggplot2")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.4'
## (as 'lib' is unspecified)
```

```
install.packages("caret")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.4'
## (as 'lib' is unspecified)
```

```
install.packages("dplyr")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.4'
## (as 'lib' is unspecified)
```

```
install.packages("sqldf")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.4'
## (as 'lib' is unspecified)
```

```
install.packages("prophet")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.4'
## (as 'lib' is unspecified)
```

```r
install.packages("readr")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.4'
## (as 'lib' is unspecified)
```

```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(data.table)
```

```
##
## Attaching package: 'data.table'
##
## The following objects are masked from 'package:lubridate':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year
##
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
##
## The following object is masked from 'package:purrr':
##
##     transpose
```

```r
library(lubridate)
library(ggplot2)
library(caret)
```

```
## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```r
library(dplyr)
library(sqldf)
```

```
## Loading required package: gsubfn
## Loading required package: proto
```

```
## Warning in fun(libname, pkgname): couldn't connect to display ":0"
```

```
## Loading required package: RSQLite
```

```
library(prophet)
```

```
## Loading required package: Rcpp
## Loading required package: rlang
##
## Attaching package: 'rlang'
##
## The following object is masked from 'package:data.table':
##
##     :=
##
## The following objects are masked from 'package:purrr':
##
##     %@%, flatten, flatten_chr, flatten_dbl, flatten_int, flatten_lgl,
##     flatten_raw, invoke, splice
```

```
library(readr)
```

Step 2: Load and Explore the Data

Loading the dataset into R and exploring its structure and summary statistics.

```
#loading and exploring dataset
data<- read.csv("Mall_Customers.csv")
str(data)
```

```
## 'data.frame':    200 obs. of  5 variables:
##  $ CustomerID            : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Genre                 : chr  "Male" "Male" "Female" "Female" ...
##  $ Age                   : int  19 21 20 23 31 22 35 23 64 30 ...
##  $ Annual.Income..k..    : int  15 15 16 16 17 17 18 18 19 19 ...
##  $ Spending.Score..1.100.: int  39 81 6 77 40 76 6 94 3 72 ...
```

```
summary(data)
```

```
##    CustomerID        Genre                Age        Annual.Income..k..
##  Min.   :  1.00   Length:200         Min.   :18.00   Min.   : 15.00
##  1st Qu.: 50.75   Class :character   1st Qu.:28.75   1st Qu.: 41.50
##  Median :100.50   Mode  :character   Median :36.00   Median : 61.50
##  Mean   :100.50                      Mean   :38.85   Mean   : 60.56
##  3rd Qu.:150.25                      3rd Qu.:49.00   3rd Qu.: 78.00
##  Max.   :200.00                      Max.   :70.00   Max.   :137.00
##  Spending.Score..1.100.
##  Min.   : 1.00
##  1st Qu.:34.75
##  Median :50.00
##  Mean   :50.20
##  3rd Qu.:73.00
##  Max.   :99.00
```

Step 3: Cleaning and processing

Found some minor data inconsistencies and cleaned those up.

```
#correcting mistake found in raw data file
colnames(data)[2] <- "Gender"
head(data)
```

```
##   CustomerID Gender Age Annual.Income..k.. Spending.Score..1.100.
## 1          1   Male  19                 15                     39
## 2          2   Male  21                 15                     81
## 3          3 Female  20                 16                      6
## 4          4 Female  23                 16                     77
## 5          5 Female  31                 17                     40
## 6          6 Female  22                 17                     76
```

Step 4: Funnel Analysis

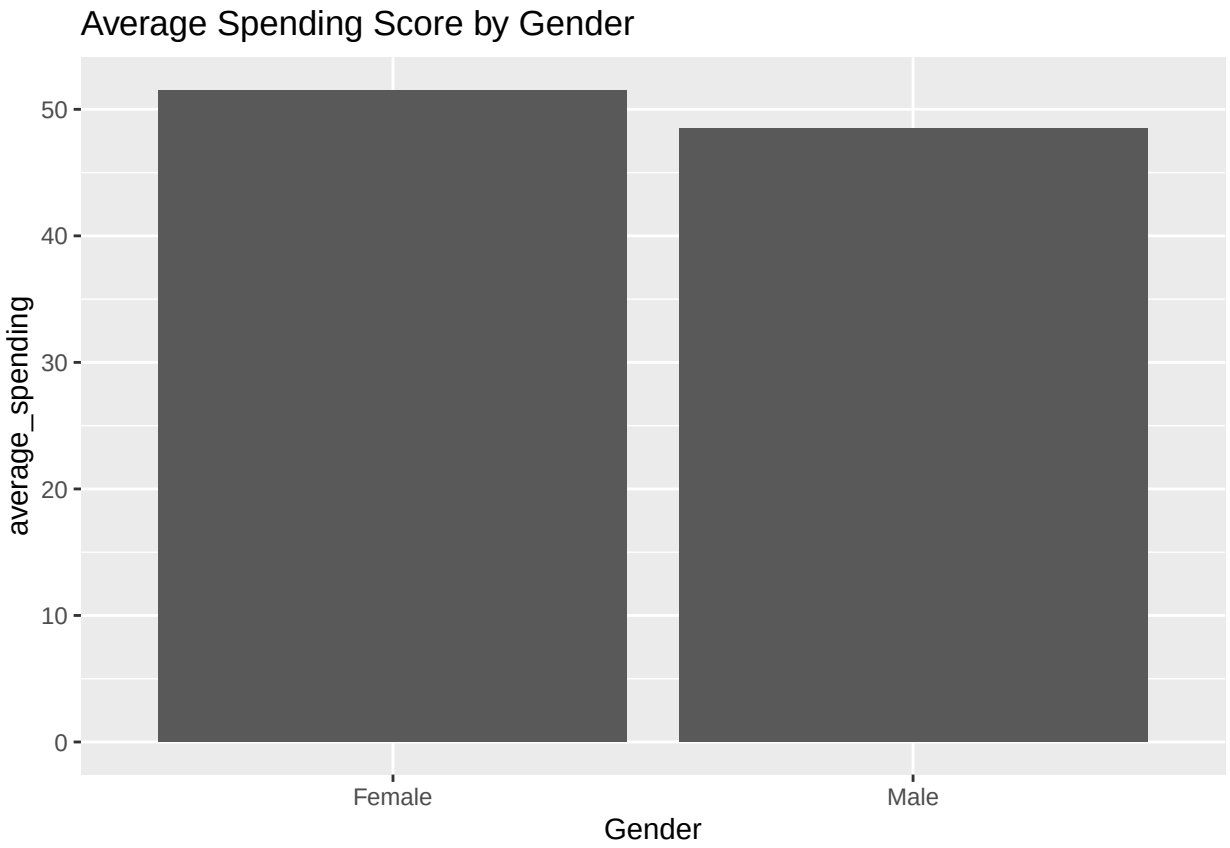
Since this dataset doesn't have a specific shopping funnel, let's focus on customer behavior. Identify patterns such as high or low spenders based on "Annual Income (k$)" and "Spending Score (1-100)."

```
#using funnel to focus on customer behavior

funnel <- data %>%
  mutate(
    high_income = Annual.Income..k.. > 75,
    high_spending = Spending.Score..1.100. > 75
  ) %>%
  summarize(
    total_customers = n(),
    high_income_count = sum(high_income),
    high_spending_count = sum(high_spending),
    high_income_spending = sum(high_income & high_spending)
  )

funnel_conversion_rate <- funnel %>%
  summarize(
    income_to_spending = high_income_spending / high_income_count
  )
```

Step 5: User Segmentation

We then segment customers based on demographic characteristics.

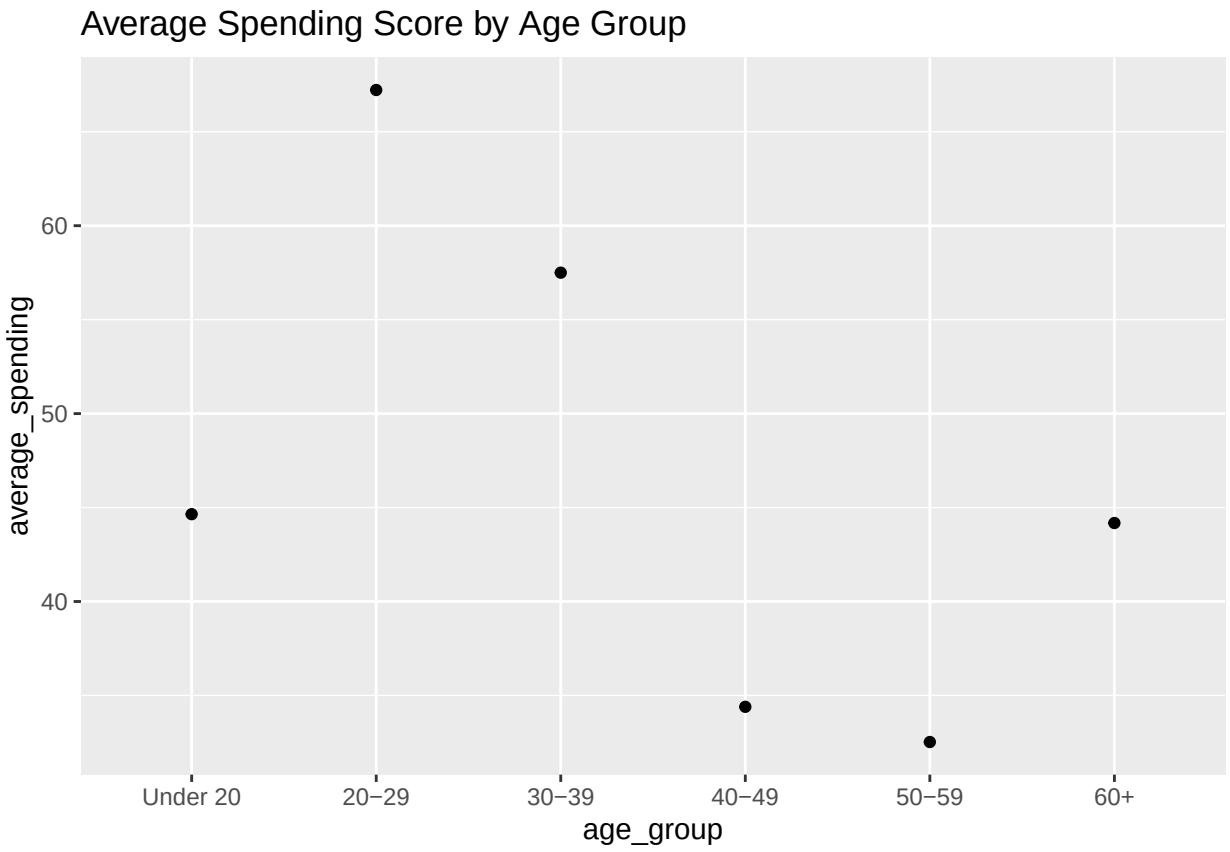
```
#user segmentation
user_segments <- data %>%
  group_by(Gender) %>%
  summarize(
    average_income = mean(Annual.Income..k..),
    average_spending = mean(Spending.Score..1.100.)
  )

ggplot(user_segments, aes(x = Gender, y = average_spending)) +
  geom_bar(stat = "identity") +
  ggtitle("Average Spending Score by Gender")
```



Step 6: Cohort Analysis

We then examine customer behavior over different age groups.

```
#cohort analysis
cohort_analysis <- data %>%
  mutate(age_group = cut(Age, breaks = c(0, 20, 30, 40, 50, 60, 100), labels = c("Under 20", "20-29", "3
  group_by(age_group) %>%
  summarize(
    average_income = mean(Annual.Income..k..),
    average_spending = mean(Spending.Score..1.100.)
  )

ggplot(cohort_analysis, aes(x = age_group, y = average_spending)) +
  geom_point() +
  ggtitle("Average Spending Score by Age Group")
```



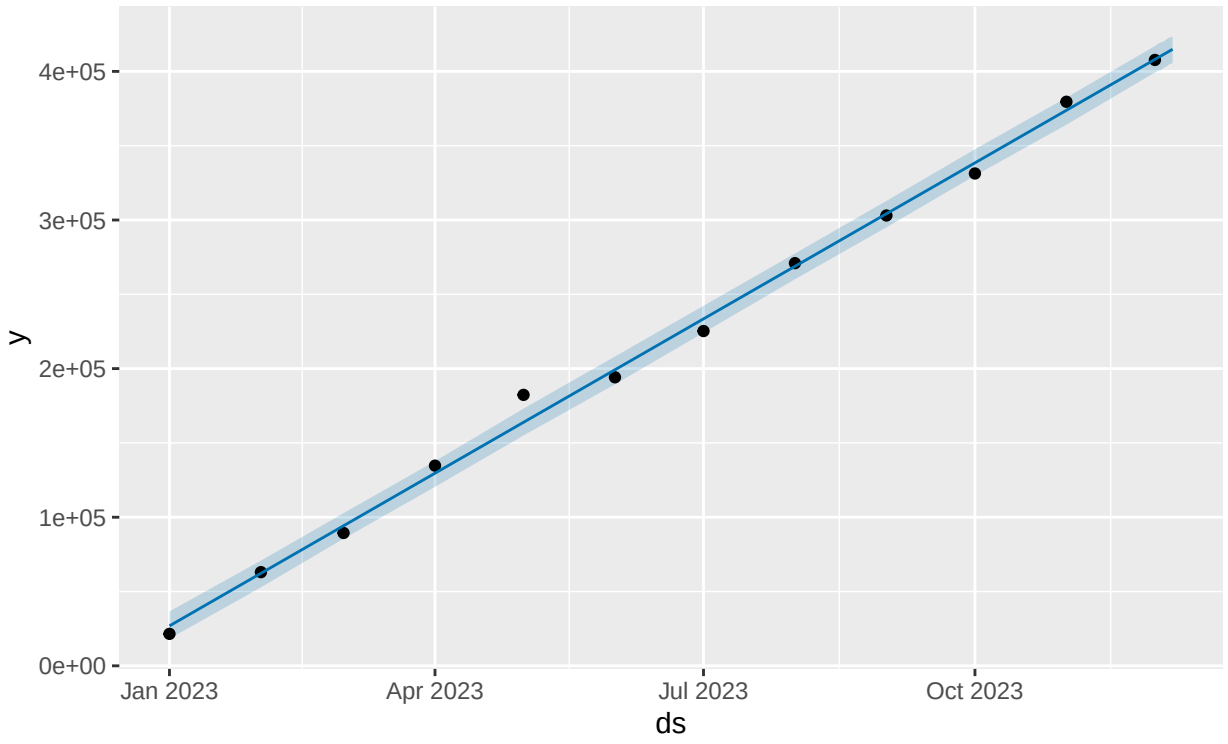
Step 7: Time Series Analysis

Although this dataset lacks clear time-based data, we create a hypothetical time series by simulating data.

```
set.seed(123)
time_series <- data.frame(
  date = seq(as.Date("2023-01-01"), by = "month", length.out = 12),
  revenue = cumsum(runif(12, 10000, 50000))
)

m <- prophet(time_series %>% rename(ds = date, y = revenue))
```

```
## Disabling yearly seasonality. Run prophet with yearly.seasonality=TRUE to override this.
## Disabling weekly seasonality. Run prophet with weekly.seasonality=TRUE to override this.
## Disabling daily seasonality. Run prophet with daily.seasonality=TRUE to override this.
## n.changepoints greater than number of observations. Using 8
```

```
forecast <- predict(m, make_future_dataframe(m, periods = 6))

plot(m, forecast)
```



Step 8: A/B Testing and Statistical Analysis

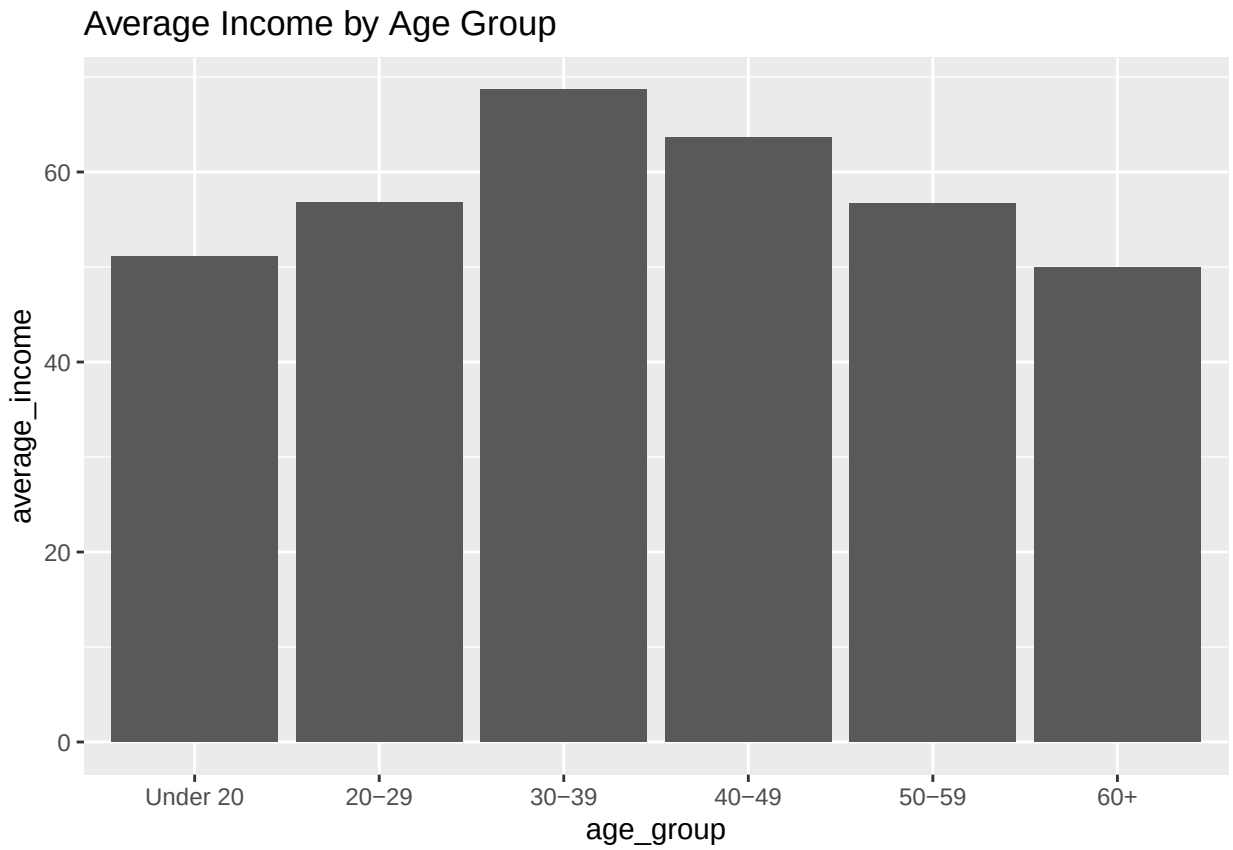
Conducting a simple A/B test to simulate the effect of different promotional campaigns.

```
#A/B testing and statistical analysis
set.seed(123)
ab_test <- data.frame(
  CustomerID = sample(1:200, 100),
  group = sample(c("Control", "Promotion"), 100, replace = TRUE),
  spending = rnorm(100, mean = 50, sd = 10)
)

ab_test$group_spending <- ifelse(ab_test$group == "Promotion", ab_test$spending * 1.2, ab_test$spending)

# Perform a t-test to compare spending between groups
t_test_result <- t.test(group_spending ~ group, data = ab_test)
```

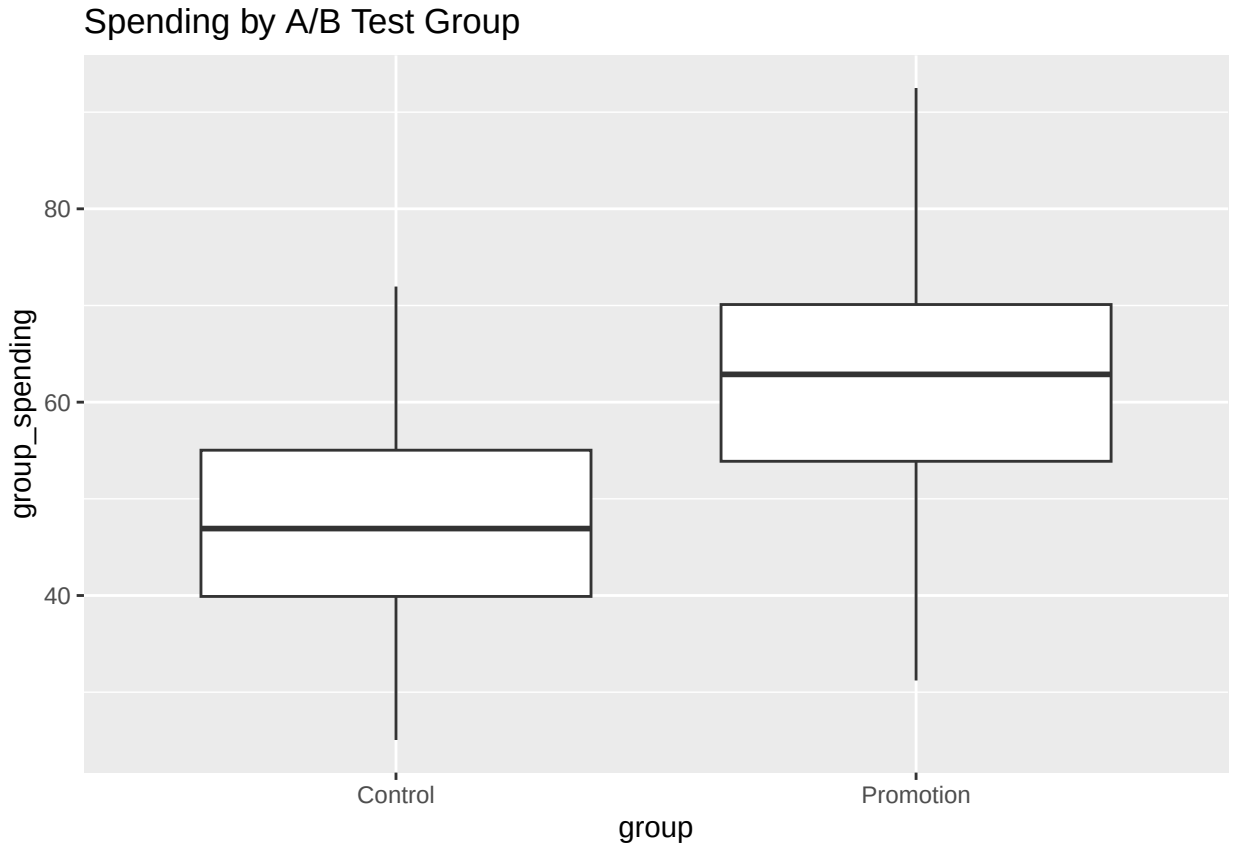
Step 9: Create Visualizations

Creating some visualizations to explain the analysis.

```
#More visualizations
ggplot(cohort_analysis, aes(x = age_group, y = average_income)) +
  geom_bar(stat = "identity") +
  ggtitle("Average Income by Age Group")
```




```
ggplot(ab_test, aes(x = group, y = group_spending)) +
  geom_boxplot() +
  ggtitle("Spending by A/B Test Group")
```



‘‘‘

Conclusion

Looking at our final visualizations, we see that our testing shows us that it would be benificial for the mall to run a promotion, specifically targeting over 40’s males in an attempt to boost the sales in those specific demograpics within the mall. We found that the over 40 age group accounts for a little less than than half of the average income of customers coming into the mall.