STAT625_Project

Brian Zhang, Vi Mai, Xinyu Zhou (Anna), Ziyan Zhao

2023-11-30

```
rm(list = ls())
walmart <- read.csv("~/Documents/Stat 625/Project/Walmart.csv")
head(walmart)
```

```
##   Store       Date Weekly_Sales Holiday_Flag Temperature Fuel_Price      CPI
## 1     1 05-02-2010      1643691            0       42.31      2.572 211.0964
## 2     1 12-02-2010      1641957            1       38.51      2.548 211.2422
## 3     1 19-02-2010      1611968            0       39.93      2.514 211.2891
## 4     1 26-02-2010      1409728            0       46.63      2.561 211.3196
## 5     1 05-03-2010      1554807            0       46.50      2.625 211.3501
## 6     1 12-03-2010      1439542            0       57.79      2.667 211.3806
##   Unemployment
## 1        8.106
## 2        8.106
## 3        8.106
## 4        8.106
## 5        8.106
## 6        8.106
```

Data Preprocessing

Since dates are strings, they must be converted to parsed and converted to days. Use days since the first day rather than the actual date to make computation easier.

```
# Convert the dates from character strings into days since the first date
asDate_result <- as.Date(walmart$Date, "%d-%m-%Y")
first_date <- min(asDate_result)
days_elapsed <- asDate_result-first_date
walmart["Days_since"] <- days_elapsed
head(walmart)
```

```
##   Store       Date Weekly_Sales Holiday_Flag Temperature Fuel_Price      CPI
## 1     1 05-02-2010      1643691            0       42.31      2.572 211.0964
## 2     1 12-02-2010      1641957            1       38.51      2.548 211.2422
## 3     1 19-02-2010      1611968            0       39.93      2.514 211.2891
## 4     1 26-02-2010      1409728            0       46.63      2.561 211.3196
## 5     1 05-03-2010      1554807            0       46.50      2.625 211.3501
## 6     1 12-03-2010      1439542            0       57.79      2.667 211.3806
##   Unemployment Days_since
## 1        8.106     0 days
```

```
## 2        8.106     7 days
## 3        8.106    14 days
## 4        8.106    21 days
## 5        8.106    28 days
## 6        8.106    35 days
```

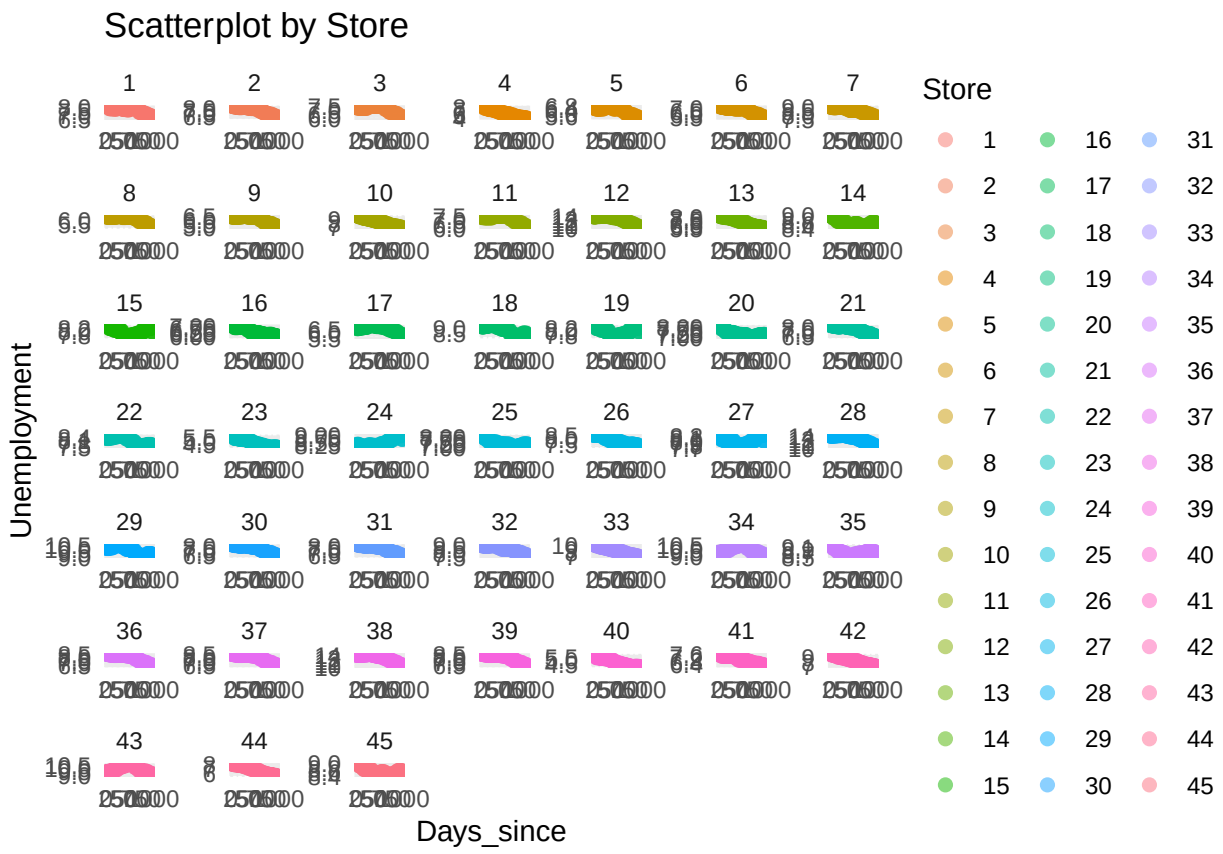
```
# Convert the dates from character strings into days since the first date
asDate_result <- as.Date(walmart$Date, "%d-%m-%Y")
first_date <- min(asDate_result)
days_elapsed <- asDate_result-first_date
walmart["Days_since"] <- days_elapsed
walmart["Weeks_since"] <- ceiling(days_elapsed / 7)
head(walmart)
```

```
##   Store       Date Weekly_Sales Holiday_Flag Temperature Fuel_Price      CPI
## 1     1 05-02-2010      1643691            0       42.31      2.572 211.0964
## 2     1 12-02-2010      1641957            1       38.51      2.548 211.2422
## 3     1 19-02-2010      1611968            0       39.93      2.514 211.2891
## 4     1 26-02-2010      1409728            0       46.63      2.561 211.3196
## 5     1 05-03-2010      1554807            0       46.50      2.625 211.3501
## 6     1 12-03-2010      1439542            0       57.79      2.667 211.3806
##   Unemployment Days_since Weeks_since
## 1        8.106     0 days      0 days
## 2        8.106     7 days      1 days
## 3        8.106    14 days      2 days
## 4        8.106    21 days      3 days
## 5        8.106    28 days      4 days
## 6        8.106    35 days      5 days
```

Scatterplot with Unemployment vs others

```
library(ggplot2)

# Assuming 'Store' is a factor variable
walmart$Store <- as.factor(walmart$Store)

ggplot(walmart, aes(x = Days_since, y = Unemployment)) +
  geom_point(aes(color = Store), alpha = 0.5, size = 2) +
  facet_wrap(~ Store, scales = "free") +
  labs(title = "Scatterplot by Store",
       x = "Days_since",
       y = "Unemployment") +
  theme_minimal()
```

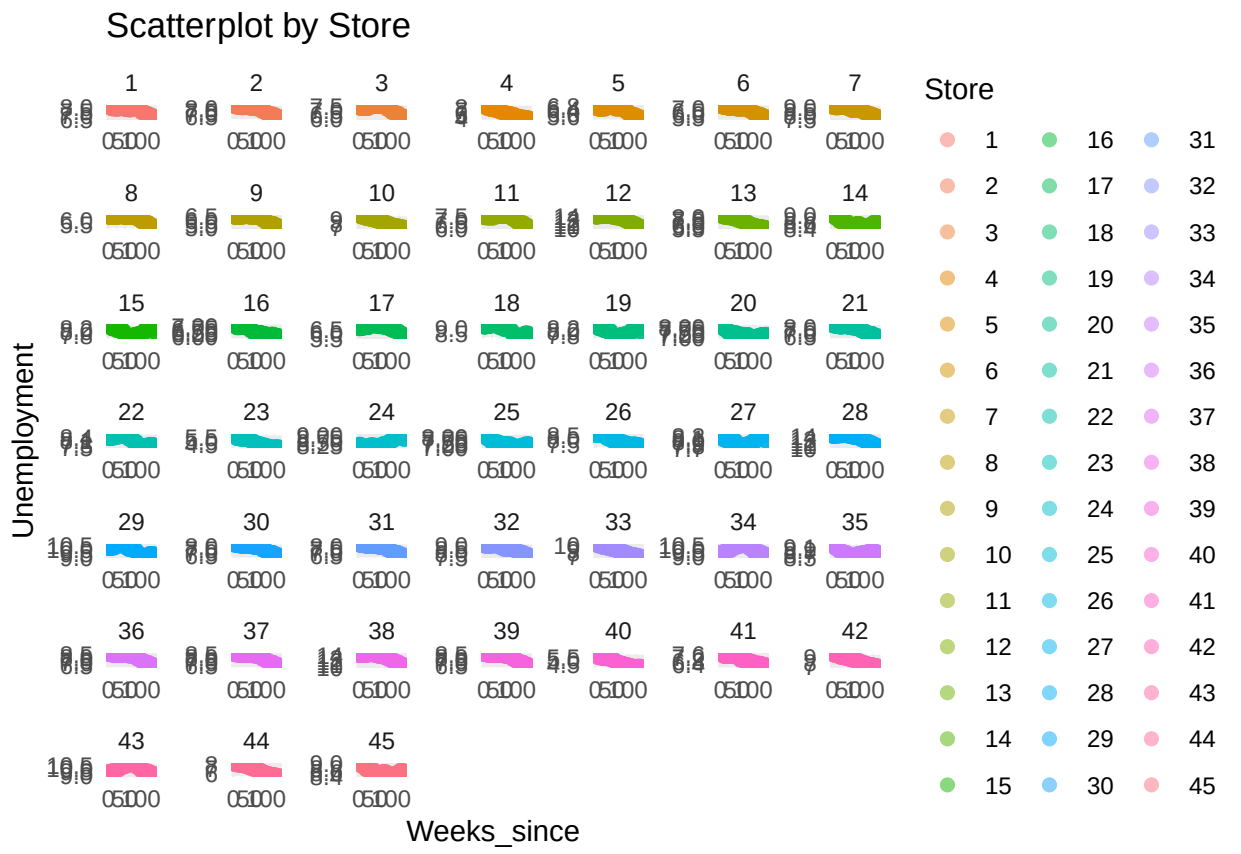
```
## Don't know how to automatically pick scale for object of type <difftime>.
## Defaulting to continuous.
```



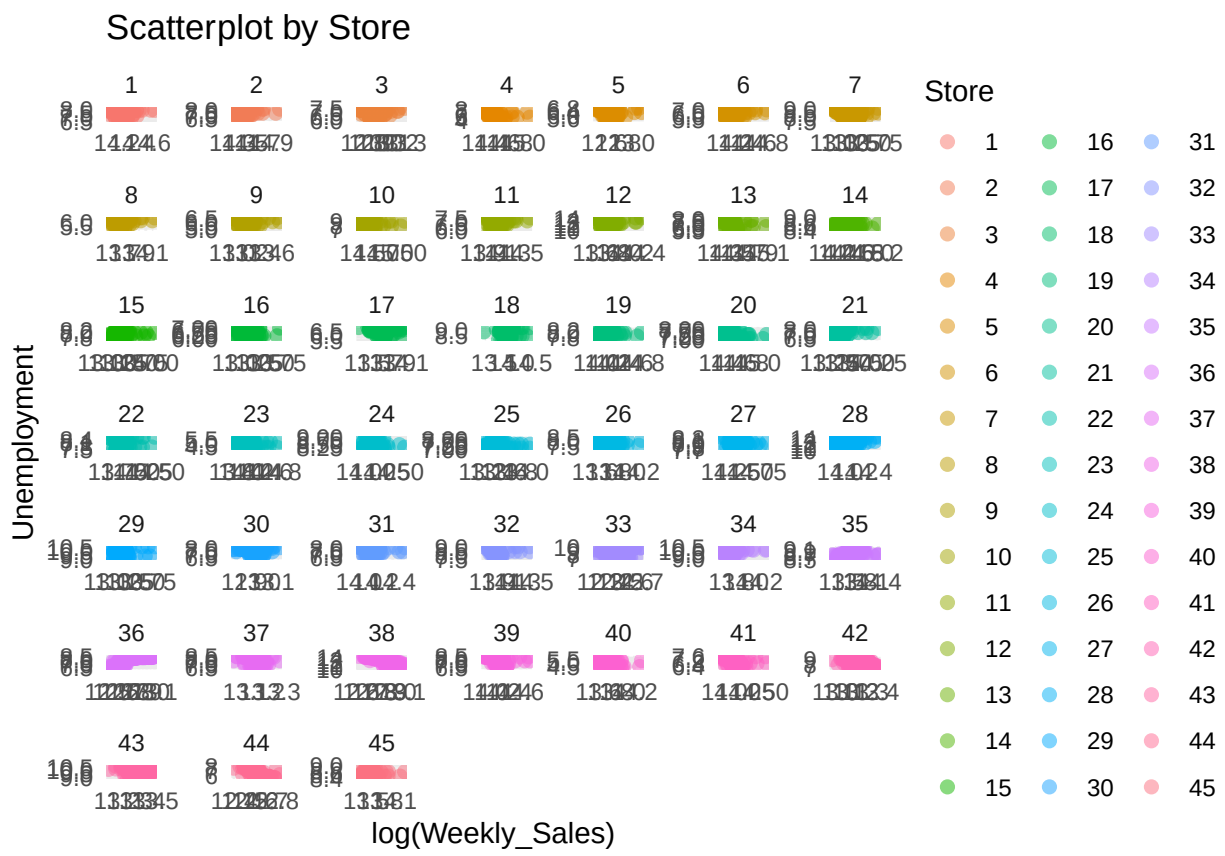
Weeks since

```
library(ggplot2)

# Assuming 'Store' is a factor variable
walmart$Store <- as.factor(walmart$Store)

ggplot(walmart, aes(x = Weeks_since, y = Unemployment)) +
  geom_point(aes(color = Store), alpha = 0.5, size = 2) +
  facet_wrap(~ Store, scales = "free") +
  labs(title = "Scatterplot by Store",
       x = "Weeks_since",
       y = "Unemployment") +
  theme_minimal()
```

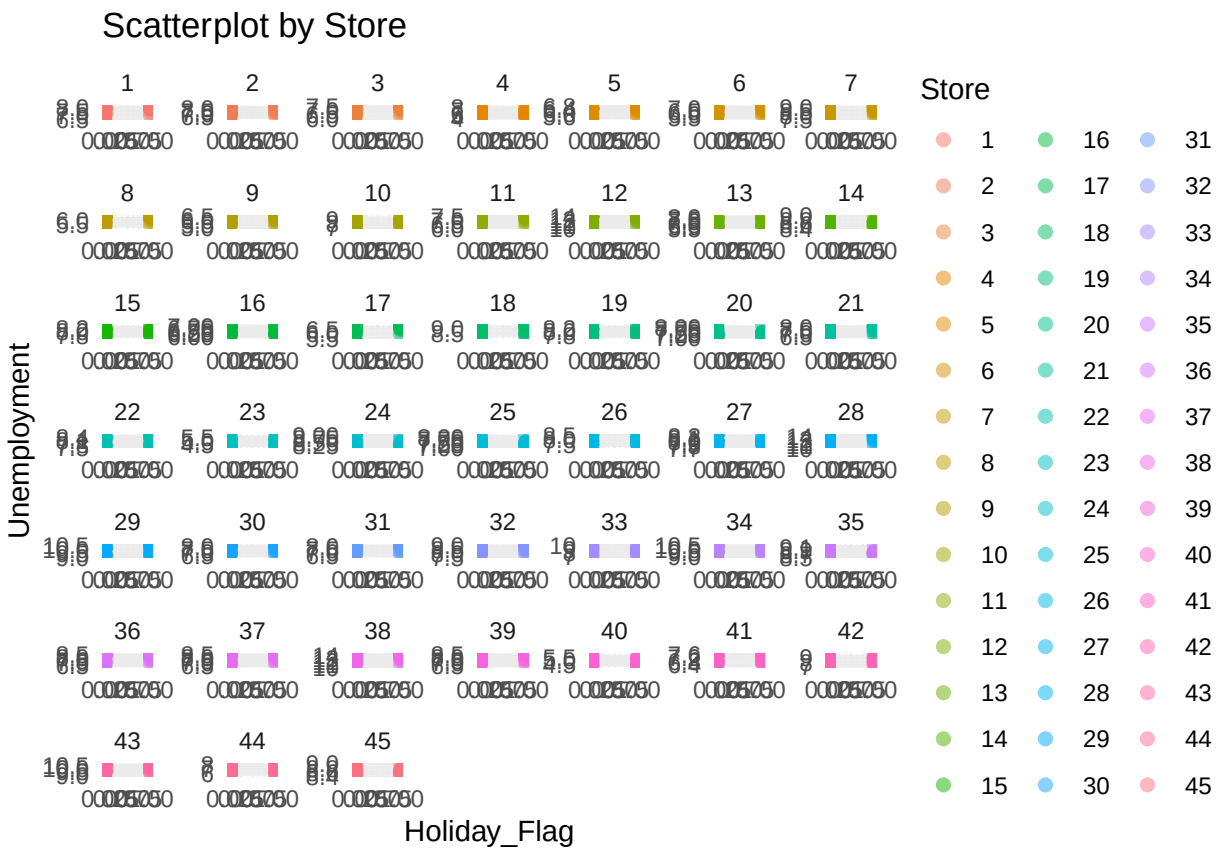
```
## Don't know how to automatically pick scale for object of type <difftime>.
## Defaulting to continuous.
```




```r
# Assuming 'Store' is a factor variable
walmart$Store <- as.factor(walmart$Store)

ggplot(walmart, aes(x = log(Weekly_Sales), y = Unemployment)) +
  geom_point(aes(color = Store), alpha = 0.5, size = 2) +
  facet_wrap(~ Store, scales = "free") +
  labs(title = "Scatterplot by Store",
       x = "log(Weekly_Sales)",
       y = "Unemployment") +
  theme_minimal()
```

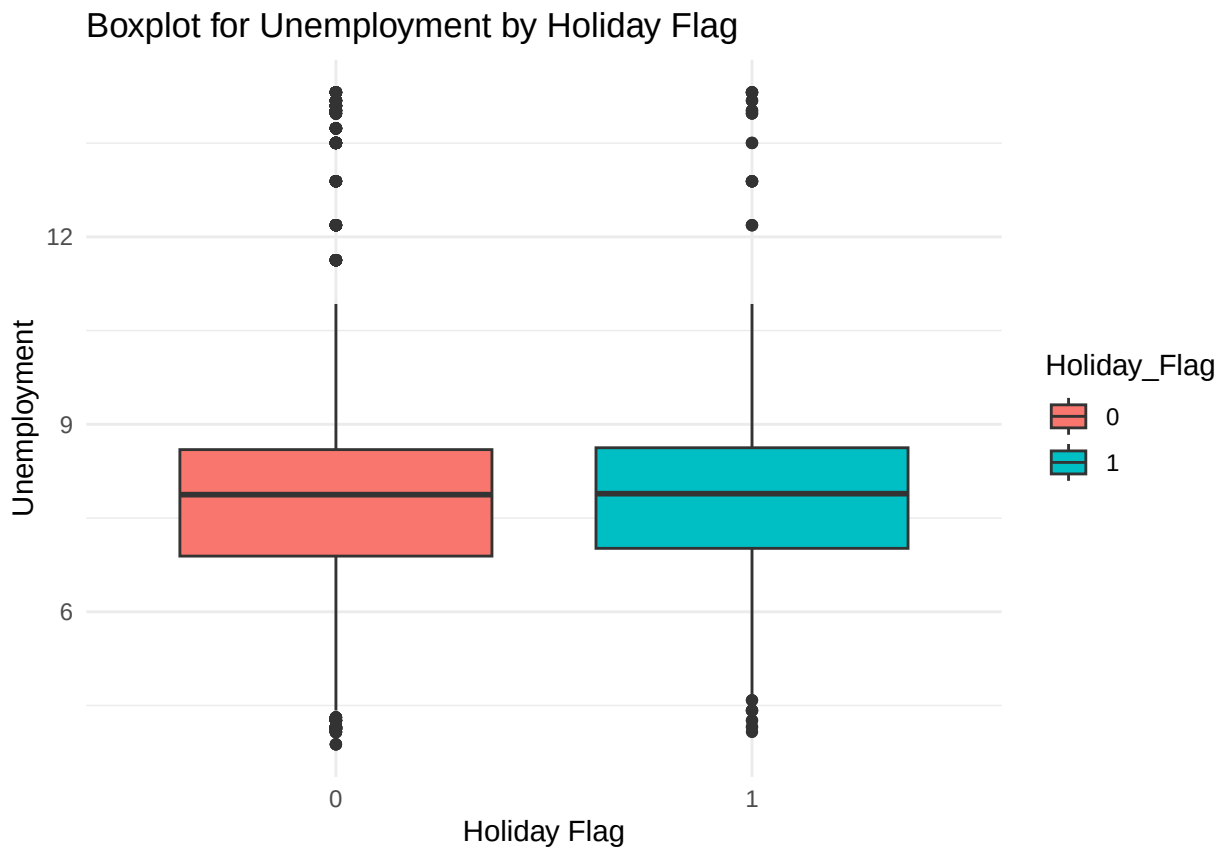



```
# Assuming 'Store' is a factor variable
walmart$Store <- as.factor(walmart$Store)

ggplot(walmart, aes(x = Holiday_Flag, y = Unemployment)) +
  geom_point(aes(color = Store), alpha = 0.5, size = 2) +
  facet_wrap(~ Store, scales = "free") +
  labs(title = "Scatterplot by Store",
       x = "Holiday_Flag",
       y = "Unemployment") +
  theme_minimal()
```

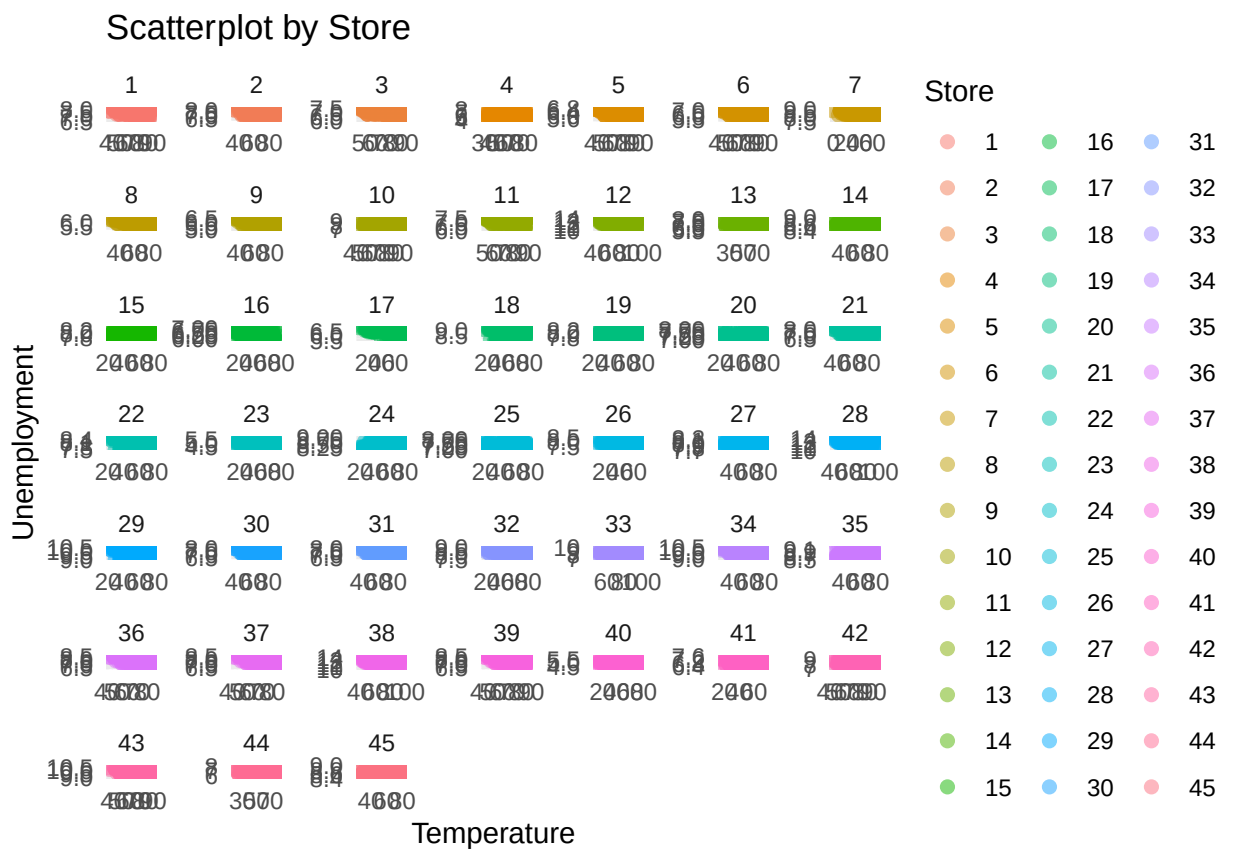


Boxplot for Unemployment based on Holiday_Flag

```
library(ggplot2)

# Assuming 'Holiday_Flag' is a factor variable
walmart$Holiday_Flag <- as.factor(walmart$Holiday_Flag)

# Boxplot for Unemployment based on Holiday_Flag
ggplot(walmart, aes(x = Holiday_Flag, y = Unemployment, fill = Holiday_Flag)) +
  geom_boxplot() +
  labs(title = "Boxplot for Unemployment by Holiday Flag",
       x = "Holiday Flag",
       y = "Unemployment") +
  theme_minimal()
```

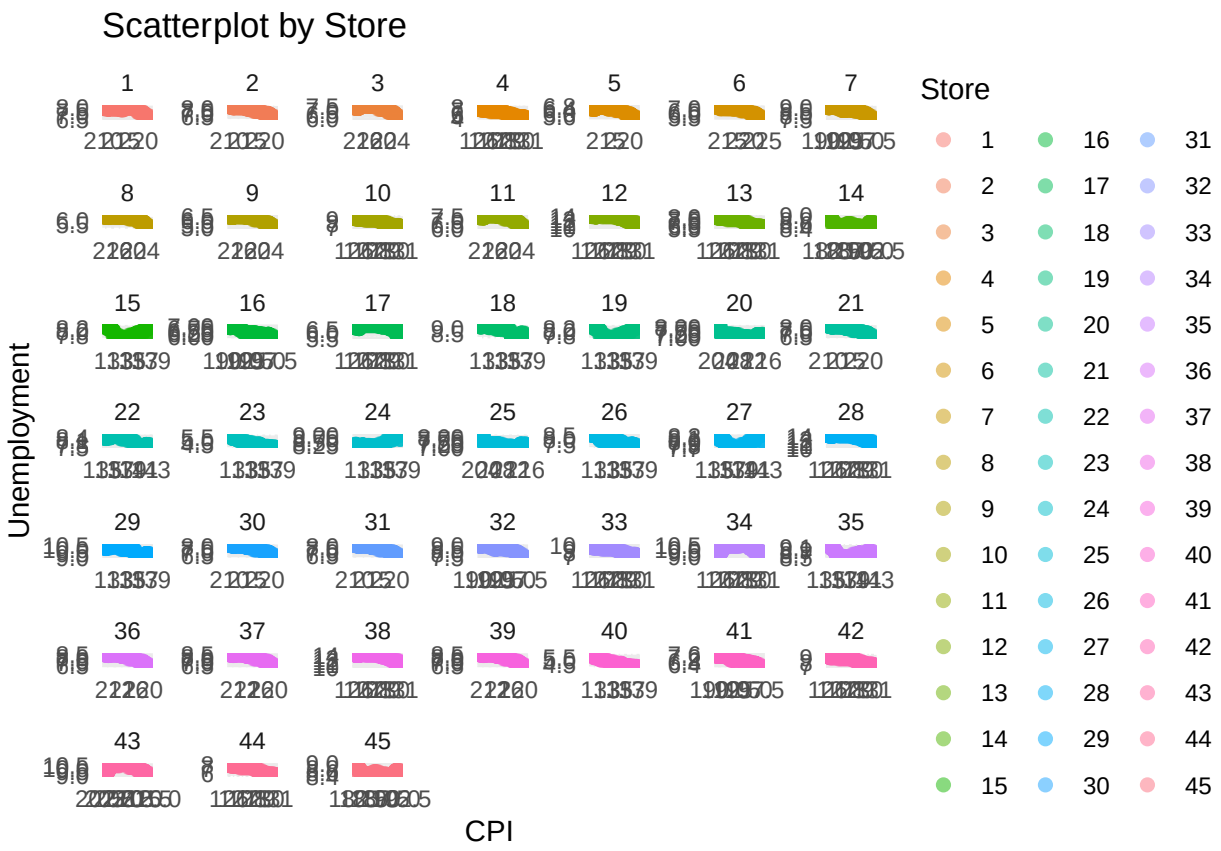



```
# Assuming 'Store' is a factor variable
walmart$Store <- as.factor(walmart$Store)

ggplot(walmart, aes(x = Temperature, y = Unemployment)) +
  geom_point(aes(color = Store), alpha = 0.5, size = 2) +
  facet_wrap(~ Store, scales = "free") +
  labs(title = "Scatterplot by Store",
       x = "Temperature",
       y = "Unemployment") +
  theme_minimal()
```




```r
# Assuming 'Store' is a factor variable
walmart$Store <- as.factor(walmart$Store)

ggplot(walmart, aes(x = Fuel_Price, y = Unemployment)) +
  geom_point(aes(color = Store), alpha = 0.5, size = 2) +
  facet_wrap(~ Store, scales = "free") +
  labs(title = "Scatterplot by Store",
       x = "Fuel_Price",
       y = "Unemployment") +
  theme_minimal()
```

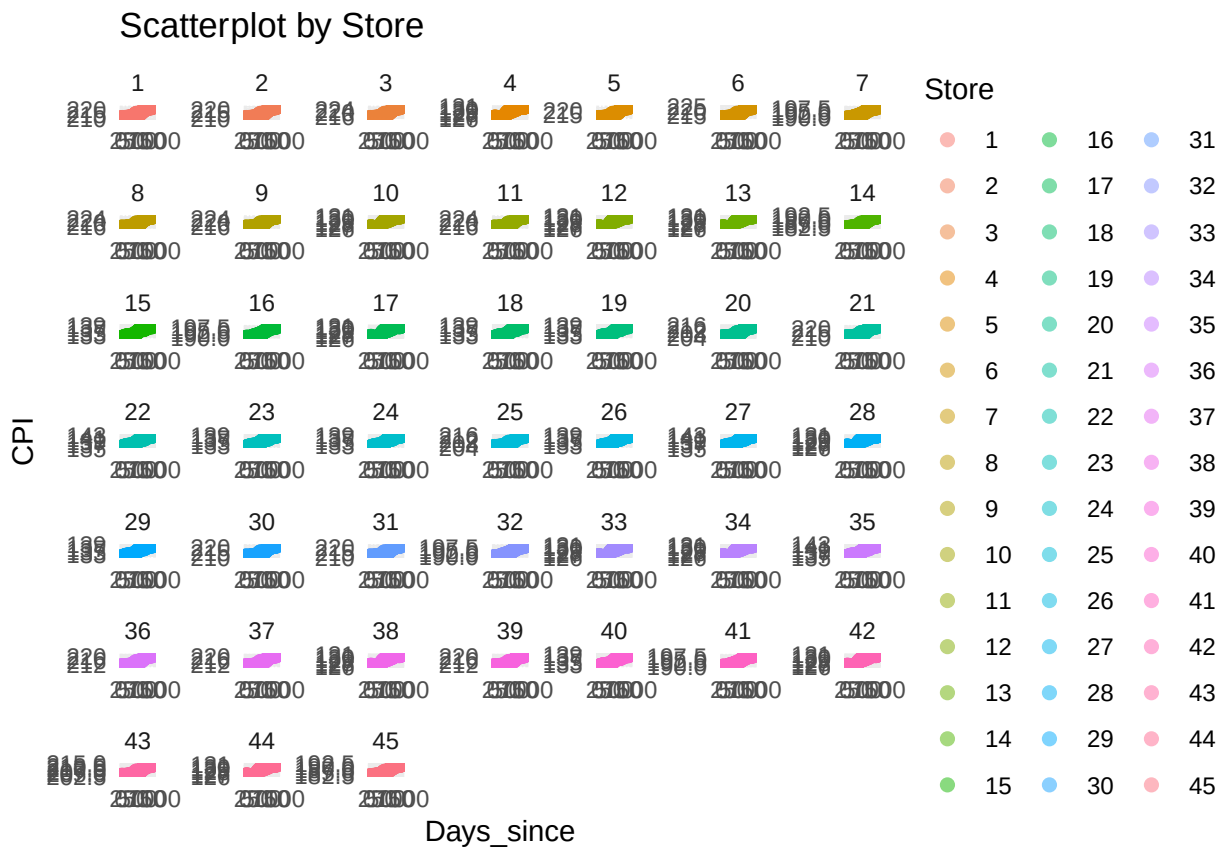

```r
# Assuming 'Store' is a factor variable
walmart$Store <- as.factor(walmart$Store)

ggplot(walmart, aes(x = CPI, y = Unemployment)) +
  geom_point(aes(color = Store), alpha = 0.5, size = 2) +
  facet_wrap(~ Store, scales = "free") +
  labs(title = "Scatterplot by Store",
       x = "CPI",
       y = "Unemployment") +
  theme_minimal()
```



Scatterplot with CPI vs others

```
# Assuming 'Store' is a factor variable
walmart$Store <- as.factor(walmart$Store)

ggplot(walmart, aes(x = Days_since, y = CPI)) +
  geom_point(aes(color = Store), alpha = 0.5, size = 2) +
  facet_wrap(~ Store, scales = "free") +
  labs(title = "Scatterplot by Store",
       x = "Days_since",
       y = "CPI") +
  theme_minimal()
```

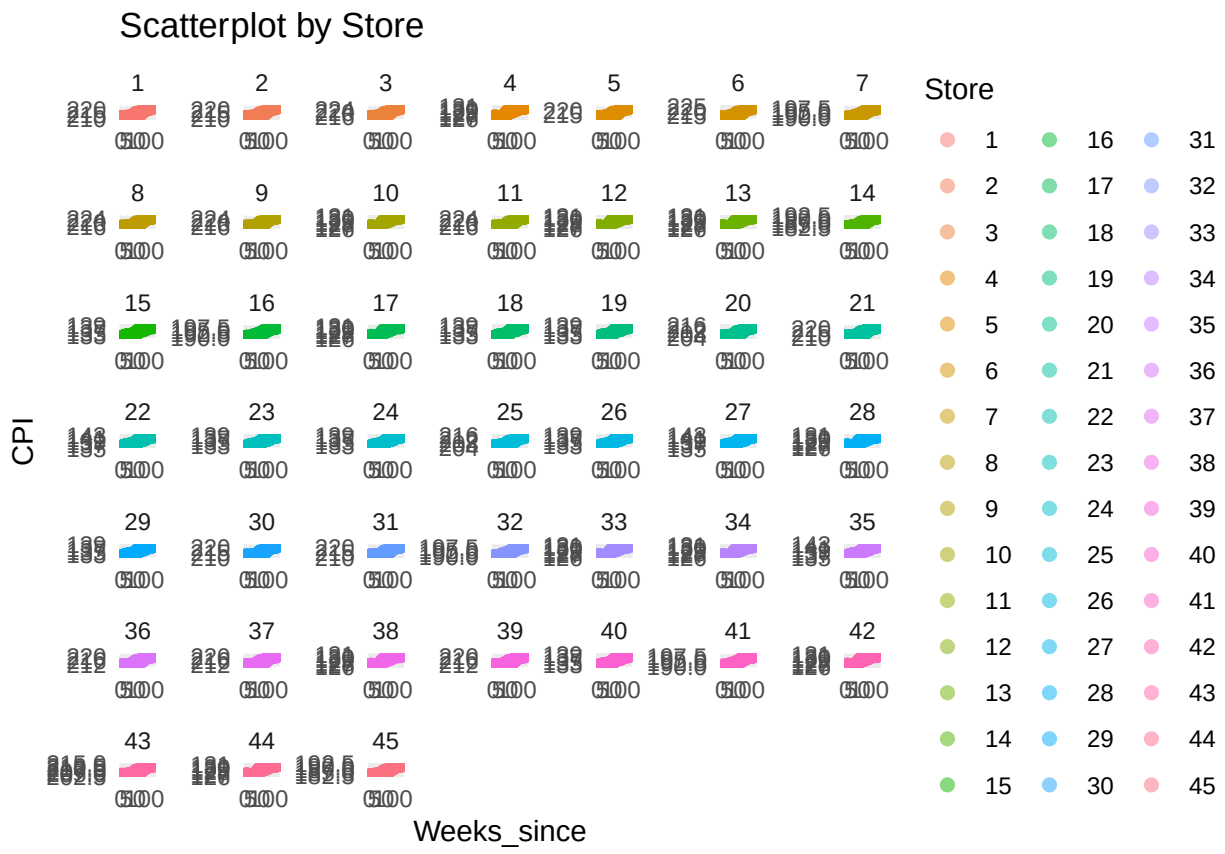
```
## Don't know how to automatically pick scale for object of type <difftime>.
## Defaulting to continuous.
```



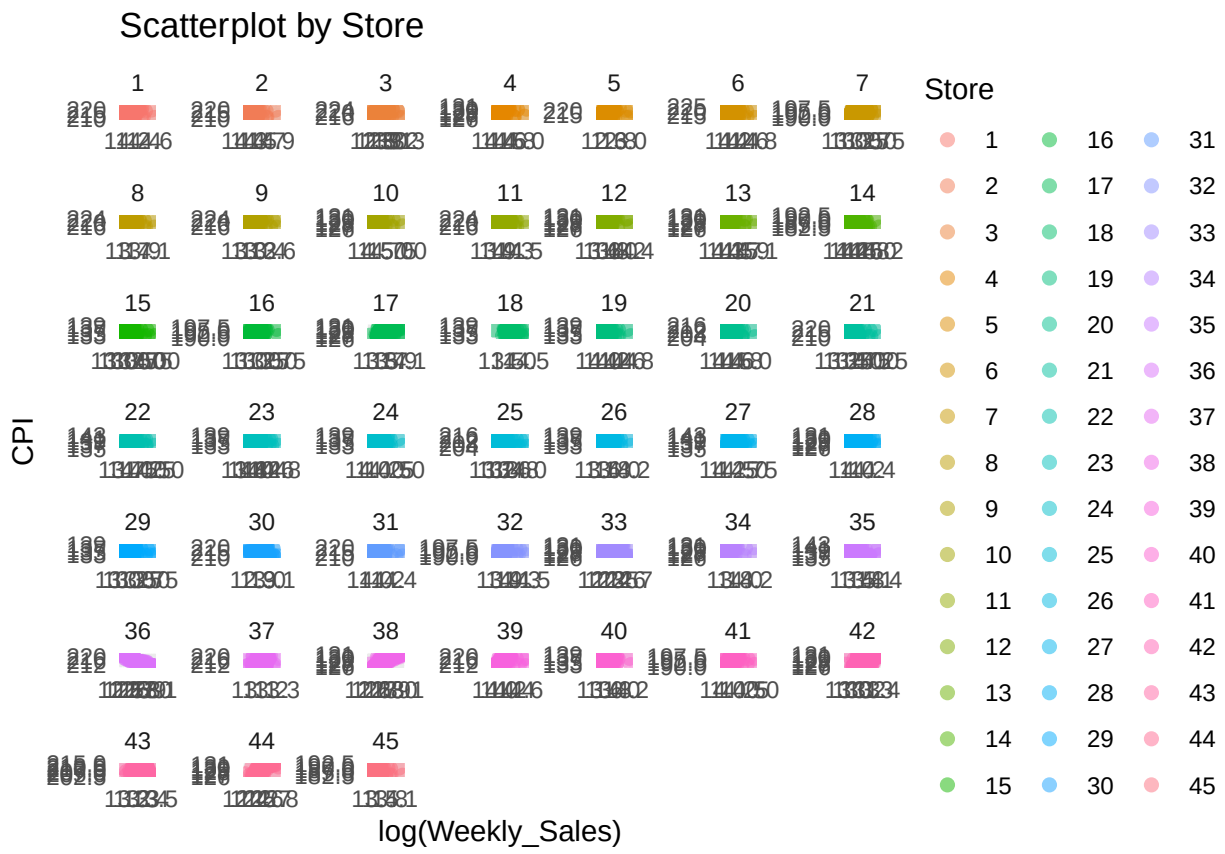
Weeks since

```
# Assuming 'Store' is a factor variable
walmart$Store <- as.factor(walmart$Store)

ggplot(walmart, aes(x = Weeks_since, y = CPI)) +
  geom_point(aes(color = Store), alpha = 0.5, size = 2) +
  facet_wrap(~ Store, scales = "free") +
  labs(title = "Scatterplot by Store",
       x = "Weeks_since",
       y = "CPI") +
  theme_minimal()
```

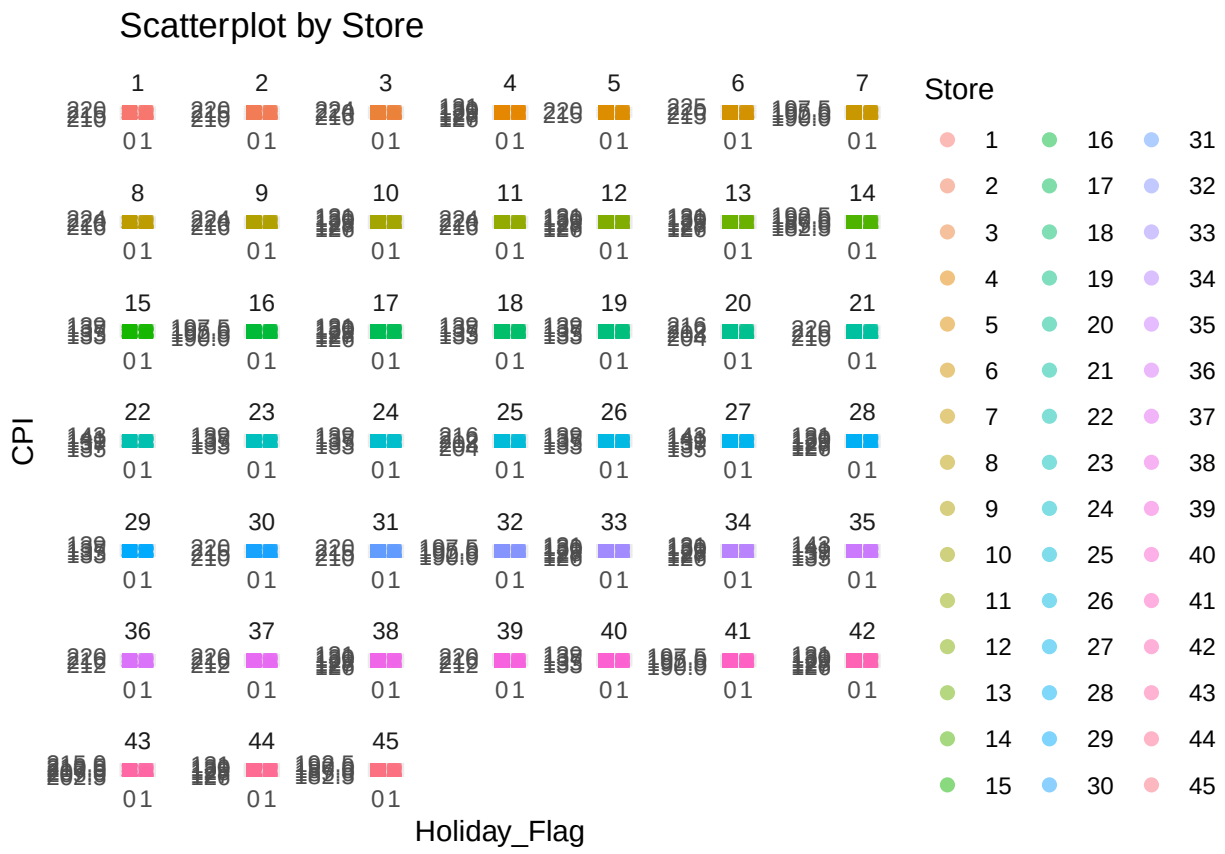
```
## Don't know how to automatically pick scale for object of type <difftime>.
## Defaulting to continuous.
```




```r
# Assuming 'Store' is a factor variable
walmart$Store <- as.factor(walmart$Store)

ggplot(walmart, aes(x = log(Weekly_Sales), y = CPI)) +
  geom_point(aes(color = Store), alpha = 0.5, size = 2) +
  facet_wrap(~ Store, scales = "free") +
  labs(title = "Scatterplot by Store",
       x = "log(Weekly_Sales)",
       y = "CPI") +
  theme_minimal()
```

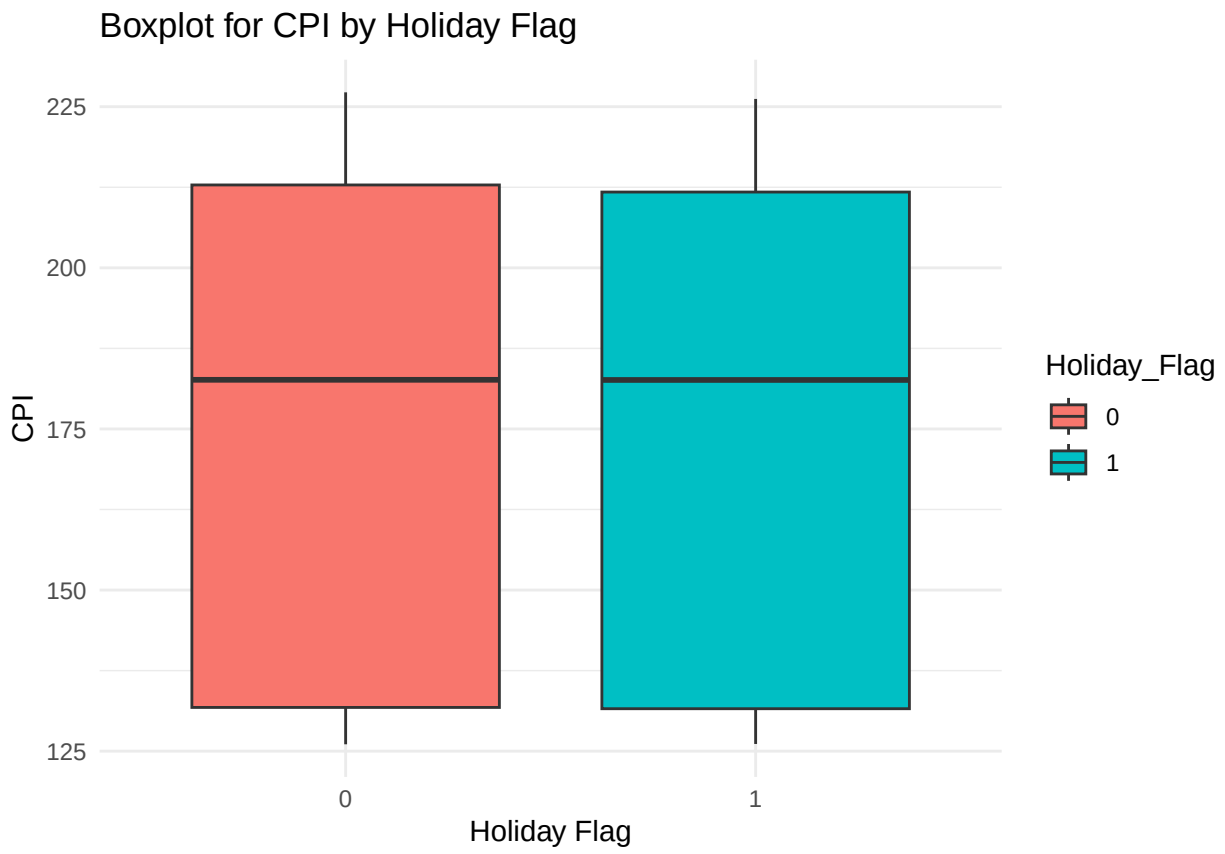



```
# Assuming 'Store' is a factor variable
walmart$Store <- as.factor(walmart$Store)

ggplot(walmart, aes(x = Holiday_Flag, y = CPI)) +
  geom_point(aes(color = Store), alpha = 0.5, size = 2) +
  facet_wrap(~ Store, scales = "free") +
  labs(title = "Scatterplot by Store",
       x = "Holiday_Flag",
       y = "CPI") +
  theme_minimal()
```

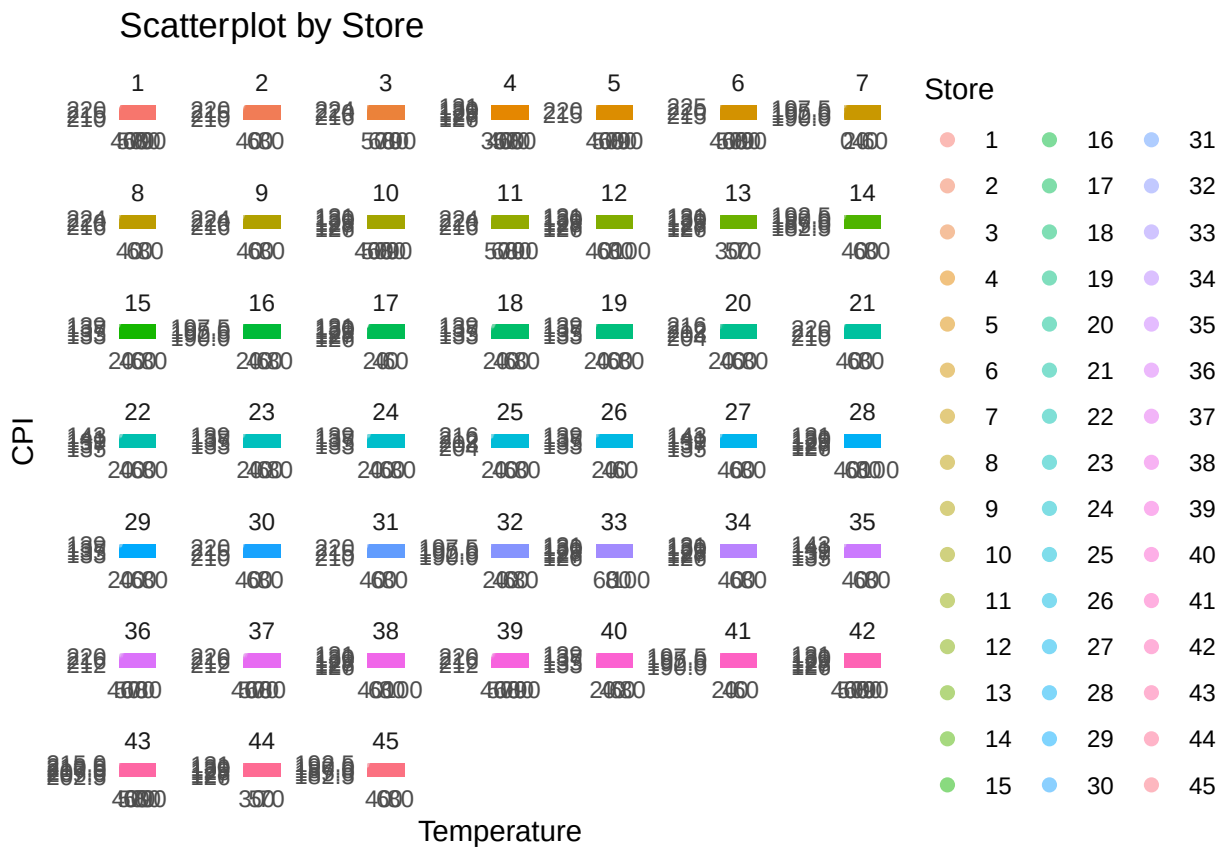


Boxplot for CPI based on Holiday_Flag

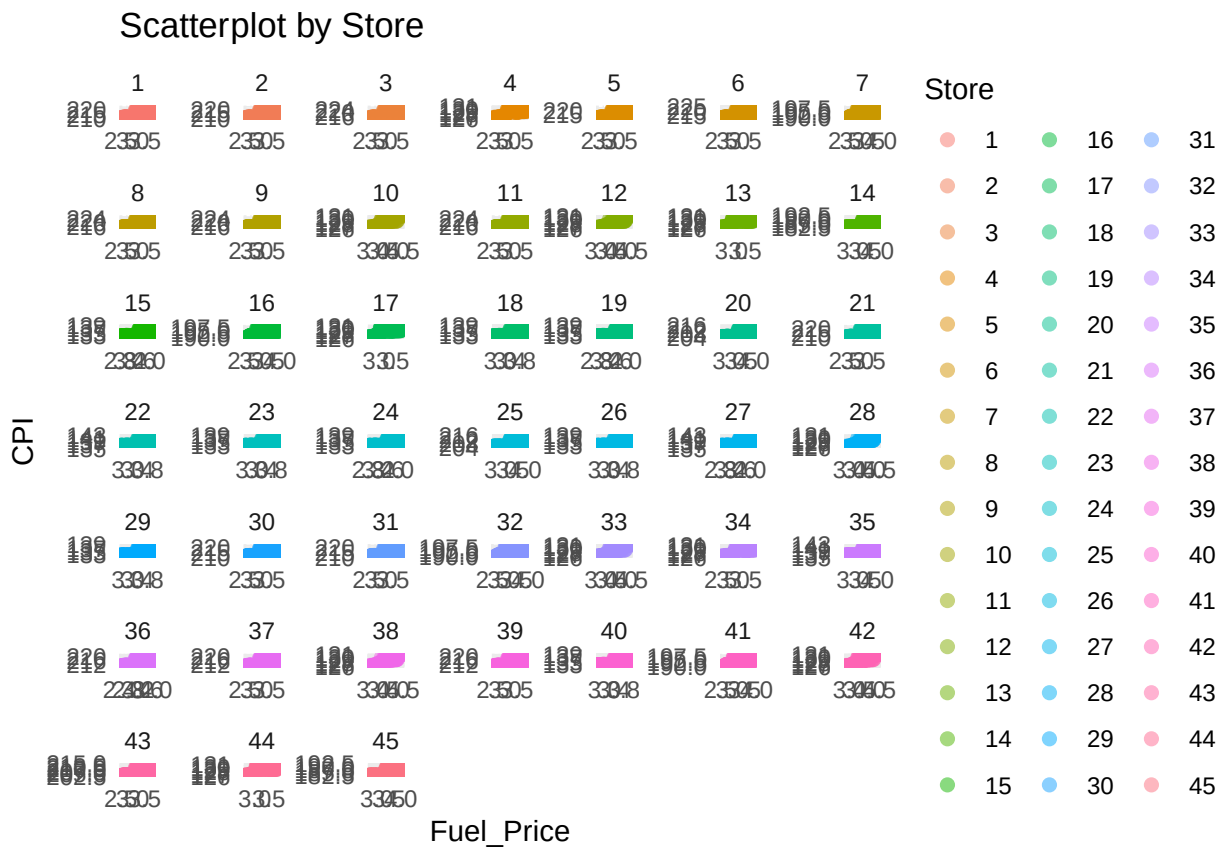
```
library(ggplot2)

# Assuming 'Holiday_Flag' is a factor variable
walmart$Holiday_Flag <- as.factor(walmart$Holiday_Flag)

# Boxplot for CPI based on Holiday_Flag
ggplot(walmart, aes(x = Holiday_Flag, y = CPI, fill = Holiday_Flag)) +
  geom_boxplot() +
  labs(title = "Boxplot for CPI by Holiday Flag",
       x = "Holiday Flag",
       y = "CPI") +
  theme_minimal()
```




```
# Assuming 'Store' is a factor variable
walmart$Store <- as.factor(walmart$Store)

ggplot(walmart, aes(x = Temperature, y = CPI)) +
  geom_point(aes(color = Store), alpha = 0.5, size = 2) +
  facet_wrap(~ Store, scales = "free") +
  labs(title = "Scatterplot by Store",
       x = "Temperature",
       y = "CPI") +
  theme_minimal()
```



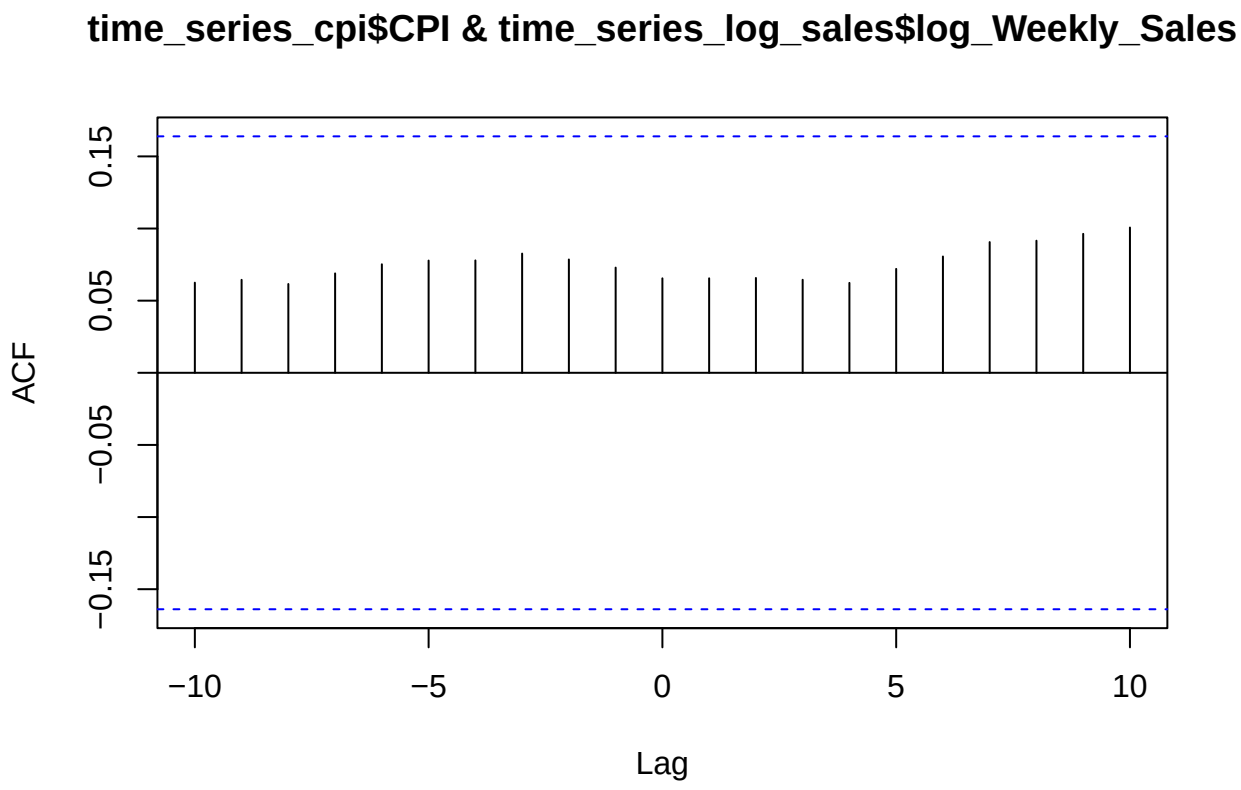

```r
# Assuming 'Store' is a factor variable
walmart$Store <- as.factor(walmart$Store)

ggplot(walmart, aes(x = Fuel_Price, y = CPI)) +
  geom_point(aes(color = Store), alpha = 0.5, size = 2) +
  facet_wrap(~ Store, scales = "free") +
  labs(title = "Scatterplot by Store",
       x = "Fuel_Price",
       y = "CPI") +
  theme_minimal()
```




```r
# Assuming 'Store' is a factor variable
walmart$Store <- as.factor(walmart$Store)

ggplot(walmart, aes(x = Unemployment, y = CPI)) +
  geom_point(aes(color = Store), alpha = 0.5, size = 2) +
  facet_wrap(~ Store, scales = "free") +
  labs(title = "Scatterplot by Store",
       x = "Unemployment",
       y = "CPI") +
  theme_minimal()
```

lagged model

```
# Assuming CPI is in a column called "CPI" and Weekly_Sales in "log_Weekly_Sales"
walmart$Date <- as.Date(walmart$Date, format="%d-%m-%Y")
walmart$Weeks_since <- as.numeric(difftime(walmart$Date, min(walmart$Date), units = "days")) / 7

# Extract time series data for CPI
time_series_cpi <- data.frame(Weeks_since = unique(walmart$Weeks_since), CPI = numeric(length(unique(wal
for (week in unique(walmart$Weeks_since)) {
  time_series_cpi$CPI[time_series_cpi$Weeks_since == week] <- walmart$CPI[walmart$Weeks_since == week][1
}

# Extract time series data for log_Weekly_Sales
time_series_log_sales <- data.frame(Weeks_since = unique(walmart$Weeks_since), log_Weekly_Sales = numeri
for (week in unique(walmart$Weeks_since)) {
  time_series_log_sales$log_Weekly_Sales[time_series_log_sales$Weeks_since == week] <- mean(log(walmart$
}

ccf_result <- ccf(time_series_cpi$CPI, time_series_log_sales$log_Weekly_Sales, lag.max = 10)
```




```
ccf_result
```

```
##
## Autocorrelations of series 'X', by lag
##
##   -10    -9    -8    -7    -6    -5    -4    -3    -2    -1     0     1     2
## 0.062 0.064 0.062 0.069 0.075 0.078 0.078 0.083 0.079 0.073 0.065 0.066 0.066
##     3     4     5     6     7     8     9    10
## 0.064 0.062 0.072 0.081 0.091 0.092 0.096 0.101
```