Assignment 2 - MRA

Albert Kaczmarek Maciej Kilijański Balazs Kiss

```
library(ggplot2)
library(dplyr)
library(tidyverse)
library(corrplot)
library(lubridate)
```

```
# Task 1
dataset <- read.csv("dataset.csv")
# Overview of the data structure
summary(dataset)
```

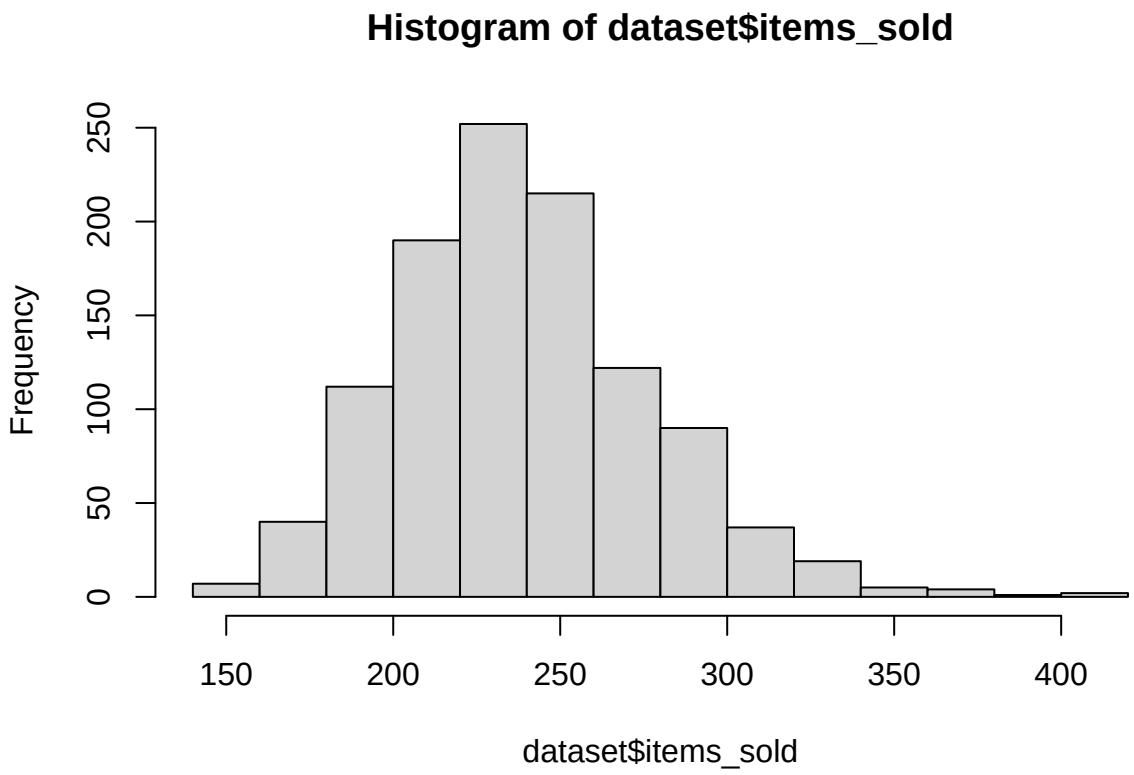
```
##      Date              weekday              price           flyer
##  Length:1096        Length:1096        Min.   :4.090   Min.   :0.0000
##  Class :character   Class :character   1st Qu.:4.290   1st Qu.:0.0000
##  Mode  :character   Mode  :character   Median :4.470   Median :0.0000
##                                        Mean   :4.637   Mean   :0.2993
##                                        3rd Qu.:4.880   3rd Qu.:1.0000
##                                        Max.   :5.720   Max.   :1.0000
##    items_sold
##  Min.   :141.0
##  1st Qu.:213.0
##  Median :236.5
##  Mean   :239.2
##  3rd Qu.:261.0
##  Max.   :408.0
```

```
## No nulls, need to transform dates to date
## Flier to be transformed as factor

dataset$Date = parse_date(dataset$Date) #Convert Date to date format
dataset$flyer = as.factor(dataset$flyer) #Convert Date to date format
summary(dataset)
```

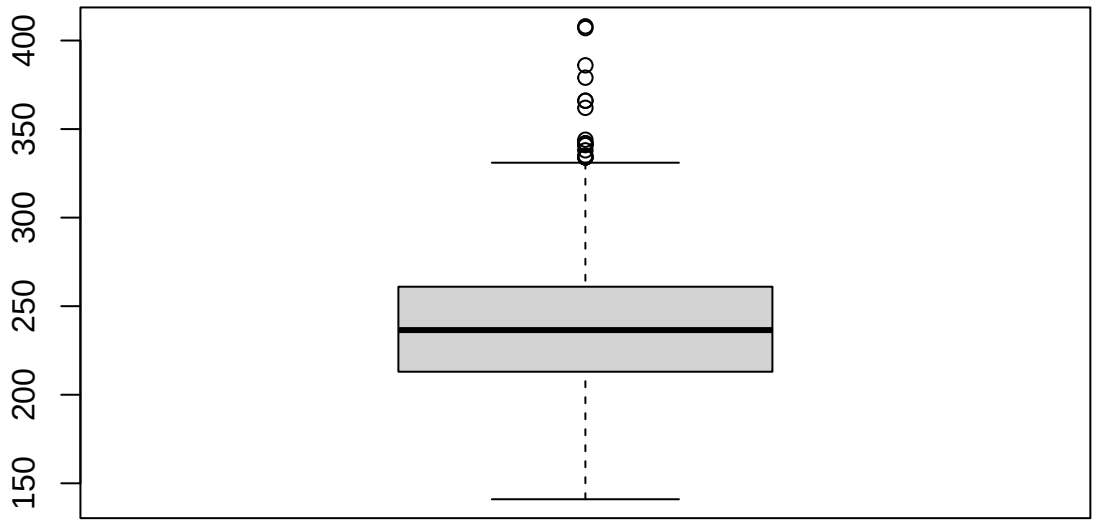
```
##       Date              weekday              price       flyer
##  Min.   :2020-01-01   Length:1096        Min.   :4.090   0:768
##  1st Qu.:2020-09-30   Class :character   1st Qu.:4.290   1:328
##  Median :2021-07-01   Mode  :character   Median :4.470
##  Mean   :2021-07-01                      Mean   :4.637
##  3rd Qu.:2022-04-01                      3rd Qu.:4.880
##  Max.   :2022-12-31                      Max.   :5.720
##    items_sold
##  Min.   :141.0
##  1st Qu.:213.0
```

```
##  Median :236.5
##  Mean   :239.2
##  3rd Qu.:261.0
##  Max.   :408.0
```

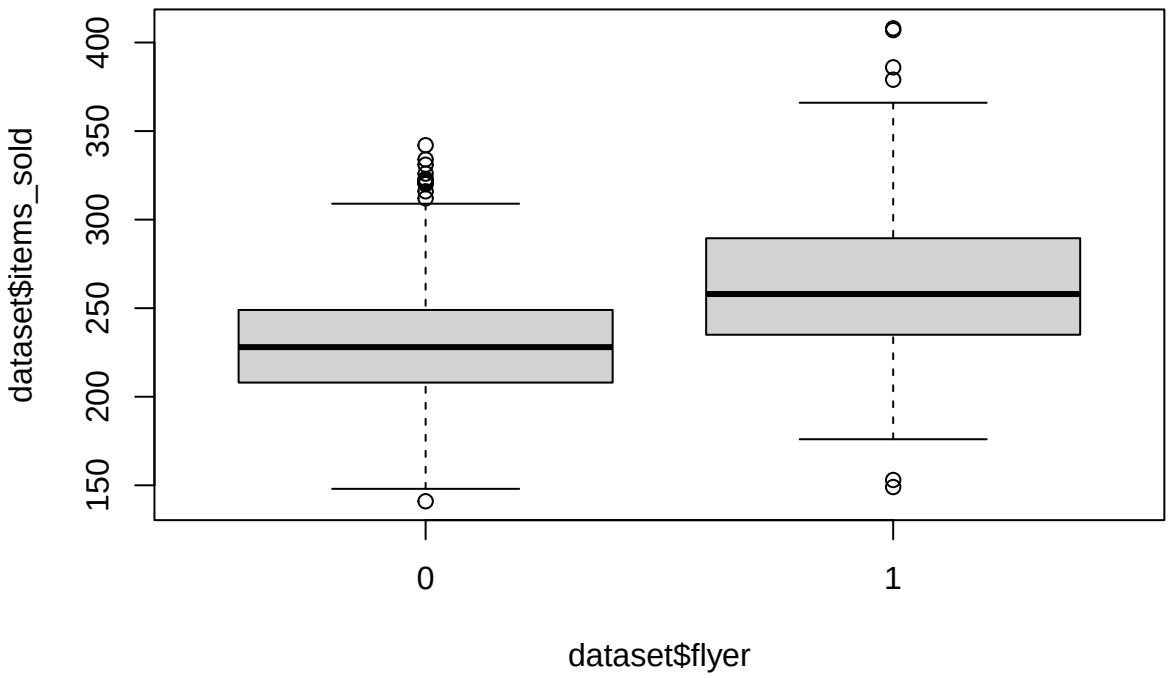
```
# No need to analyze the price as it is a dimension, not a measure

hist(dataset$items_sold)
```




```
boxplot(dataset$items_sold) # Just number of items sold
```

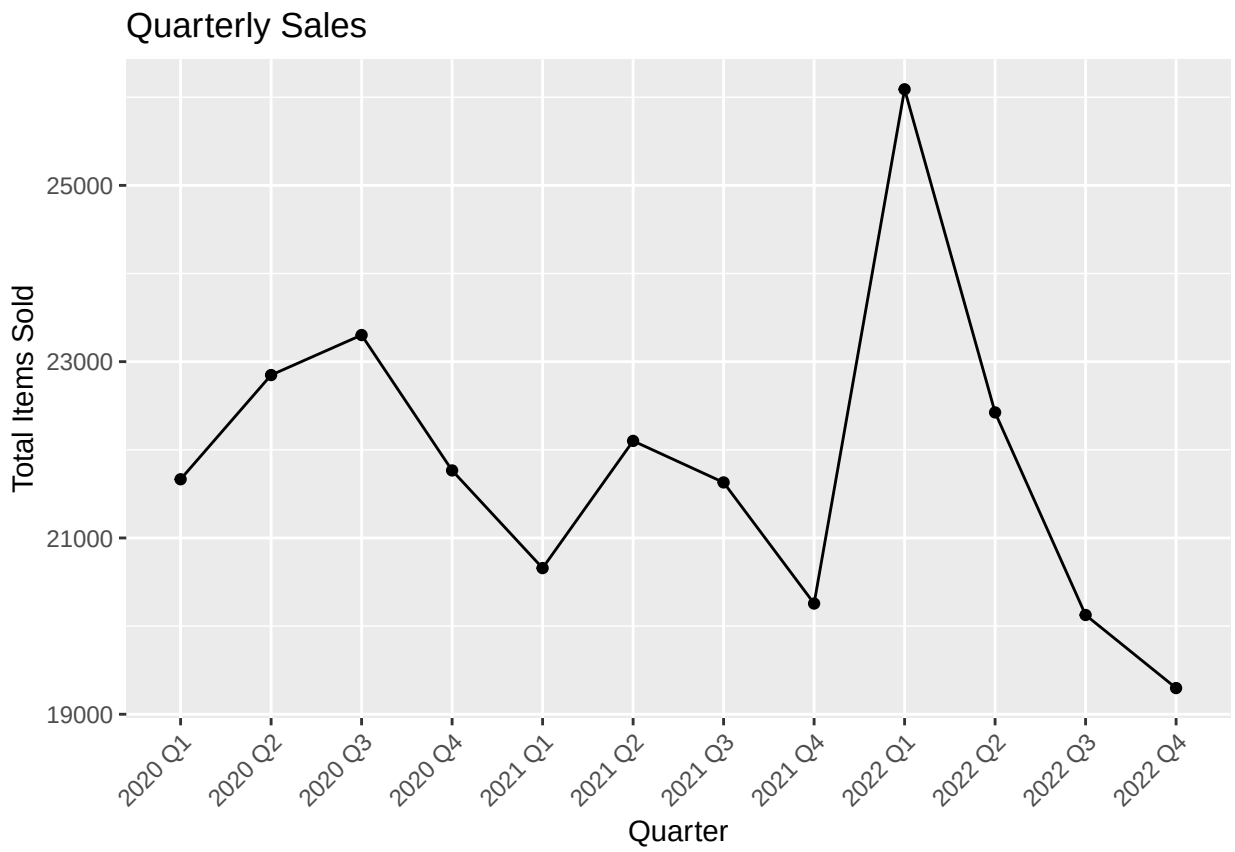



```
boxplot(dataset$items_sold ~ dataset$flyer) # Including the flyer as factor
```




```r
# Create a new column for quarter and year
dataset <- dataset %>%
  mutate(Quarter_Year = paste0(year(Date), " Q", quarter(Date)))

dataset$Month_Year <- paste0(year(dataset$Date), '-', month(dataset$Date))

# Aggregate sales by quarter and year
quarterly_sales <- dataset %>%
  group_by(Quarter_Year) %>%
  summarise(total_items_sold = sum(items_sold))

monthly_sales <- dataset %>%
  group_by(Month_Year) %>%
  summarise(total_items_sold = sum(items_sold))

weekday_sales <- dataset %>%
  group_by(weekday) %>%
  summarise(total_items_sold = mean(items_sold))

weekday_sales$day_number <- recode(weekday_sales$weekday,
       "Montag"=0,
       "Dienstag"=1,
       "Mittwoch"=2,
       "Donnerstag"=3,
       "Freitag"=4,
       "Samstag"=5,
```

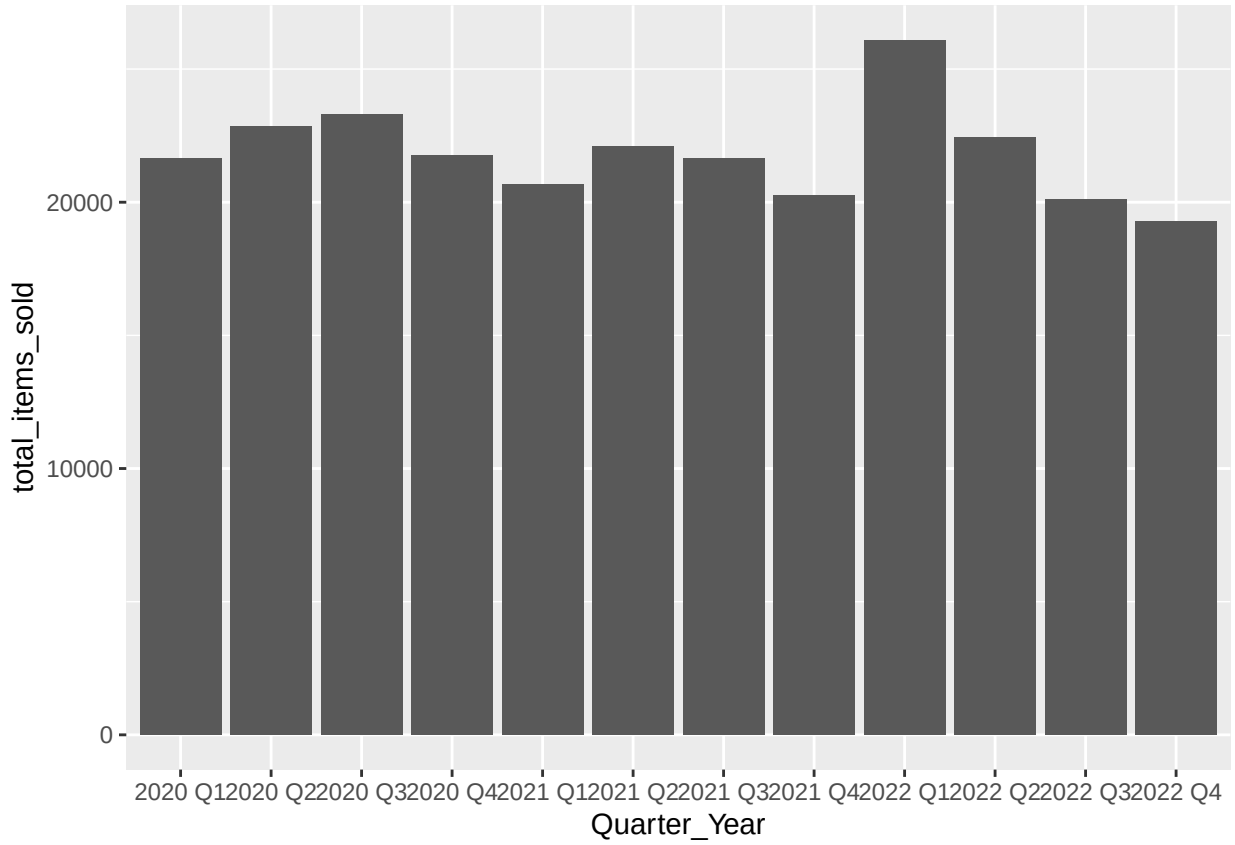
```
       "Sonntag"=6)

weekday_sales <- weekday_sales %>% arrange(day_number)

# Convert Quarter_Year to a factor with the correct order
quarterly_sales <- quarterly_sales %>%
  mutate(Quarter_Year = factor(Quarter_Year, levels = sort(unique(Quarter_Year))))
# Convert Month_Year to a factor with the correct order
monthly_sales <- monthly_sales %>%
  mutate(Month_Year = factor(Month_Year, levels = sort(unique(Month_Year))))

# Plot the aggregated sales
plot_quarterly <- ggplot(quarterly_sales, aes(x = Quarter_Year, y = total_items_sold)) +
  geom_line(group = 1) +
  geom_point() +
  labs(x = "Quarter", y = "Total Items Sold", title = "Quarterly Sales") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

plot_quarterly
```



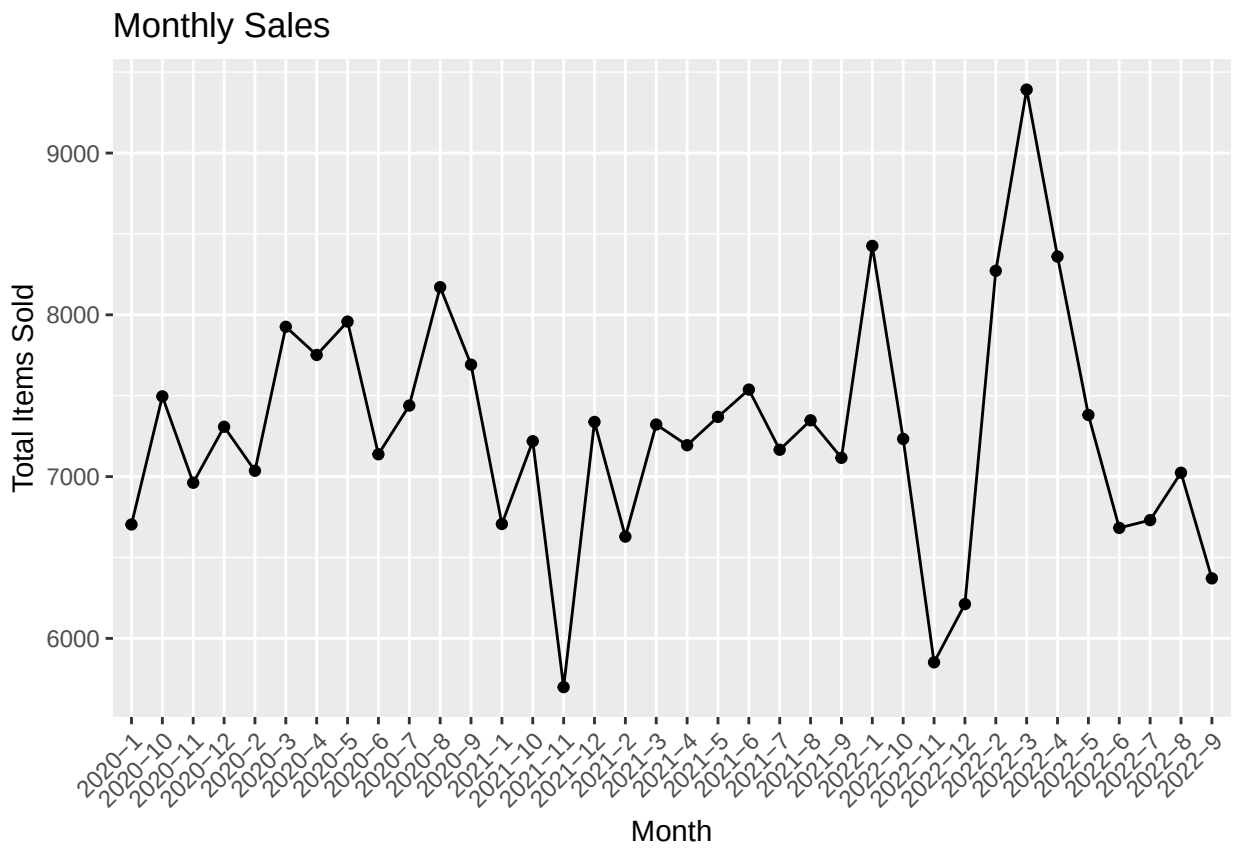

```
plot_quarterly_col <- ggplot(quarterly_sales, aes(x = Quarter_Year, y = total_items_sold)) +
  geom_col()
  labs(x = "Quarter", y = "Total Items Sold", title = "Quarterly Sales") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

```
## NULL
```

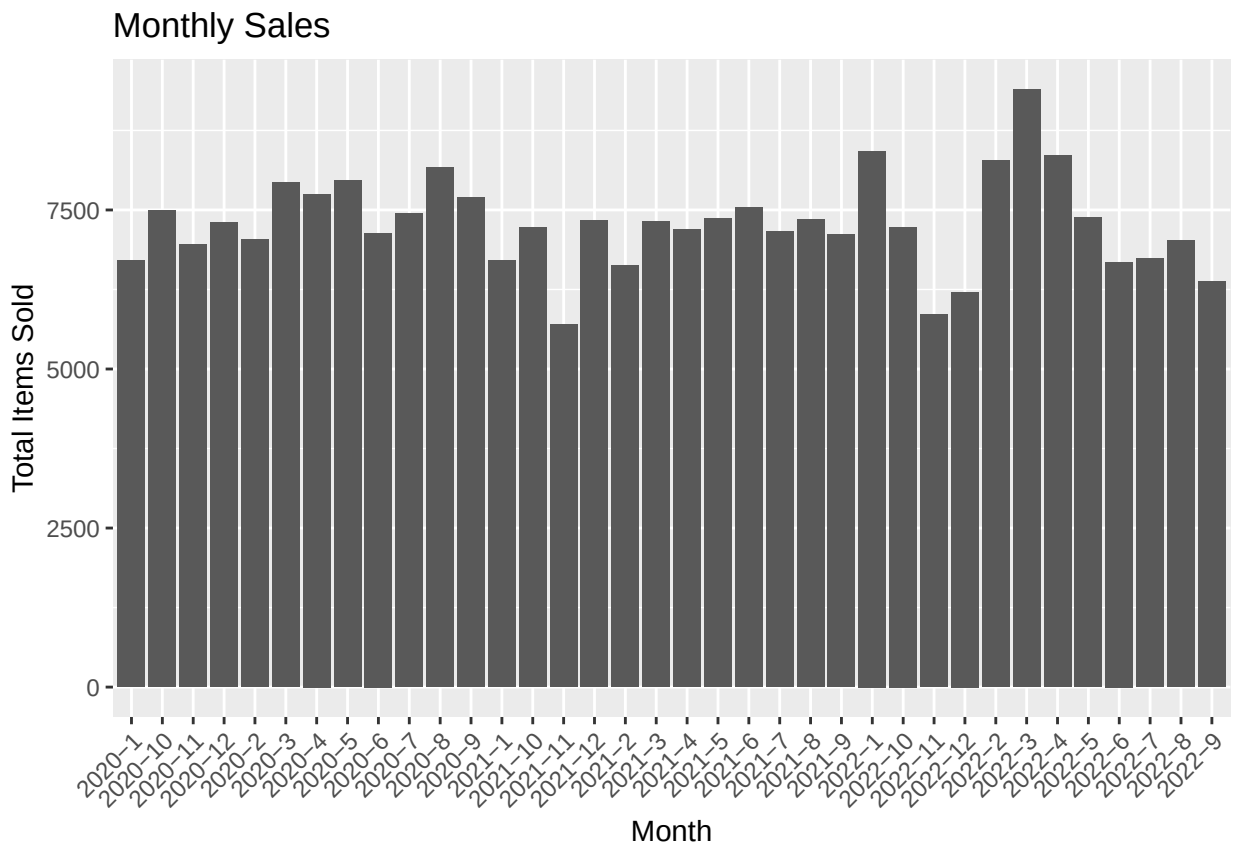
```
plot_quarterly_col
```



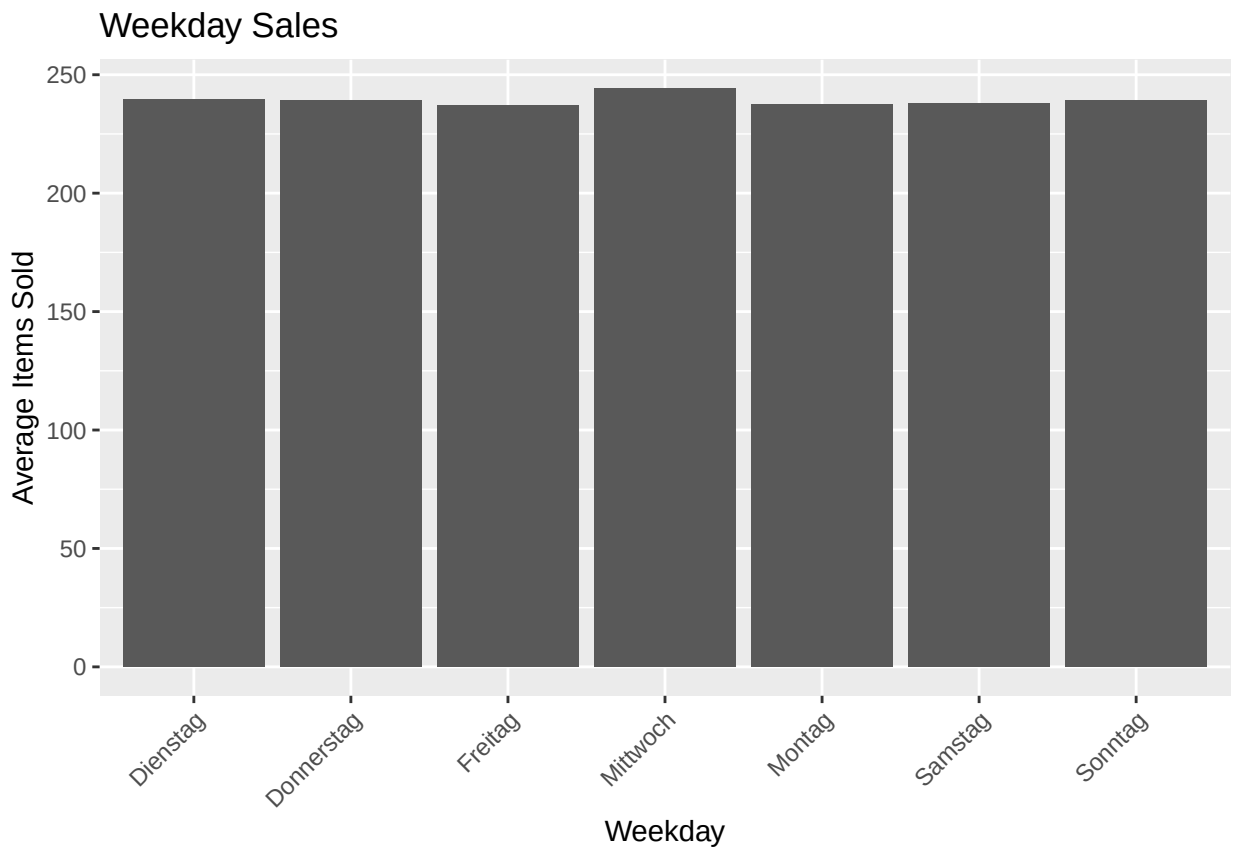

```
## There is some seasonal decrease in Q4, overall trend is decreasing

plot_monthly <- ggplot(monthly_sales, aes(x = Month_Year, y = total_items_sold)) +
  geom_line(group = 1) +
  geom_point() +
  labs(x = "Month", y = "Total Items Sold", title = "Monthly Sales") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

plot_monthly
```



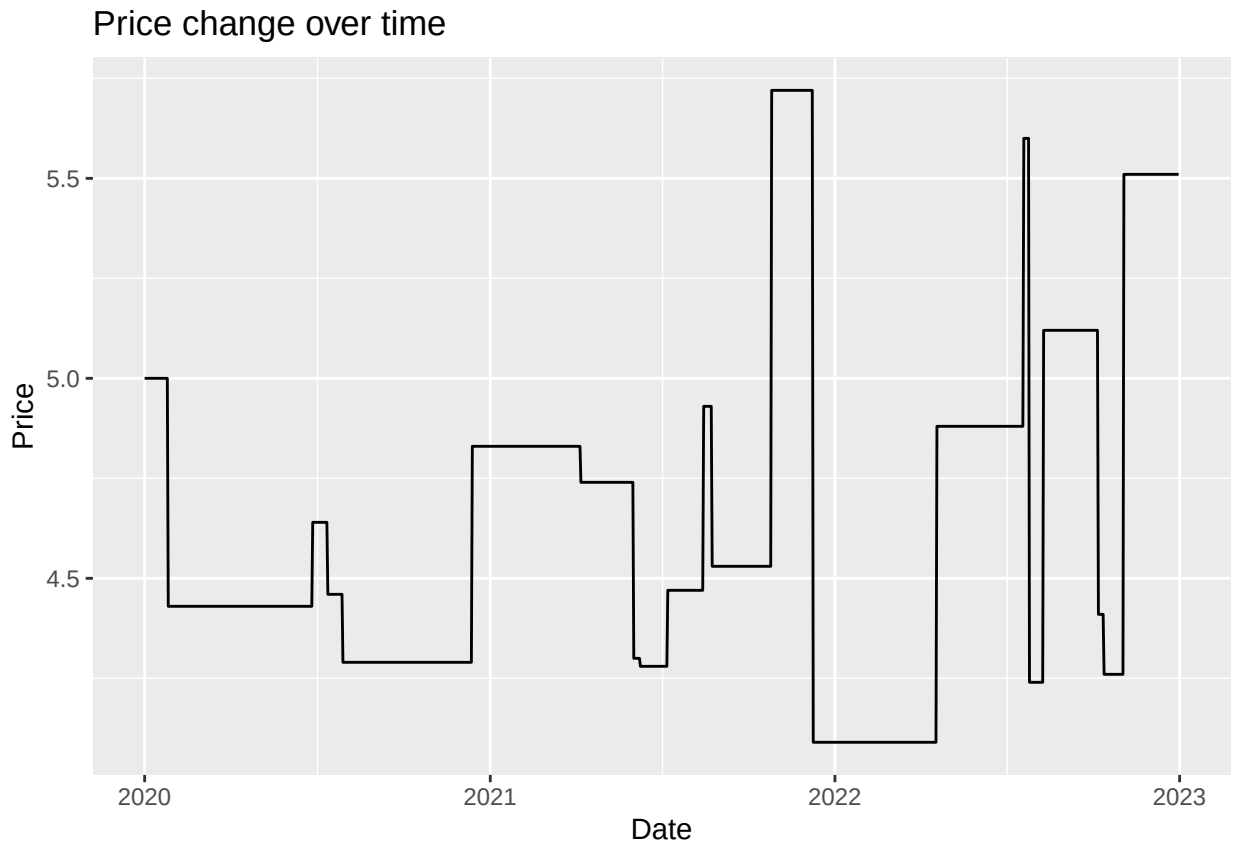

```
plot_monthly_col <- ggplot(monthly_sales, aes(x = Month_Year, y = total_items_sold)) +
  geom_col() +
  labs(x = "Month", y = "Total Items Sold", title = "Monthly Sales") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

plot_monthly_col
```




```
## Hard to see the decreasing trend here, but quarters suggest so

plot_weekday_col <- ggplot(weekday_sales, aes(x = weekday, y = total_items_sold)) +
  geom_col() +
  labs(x = "Weekday", y = "Average Items Sold", title = "Weekday Sales") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

plot_weekday_col
```

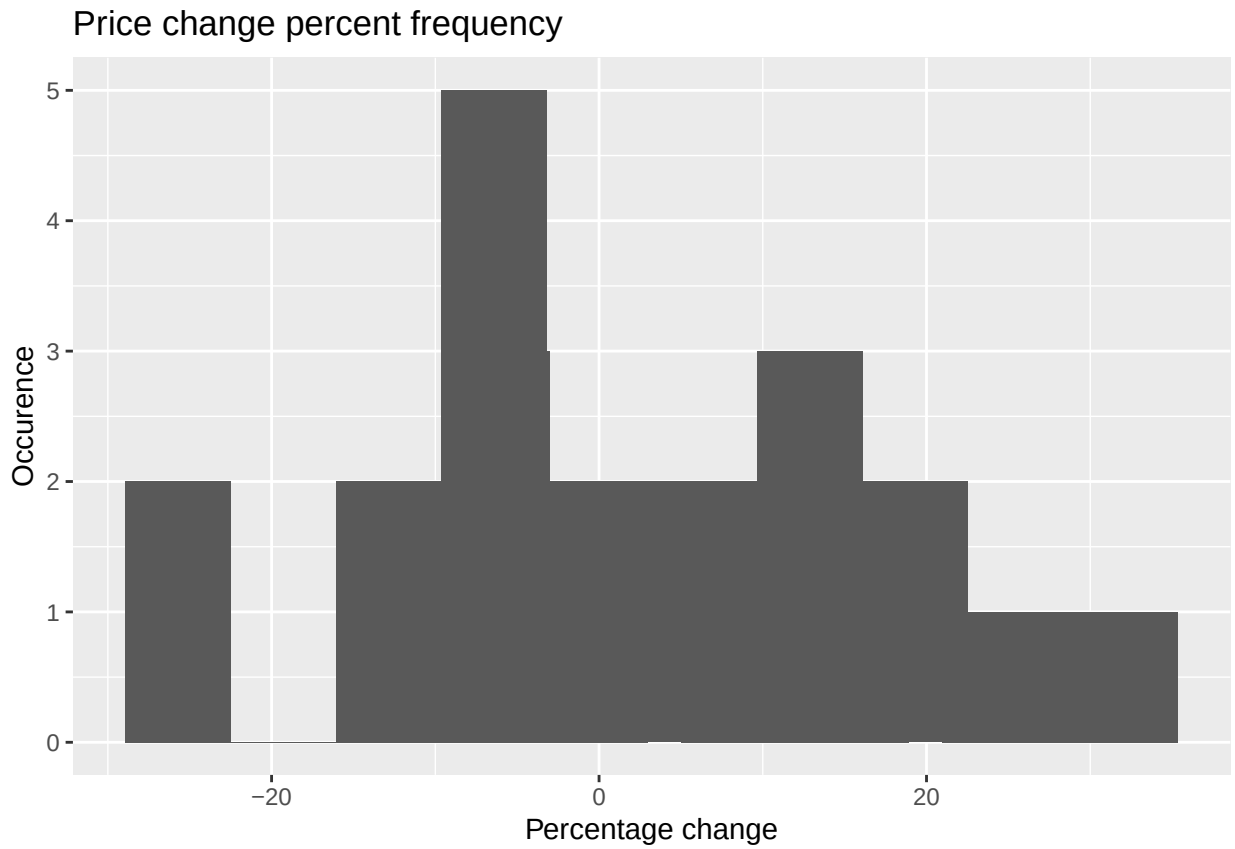



```
## Days seem to be quite equal, with Wednesdays being the peak days in terms of average sales volume

## Frequency of price changes

price_change_plot <- ggplot(dataset, aes(x = Date, y = price)) +
    geom_line() +
    labs(x = 'Date', y = 'Price', title = 'Price change over time')

price_change_plot
```




```
# Calculate percentage price change from the previous price
dataset <- dataset %>%
  arrange(Date) %>%
  mutate(price_change_pct = (price - lag(price)) / lag(price) * 100)

price_changes_only <- dataset %>%
    filter(price_change_pct != 0)

price_change_percent_hist <- ggplot(price_changes_only, aes(x = price_change_pct)) +
    geom_histogram() + stat_bin(bins = 10) +
    labs(x = 'Percentage change', y = 'Occurence', title = 'Price change percent frequency')

price_change_percent_hist
```

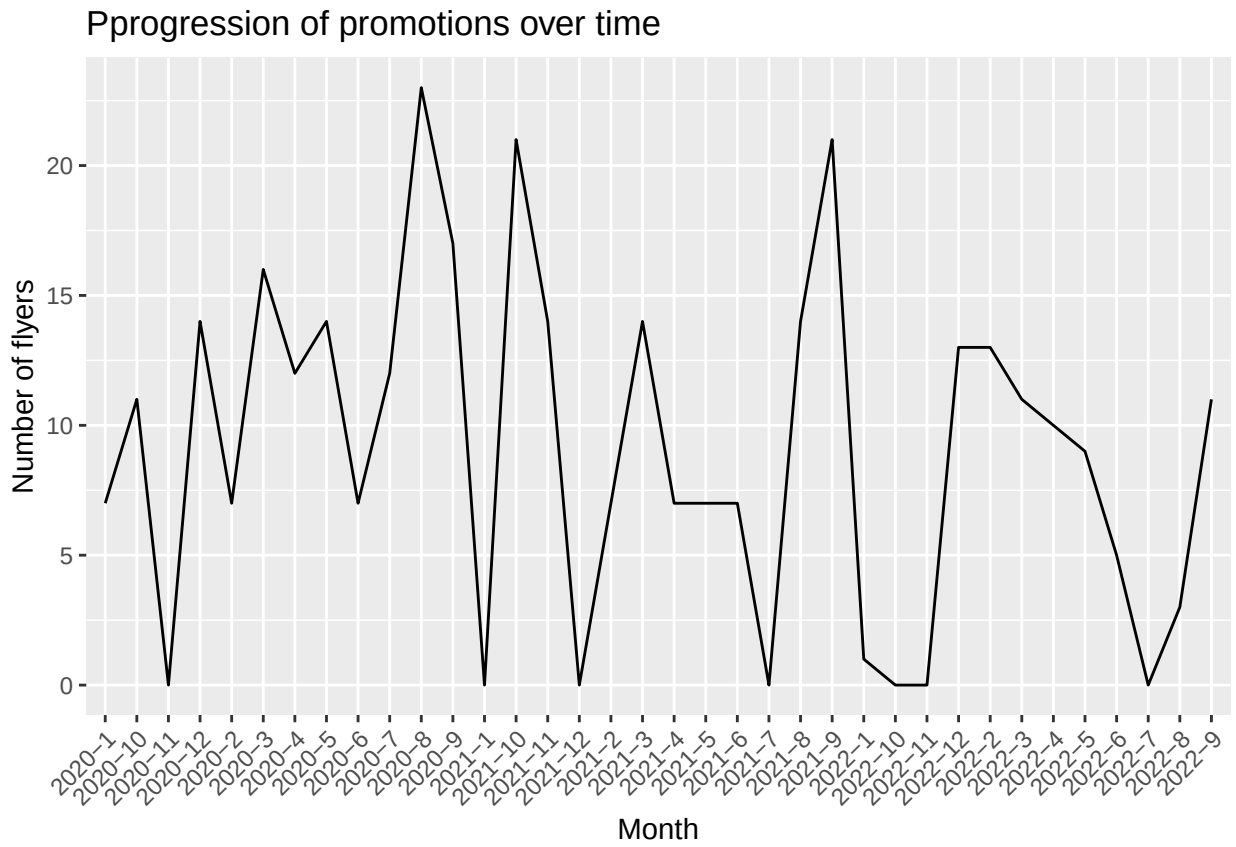
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
mean_price_change <- mean(abs(price_changes_only$price_change_pct))
## Prices change 12% on average

average_price_duration <- nrow(dataset) / nrow(price_changes_only)

## We assume that promotions progression = frequency of flyers at a time
flyer_days_per_month <- dataset %>%
    group_by(year = year(Date), month = month(Date)) %>%
    summarise(days_with_flyer = sum(flyer == 1)) %>%
    arrange(year, month)
```

```
## `summarise()` has grouped output by 'year'. You can override using the
## `.groups` argument.
```

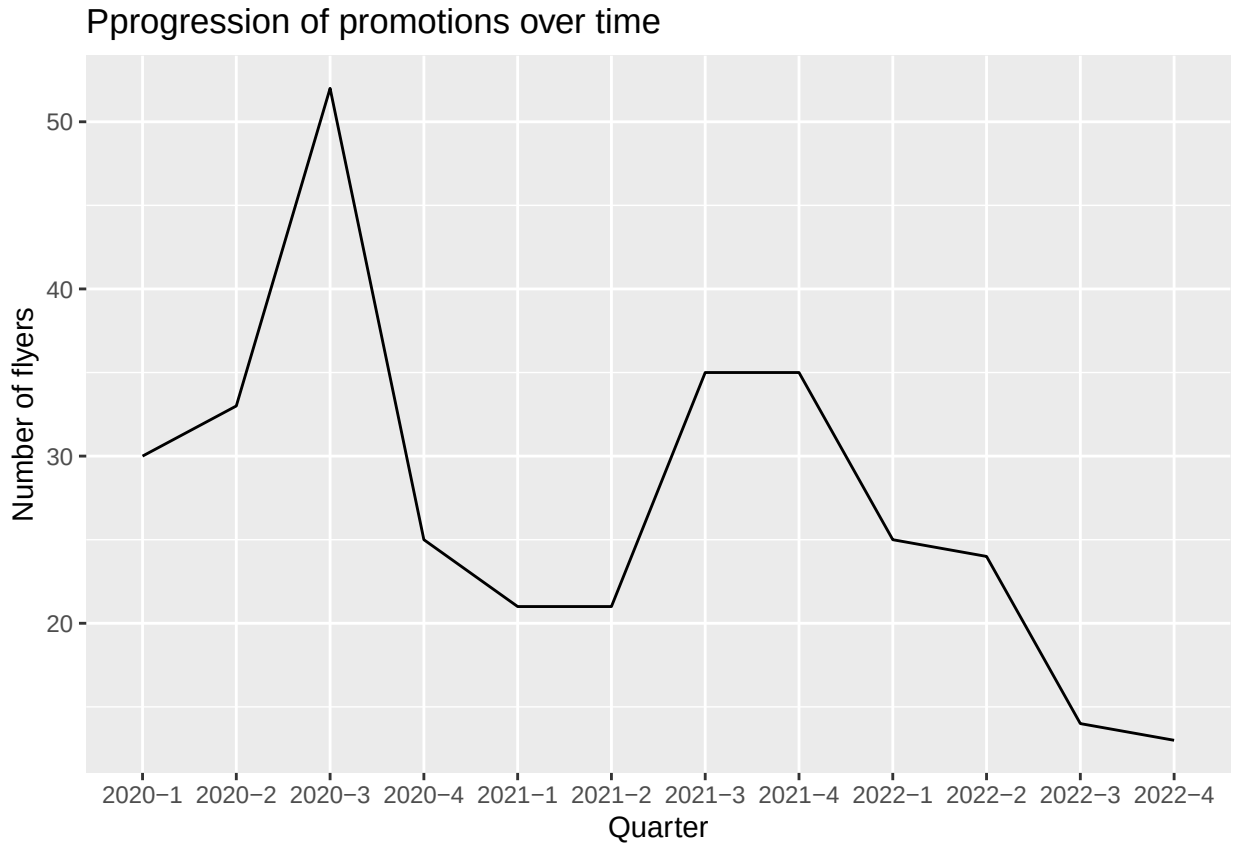
```
flyer_days_per_month_plot <- ggplot(flyer_days_per_month, aes(x = paste0(year, '-', month), y = days_wi
    geom_line() +
    labs(x = 'Month', y = 'Number of flyers', title = 'Pprogression of promotions over time') +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))

flyer_days_per_month_plot
```




```
## No pattern visible

flyer_days_per_quarter <- dataset %>%
    group_by(year = year(Date), quarter = quarter(Date)) %>%
    summarise(days_with_flyer = sum(flyer == 1)) %>%
    arrange(year, quarter)
```

```
## `summarise()` has grouped output by 'year'. You can override using the
## `.groups` argument.
```

```
flyer_days_per_quarter_plot <- ggplot(flyer_days_per_quarter, aes(x = paste0(year, '-', quarter), y = da
    geom_line() +
    labs(x = 'Quarter', y = 'Number of flyers', title = 'Pprogression of promotions over time')

flyer_days_per_quarter_plot
```

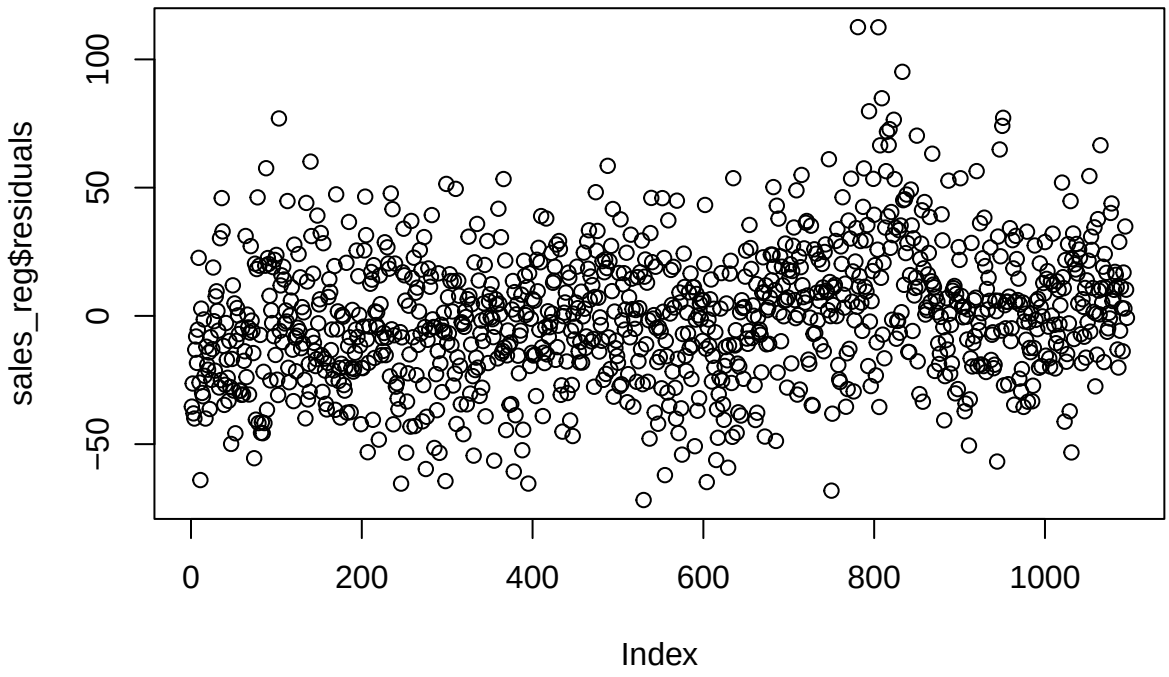



```
## A clear decreasing trend visible, with a slight increase in Q3 2021.

# Regression analysis
dataset <- read.csv("dataset.csv")
dataset$Date = parse_date(dataset$Date) #Convert Date to date format
dataset$flyer = as.factor(dataset$flyer) #Convert Date to date format

dataset_reg <- dataset %>%
    mutate(month = month(Date), quarter = quarter(Date)) %>%
    mutate(weekday = as.factor(weekday))

sales_reg = lm(items_sold ~ weekday + month + quarter + price + flyer, dataset_reg)
summary(sales_reg)
```

```
##
## Call:
## lm(formula = items_sold ~ weekday + month + quarter + price +
##     flyer, data = dataset_reg)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -71.79 -17.36  -0.17  16.20 112.58
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       470.86795    8.88780  52.979   <2e-16 ***
```

```
## weekdayDonnerstag   0.04289    2.94546   0.015    0.988
## weekdayFreitag     -2.31289    2.94631  -0.785    0.433
## weekdayMittwoch     5.04479    2.94524   1.713    0.087 .
## weekdayMontag      -2.29155    2.95011  -0.777    0.437
## weekdaySamstag     -1.32119    2.94646  -0.448    0.654
## weekdaySonntag     -0.16073    2.95066  -0.054    0.957
## month              -0.27683    0.97334  -0.284    0.776
## quarter            -4.34671    2.99458  -1.452    0.147
## price             -49.18354    1.89638 -25.935   <2e-16 ***
## flyer1             30.93849    1.73597  17.822   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 26.05 on 1085 degrees of freedom
## Multiple R-squared:  0.5344, Adjusted R-squared:  0.5301
## F-statistic: 124.5 on 10 and 1085 DF,  p-value: < 2.2e-16
```

```
## The model has an "ok" R^2 of .5344. And a very low p-value close to 0.00
plot(sales_reg$residuals)
```




```
sd(sales_reg$residuals)
```

```
## [1] 25.93334
```

```r
## Most of variables are insignificant, hence we will start removing them
## Factors for weekdays are largely insignificant. We can leave wednesday as it has the only acceptable

sales_reg2 = lm(items_sold ~ as.factor(weekday == 'Mittwoch') + month + quarter + price + flyer, dataset
summary(sales_reg2)
```

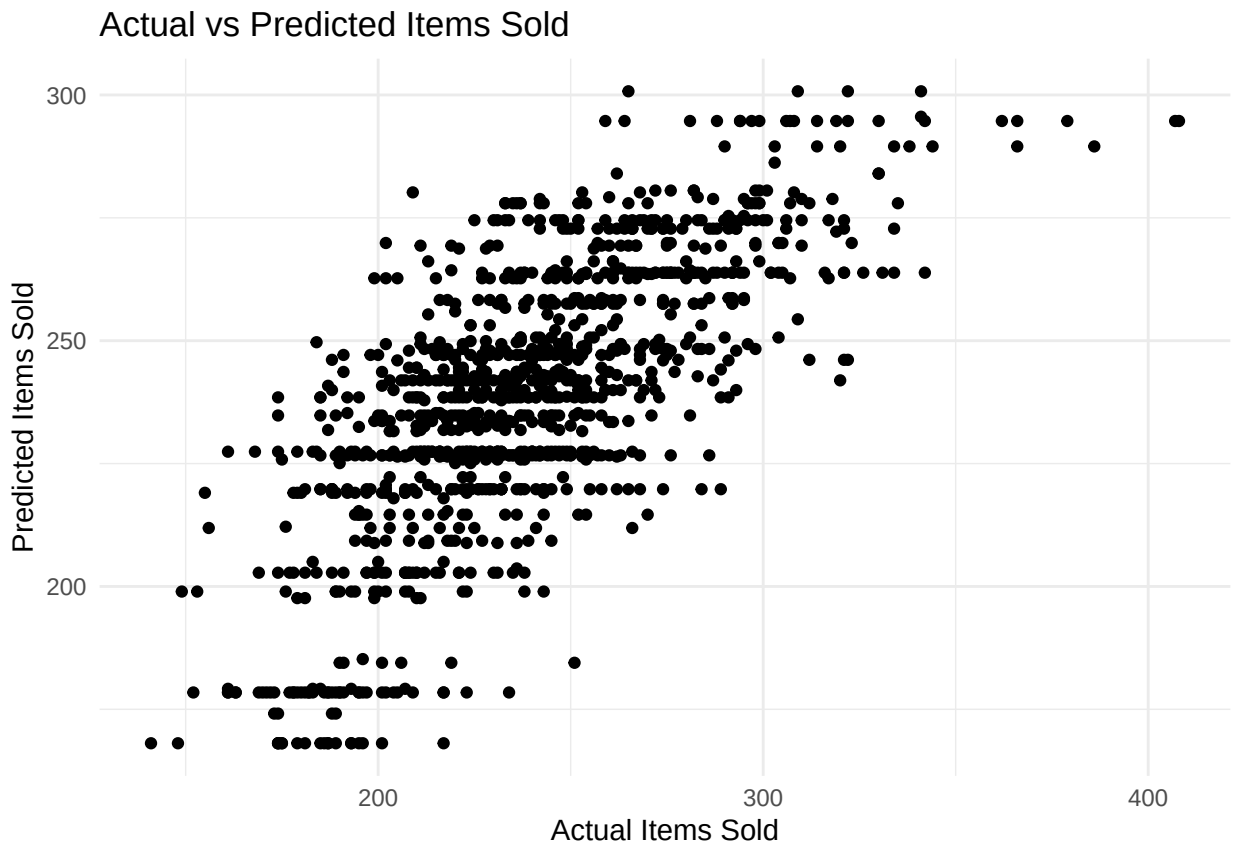
```
##
## Call:
## lm(formula = items_sold ~ as.factor(weekday == "Mittwoch") +
##     month + quarter + price + flyer, data = dataset_reg)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -70.95 -17.38  -0.50  16.27 113.53
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                        469.8737     8.6782  54.144  < 2e-16 ***
## as.factor(weekday == "Mittwoch")TRUE   6.0519     2.2430   2.698  0.00708 **
## month                               -0.2580     0.9712  -0.266  0.79053
## quarter                             -4.4043     2.9879  -1.474  0.14076
## price                              -49.1813     1.8933 -25.977  < 2e-16 ***
## flyer1                              30.9311     1.7331  17.847  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 26.01 on 1090 degrees of freedom
## Multiple R-squared:  0.5338, Adjusted R-squared:  0.5317
## F-statistic: 249.6 on 5 and 1090 DF,  p-value: < 2.2e-16
```

```r
## The model has improved. We have a very similar performance in terms of R^2
## We can proceed with variable removal. Let's start with month, as it is also insignificant.

sales_reg3 = lm(items_sold ~ as.factor(weekday == 'Mittwoch') + quarter + price + flyer, dataset_reg)
summary(sales_reg3)
```

```
##
## Call:
## lm(formula = items_sold ~ as.factor(weekday == "Mittwoch") +
##     quarter + price + flyer, data = dataset_reg)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -71.179 -17.418  -0.544  16.282 113.296
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                        470.3129     8.5156  55.229  < 2e-16 ***
## as.factor(weekday == "Mittwoch")TRUE   6.0417     2.2418   2.695  0.00715 **
## quarter                             -5.1741     0.7299  -7.089 2.43e-12 ***
## price                              -49.2181     1.8874 -26.077  < 2e-16 ***
## flyer1                              30.8673     1.7157  17.991  < 2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 26 on 1091 degrees of freedom
## Multiple R-squared:  0.5338, Adjusted R-squared:  0.5321
## F-statistic: 312.3 on 4 and 1091 DF,  p-value: < 2.2e-16
```

```
## Now all variables are significant, the R^2 is still high at .5338
## Model's p-value is low and acceptable

dataset_reg$regression_result <- predict(sales_reg3, dataset_reg)

# Plot the actual items_sold on x and regression_result on y as points with different colors
plot_actual_vs_predicted <- ggplot(dataset_reg, aes(x = items_sold, y = regression_result)) +
  geom_point() +
  labs(x = "Actual Items Sold", y = "Predicted Items Sold", title = "Actual vs Predicted Items Sold") +
  theme_minimal()

plot_actual_vs_predicted
```

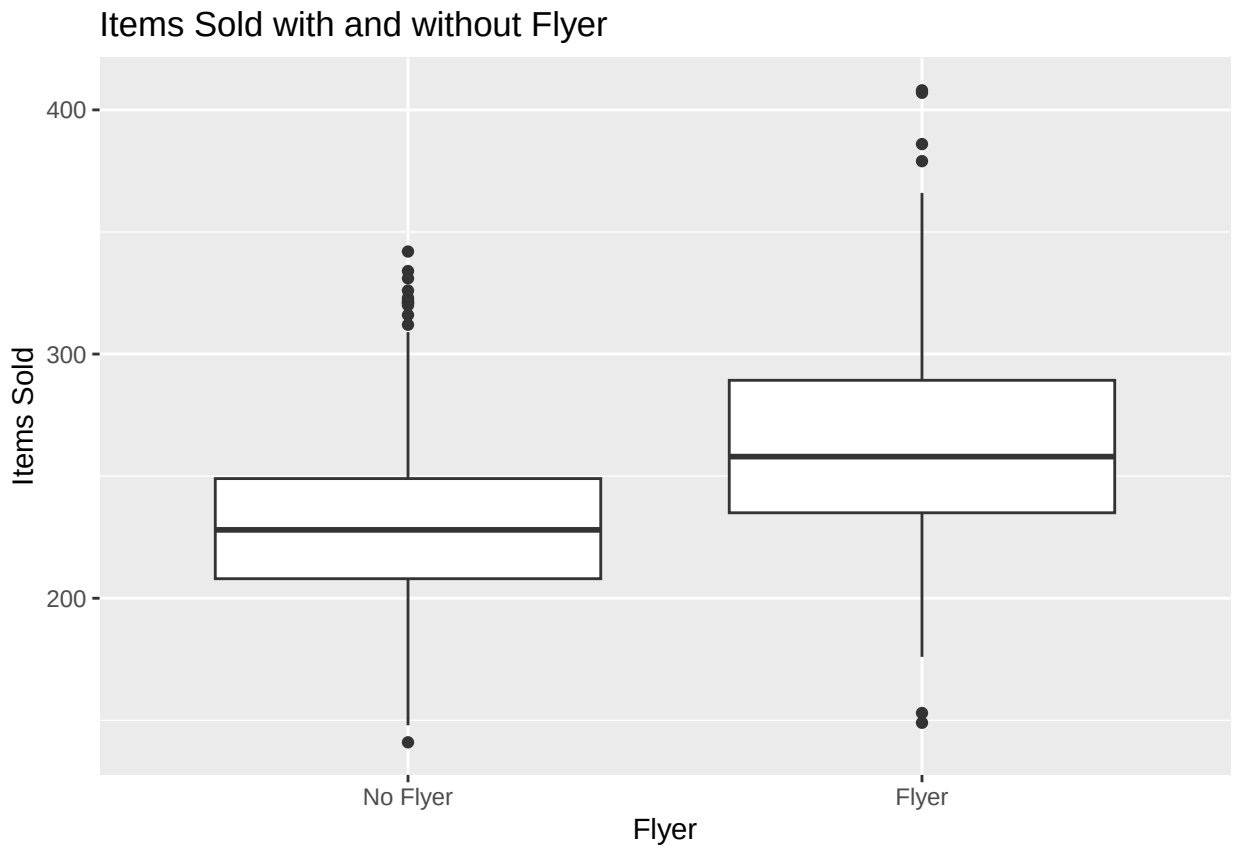



```
# Group comparisons
t.test(dataset[dataset$flyer==1,]$items_sold,
    dataset[dataset$flyer==0,]$items_sold)
```

```
##
##  Welch Two Sample t-test
```

```
##
## data:  dataset[dataset$flyer == 1, ]$items_sold and dataset[dataset$flyer == 0, ]$items_sold
## t = 12.76, df = 522.38, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  27.05966 36.90814
## sample estimates:
## mean of x mean of y
##  261.6128  229.6289
```

```
## The t-test rejects the null hypothesis with p-value < 0.05
## meaning that we can assume that there is a difference in sample means
boxplot_flyer <- ggplot(dataset, aes(x = as.factor(flyer), y = items_sold)) +
  geom_boxplot() +
  labs(x = "Flyer", y = "Items Sold", title = "Items Sold with and without Flyer") +
  scale_x_discrete(labels = c("0" = "No Flyer", "1" = "Flyer"))

boxplot_flyer
```




```
## anova test
anova_test <- aov(items_sold ~ weekday, data = dataset)
summary(anova_test)
```

```
##               Df  Sum Sq Mean Sq F value Pr(>F)
## weekday        6    5343   890.4   0.615  0.718
## Residuals   1089 1576401  1447.6
```

```r
## p-value > 0.05, hence we reject the null hypothesis
boxplot_weekday <- ggplot(dataset, aes(x = as.factor(weekdays(Date)), y = items_sold)) +
  geom_boxplot() +
  labs(x = "Weekday", y = "Items Sold", title = "Items Sold on Weekdays") +
  scale_x_discrete()

boxplot_weekday
```