```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

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

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

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

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

```
library(ggplot2)
library(tidyr)

# Read the CSV file
data <- read.csv("trashwheel.csv")
data$Date <- as.Date(data$Date, format = "%m/%d/%Y")
```

```
# Filter data for the last 6 months
recent_data <- data %>%
  filter(Date > max(Date) %m-% months(6))
```

```
sums_recent <- recent_data %>%
  select(-Year) %>%
  select(-Date) %>%
  summarise(across(where(is.numeric), sum, na.rm = TRUE))
```

```
## Warning: There was 1 warning in `summarise()`.
## ℹ In argument: `across(where(is.numeric), sum, na.rm = TRUE)`.
## Caused by warning:
## ! The `...` argument of `across()` is deprecated as of dplyr 1.1.0.
## Supply arguments directly to `.fns` through an anonymous function instead.
##
##   # Previously
##   across(a:b, mean, na.rm = TRUE)
##
##   # Now
##   across(a:b, \(x) mean(x, na.rm = TRUE))
```

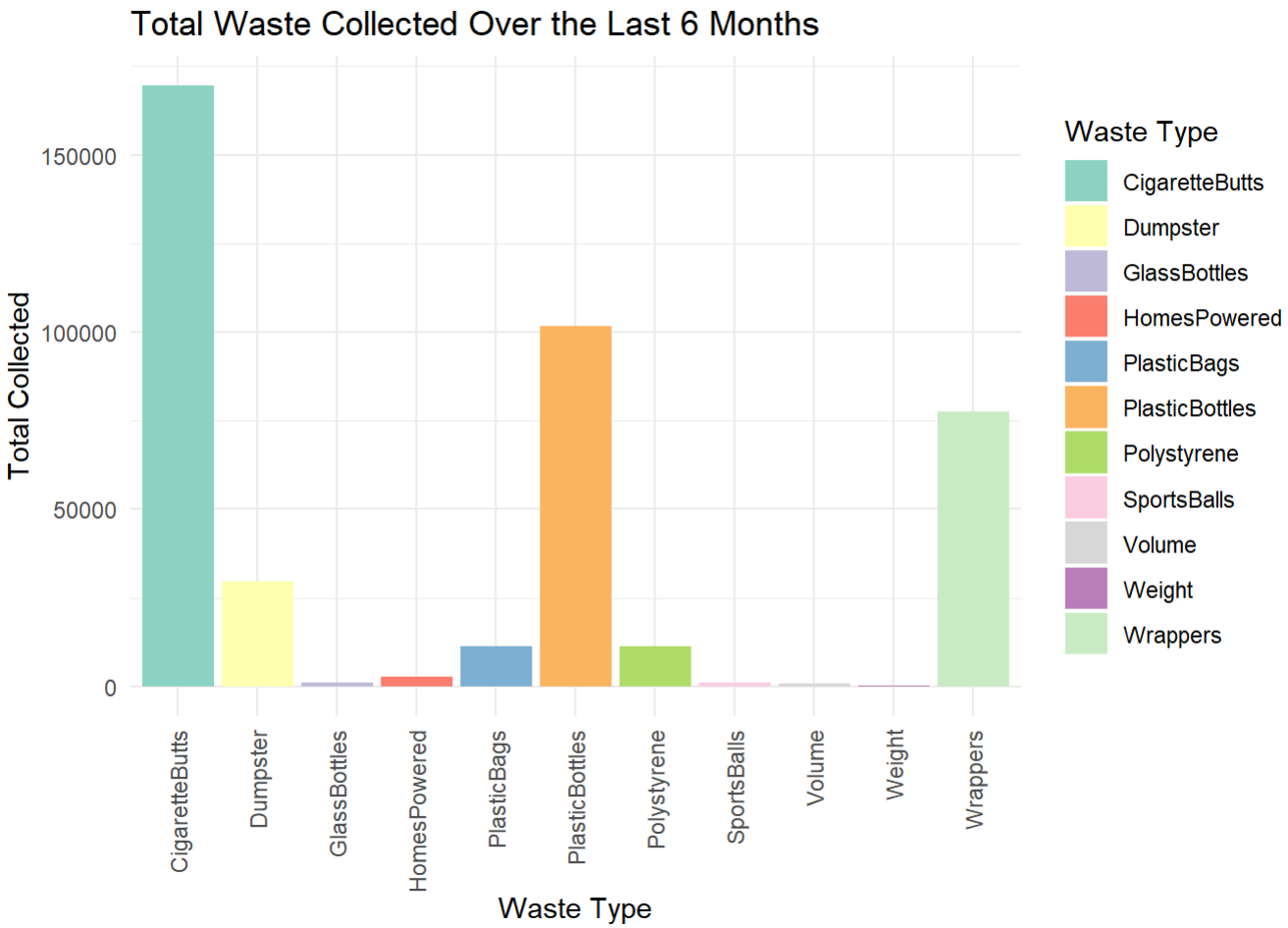
```
head(sums_recent)
```

```
##   Dumpster Weight Volume PlasticBottles Polystyrene CigaretteButts GlassBottles
## 1    29645 157.97    710         101710       11453         169480          983
##   PlasticBags Wrappers SportsBalls HomesPowered
## 1       11298    77670         964         2638
```

```
sums_recent <- pivot_longer(sums_recent, cols = everything(), names_to = "Waste_Type", values_to
= "Total")

print(sums_recent)
```

```
## # A tibble: 11 × 2
##    Waste_Type       Total
##    <chr>            <dbl>
##  1 Dumpster         29645
##  2 Weight             158.
##  3 Volume             710
##  4 PlasticBottles  101710
##  5 Polystyrene      11453
##  6 CigaretteButts  169480
##  7 GlassBottles       983
##  8 PlasticBags      11298
##  9 Wrappers         77670
## 10 SportsBalls        964
## 11 HomesPowered      2638
```

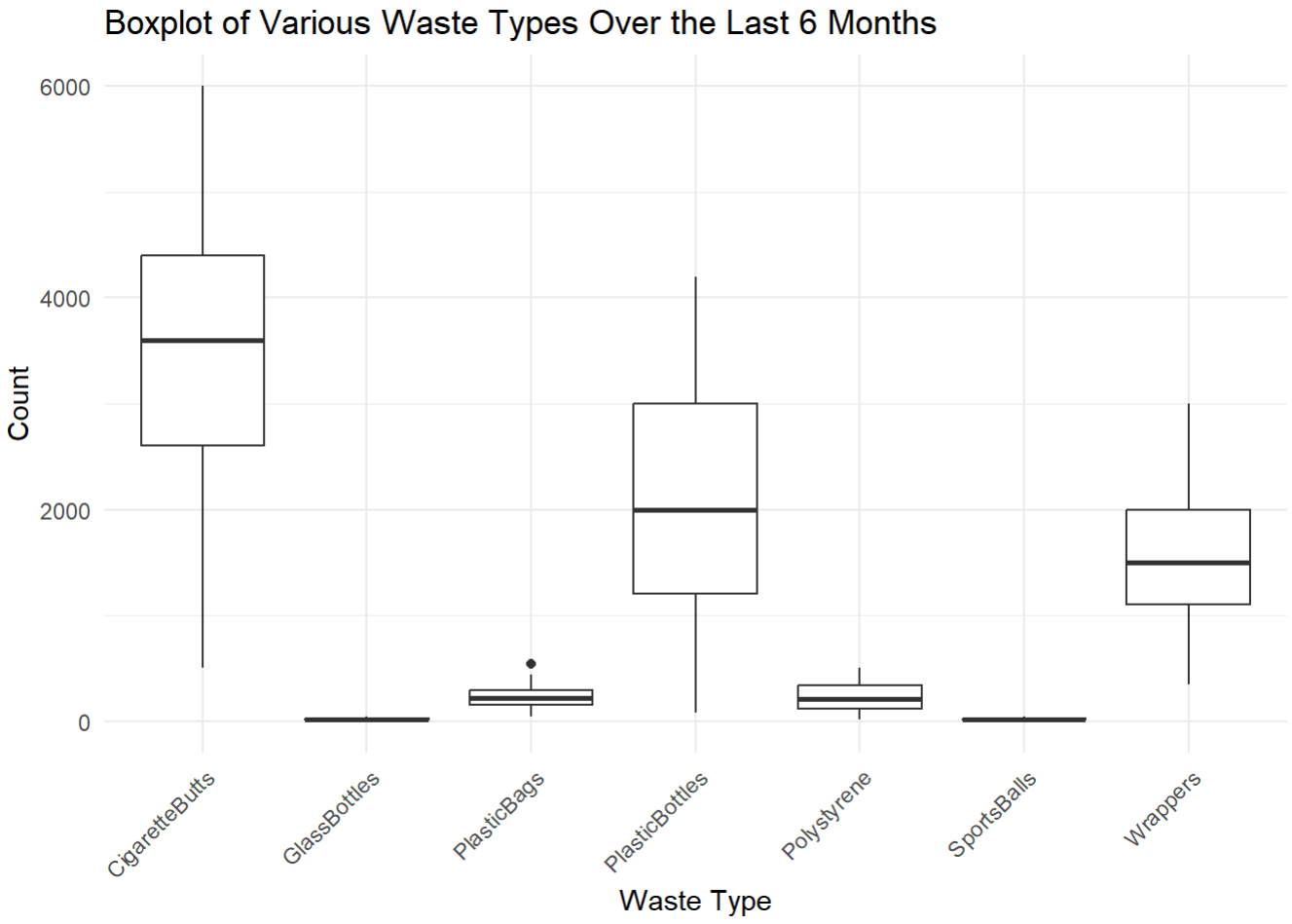
```
ggplot(sums_recent, aes(x = Waste_Type, y = Total, fill = Waste_Type)) +
  geom_bar(stat = "identity") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) +
  labs(title = "Total Waste Collected Over the Last 6 Months",
       x = "Waste Type",
       y = "Total Collected",
       fill = "Waste Type") +
  scale_fill_brewer(palette = "Set3")
```




```
recent_data_long <- recent_data %>%
  select(Date, PlasticBottles, Polystyrene, CigaretteButts, GlassBottles,
         PlasticBags, Wrappers, SportsBalls) %>%
  gather(key = "Waste_Type", value = "Count", -Date)

head(recent_data_long)
```

```
##         Date     Waste_Type Count
## 1 2023-06-28 PlasticBottles  3400
## 2 2023-06-28 PlasticBottles  4000
## 3 2023-06-28 PlasticBottles  2100
## 4 2023-06-29 PlasticBottles  1900
## 5 2023-07-03 PlasticBottles  2300
## 6 2023-07-05 PlasticBottles  3200
```

```
ggplot(recent_data_long, aes(x = Waste_Type, y = Count)) +
  geom_boxplot() +
  theme_minimal() +
  labs(title = "Boxplot of Various Waste Types Over the Last 6 Months",
       x = "Waste Type",
       y = "Count") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



```
summary_stats <- recent_data_long %>%
  group_by(Waste_Type) %>%
  summarise(Median_Count = median(Count),
            IQR = IQR(Count),
            Lower_Whisker = quantile(Count, 0.25) - 1.5 * IQR(Count),
            Upper_Whisker = quantile(Count, 0.75) + 1.5 * IQR(Count),
            Mean_Count = mean(Count))

# Print the summary statistics
print(summary_stats)
```

```
## # A tibble: 7 × 6
##   Waste_Type     Median_Count   IQR Lower_Whisker Upper_Whisker Mean_Count
##   <chr>                 <int> <dbl>         <dbl>         <dbl>      <dbl>
## 1 CigaretteButts         3600  1800        -100          7100       3459.
## 2 GlassBottles             20    15         -10.5          49.5       20.1
## 3 PlasticBags             220   140         -60            500        231.
## 4 PlasticBottles         2000  1800       -1500          5700       2076.
## 5 Polystyrene             210   220        -210            670        234.
## 6 SportsBalls              20    15         -10.5          49.5       19.7
## 7 Wrappers               1500   900        -250          3350       1585.
```

```
recent_data_long$Date_numeric <- as.numeric(recent_data_long$Date - min(recent_data_long$Date))

# List of waste types to model
waste_types <- c("Polystyrene", "CigaretteButts", "GlassBottles", "PlasticBags", "Wrappers", "Sp
ortsBalls")

# Initialize an empty list to store models
models <- list()

# Loop over waste types, fit model, and store
for(waste in waste_types) {
  model <- lm(Count ~ Date_numeric, data = filter(recent_data_long, Waste_Type == waste))
  models[[waste]] <- model
}

# Display summaries of all models
lapply(models, summary)
```

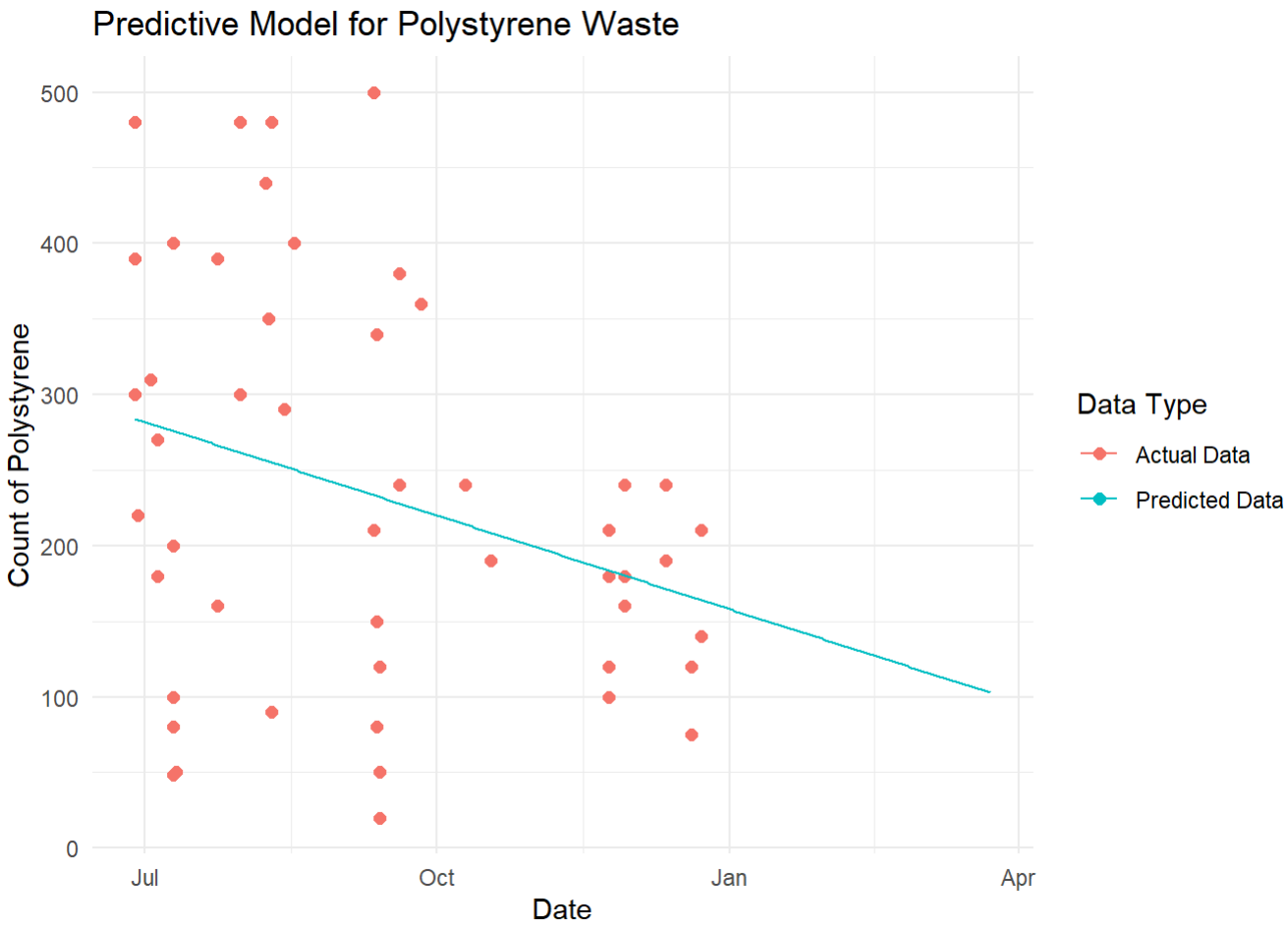
```
## $Polystyrene
##
## Call:
## lm(formula = Count ~ Date_numeric, data = filter(recent_data_long,
##     Waste_Type == waste))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -227.947  -83.583   -0.212   94.278  266.527
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  284.0374    29.0047   9.793 6.27e-13 ***
## Date_numeric  -0.6742     0.3032  -2.223    0.031 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 127.1 on 47 degrees of freedom
## Multiple R-squared:  0.09517,    Adjusted R-squared:  0.07592
## F-statistic: 4.944 on 1 and 47 DF,  p-value: 0.03103
##
##
## $CigaretteButts
##
## Call:
## lm(formula = Count ~ Date_numeric, data = filter(recent_data_long,
##     Waste_Type == waste))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2951.2  -542.4   167.7   867.1  2438.4
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3693.931    305.549  12.089 4.98e-16 ***
## Date_numeric   -3.152      3.194  -0.987    0.329
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1338 on 47 degrees of freedom
## Multiple R-squared:  0.02029,    Adjusted R-squared:  -0.000552
## F-statistic: 0.9735 on 1 and 47 DF,  p-value: 0.3289
##
##
## $GlassBottles
##
## Call:
## lm(formula = Count ~ Date_numeric, data = filter(recent_data_long,
##     Waste_Type == waste))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -22.055  -7.985  -0.017   6.467  23.155
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  22.43726    2.39036   9.387 2.37e-12 ***
## Date_numeric -0.03185    0.02499  -1.274    0.209
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.47 on 47 degrees of freedom
## Multiple R-squared:  0.0334, Adjusted R-squared:  0.01283
## F-statistic: 1.624 on 1 and 47 DF,  p-value: 0.2088
##
##
## $PlasticBags
##
## Call:
## lm(formula = Count ~ Date_numeric, data = filter(recent_data_long,
##     Waste_Type == waste))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -203.165  -66.105   -4.461   55.539  309.556
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  255.1115    24.5069  10.410 8.62e-14 ***
## Date_numeric  -0.3289     0.2562  -1.284    0.206
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 107.3 on 47 degrees of freedom
## Multiple R-squared:  0.03388,    Adjusted R-squared:  0.01332
## F-statistic: 1.648 on 1 and 47 DF,  p-value: 0.2055
##
##
## $Wrappers
##
## Call:
## lm(formula = Count ~ Date_numeric, data = filter(recent_data_long,
##     Waste_Type == waste))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1231.69  -473.20   -41.56   421.26  1375.42
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1691.788    145.247  11.648 1.86e-15 ***
## Date_numeric   -1.430      1.518  -0.942    0.351
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

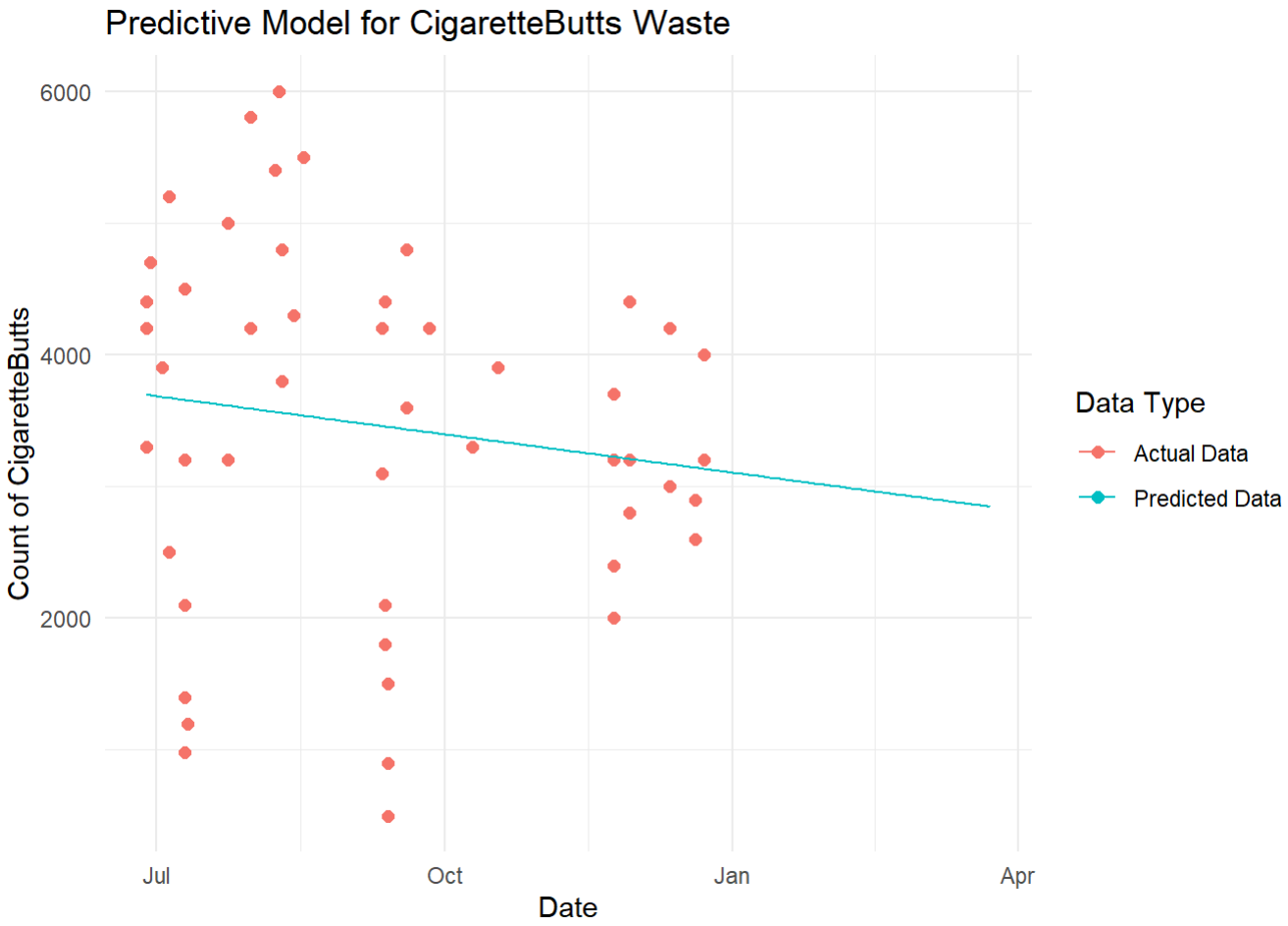
```
##
## Residual standard error: 636.2 on 47 degrees of freedom
## Multiple R-squared:  0.01852,    Adjusted R-squared:  -0.002365
## F-statistic: 0.8867 on 1 and 47 DF,  p-value: 0.3512
##
##
## $SportsBalls
##
## Call:
## lm(formula = Count ~ Date_numeric, data = filter(recent_data_long,
##     Waste_Type == waste))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.6224  -7.1185   0.0232   5.9884  21.6509
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  21.26811    2.37402   8.959 9.84e-12 ***
## Date_numeric -0.02137    0.02482  -0.861    0.394
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.4 on 47 degrees of freedom
## Multiple R-squared:  0.01553,    Adjusted R-squared:  -0.005413
## F-statistic: 0.7416 on 1 and 47 DF,  p-value: 0.3935
```

```
# Function to create prediction plots for a given waste type
create_prediction_plot <- function(waste) {
  # Extending the prediction range to include the next 3 months
  max_date_numeric <- max(recent_data_long$Date_numeric)
  future_extend <- as.numeric((max(recent_data_long$Date) %m+% months(3)) - max(recent_data_long
$Date))
  prediction_data <- data.frame(Date_numeric = seq(min(recent_data_long$Date_numeric), max_date_
numeric + future_extend, by = 1))
  prediction_data$Predicted_Count <- predict(models[[waste]], newdata = prediction_data)
  prediction_data$Date <- min(recent_data_long$Date) + prediction_data$Date_numeric * days(1)

  # Plotting
  p <- ggplot(filter(recent_data_long, Waste_Type == waste), aes(x = Date, y = Count)) +
    geom_point(aes(color = "Actual Data"), size = 2) +
    geom_line(data = prediction_data, aes(x = Date, y = Predicted_Count, color = "Predicted Dat
a")) +
    theme_minimal() +
    labs(title = paste("Predictive Model for", waste, "Waste"),
         x = "Date",
         y = paste("Count of", waste),
         color = "Data Type")

  return(p)
}
```

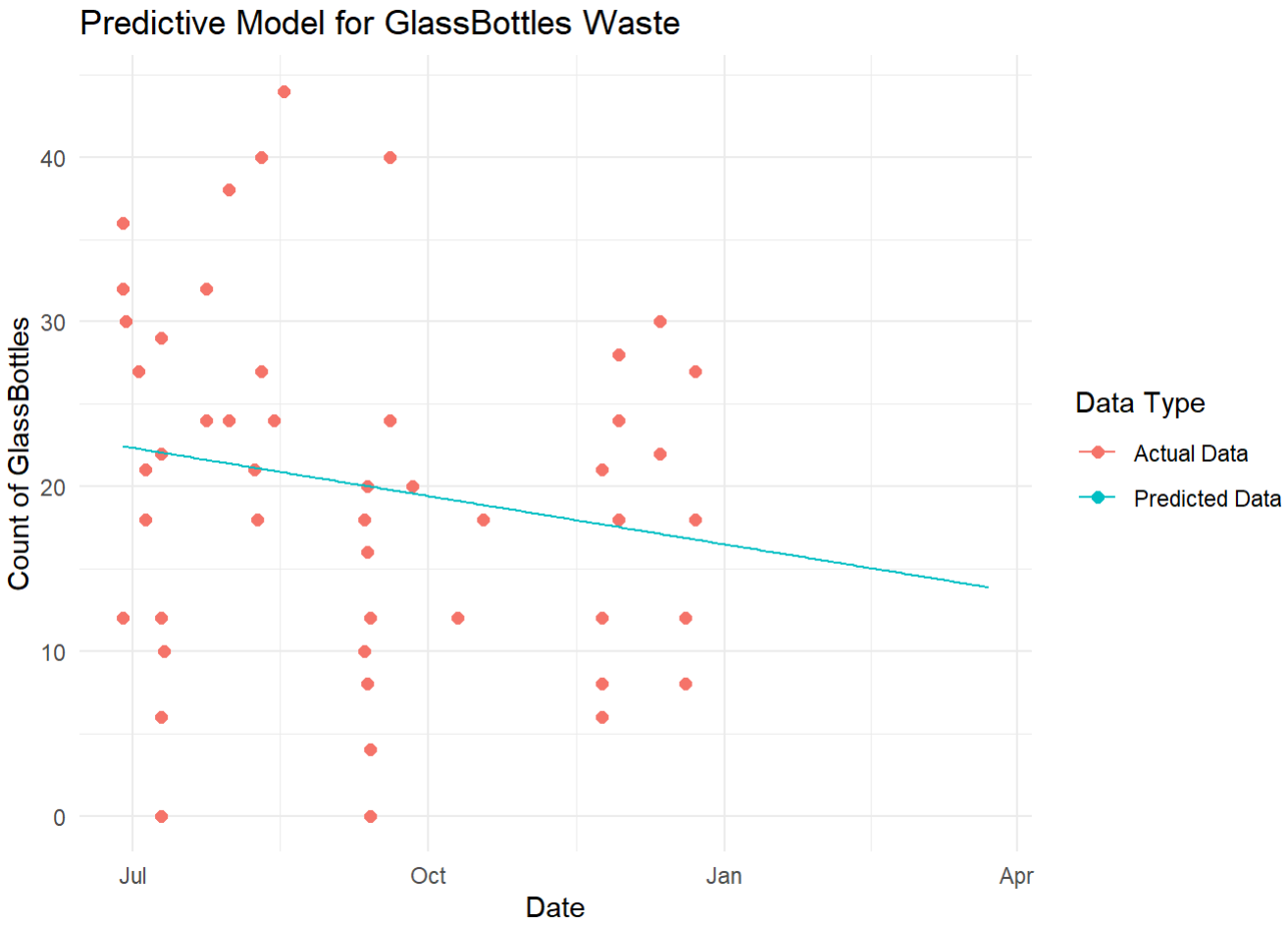
```
# Use the function to create plots
plot_list <- lapply(waste_types, create_prediction_plot)

plot_list[[1]]
```



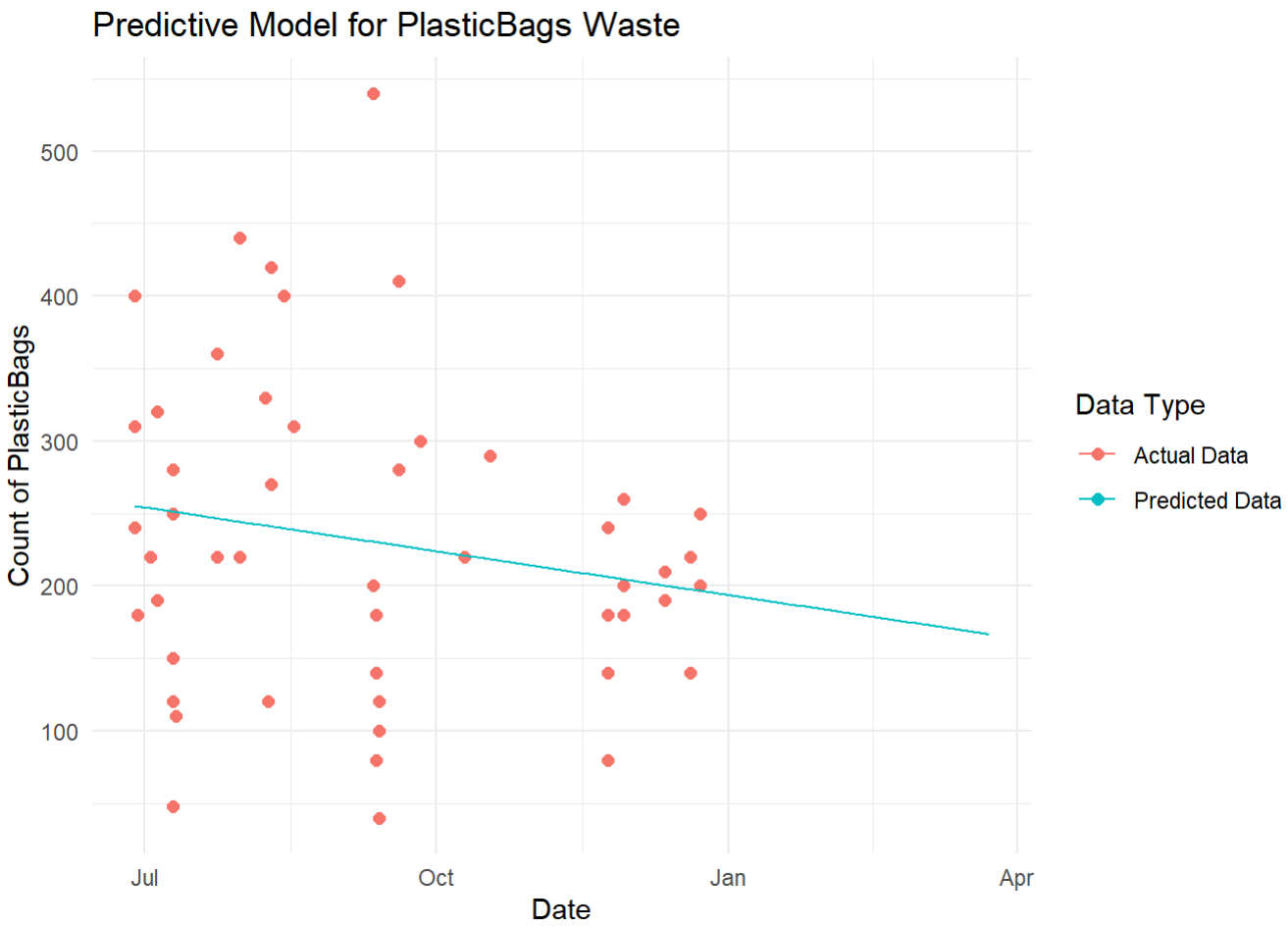

```
plot_list[[2]]
```


Predictive Model for CigaretteButts Waste
6000
4000
2000
Count of CigaretteButts
Jul
Oct
Jan
Apr
Date
Data Type
Actual Data
Predicted Data

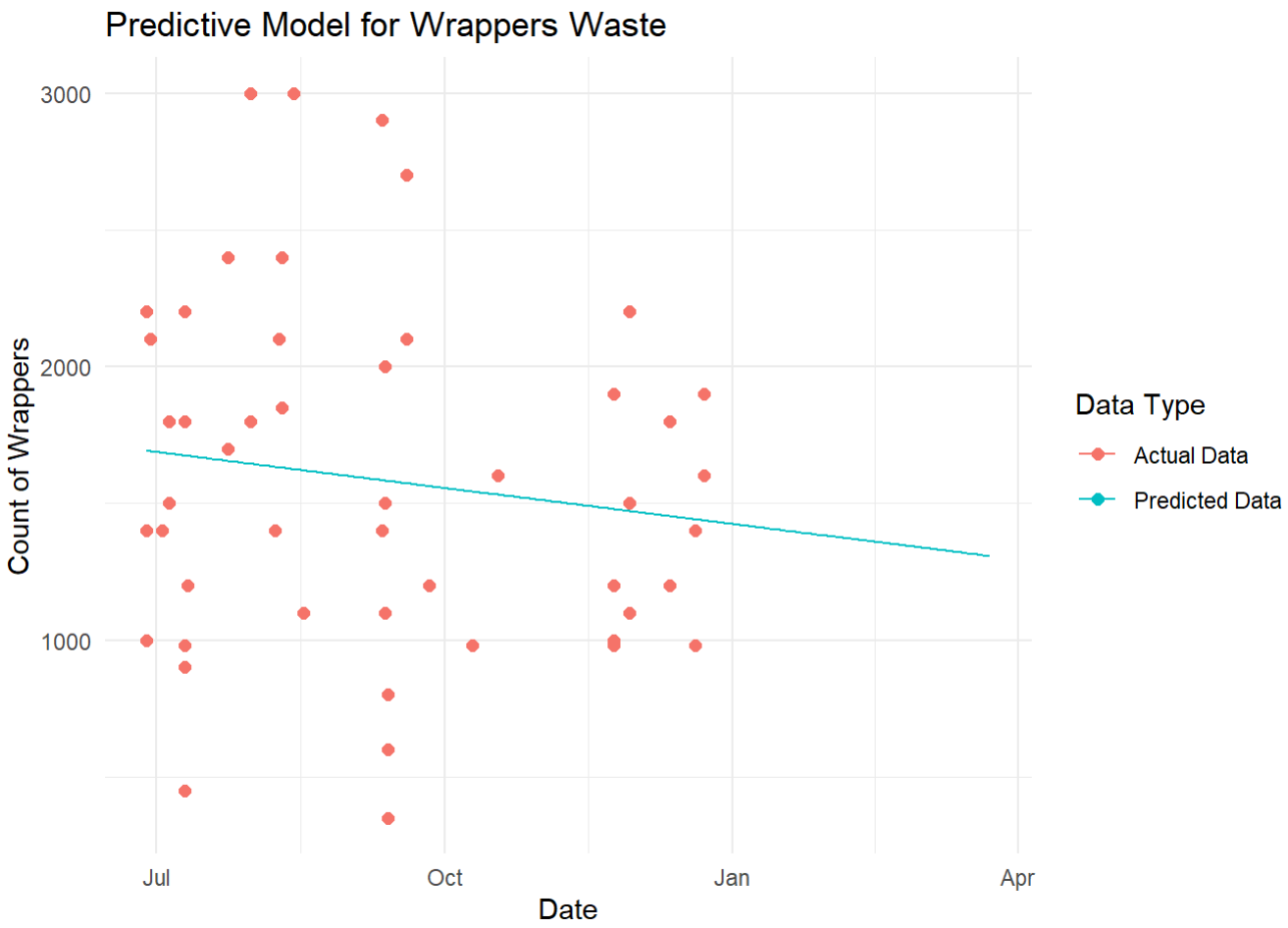
```
plot_list[[3]]
```


Predictive Model for GlassBottles Waste
Count of GlassBottles
40
30
20
10
0
Jul
Oct
Jan
Apr
Date
Data Type
Actual Data
Predicted Data

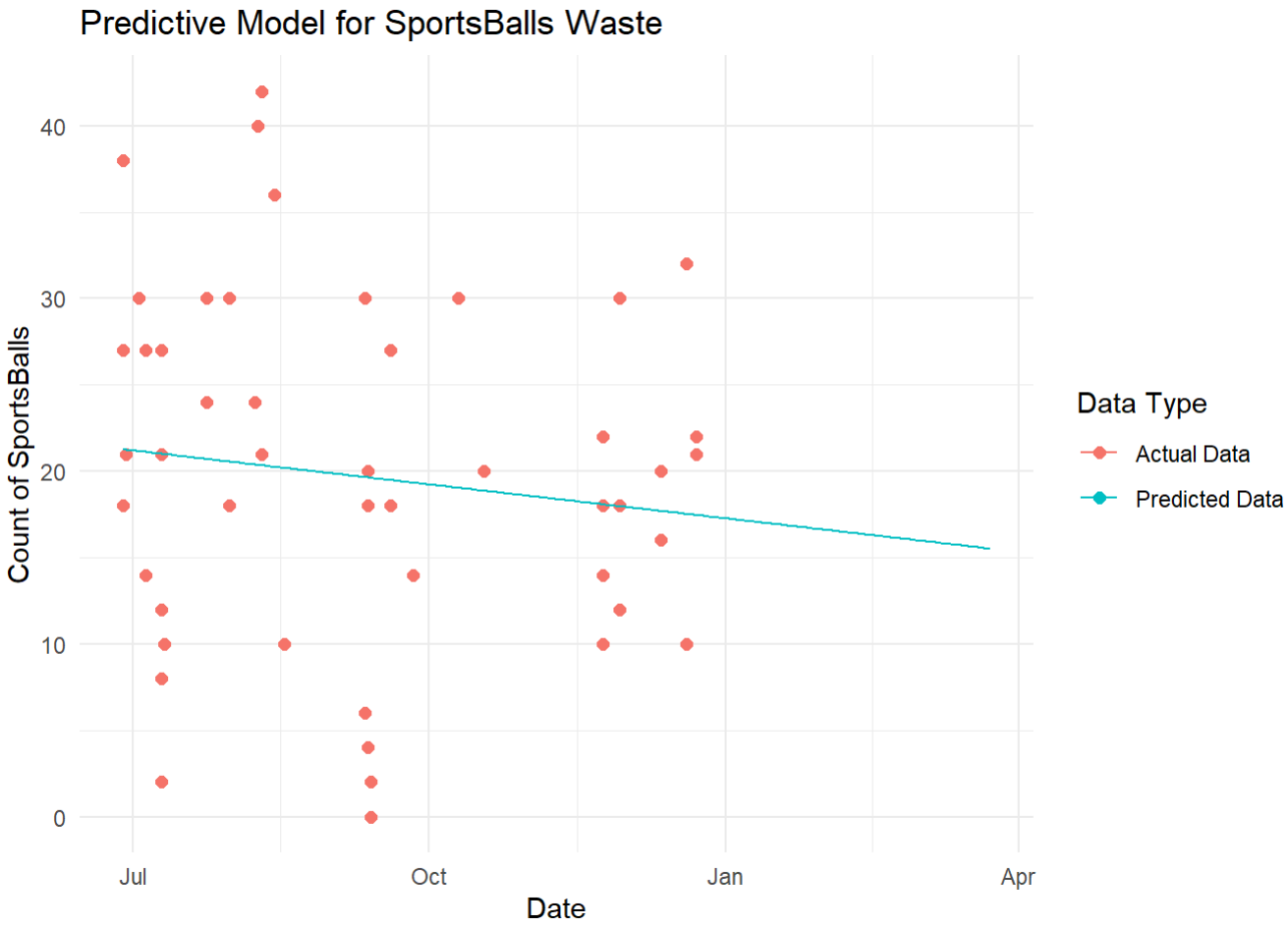
```
plot_list[[4]]
```


Predictive Model for PlasticBags Waste
Count of PlasticBags
500
400
300
200
100
Jul
Oct
Jan
Apr
Date
Data Type
Actual Data
Predicted Data

```
plot_list[[5]]
```


Predictive Model for Wrappers Waste
Count of Wrappers
3000
2000
1000
Jul
Oct
Jan
Apr
Date
Data Type
Actual Data
Predicted Data

```
plot_list[[6]]
```


Predictive Model for SportsBalls Waste
Count of SportsBalls
40
30
20
10
0
Jul
Oct
Jan
Apr
Date
Data Type
Actual Data
Predicted Data