Regional Analysis of a Supermarket EDA

MTTP Team 3

2024-02-27

Setup

Loading in packages

```
library(tidyverse)
library(RColorBrewer) #Make colors for ggplots
library(readxl) #Reading Excel Files In
library(vcd) #Mosaic
library(caret)#Machine Learning
library(randomForest)#Machine Learning
library(ggpmisc)#R and R^2 statistics
library(gplots)#Balloonplots
```

Loading in Data

```
data_set <- read_excel("(US) Sample - Superstore.xls")

returns <- read_excel("(US) Sample - Superstore.xls", sheet = "Returns")
```

Filtering out duplicates from returns dataset

```
returns <- distinct(returns)
```

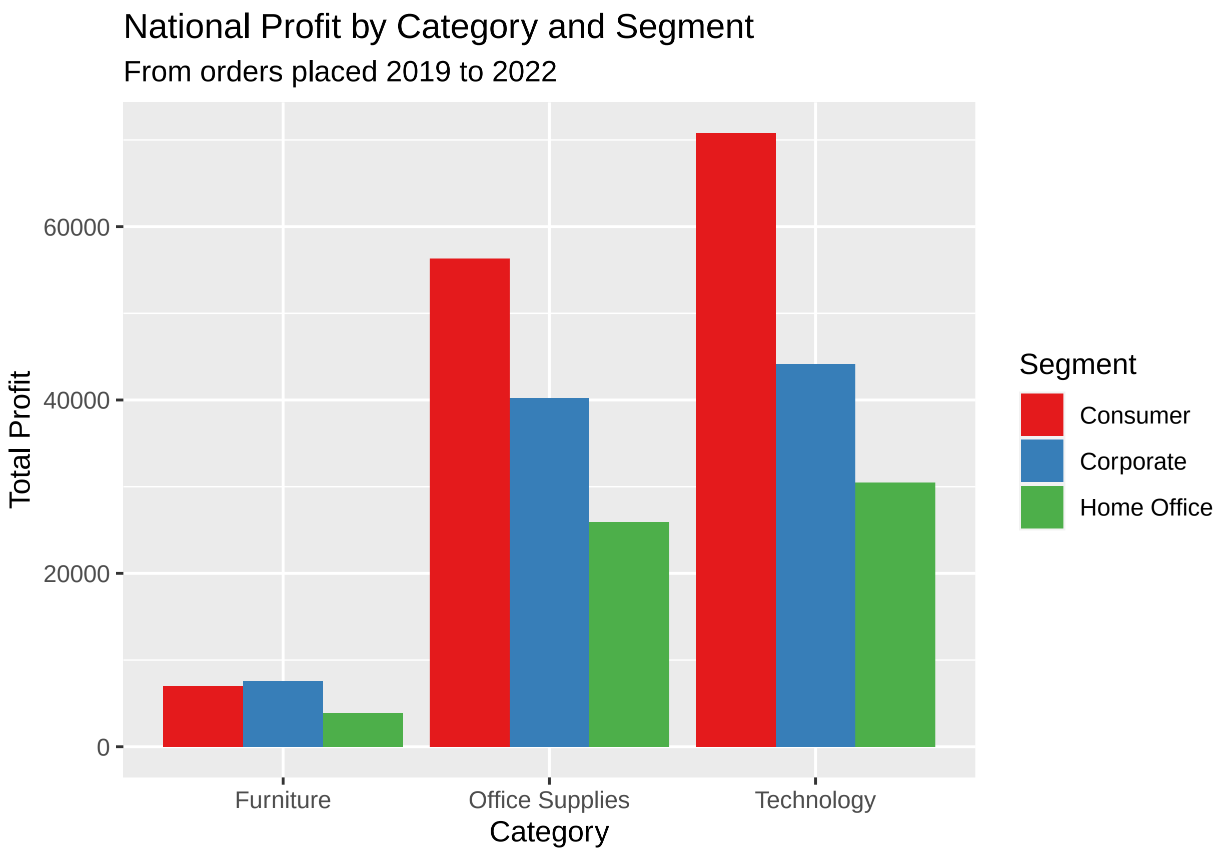
Left-joining Returns

```
data_set_returns <- data_set %>%
  left_join(returns, "Order ID") %>%
  mutate(Returned = coalesce(Returned, "No"))
```

National data

National Profit by Category and Segment

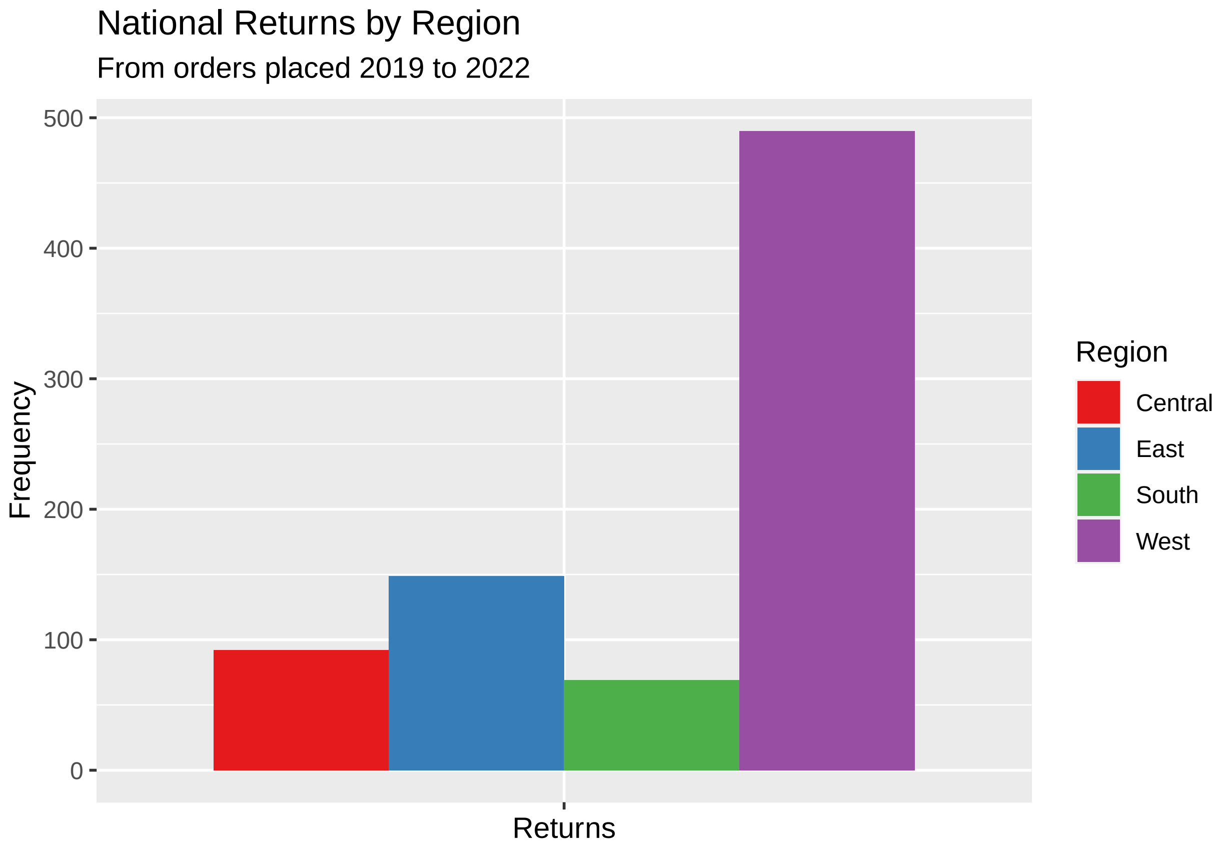
```r
#Colors
brewer_palette <- brewer.pal(n = 3, name = "Set1")

#Getting the total profit
national_profit <- data_set %>%
  select(Segment, Category, Profit) %>%
  group_by(Segment, Category) %>%
  mutate(total_profit = sum(Profit)) %>%

  #Graphing results
  ggplot(aes(x = Category, y = total_profit, fill = Segment)) +
  geom_bar(position = "dodge", stat = "identity") +
  labs(
    x = "Category",
    y = "Total Profit",
    title = "National Profit by Category and Segment",
    subtitle = "From orders placed 2019 to 2022"
  ) +
  scale_fill_manual(values = brewer_palette)

#Saving the ggplot
ggsave("National Profit by Category and Segment.png",
       plot = national_profit,
       width = 6,
       height = 4)

national_profit
```



National Returns by Region

```
#Colors
brewer_palette <- brewer.pal(n = 4, name = "Set1")

#Getting the total returns
national_returns <- data_set_returns %>%
  select(Region, Returned) %>%
  filter(Returned == "Yes") %>%
  group_by(Region) %>%

  #Graphing the results
  ggplot(aes(x = Returned, fill = Region)) +
  geom_bar(position = "dodge") +
  labs(
    x = "Returns",
    y = "Frequency",
    title = "National Returns by Region",
    subtitle = "From orders placed 2019 to 2022"
  ) +
  #Removing the "Yes's from x axis"
  theme(axis.text.x = element_blank()) +
  scale_fill_manual(values = brewer_palette)
```

```r
#Saving the ggplot
ggsave("National Returns by Region.png",
       plot = national_returns,
       width = 6,
       height = 4)

national_returns
```



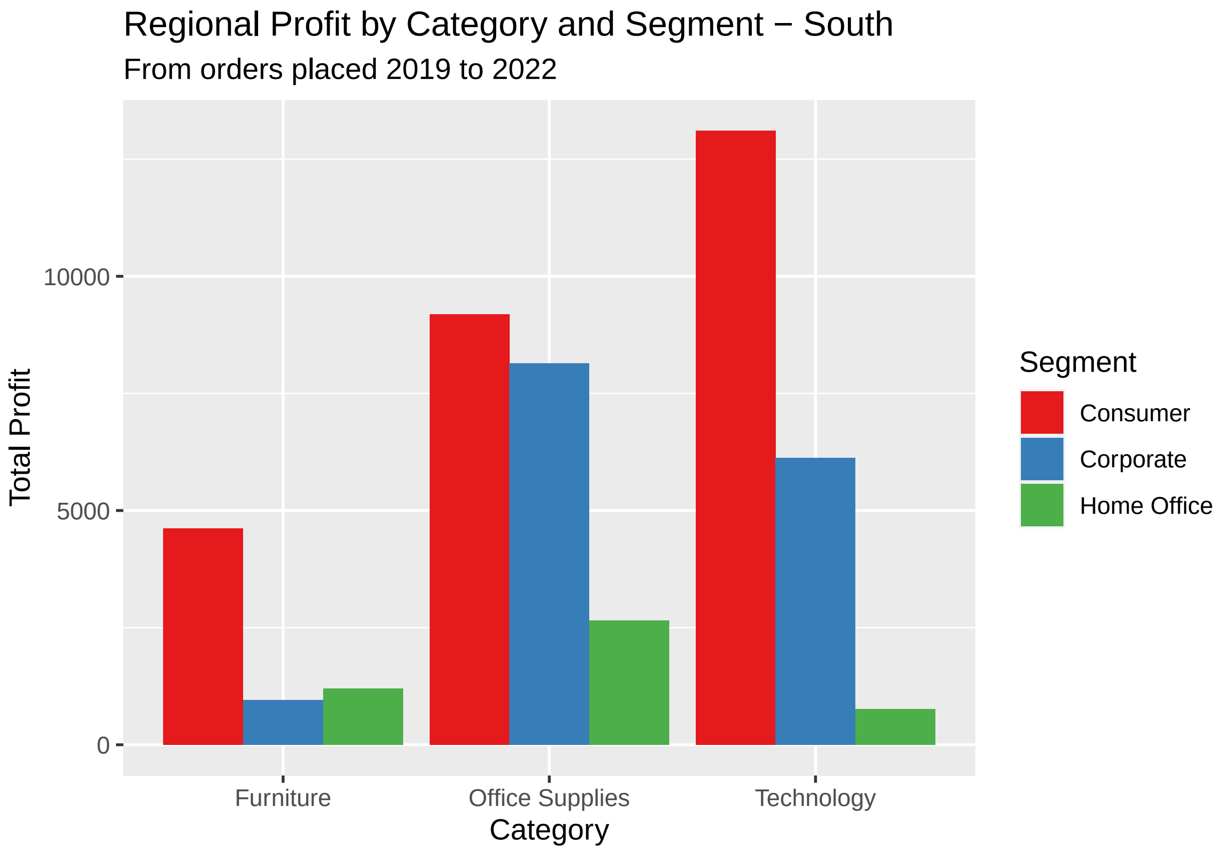
Southern Analysis - Ethan

Filtering for Southern Region

```r
south_returns <- data_set_returns %>%
  filter(Region == "South")
```

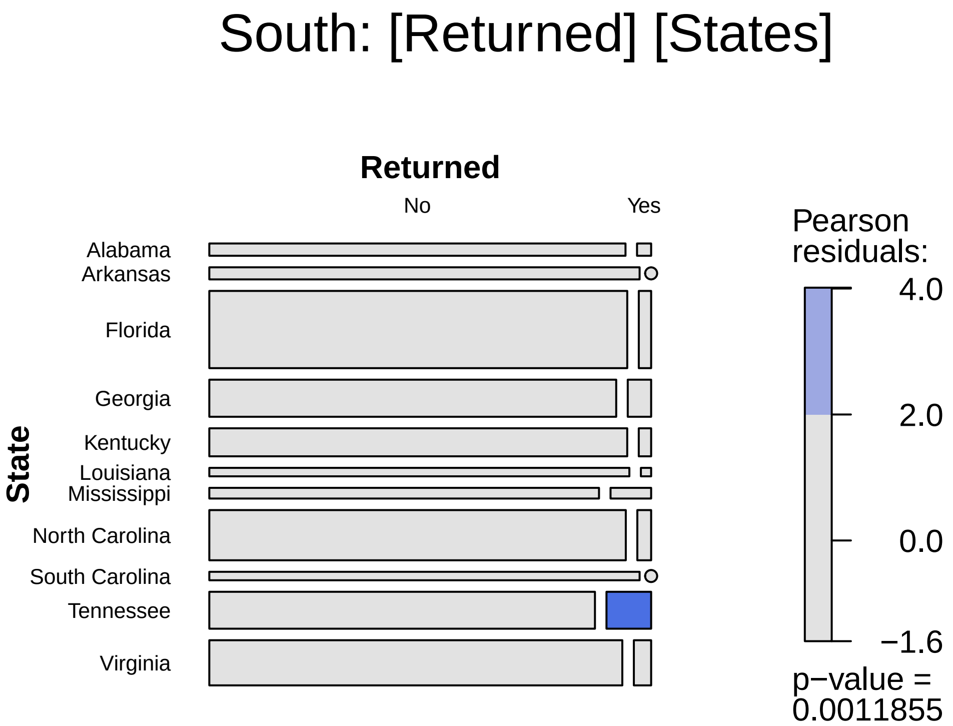
Regional Profit by Category and Segment | South

```r
regional_profit <- south_returns %>%
  filter(Region == "South") %>%
  select(Segment, Category, Profit) %>%
  group_by(Segment, Category) %>%
  mutate(total_profit = sum(Profit)) %>%

  ggplot(aes(x = Category, y = total_profit, fill = Segment)) +
  geom_bar(position = "dodge", stat = "identity") +
  labs(
    x = "Category",
    y = "Total Profit",
    title = "Regional Profit by Category and Segment - South",
    subtitle = "From orders placed 2019 to 2022"
  ) +
  scale_fill_manual(values = brewer_palette)

ggsave("Regional Profit by Category and Segment - South.png",
       plot = regional_profit,
       width = 6,
       height = 4)

regional_profit
```



Regional Returns by State | South

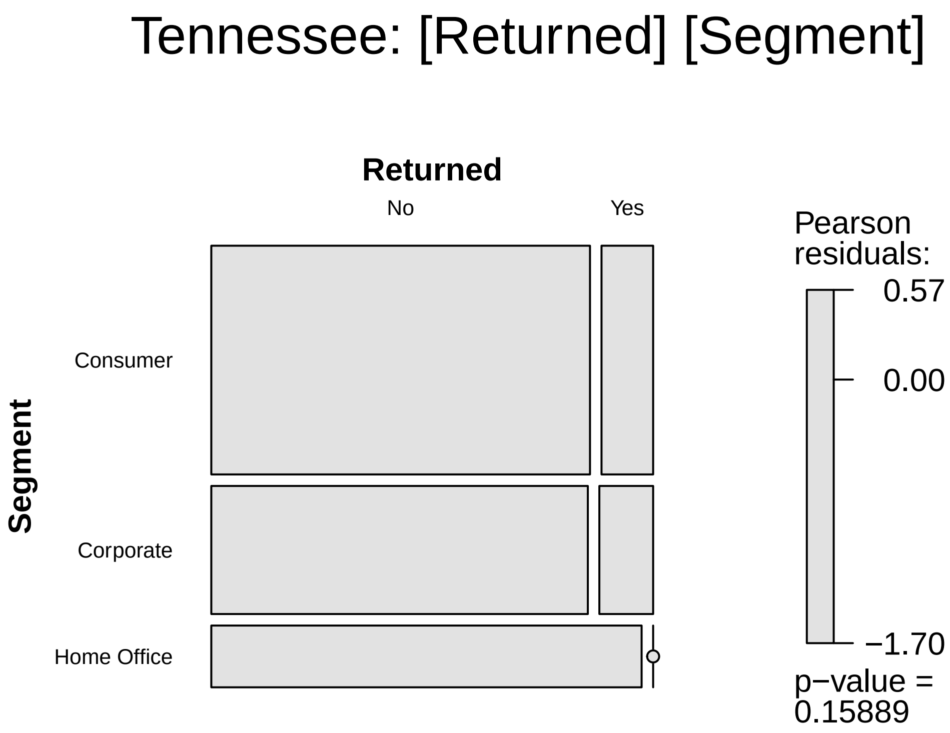
```r
state_v_returns_S <-  xtabs(~Returned + State, data = south_returns)
#S = South

mosaic(
  t(state_v_returns_S),
  gp = shading_hcl,
  main = "South: [Returned] [States]",
  labeling = labeling_border
    (
      varnames = c(TRUE, TRUE),
      offset_varnames = c(0, 0, 0, 3),
      rot_labels = c(0,0, 0, 0),
      offset_label = c(0.5,0,0, 0.5),
      just_labels = c("center","right"),
      gp_labels = gpar(fontsize = 8)
    )
)
```



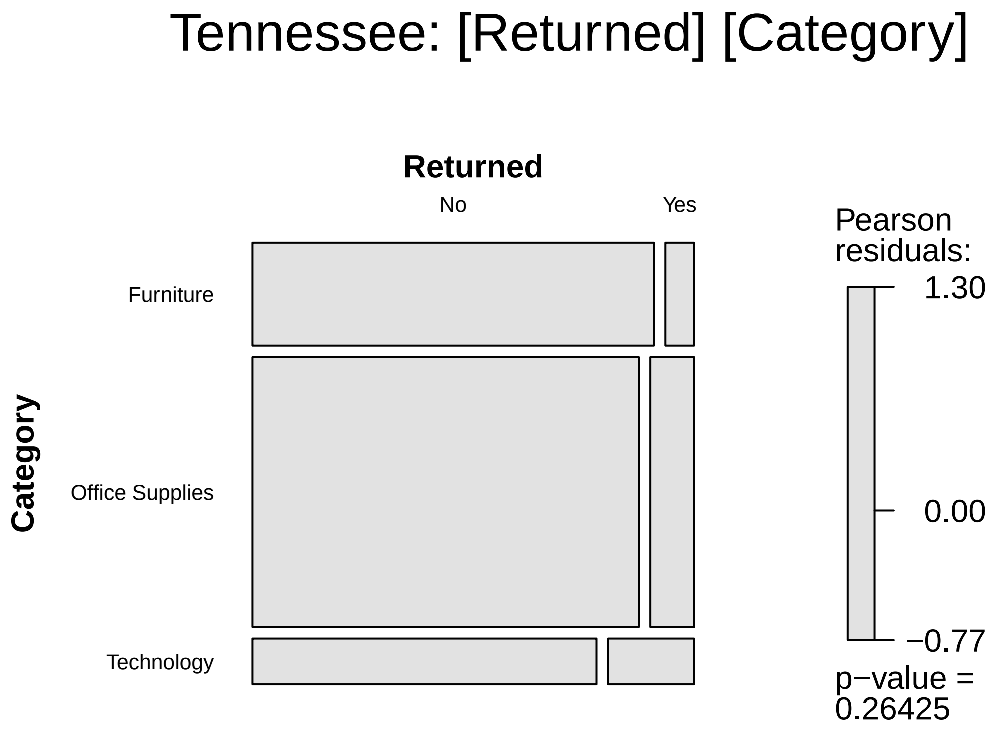
Mosiac plots within Tennessee | South

```r
tennessee_returns <- south_returns %>%
  filter(State == "Tennessee")
```

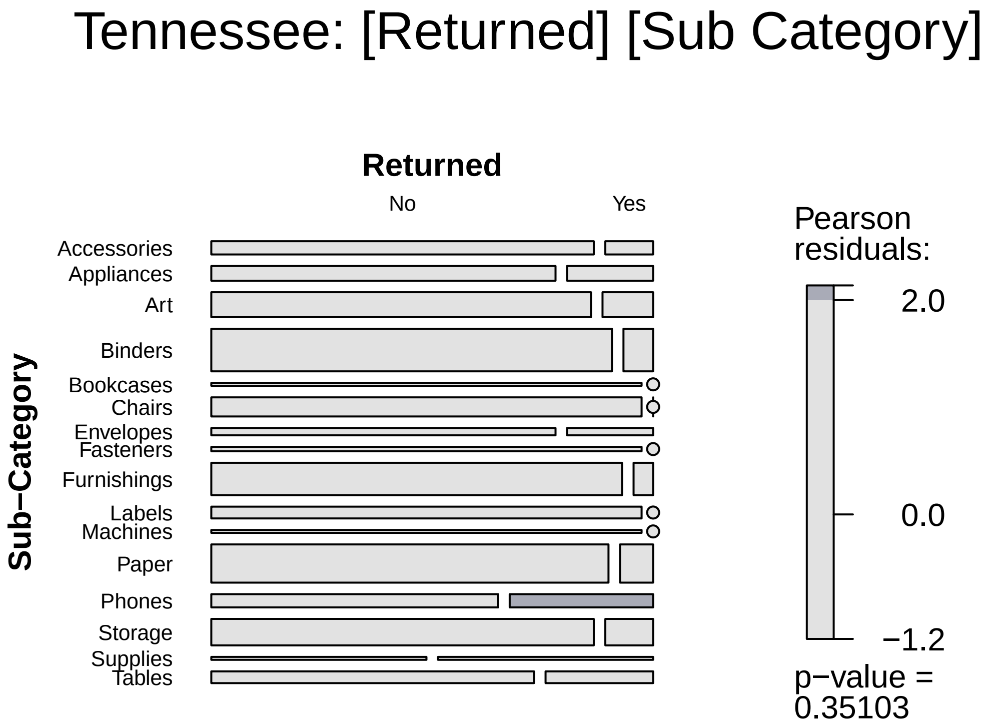
```
# Segment_v_returns

segment_v_returns_TN <- xtabs(~Returned + Segment, data = tennessee_returns)

  mosaic(
  t(segment_v_returns_TN),
  gp = shading_hcl,
  main = "Tennessee: [Returned] [Segment]",
  labeling = labeling_border
    (
      varnames = c(TRUE, TRUE),
      offset_varnames = c(0, 0, 0, 3),
      rot_labels = c(0,0, 0, 0),
      offset_label = c(0.5,0,0, 0.5),
      just_labels = c("center","right"),
      gp_labels = gpar(fontsize = 8)
    )
)
```



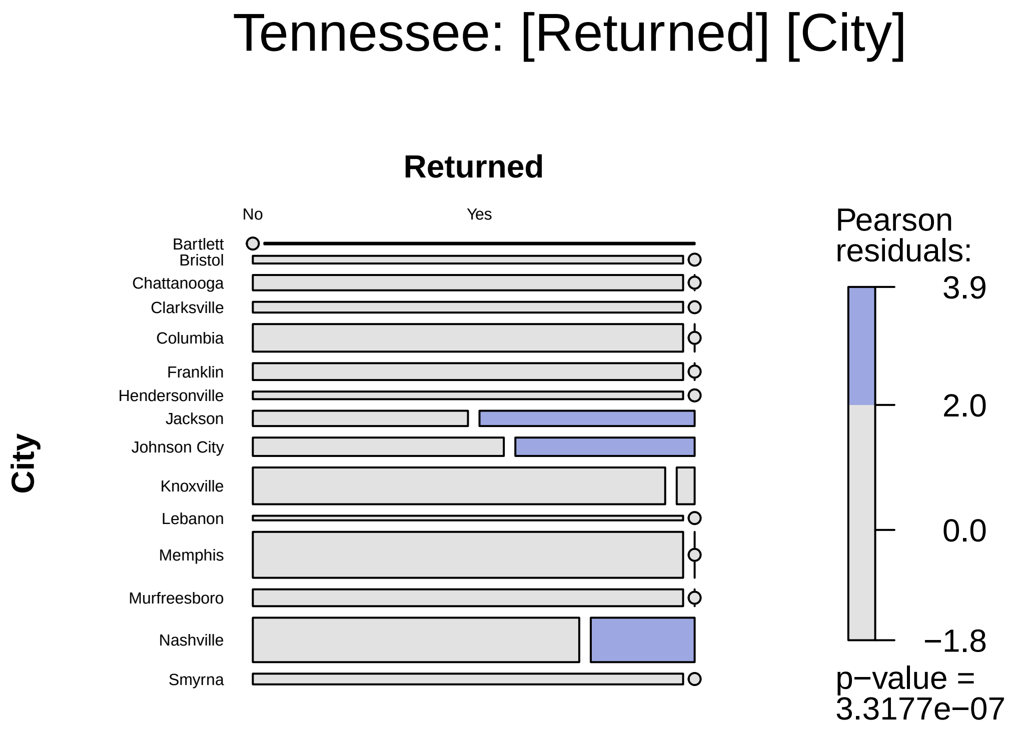
```
# Category_v_returns

category_v_returns_TN <- xtabs(~Returned + Category, data = tennessee_returns)

mosaic(
  t(category_v_returns_TN),
  gp = shading_hcl,
  main = "Tennessee: [Returned] [Category]",
  labeling = labeling_border
    (
      varnames = c(TRUE, TRUE),
      offset_varnames = c(0, 0, 0, 4),
      rot_labels = c(0,0, 0, 0),
      offset_label = c(0.5,0,0, 0.5),
      just_labels = c("center","right"),
      gp_labels = gpar(fontsize = 8)
    )
)
```



```
# Sub_category_v_returns

sub_category_v_returns_TN <- xtabs(~Returned + `Sub-Category`,
                                   data = tennessee_returns)

mosaic(
  t(sub_category_v_returns_TN),
  gp = shading_hcl,
  main = "Tennessee: [Returned] [Sub Category]",
  labeling = labeling_border
    (
      varnames = c(TRUE, TRUE),
      offset_varnames = c(0, 0, 0, 3),
      rot_labels = c(0,0, 0, 0),
      offset_label = c(0.5,0,0, 0.5),
      just_labels = c("center","right"),
      gp_labels = gpar(fontsize = 8)
    )
)
```

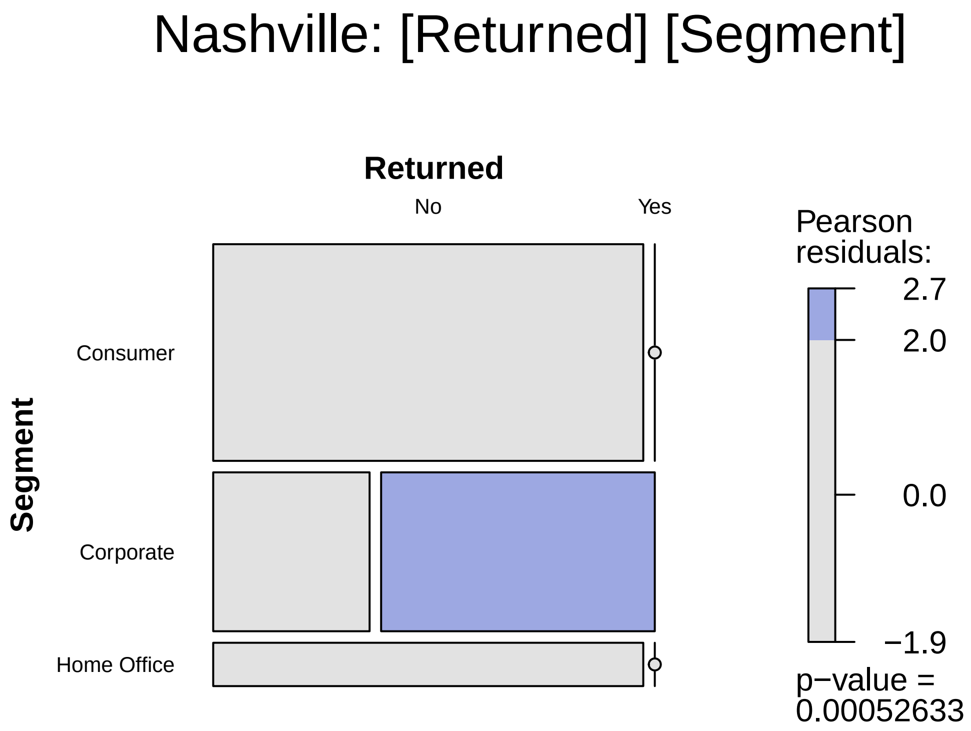


```
# City_v_returns

city_v_returns_TN <- xtabs(~Returned + City, data = tennessee_returns)

mosaic(
  t(city_v_returns_TN),
  gp = shading_hcl,
  main = "Tennessee: [Returned] [City]",
  labeling = labeling_border
    (
      varnames = c(TRUE, TRUE),
      offset_varnames = c(0, 0, 0, 4),
      rot_labels = c(0,0, 0, 0),
      offset_label = c(0.5,0,0, 0.5),
      just_labels = c("center","right"),
      gp_labels = gpar(fontsize = 6)
    )
)
```

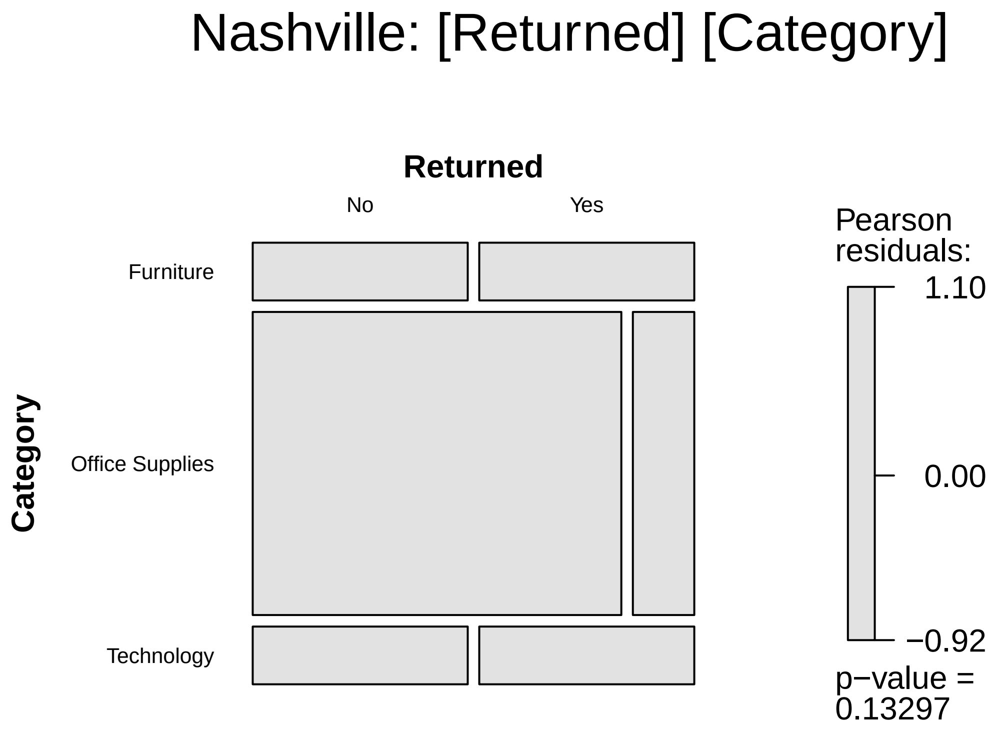


Mosiac plots within Nashville | South

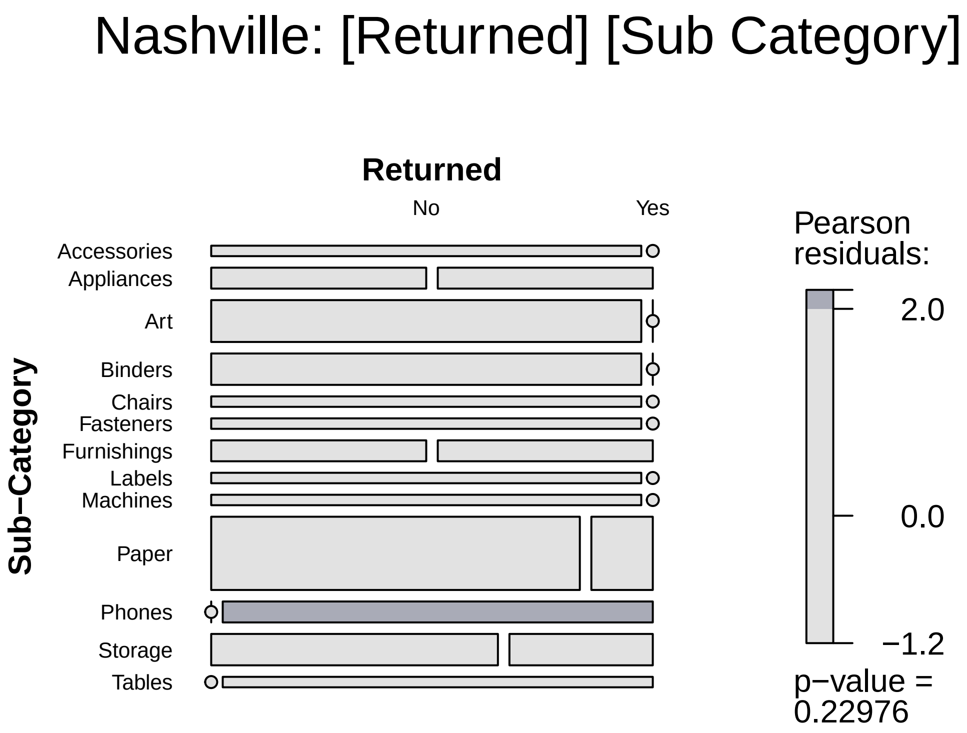
```
nashville_returns <- tennessee_returns %>%
  filter(City == "Nashville")

# Segment_v_returns

segment_v_returns_nashville <- xtabs(~Returned + Segment,
                                     data = nashville_returns)

mosaic(
  t(segment_v_returns_nashville),
  gp = shading_hcl,
  main = "Nashville: [Returned] [Segment]",
  labeling = labeling_border
    (
      varnames = c(TRUE, TRUE),
      offset_varnames = c(0, 0, 0, 3),
      rot_labels = c(0,0, 0, 0),
      offset_label = c(0.5,0,0, 0.5),
      just_labels = c("center","right"),
      gp_labels = gpar(fontsize = 8)
    )
)
```



```
# Category_v_returns

category_v_returns_nashville <- xtabs(~Returned + Category,
                                      data = nashville_returns)

mosaic(
  t(category_v_returns_nashville),
  gp = shading_hcl,
  main = "Nashville: [Returned] [Category]",
  labeling = labeling_border
    (
      varnames = c(TRUE, TRUE),
      offset_varnames = c(0, 0, 0, 4),
      rot_labels = c(0,0, 0, 0),
      offset_label = c(0.5,0,0, 0.5),
      just_labels = c("center","right"),
      gp_labels = gpar(fontsize = 8)
    )
)
```

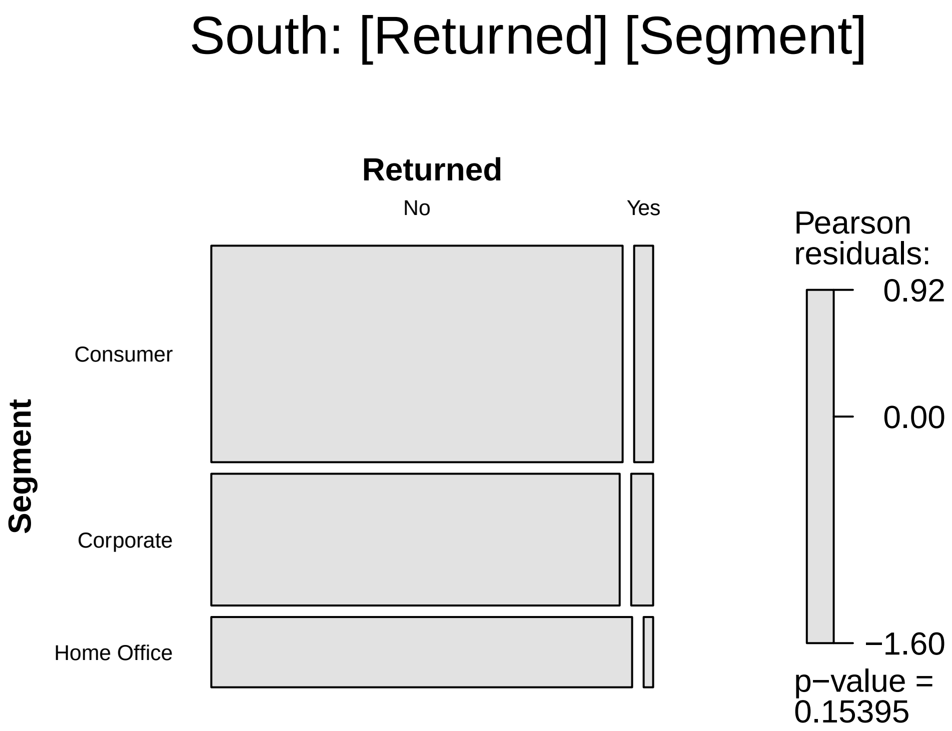


```r
# Sub_category_v_returns

sub_category_v_returns_nashville <- xtabs(~Returned + `Sub-Category`,
                                          data = nashville_returns)

mosaic(
  t(sub_category_v_returns_nashville),
  gp = shading_hcl,
  main = "Nashville: [Returned] [Sub Category]",
  labeling = labeling_border
    (
      varnames = c(TRUE, TRUE),
      offset_varnames = c(0, 0, 0, 3),
      rot_labels = c(0,0, 0, 0),
      offset_label = c(0.5,0,0, 0.5),
      just_labels = c("center","right"),
      gp_labels = gpar(fontsize = 8)
    )
)
```



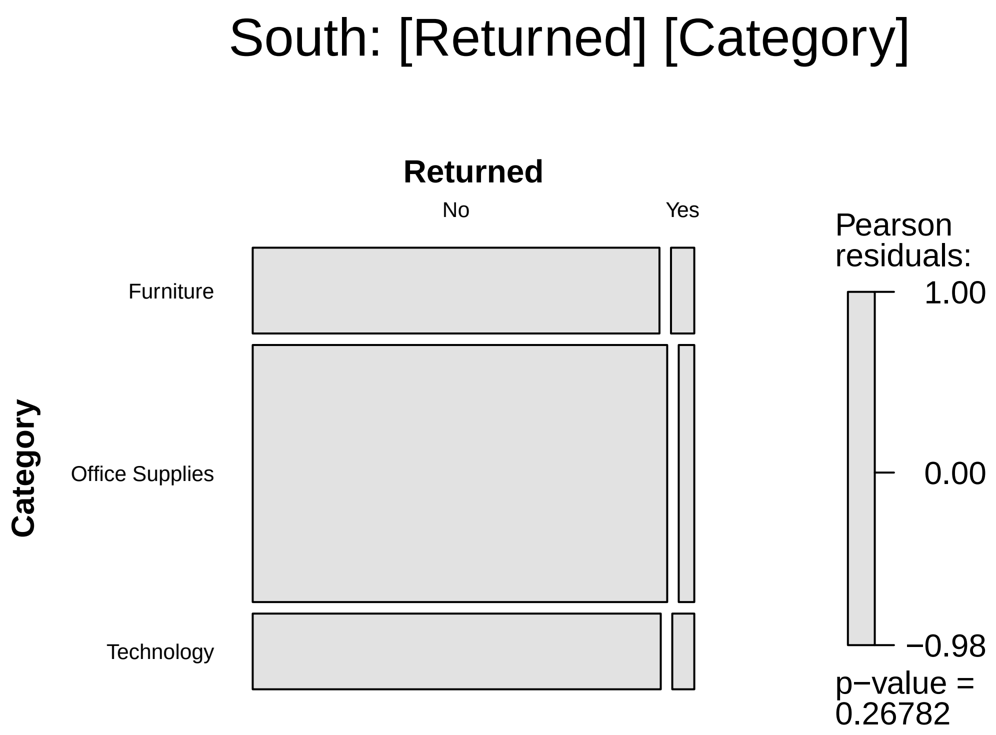
Regional Segments vs Returns mosiac | South

```
segment_v_returns_S <-  xtabs(~Returned + Segment, data = south_returns)
#S = South

mosaic(
  t(segment_v_returns_S),
  gp = shading_hcl,
  main = "South: [Returned] [Segment]",
  labeling = labeling_border
    (
      varnames = c(TRUE, TRUE),
      offset_varnames = c(0, 0, 0, 3),
      rot_labels = c(0,0, 0, 0),
      offset_label = c(0.5,0,0, 0.5),
      just_labels = c("center","right"),
      gp_labels = gpar(fontsize = 8)
    )
)
```



Regional Categories vs Returns mosiac | South

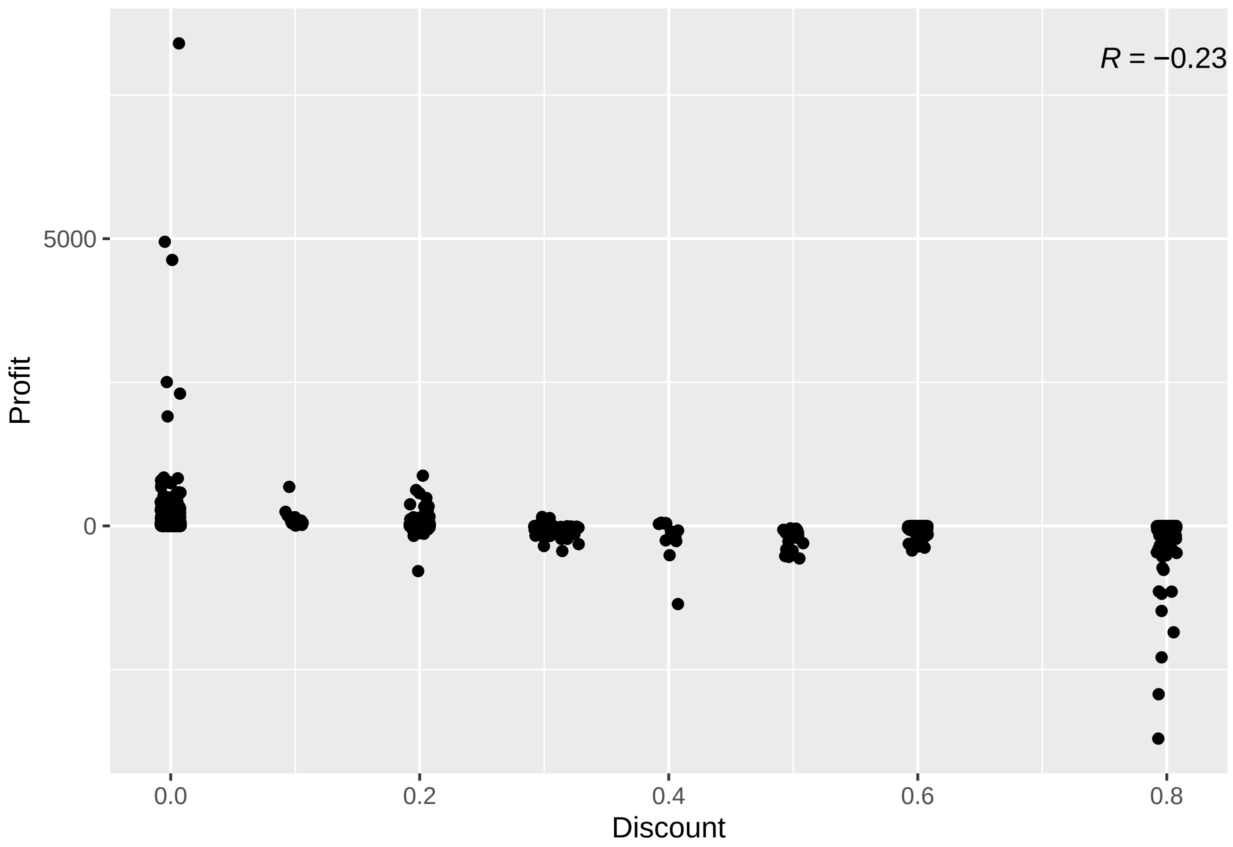
```
category_v_returns_S <-  xtabs(~Returned + Category, data = south_returns)
#S = South

mosaic(
  t(category_v_returns_S),
  gp = shading_hcl,
  main = "South: [Returned] [Category]",
  labeling = labeling_border
    (
      varnames = c(TRUE, TRUE),
      offset_varnames = c(0, 0, 0, 4),
      rot_labels = c(0,0, 0, 0),
      offset_label = c(0.5,0,0, 0.5),
      just_labels = c("center","right"),
      gp_labels = gpar(fontsize = 8)
    )
)
```


South: [Returned] [Category]

Central Analysis - Jason

Profit and discount rate point chart | Central

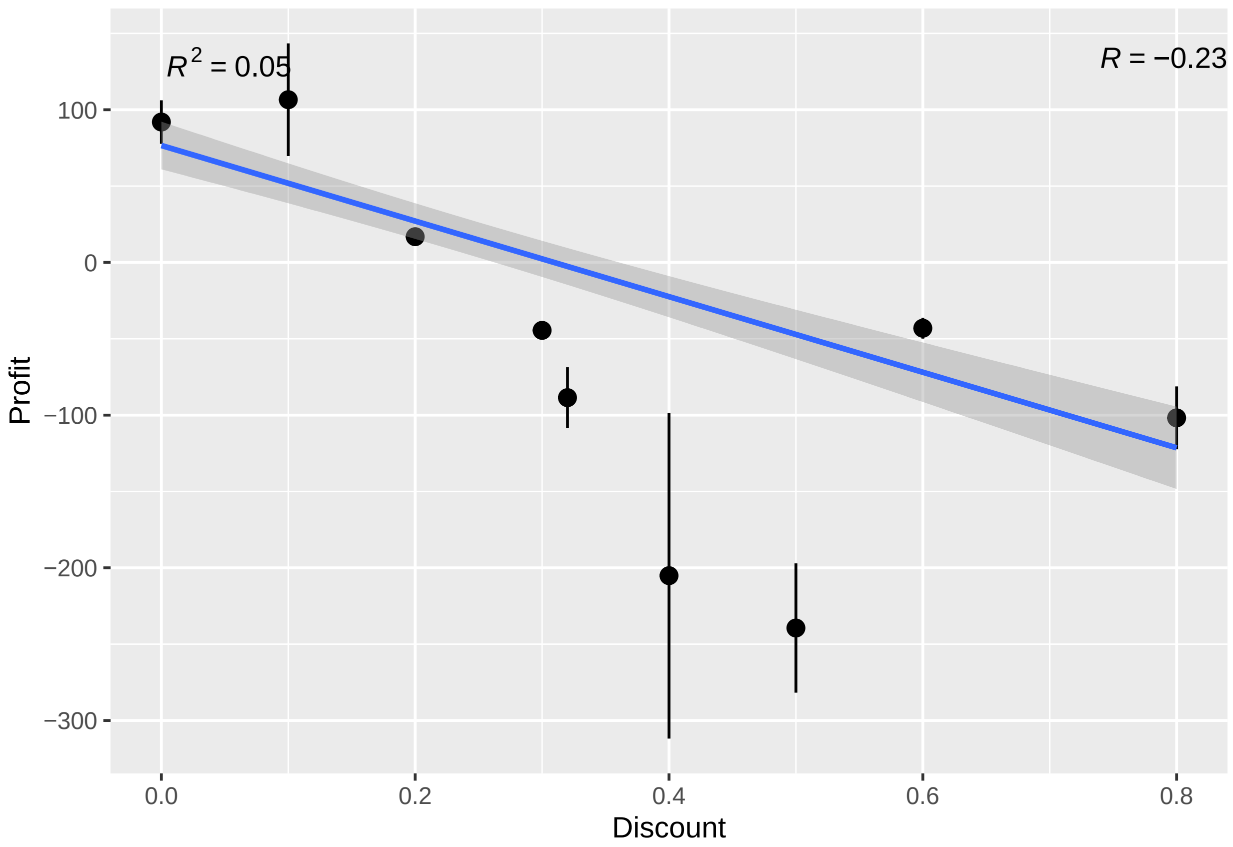
```
data_set_returns %>%
  select("Order Date", "Region", "Product ID",
         "Category", "Quantity", "Discount",
         "Profit") %>%
  filter(Region == "Central") %>%
  ggplot(aes(Discount, Profit)) +
  geom_jitter() +
  stat_correlation(label.x = 1)
```



Mean profit by discount rate | Central

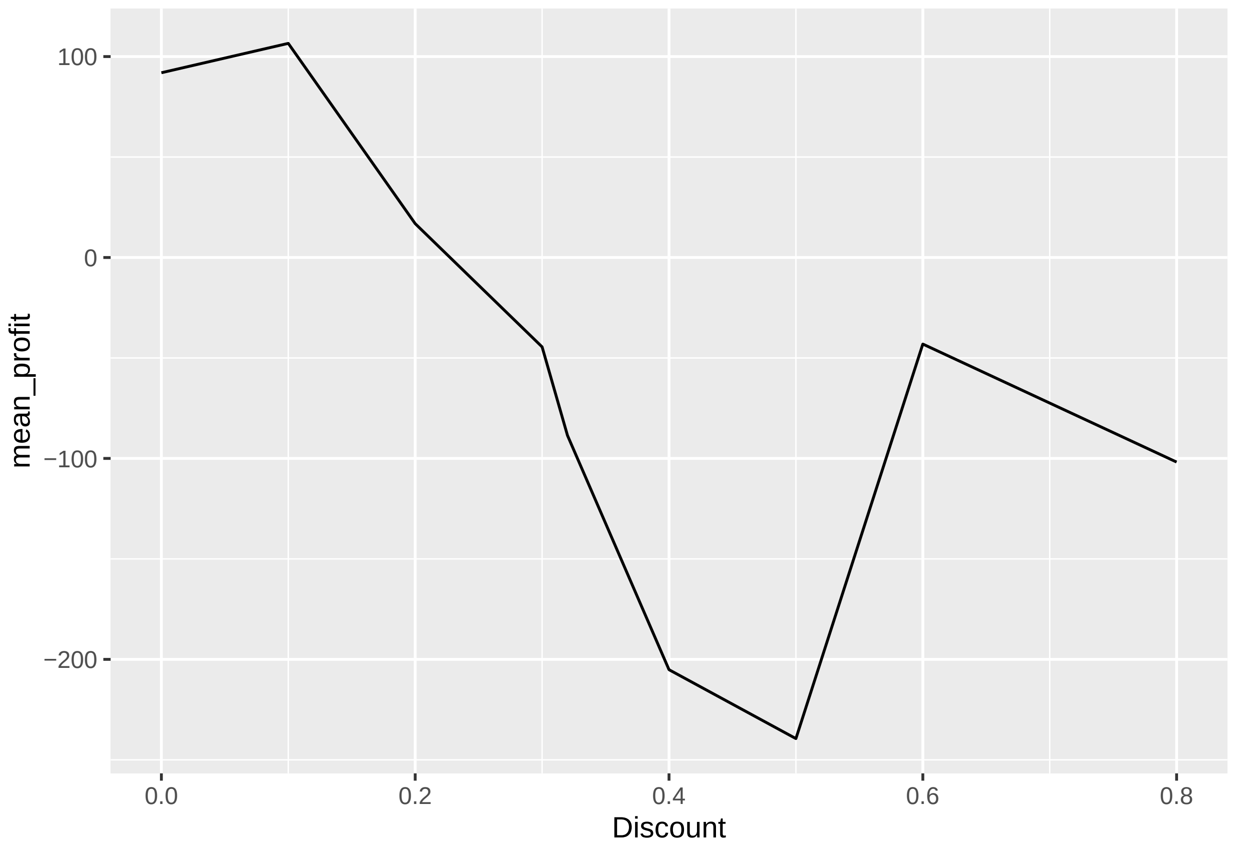
```
data_set_returns %>%
  select("Order Date", "Region", "Product ID",
         "Category", "Quantity", "Discount",
         "Profit") %>%
  filter(Region == "Central") %>%
  ggplot(aes(Discount, Profit)) +
  stat_summary() +
  stat_correlation(label.x = 1) +
  stat_poly_line() +
  stat_poly_eq()
```

```
## No summary function supplied, defaulting to ‘mean_se()‘
```



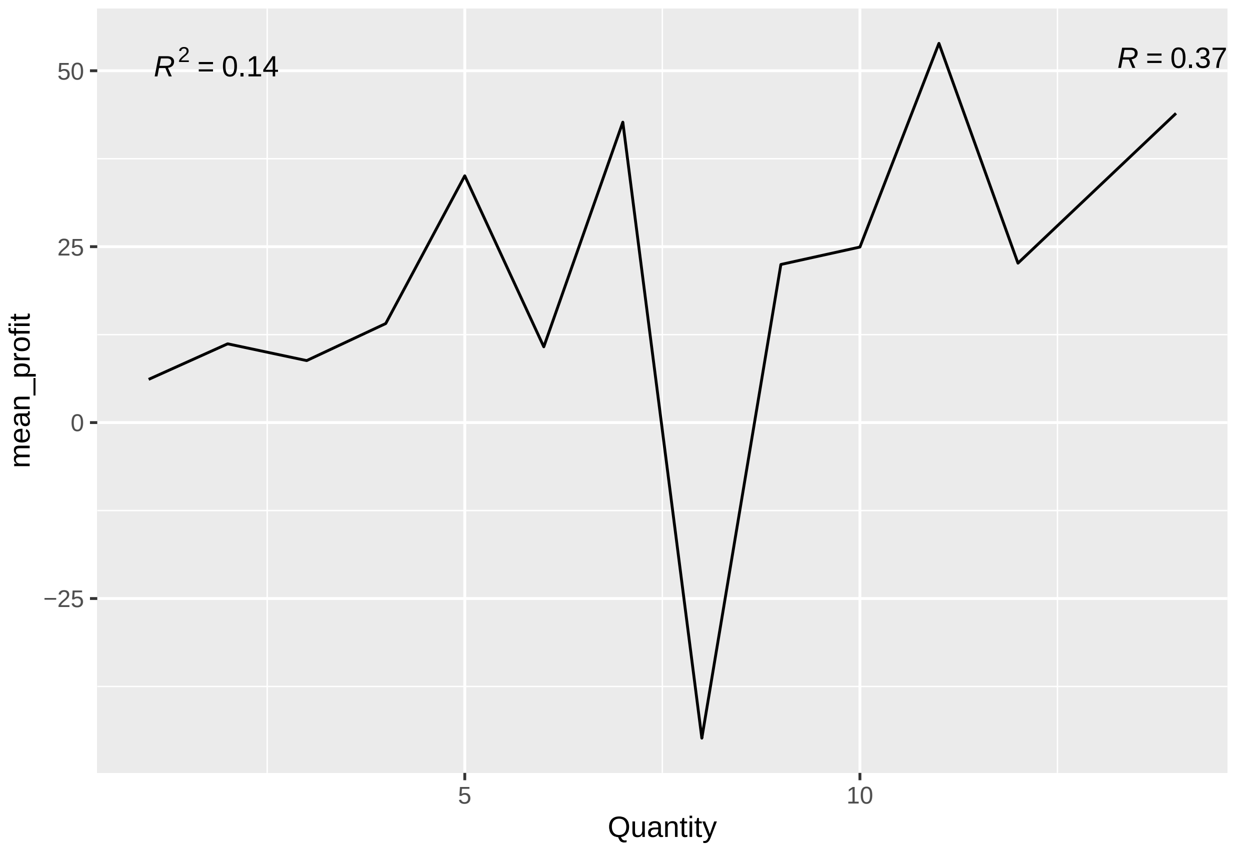
Mean profit by discount rate line chart | Central

```
data_set_returns %>%
  select("Region", "Discount", "Profit") %>%
  filter(Region == "Central") %>%
  group_by(Discount) %>%
  summarize(mean_profit = mean(Profit)) %>%
  ggplot() +
  geom_line(aes(Discount, mean_profit))
```



Profits by quantity customer bought | Central

```
data_set_returns %>%
  select("Order Date", "Region", "Product ID",
         "Category", "Quantity", "Discount",
         "Profit") %>%
  filter(Region == "Central") %>%
  group_by(Quantity) %>%
  summarize(mean_profit = mean(Profit)) %>%
  filter(Quantity != 13) %>%
  ggplot(aes(Quantity, mean_profit)) +
  geom_line() +
  stat_correlation(label.x = 1) +
  stat_poly_eq()
```



Chi squared tested based on returns based on shipping type | Central

```
central_data <- data_set_returns %>%
  select("Ship Mode", "Segment", "State",
         "Region", "Category", "Profit",
         "Returned") %>%
  filter(Region == "Central")

data_set_returns %>%
  select("Ship Mode", "Segment", "State",
         "Region", "Category", "Profit",
         "Returned") %>%
  filter(Region == "Central") %>%
  group_by('Ship Mode')
```

```
## # A tibble: 2,323 x 8
## # Groups:   "Ship Mode" [1]
##    `Ship Mode`    Segment   State Region Category  Profit Returned `"Ship Mode"`
##    <chr>          <chr>     <chr> <chr>  <chr>      <dbl> <chr>    <chr>
##  1 Standard Class Home Off~ Texas Centr~ Office ~ -124.   No       Ship Mode
##  2 Standard Class Home Off~ Texas Centr~ Office ~   -3.82 No       Ship Mode
##  3 Standard Class Consumer  Wisc~ Centr~ Office ~   13.3  No       Ship Mode
##  4 Standard Class Corporate Nebr~ Centr~ Office ~    5.06 No       Ship Mode
##  5 Standard Class Corporate Nebr~ Centr~ Office ~   15.7  No       Ship Mode
##  6 Second Class   Home Off~ Texas Centr~ Office ~    9.95 No       Ship Mode
##  7 First Class    Corporate Texas Centr~ Technol~  123.   No       Ship Mode
##  8 First Class    Corporate Texas Centr~ Furnitu~ -148.   No       Ship Mode
##  9 Standard Class Home Off~ Texas Centr~ Office ~   35.4  No       Ship Mode
## 10 Standard Class Home Off~ Texas Centr~ Furnitu~  -47.0  No       Ship Mode
## # i 2,313 more rows
```

```r
cont_table = table(data_set_returns$Returned, data_set_returns$`Ship Mode`)

cont_table
```

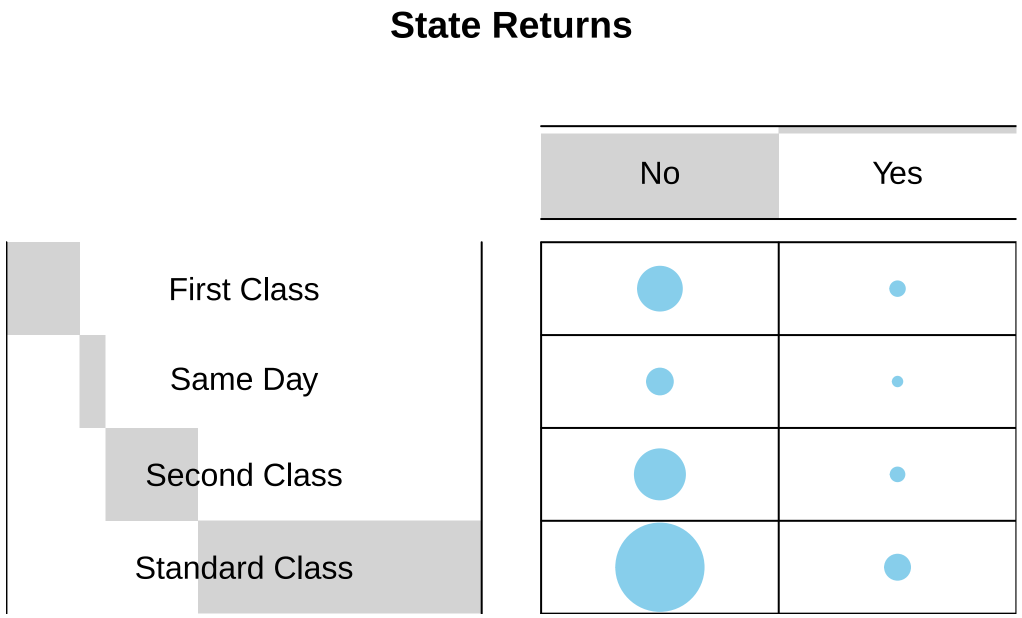
```
##
##       First Class Same Day Second Class Standard Class
##   No         1386      479         1811           5518
##   Yes         152       64          134            450
```

```r
chi_squared_result = chisq.test(cont_table)

chi_squared_result
```

```
##
##  Pearson’s Chi-squared test
##
## data:  cont_table
## X-squared = 22.947, df = 3, p-value = 4.143e-05
```

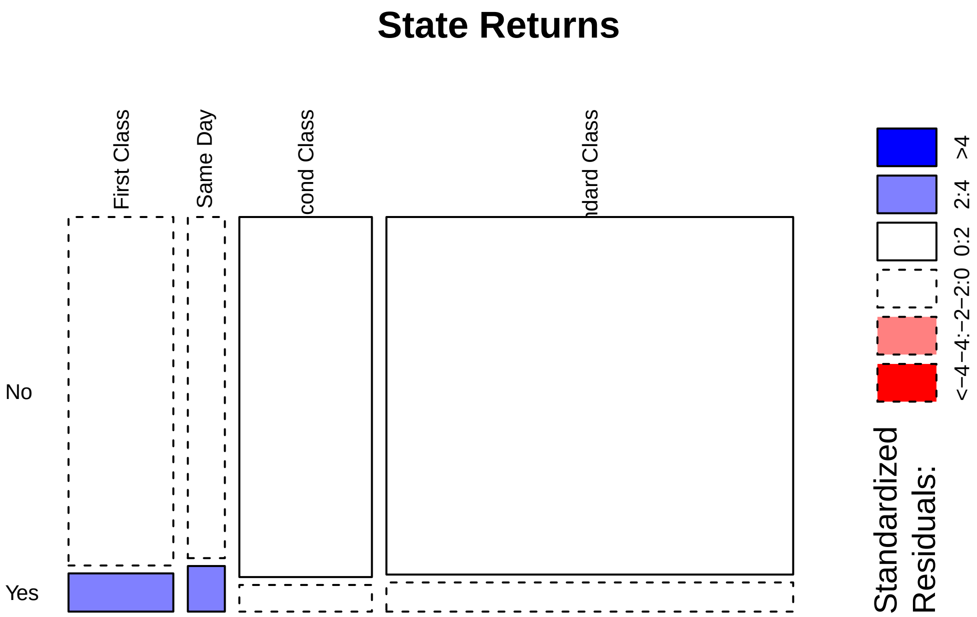
Baloon plot of chi squared test results | Central

```r
balloonplot(cont_table, main = "State Returns", xlab ="", ylab="",
            label = FALSE, show.margins = FALSE)
```



Mosiaic plot of chi squared test results | Central

```r
mosaicplot(t(cont_table), shade = TRUE, las = 2, main = "State Returns")
```



Chi squared test for same day shipping compaired to returns | Central

```
sameDay_returns <- data_set_returns %>%
  select("Ship Mode", "Segment", "State",
         "Region", "Category", "Profit",
         "Returned") %>%
  filter(Region == "Central") %>%
  filter(`Ship Mode` == "Same Day") %>%
  group_by(State)

state_return_table = table(sameDay_returns$Returned, sameDay_returns$State)

state_return_table
```

```
##
##       Illinois Indiana Iowa Kansas Michigan Minnesota Missouri Nebraska
##   No        26       3    1      1       16         4        2        2
##   Yes        0       0    0      0        2         0        0        1
##
##       Oklahoma Texas Wisconsin
##   No         6    44         3
##   Yes        1     4         4
```

```
same_day_chi_test = chisq.test(state_return_table)
```

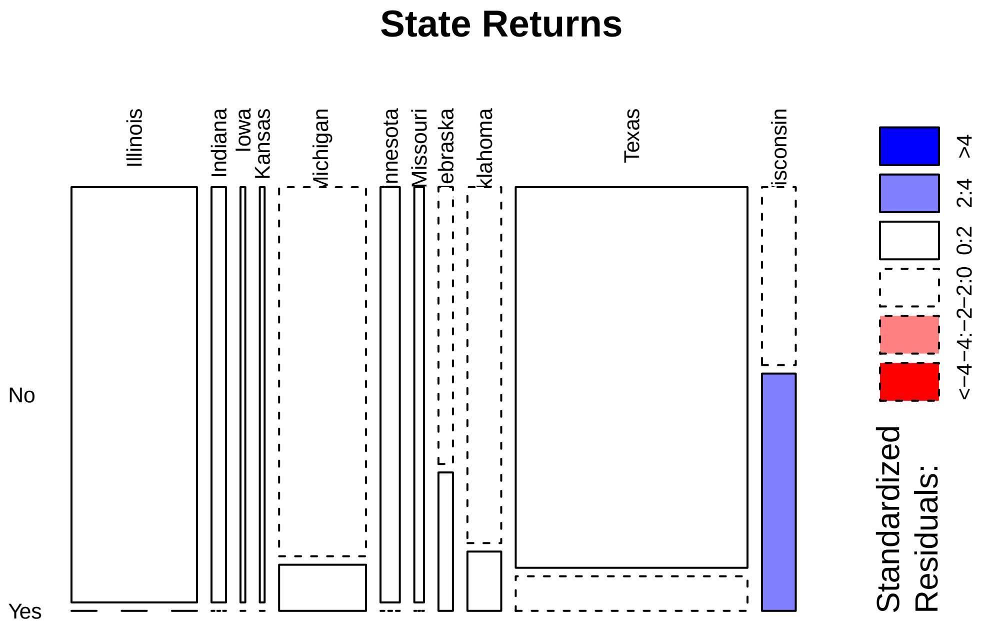
```
## Warning in chisq.test(state_return_table): Chi-squared approximation may be
## incorrect
```

```
same_day_chi_test
```

```
##
##  Pearson's Chi-squared test
##
## data:  state_return_table
## X-squared = 23.527, df = 10, p-value = 0.008959
```

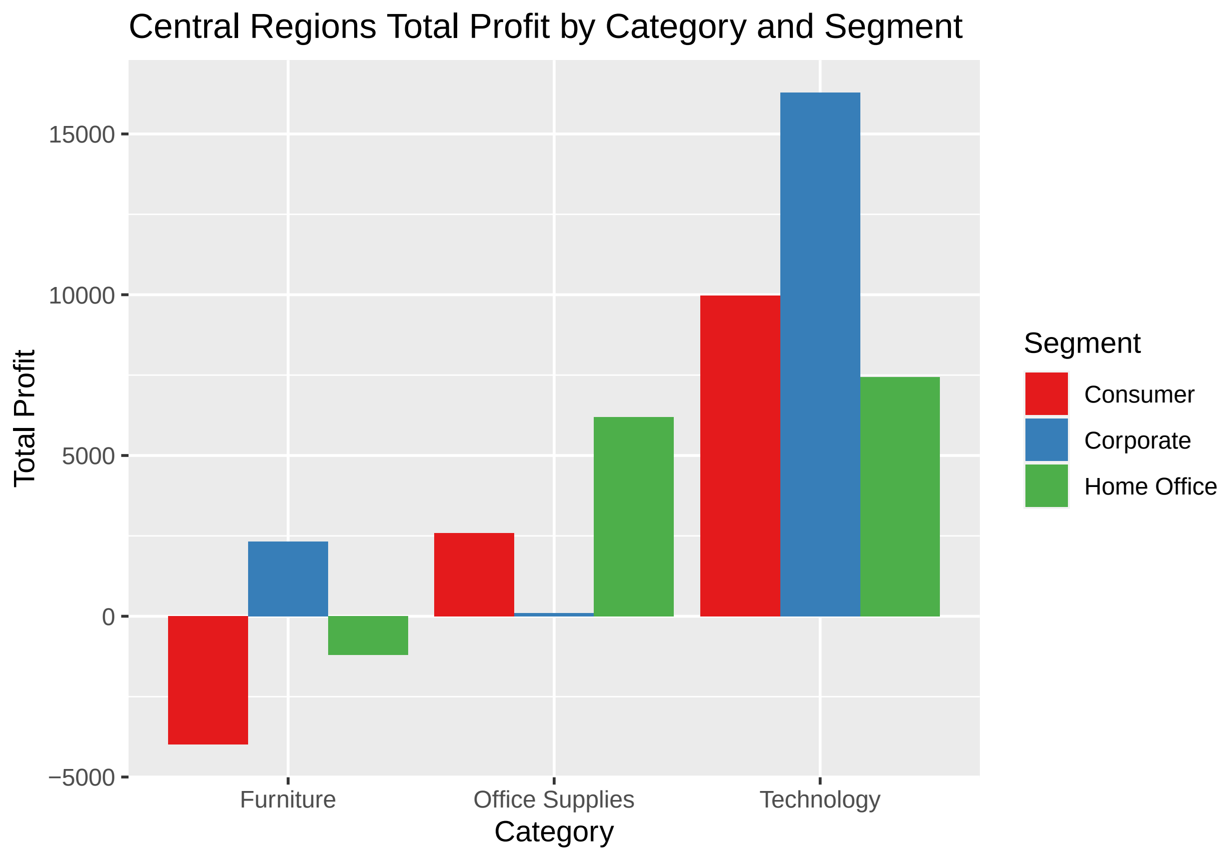
Mosiac plot of chi squared results | Central

```
mosaicplot(t(state_return_table), shade = TRUE,
           las = 2, main = "State Returns")
```



Total profit bar chart by category and segment | Central

```r
brewer_pal <- brewer.pal(n = 3, name = "Set1")

data_set_returns %>%
  filter(Region == "Central") %>%
  select(Profit, Segment, Category) %>%
  group_by(Segment, Category) %>%
  mutate(`total_profit` = sum(Profit)) %>%

  ggplot(aes(x= Category, y = total_profit, fill = Segment)) +
  geom_bar(position = "dodge", stat = "identity") +
  labs(
    x= "Category",
    y= "Total Profit",
    title = "Central Regions Total Profit by Category and Segment",
    subtile = "From orders placed 2019 to 2022"
  ) +
  scale_fill_manual(values = brewer_pal)
```



Mosaic plot of central states returned residuals | Central

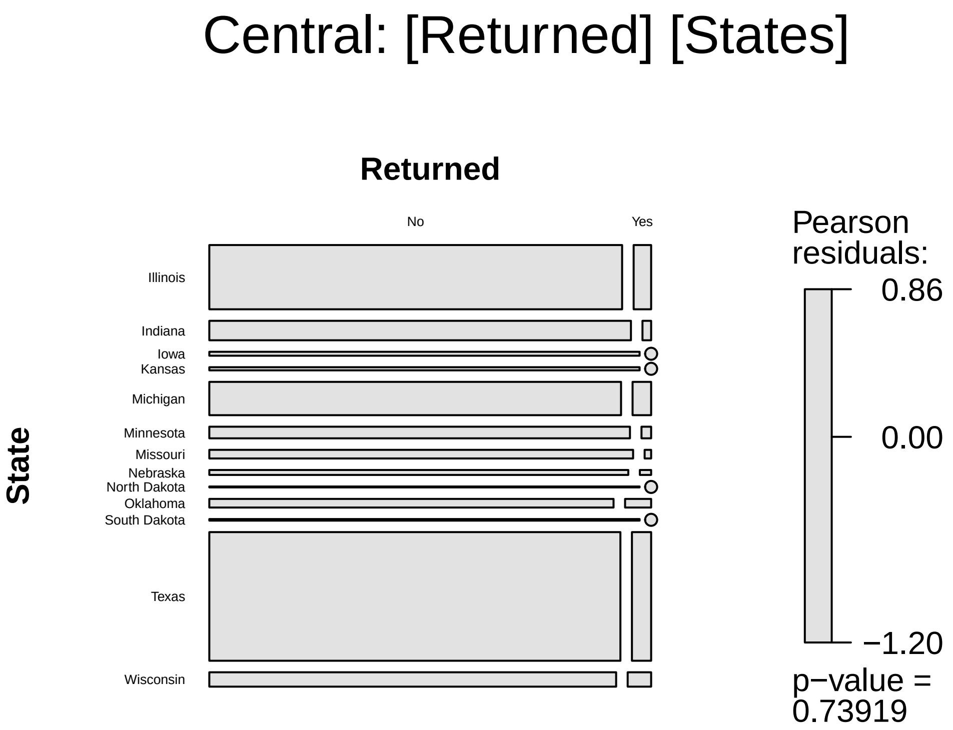
```r
state_returns <- data_set_returns %>%
  select("Ship Mode", "Segment", "State",
         "Region", "Category", "Profit",
         "Returned") %>%
  filter(Region == "Central") %>%
  group_by(State)

texas_returns <- data_set_returns %>%
  select("Ship Mode", "Segment", "State",
         "Region", "Category", "Profit",
         "Returned", "City") %>%
  filter(Region == "Central") %>%
  filter(State == "Texas") %>%
  filter(City == "Austin" | City == "Houston" |
           City == "Dallas" | City == "San Antonio" |
           City == "El Paso")

texas_returns
```

```
## # A tibble: 651 x 8
##    ‘Ship Mode‘    Segment     State Region  Category       Profit Returned City
##    <chr>          <chr>       <chr> <chr>   <chr>           <dbl> <chr>    <chr>
##  1 Second Class   Home Office Texas Central Office Suppli~   9.95 No       Hous~
##  2 Standard Class Home Office Texas Central Office Suppli~  35.4  No       Hous~
##  3 Standard Class Home Office Texas Central Furniture      -47.0  No       Hous~
##  4 Standard Class Home Office Texas Central Furniture      -15.1  No       Hous~
##  5 Standard Class Home Office Texas Central Technology      41.8  No       Hous~
##  6 First Class    Corporate   Texas Central Office Suppli~  -1.93 No       Hous~
##  7 First Class    Corporate   Texas Central Furniture       -5.82 No       Hous~
##  8 First Class    Corporate   Texas Central Office Suppli~   2.72 No       Hous~
##  9 Second Class   Consumer    Texas Central Furniture      -14.5  No       Hous~
## 10 Second Class   Home Office Texas Central Office Suppli~  13.9  No       Hous~
## # i 641 more rows
```

```
# Jason's creatively named variables
poopyerbutt = xtabs(~Returned + City, data = texas_returns)

poopybutt <- xtabs(~Returned + State, data = state_returns)


poopybutt
```

```
##          State
## Returned Illinois Indiana Iowa Kansas Michigan Minnesota Missouri Nebraska
##       No      472     146   30     24      244        87       65       37
##       Yes      20       3    0      0       11         2        1        1
##          State
## Returned North Dakota Oklahoma South Dakota Texas Wisconsin
##       No             7       62           12   941       104
##       Yes            0        4            0    44         6
```

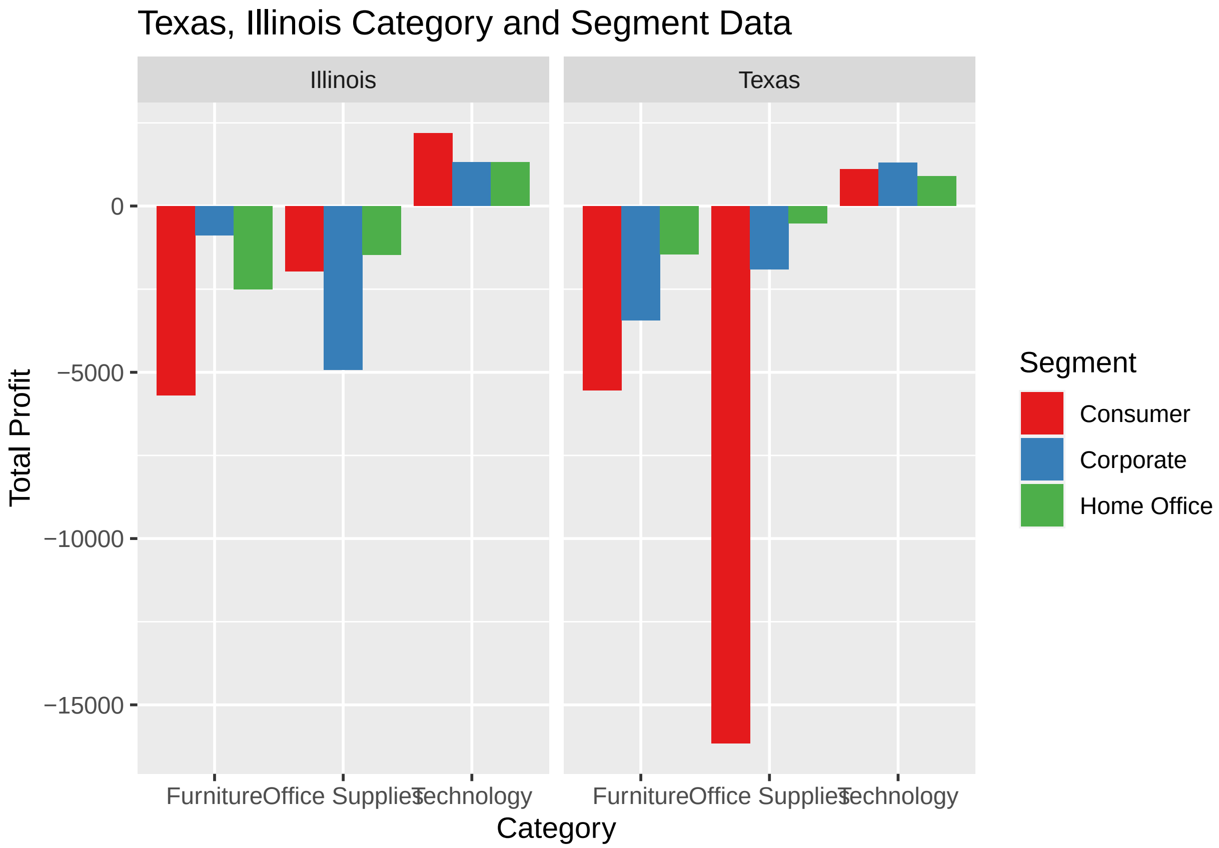
```
mosaic(
  t(poopybutt),
  shade = TRUE,
  main = "Central: [Returned] [States]",
  labeling = labeling_border
    (
      varnames = c(TRUE, TRUE),
      offset_varnames = c(0, 0, 0, 3),
      rot_labels = c(0,0, 0, 0),
      offset_label = c(0.5,0,0, 0.5),
      just_labels = c("center","right"),
      gp_labels = gpar(fontsize = 5),
      spacing = 4
    )
)
```



Bar chart of texas and illinos total profit by segment and category | Central

```
data_set_returns %>%
  select("Ship Mode", "Segment", "State",
         "Region", "Category", "Profit",
         "City") %>%
  filter(Region == "Central") %>%
  filter(State == 'Texas' | State == 'Illinois') %>%
  group_by(State, Category, Segment) %>%
  summarize("Total Profit" = sum(Profit)) %>%
  ggplot(aes(Category, `Total Profit`, fill = Segment)) +
  geom_bar(position = "dodge", stat = "identity") +
  facet_wrap(~ State) +
  labs(
    y= "Total Profit",
    title = "Texas, Illinois Category and Segment Data"
  ) +
   scale_fill_manual(values = brewer_pal)
```

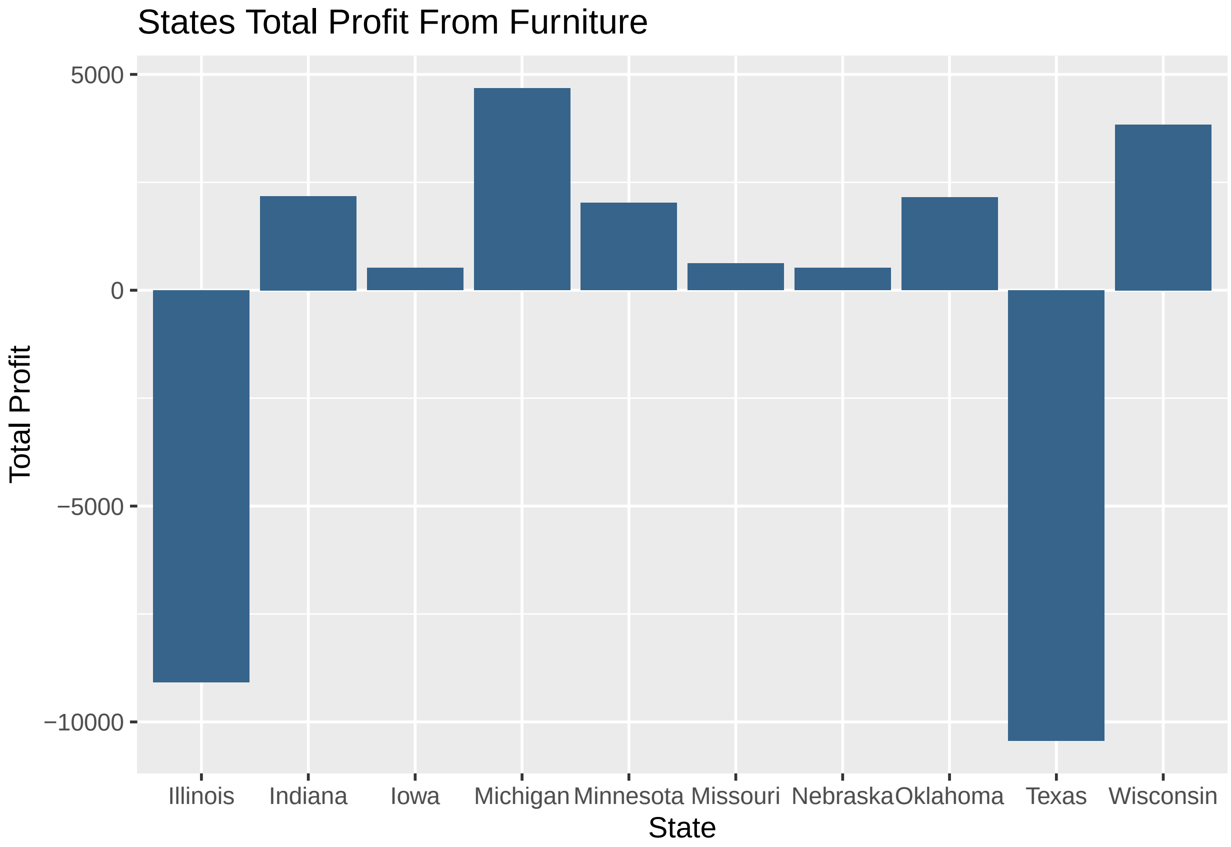
```
## ‘summarise()‘ has grouped output by ’State’, ’Category’. You can override using
## the ‘.groups‘ argument.
```



Bar chart of central states total profits in furniture category | Central

```
brewer_pal <- brewer.pal(n = 3, name = "Set1")

data_set_returns %>%
  select("Ship Mode", "Segment", "State",
         "Region", "Category", "Profit",
         "City") %>%
  filter(Region == "Central") %>%
  filter(Category == "Furniture") %>%
  filter(State != "Kansas") %>%
  filter(State != "South Dakota") %>%
  group_by(State, Category) %>%
  summarize("Total Profit" = sum(Profit)) %>%

  ggplot(aes(`State`, `Total Profit`)) +
  geom_bar(position = "dodge", stat = "identity", fill = "steelblue4") +
  stat_correlation(label.x = 1) + stat_poly_eq() +
  labs(
    x= "State",
    y= "Total Profit",
    title = "States Total Profit From Furniture",
    subtile = "From orders placed 2019 to 2022"
  )
```

```
## ‘summarise()‘ has grouped output by ’State’. You can override using the
## ‘.groups‘ argument.
```



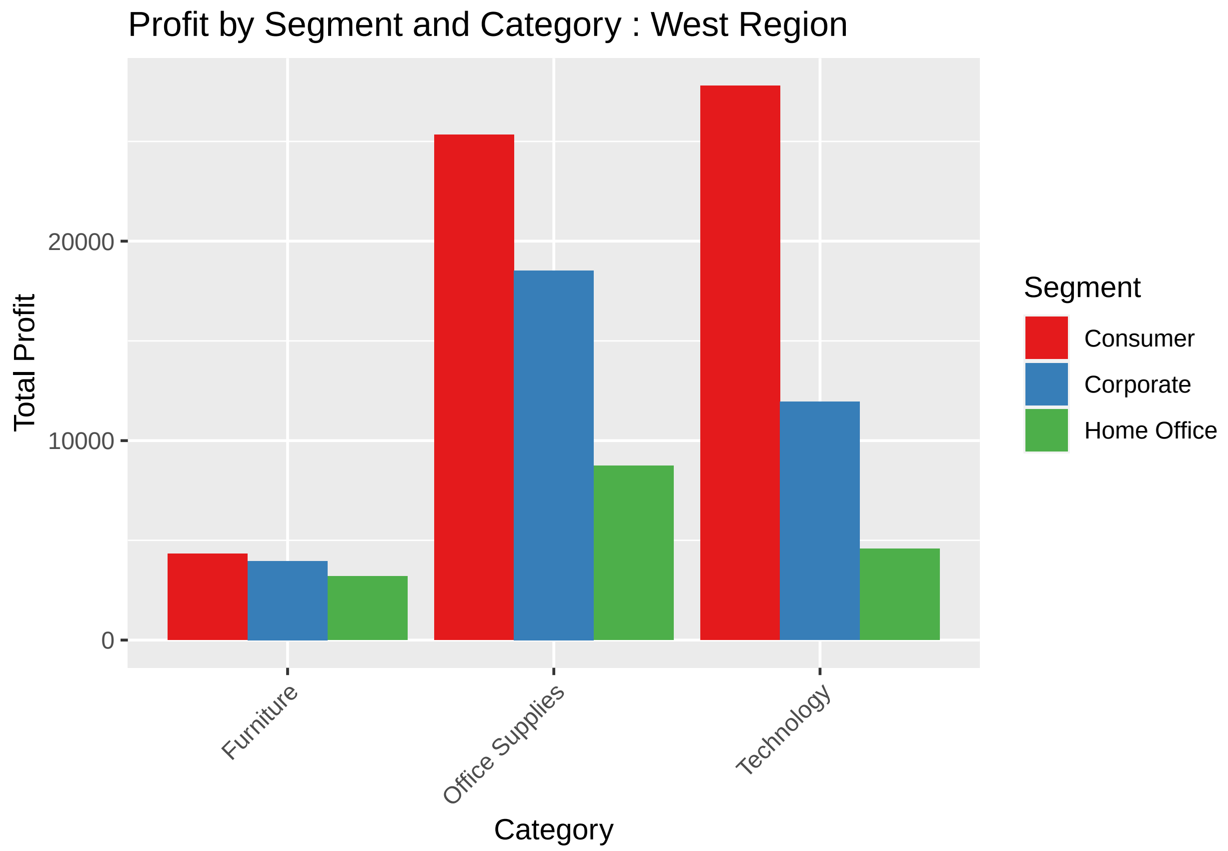
Western Analysis - Nate

Profit by Segment and Category bar chart | West

```
# Aggregate data by Category, Segment, and calculate total profit
profit_by_category_segment <- data_set_returns %>%
  filter(Region == "West") %>%
  group_by(Category, Segment) %>%
  summarise(Total_Profit = sum(Profit))
```

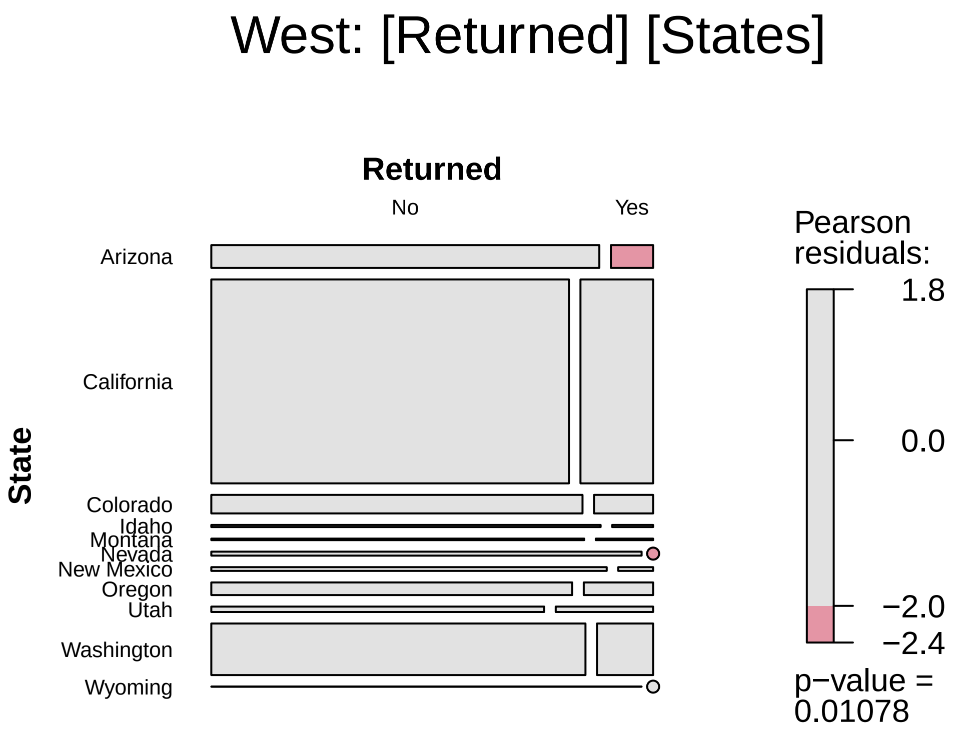
```
## ‘summarise()‘ has grouped output by ’Category’. You can override using the
## ‘.groups‘ argument.
```

```
# Plotting the bar chart
ggplot(profit_by_category_segment, aes(x = Category, y = Total_Profit,
                                       fill = Segment)) +
  geom_col(position = "dodge") +
  labs(title = "Profit by Segment and Category : West Region",
       x = "Category",
       y = "Total Profit",
       fill = "Segment") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_fill_manual(values = brewer_palette)
```



States returned mosiaic plot | West

```
west_returns <- data_set_returns %>%
  filter(Region == "West")

state_v_returns_S <-  xtabs(~Returned + State, data = west_returns)

mosaic(
  t(state_v_returns_S),
  gp = shading_hcl,
  main = "West: [Returned] [States]",
  labeling = labeling_border
    (
      varnames = c(TRUE, TRUE),
      offset_varnames = c(0, 0, 0, 3),
      rot_labels = c(0,0, 0, 0),
      offset_label = c(0.5,0,0, 0.5),
      just_labels = c("center","right"),
      gp_labels = gpar(fontsize = 8)
    )
)
```



Eastern Analysis - Jacob

Making data for eastern analysis | East

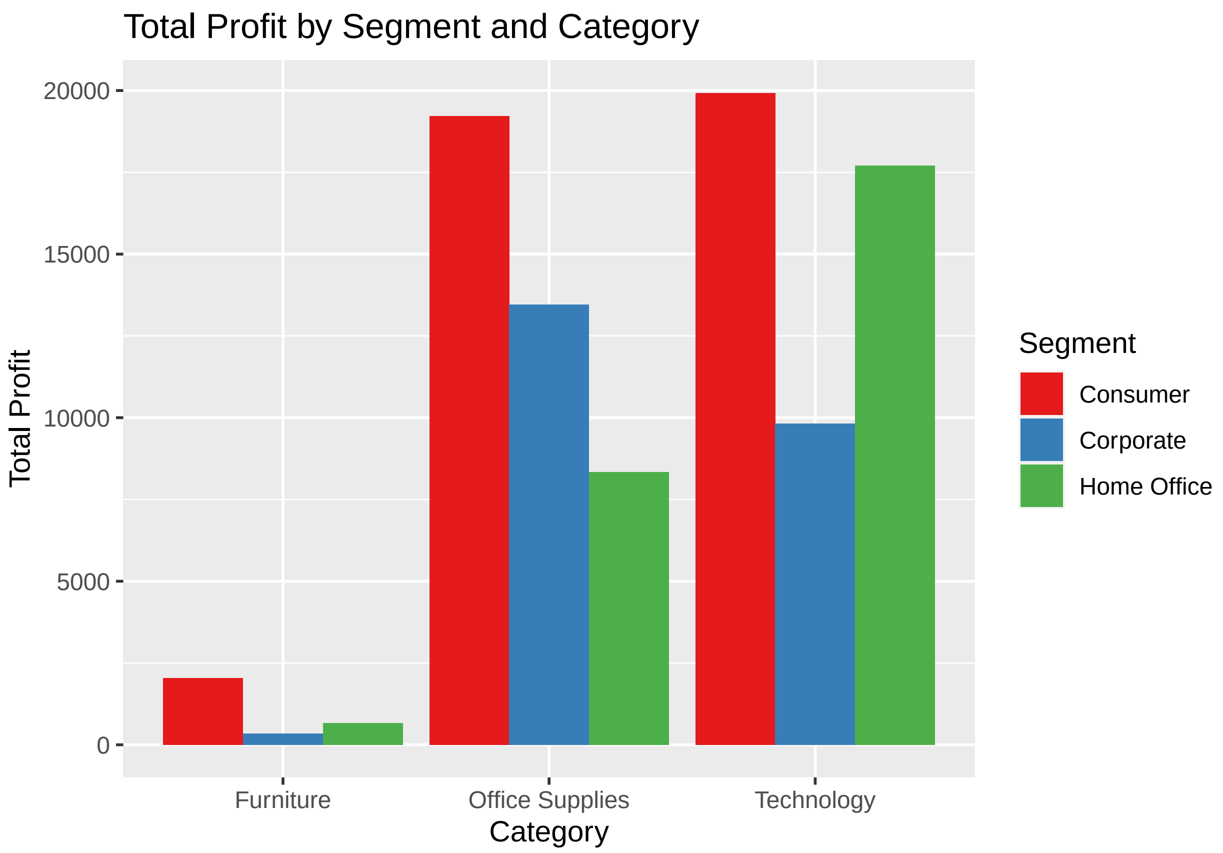
```
east_data <- data_set_returns %>%
  filter(Region == "East")
```

Profit by segment and category bar chart | East

```
brewer_palette <- brewer.pal(n=3, name = "Set1")

total_profit <- east_data %>%
  group_by(Segment, Category) %>%
  summarize(total_profit = sum(Profit))
```

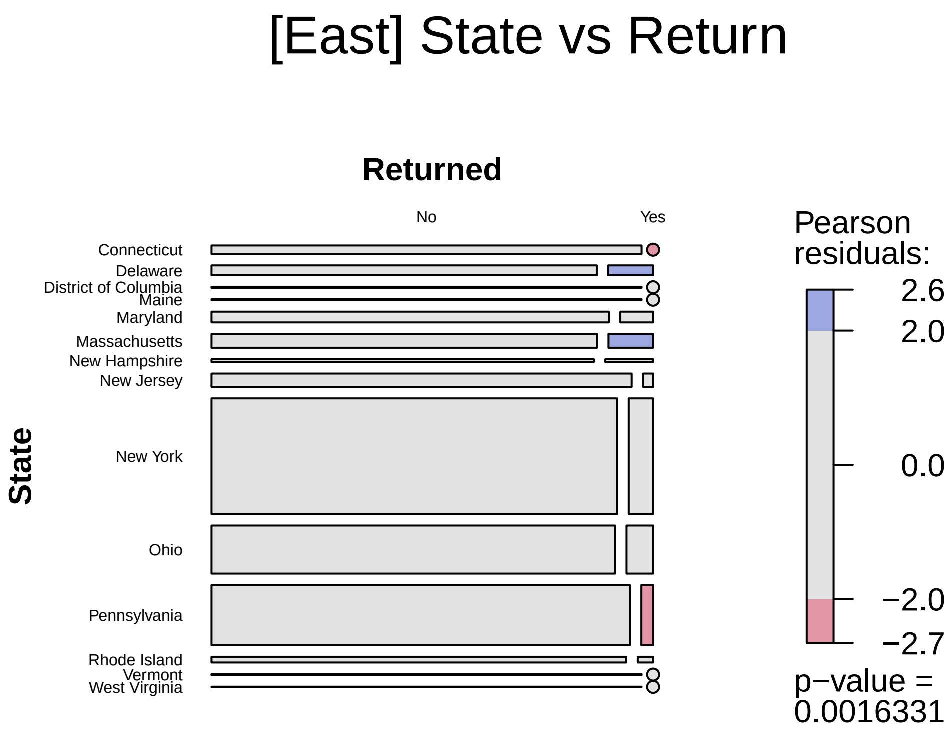
```
## ‘summarise()‘ has grouped output by ’Segment’. You can override using the
## ‘.groups‘ argument.
```

```
ggplot(total_profit, aes(x = Category, y = total_profit, fill = Segment)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Total Profit by Segment and Category",
       x = "Category",
       y = "Total Profit",
      fill = "Segment")+
      scale_fill_manual(values = brewer_palette)
```



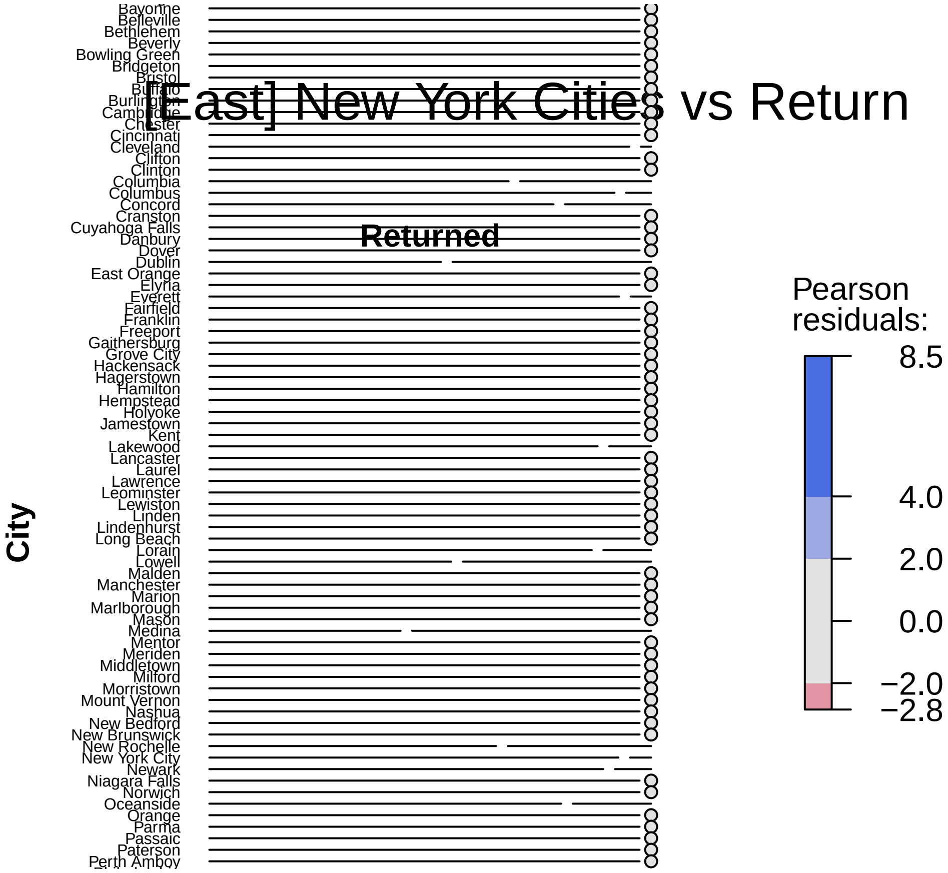
Returns by State | East

```
state_v_return_table <-  xtabs(~Returned + State, data = east_data)

mosaic(
  t(state_v_return_table),
  gp= shading_hcl,
  main = "[East] State vs Return",
  labeling = labeling_border
    (
      varnames = c(TRUE, TRUE),
      offset_varnames = c(0, 0, 0, 3),
      rot_labels = c(0,0, 0, 0),
      offset_label = c(0.5,0,0, 0.5),
      just_labels = c("center","right"),
      gp_labels = gpar(fontsize = 6)
    )
)
```



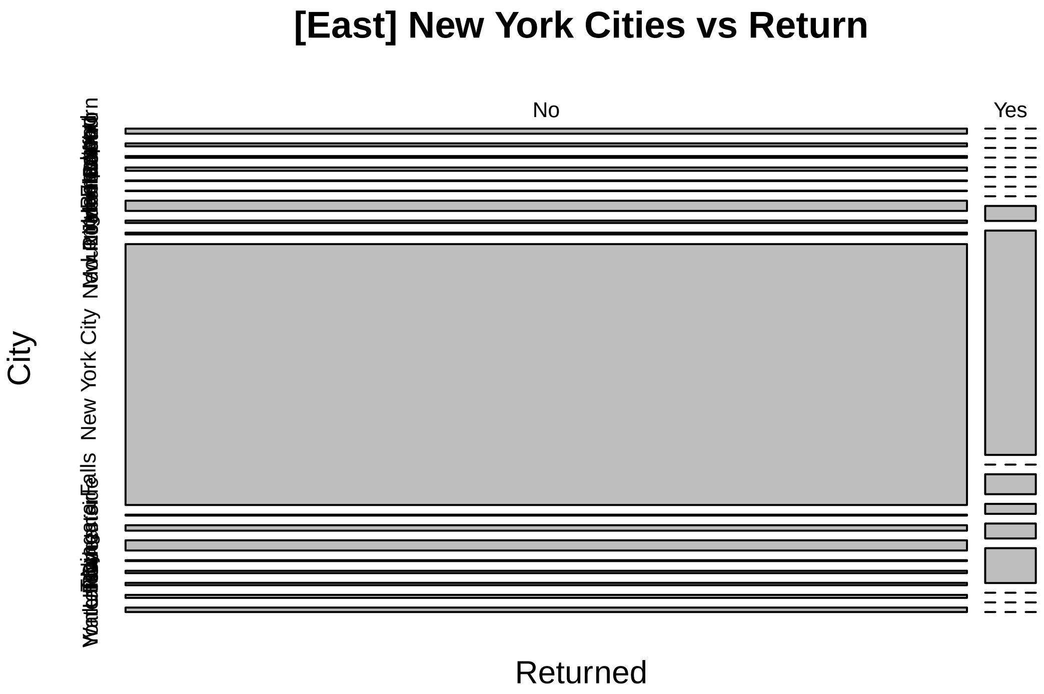
Attempted New York chi squared test and mosiac plot | East

```
new_york_data_set_returns <- east_data %>% filter(east_data$State=="New York")
ny_v_return_table <-  xtabs(~Returned + City, data = east_data)

mosaic(
  t(ny_v_return_table),
  gp= shading_hcl(t(ny_v_return_table),p.value=NA),
  main = "[East] New York Cities vs Return",
  labeling = labeling_border
    (
      varnames = c(TRUE, TRUE),
      offset_varnames = c(0, 0, 0, 3),
      rot_labels = c(0,0, 0, 0),
      offset_label = c(0.5,0,0, 0.5),
      just_labels = c("center","right"),
      gp_labels = gpar(fontsize = 6)
    )
)
```

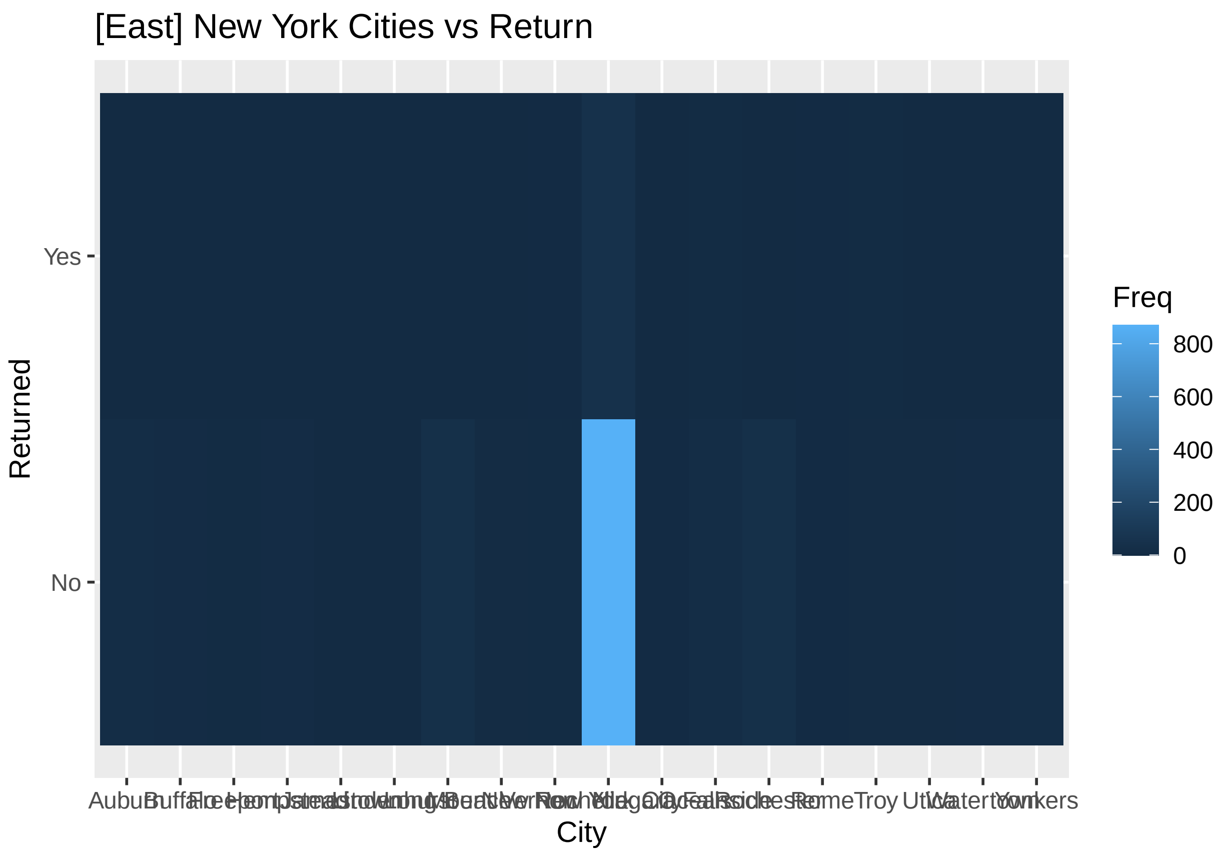


More New York data | East

```r
# Filter data for New York state
new_york_data_set_returns <- east_data %>% filter(State == "New York")

# Create a contingency table
ny_v_return_table <- xtabs(~ Returned + City, data = new_york_data_set_returns)

# Create a mosaic plot
mosaicplot(ny_v_return_table, main = "[East] New York Cities vs Return")
```



```
# Convert contingency table to data frame
ny_v_return_data_set_returns <- as.data.frame.table(ny_v_return_table)

# Create ggplot
ggplot(ny_v_return_data_set_returns, aes(x = City, y = Returned, fill = Freq)) +
  geom_tile() +
  labs(title = "[East] New York Cities vs Return",
       x = "City",
       y = "Returned")
```



Machine learning

Target Guided Ordinal Encoding for Segement, Category, Returned

Segment

```
# Segment Encoding

# Finding the mean of the target variable (Profit) for each segment
segment_mean_profit <- data_set_returns %>%
  group_by(Segment) %>%
  summarize(mean_profit = mean(Profit))

# Ranking the segments by mean profit
segment_mean_profit <- segment_mean_profit %>%
  mutate(segment_rank = rank(mean_profit))

# Encoding the variables
data_set_returns <- data_set_returns %>%
  left_join(segment_mean_profit, by = "Segment") %>%
  select(-mean_profit)
```

Category

```r
# Category Encoding

# Finding the mean of the target variable (Profit) for each segment
category_mean_profit <- data_set_returns %>%
  group_by(Category) %>%
  summarize(mean_profit = mean(Profit))

# Ranking the segments by mean profit
category_mean_profit <- category_mean_profit %>%
  mutate(category_rank = rank(mean_profit))

# Encoding the variables
data_set_returns <- data_set_returns %>%
  left_join(category_mean_profit, by = "Category") %>%
  select(-mean_profit)
```

Returned

```r
# Returned Encoding

# Finding the mean of the target variable (Profit) for each segment
returned_mean_profit <- data_set_returns %>%
  group_by(Returned) %>%
  summarize(mean_profit = mean(Profit))

# Ranking the segments by mean profit
returned_mean_profit <- returned_mean_profit %>%
  mutate(returned_rank = rank(mean_profit))

# Encoding the variables
data_set_returns <- data_set_returns %>%
  left_join(returned_mean_profit, by = "Returned") %>%
  select(-mean_profit)

view(data_set_returns)
```

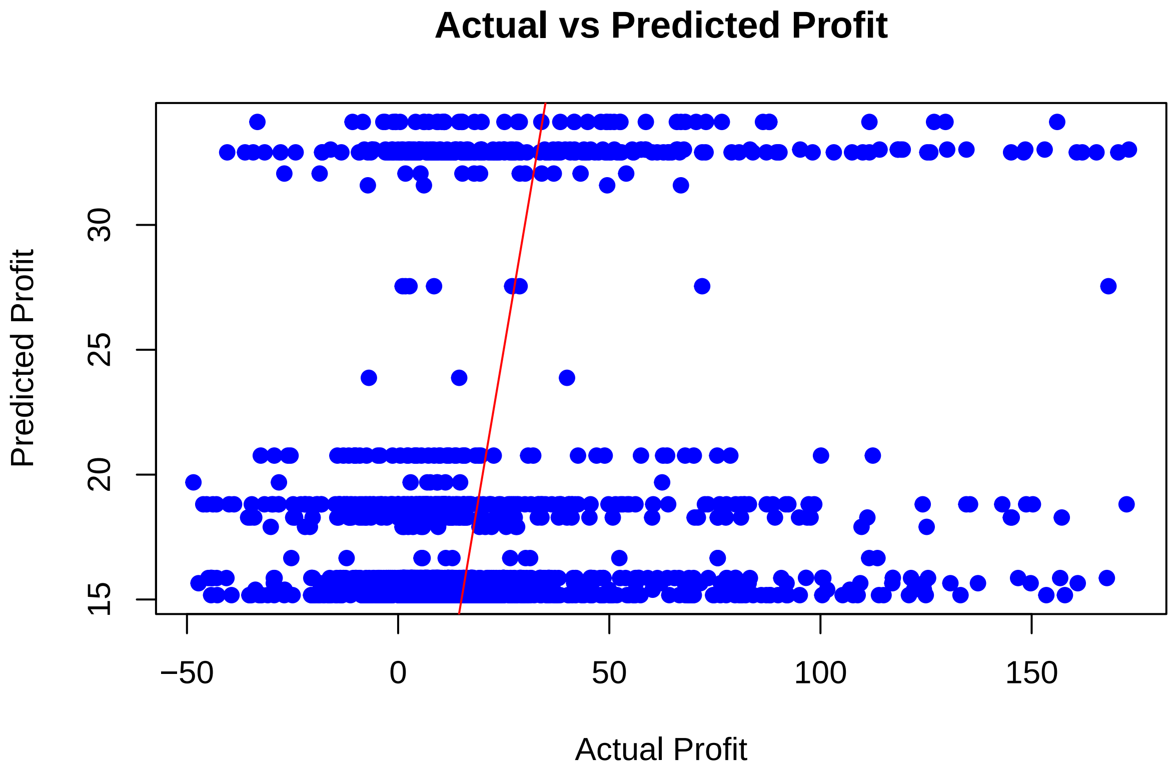
Random Forest Regression Analysis

```r
# Creating the training and testing data
set.seed(123) # Setting a random seed for reproducability
trainIndex <- createDataPartition(data_set$Profit, p = 0.8, list = FALSE)
train_data <- data_set_returns[trainIndex, ]
test_data <- data_set_returns[-trainIndex, ]

# Trimming outliers of train data
lower_percentile <- 0.05
upper_percentile <- 0.95
```

```
# Calculating the lower and upper quantiles
lower_threshold <- quantile(train_data$Profit, lower_percentile)
upper_threshold <- quantile(train_data$Profit, upper_percentile)

# Trimming outliers from the training data
train_data_trimmed <- train_data[train_data$Profit >= lower_threshold &
                                   train_data$Profit <= upper_threshold, ]

#Trimming outliers of test data
lower_threshold <- quantile(test_data$Profit, lower_percentile)
upper_threshold <- quantile(test_data$Profit, upper_percentile)

# Trimming outliers from the training data
test_data_trimmed <- test_data[test_data$Profit >= lower_threshold &
                                 test_data$Profit <= upper_threshold, ]

# Training the model
model <- train(Profit ~ Segment + Category + Returned,
               data = train_data_trimmed, method = "rf",
               na.action = na.omit,
               preProcess=c("scale","center"))

# Evaluating the model with Mean Squared Error (MSE)
predictions <- predict(model, newdata = test_data_trimmed)
mse <- mean((predictions - test_data_trimmed$Profit)^2)
print(paste("MSE:", mse))
```

```
## [1] "MSE: 1080.94646326448"
```

```
# Plot
results <- data.frame(Actual = test_data_trimmed$Profit,
                      Predicted = predictions)

# Plot actual vs predicted values
plot(results$Actual, results$Predicted,
     xlab = "Actual Profit",
     ylab = "Predicted Profit",
     main = "Actual vs Predicted Profit",
     col = "blue",
     pch = 19)

# Adding a diagonal line for comparison
abline(0, 1, col = "red")
```



```
# Predicting profit given "new data"
new_data <- data.frame(Segment = "Consumer", Category = "Technology",
                       Returned = "No")
predicted_profit <- predict(model, newdata = new_data)

predicted_profit
```

```
##        1
## 32.90644
```