# Univariate - MA Data Analysis

Akihiko Mori

## Library —

## Import Data from file —

### Check imported data

```
head(df)
```

```
## # A tibble: 6 x 58
##   date       day   week_end is_closed food_loss_kg food_waste_kg solid_waste_kg
##   <date>     <chr>    <dbl> <lgl>            <dbl>         <dbl>          <dbl>
## 1 2022-09-16 Fri          1 FALSE             9.5           6.55           2.5
## 2 2022-09-17 Sat          1 FALSE            12.2           2.8            0.6
## 3 2022-09-18 Sun          1 FALSE             6.5           3.25           0.85
## 4 2022-09-20 Tue         -1 FALSE            13.1           0.7            0.3
## 5 2022-09-21 Wed         -1 FALSE             5.7           1.1            0.45
## 6 2022-09-22 Thu         -1 FALSE             7.25          0.8            0.35
## # i 51 more variables: liquid_waste_kg <dbl>, customers <dbl>, fulls <dbl>,
## #   halfs <dbl>, takeouts <dbl>, liquors <dbl>, sales <dbl>, container <dbl>,
## #   temp_c <dbl>, humi_p <dbl>, prcp_mm <dbl>, TS_noodle_kg <dbl>,
## #   TS_water_kg <dbl>, TS_bones_kg <dbl>, TS_veg_kg <dbl>, TS_meat_kg <dbl>,
## #   TS_condi_kg <dbl>, TS_Broth_kg <dbl>, TS_Stock_kg <dbl>, TS_FL_kg <dbl>,
## #   TS_FL_bone_kg <dbl>, TS_FL_veg_kg <dbl>, TS_FL_meat_kg <dbl>,
## #   TS_FP_kg <dbl>, FL_noodle_kg <dbl>, FL_water_kg <dbl>, ...
```

```
str(df)
```

```
## spc_tbl_ [169 x 58] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ date             : Date[1:169], format: "2022-09-16" "2022-09-17" ...
##  $ day              : chr [1:169] "Fri" "Sat" "Sun" "Tue" ...
##  $ week_end         : num [1:169] 1 1 1 -1 -1 -1 1 1 1 -1 ...
##  $ is_closed        : logi [1:169] FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ food_loss_kg     : num [1:169] 9.5 12.2 6.5 13.1 5.7 ...
##  $ food_waste_kg    : num [1:169] 6.55 2.8 3.25 0.7 1.1 0.8 1.5 2.65 2.55 2.2 ...
##  $ solid_waste_kg   : num [1:169] 2.5 0.6 0.85 0.3 0.45 0.35 0.65 0.7 0.8 0.8 ...
##  $ liquid_waste_kg  : num [1:169] 4.05 2.2 2.4 0.4 0.65 0.45 0.85 1.95 1.75 1.4 ...
##  $ customers        : num [1:169] 42 42 27 13 15 14 12 35 24 26 ...
##  $ fulls            : num [1:169] 36 30 24 10 10 10 11 35 18 25 ...
##  $ halfs            : num [1:169] 4 6 2 2 3 2 2 2 3 3 ...
##  $ takeouts         : num [1:169] 15 12 10 12 10 16 28 23 25 13 ...
##  $ liquors          : num [1:169] 2 2 1 4 1 1 2 3 6 3 ...
```

```
##  $ sales            : num [1:169] 1080 862 629 635 533 ...
##  $ container        : num [1:169] 0 0 0 0 0 0 0 0 0 0 ...
##  $ temp_c           : num [1:169] 9.04 7 9.61 5.66 7.35 ...
##  $ humi_p           : num [1:169] 89.5 92.5 81.1 74.1 76.7 66.7 75.6 71.3 70.1 74.7 ...
##  $ prcp_mm          : num [1:169] 4.1 1 0 0 0 0 0 0 0 0 ...
##  $ TS_noodle_kg     : num [1:169] -7.95 -6.75 -5.25 -3.45 -3.23 ...
##  $ TS_water_kg      : num [1:169] -34.5 -29.2 -22.8 -14.9 -14 ...
##  $ TS_bones_kg      : num [1:169] -8.74 -7.42 -5.78 -3.79 -3.55 ...
##  $ TS_veg_kg        : num [1:169] -4.98 -4.23 -3.29 -2.16 -2.02 ...
##  $ TS_meat_kg       : num [1:169] -2.12 -1.8 -1.4 -0.92 -0.86 -1.08 -1.6 -2.36 -1.78 -1.58 ...
##  $ TS_condi_kg      : num [1:169] -0.795 -0.675 -0.525 -0.345 -0.323 ...
##  $ TS_Broth_kg      : num [1:169] 1.03e-15 -1.11e-15 1.11e-15 -7.57e-16 -1.51e-16 ...
##  $ TS_Stock_kg      : num [1:169] 29.7 25.2 19.6 12.9 12 ...
##  $ TS_FL_kg         : num [1:169] 11.34 9.63 7.49 4.92 4.6 ...
##  $ TS_FL_bone_kg    : num [1:169] -8.74 -7.42 -5.78 -3.79 -3.55 ...
##  $ TS_FL_veg_kg     : num [1:169] -2.332 -1.98 -1.54 -1.012 -0.946 ...
##  $ TS_FL_meat_kg    : num [1:169] -0.265 -0.225 -0.175 -0.115 -0.107 ...
##  $ TS_FP_kg         : num [1:169] 47.7 40.5 31.5 20.7 19.4 ...
##  $ FL_noodle_kg     : num [1:169] -6.66 -8.59 -4.56 -9.18 -4 ...
##  $ FL_water_kg      : num [1:169] -28.9 -37.2 -19.7 -39.8 -17.3 ...
##  $ FL_bones_kg      : num [1:169] -7.32 -9.45 -5.01 -10.1 -4.39 ...
##  $ FL_veg_kg        : num [1:169] -4.17 -5.38 -2.86 -5.75 -2.5 ...
##  $ FL_meat_kg       : num [1:169] -1.78 -2.29 -1.21 -2.45 -1.07 ...
##  $ FL_condi_kg      : num [1:169] -0.666 -0.859 -0.456 -0.918 -0.4 ...
##  $ FL_Broth_kg      : num [1:169] -1.33e-15 1.55e-15 1.33e-15 1.37e-15 8.67e-16 ...
##  $ FL_Stock_kg      : num [1:169] 24.9 32.1 17 34.3 14.9 ...
##  $ FL_FL_kg         : num [1:169] 9.5 12.2 6.5 13.1 5.7 ...
##  $ FL_FL_bone_kg    : num [1:169] -7.32 -9.45 -5.01 -10.1 -4.39 ...
##  $ FL_FL_veg_kg     : num [1:169] -1.95 -2.52 -1.34 -2.69 -1.17 ...
##  $ FL_FL_meat_kg    : num [1:169] -0.222 -0.286 -0.152 -0.306 -0.133 ...
##  $ FL_FP_kg         : num [1:169] 40 51.5 27.3 55.1 24 ...
##  $ Broth_diff       : num [1:169] -4.82 6.86 -2.59 21.4 2.88 ...
##  $ Final_Prod_diff  : num [1:169] -7.75 11.02 -4.16 34.39 4.62 ...
##  $ daily_total_served: num [1:169] 47.7 40.5 31.5 20.7 19.4 ...
##  $ tueD             : num [1:169] 0 0 0 1 0 0 0 0 0 1 ...
##  $ wedD             : num [1:169] 0 0 0 0 1 0 0 0 0 0 ...
##  $ thuD             : num [1:169] 0 0 0 0 0 1 0 0 0 0 ...
##  $ friD             : num [1:169] 1 0 0 0 0 0 1 0 0 0 ...
##  $ satD             : num [1:169] 0 1 0 0 0 0 0 1 0 0 ...
##  $ tueE             : num [1:169] 0 0 -1 1 0 0 0 0 -1 1 ...
##  $ wedE             : num [1:169] 0 0 -1 0 1 0 0 0 -1 0 ...
##  $ thuE             : num [1:169] 0 0 -1 0 0 1 0 0 -1 0 ...
##  $ friE             : num [1:169] 1 0 -1 0 0 0 1 0 -1 0 ...
##  $ satE             : num [1:169] 0 1 -1 0 0 0 0 1 -1 0 ...
##  $ wkend            : num [1:169] 1 1 1 -1 -1 -1 1 1 1 -1 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   date = col_date(format = ""),
##   ..   day = col_character(),
##   ..   week_end = col_double(),
##   ..   is_closed = col_logical(),
##   ..   food_loss_kg = col_double(),
##   ..   food_waste_kg = col_double(),
##   ..   solid_waste_kg = col_double(),
```

```
##   ..   liquid_waste_kg = col_double(),
##   ..   customers = col_double(),
##   ..   fulls = col_double(),
##   ..   halfs = col_double(),
##   ..   takeouts = col_double(),
##   ..   liquors = col_double(),
##   ..   sales = col_double(),
##   ..   container = col_double(),
##   ..   temp_c = col_double(),
##   ..   humi_p = col_double(),
##   ..   prcp_mm = col_double(),
##   ..   TS_noodle_kg = col_double(),
##   ..   TS_water_kg = col_double(),
##   ..   TS_bones_kg = col_double(),
##   ..   TS_veg_kg = col_double(),
##   ..   TS_meat_kg = col_double(),
##   ..   TS_condi_kg = col_double(),
##   ..   TS_Broth_kg = col_double(),
##   ..   TS_Stock_kg = col_double(),
##   ..   TS_FL_kg = col_double(),
##   ..   TS_FL_bone_kg = col_double(),
##   ..   TS_FL_veg_kg = col_double(),
##   ..   TS_FL_meat_kg = col_double(),
##   ..   TS_FP_kg = col_double(),
##   ..   FL_noodle_kg = col_double(),
##   ..   FL_water_kg = col_double(),
##   ..   FL_bones_kg = col_double(),
##   ..   FL_veg_kg = col_double(),
##   ..   FL_meat_kg = col_double(),
##   ..   FL_condi_kg = col_double(),
##   ..   FL_Broth_kg = col_double(),
##   ..   FL_Stock_kg = col_double(),
##   ..   FL_FL_kg = col_double(),
##   ..   FL_FL_bone_kg = col_double(),
##   ..   FL_FL_veg_kg = col_double(),
##   ..   FL_FL_meat_kg = col_double(),
##   ..   FL_FP_kg = col_double(),
##   ..   Broth_diff = col_double(),
##   ..   Final_Prod_diff = col_double(),
##   ..   daily_total_served = col_double(),
##   ..   tueD = col_double(),
##   ..   wedD = col_double(),
##   ..   thuD = col_double(),
##   ..   friD = col_double(),
##   ..   satD = col_double(),
##   ..   tueE = col_double(),
##   ..   wedE = col_double(),
##   ..   thuE = col_double(),
##   ..   friE = col_double(),
##   ..   satE = col_double(),
##   ..   wkend = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```r
names(df)
```

```
##  [1] "date"             "day"                "week_end"
##  [4] "is_closed"        "food_loss_kg"       "food_waste_kg"
##  [7] "solid_waste_kg"   "liquid_waste_kg"    "customers"
## [10] "fulls"            "halfs"              "takeouts"
## [13] "liquors"          "sales"              "container"
## [16] "temp_c"           "humi_p"             "prcp_mm"
## [19] "TS_noodle_kg"     "TS_water_kg"        "TS_bones_kg"
## [22] "TS_veg_kg"        "TS_meat_kg"         "TS_condi_kg"
## [25] "TS_Broth_kg"      "TS_Stock_kg"        "TS_FL_kg"
## [28] "TS_FL_bone_kg"    "TS_FL_veg_kg"       "TS_FL_meat_kg"
## [31] "TS_FP_kg"         "FL_noodle_kg"       "FL_water_kg"
## [34] "FL_bones_kg"      "FL_veg_kg"          "FL_meat_kg"
## [37] "FL_condi_kg"      "FL_Broth_kg"        "FL_Stock_kg"
## [40] "FL_FL_kg"         "FL_FL_bone_kg"      "FL_FL_veg_kg"
## [43] "FL_FL_meat_kg"    "FL_FP_kg"           "Broth_diff"
## [46] "Final_Prod_diff"  "daily_total_served" "tueD"
## [49] "wedD"             "thuD"               "friD"
## [52] "satD"             "tueE"               "wedE"
## [55] "thuE"             "friE"               "satE"
## [58] "wkend"
```

## Univariable —

### Open days

```r
# sample size: open and close days ---------------------------------------
data.frame(obs_days    = nrow(df),
           open_days   = sum(df$is_closed),
           closed_days = sum(!df$is_closed))
```

```
##   obs_days open_days closed_days
## 1      169         8         161
```

```r
df %>%
  freq_table(is_closed)
```

```
## # A tibble: 2 x 3
##   is_closed     n  prop
##   <lgl>     <int> <dbl>
## 1 FALSE       161  95.3
## 2 TRUE          8   4.7
```

```r
df %>%
  select(c(date, day, is_closed))%>%
  subset(is_closed == TRUE)
```

```
## # A tibble: 8 x 3
```

```
##   date       day   is_closed
##   <date>     <chr> <lgl>
## 1 2022-10-09 Sun   TRUE
## 2 2022-11-10 Thu   TRUE
## 3 2022-11-11 Fri   TRUE
## 4 2022-12-01 Thu   TRUE
## 5 2022-12-24 Sat   TRUE
## 6 2022-12-25 Sun   TRUE
## 7 2023-01-01 Sun   TRUE
## 8 2023-03-19 Sun   TRUE
```

**Basic Summary of Dependent Variables**

```
# basic summary: dependents ----------------------------------------------------
data.frame(food_loss_waste   = c(summary(df$food_loss_kg + df$food_waste_kg)),
           food_loss         = c(summary(df$food_loss_kg)),
           food_waste_all    = c(summary(df$food_waste_kg)),
           food_waste_liquid = c(summary(df$liquid_waste_kg)),
           food_waste_solid  = c(summary(df$solid_waste_kg)))
```

```
##         food_loss_waste food_loss food_waste_all food_waste_liquid
## Min.           0.000000  0.000000       0.000000          0.000000
## 1st Qu.        8.250000  6.600000       0.950000          0.550000
## Median         9.500000  7.300000       1.950000          1.400000
## Mean           9.543491  7.460355       2.083136          1.408876
## 3rd Qu.       11.050000  8.150000       2.900000          2.000000
## Max.          17.900000 13.800000       6.550000          4.500000
##         food_waste_solid
## Min.           0.0000000
## 1st Qu.        0.3500000
## Median         0.6000000
## Mean           0.6742604
## 3rd Qu.        0.9000000
## Max.           2.9500000
```

```
df %>%
  select(c(food_loss_kg,food_waste_kg,liquid_waste_kg,solid_waste_kg)) %>%
  get_summary_stats()
```

```
## # A tibble: 4 x 13
##   variable        n   min   max median    q1    q3   iqr   mad  mean    sd    se
##   <fct>       <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 food_loss_~   169     0 13.8     7.3  6.6   8.15  1.55 1.19  7.46   2.69 0.207
## 2 food_waste~   169     0  6.55    1.95 0.95  2.9   1.95 1.48  2.08   1.45 0.111
## 3 liquid_was~   169     0  4.5     1.4  0.55  2     1.45 1.04  1.41   1.02 0.079
## 4 solid_wast~   169     0  2.95    0.6  0.35  0.9   0.55 0.445 0.674  0.51 0.039
## # i 1 more variable: ci <dbl>
```

```
library(summarytools)
```

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

```
## system might not have X11 capabilities; in case of errors when using dfSummary(), set st_options(use

##
## Attaching package: 'summarytools'

## The following object is masked from 'package:tibble':
##
##     view
```

```
df %>%
  select(c(food_loss_kg,food_waste_kg,
           liquid_waste_kg,solid_waste_kg)) %>%
  descr(order = "preserve",
        stats = c('mean', 'sd', 'min', 'q1', 'med', 'q3', 'max'),
        round.digits = 6)
```

```
## Descriptive Statistics
## df
## N: 169
##
##                 food_loss_kg   food_waste_kg   liquid_waste_kg   solid_waste_kg
## ------------- -------------- --------------- ----------------- ----------------
##          Mean       7.460355        2.083136          1.408876         0.674260
##       Std.Dev       2.693018        1.445795          1.021296         0.509818
##           Min       0.000000        0.000000          0.000000         0.000000
##            Q1       6.600000        0.950000          0.550000         0.350000
##        Median       7.300000        1.950000          1.400000         0.600000
##            Q3       8.150000        2.900000          2.000000         0.900000
##           Max      13.800000        6.550000          4.500000         2.950000
```

```
# basic summary: dependents excluding closed days ------------------------------
data.frame(food_loss_waste   = c(summary(df$food_loss_kg[!df$is_closed]
                                         + df$food_waste_kg[!df$is_closed])),
           food_loss         = c(summary(df$food_loss_kg[!df$is_closed])),
           food_waste_all    = c(summary(df$food_waste_kg[!df$is_closed])),
           food_waste_liquid = c(summary(df$liquid_waste_kg[!df$is_closed])),
           food_waste_solid  = c(summary(df$solid_waste_kg[!df$is_closed])))
```

```
##         food_loss_waste food_loss food_waste_all food_waste_liquid
## Min.             0.0000  0.000000       0.000000          0.000000
## 1st Qu.          8.4000  6.700000       1.100000          0.650000
## Median           9.6500  7.350000       2.100000          1.500000
## Mean            10.0177  7.831056       2.186646          1.478882
## 3rd Qu.         11.1500  8.400000       2.950000          2.050000
## Max.            17.9000 13.800000       6.550000          4.500000
##         food_waste_solid
## Min.            0.000000
## 1st Qu.         0.350000
## Median          0.650000
## Mean            0.707764
## 3rd Qu.         0.950000
## Max.            2.950000
```

```r
df %>%
  filter(is_closed == FALSE) %>%
  select(c(food_loss_kg,food_waste_kg,liquid_waste_kg,solid_waste_kg)) %>%
  get_summary_stats()
```

```
## # A tibble: 4 x 13
##   variable        n   min   max median    q1    q3   iqr   mad  mean    sd    se
##   <fct>       <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 food_loss_~   161     0 13.8    7.35  6.7   8.4   1.7  1.11  7.83  2.17  0.171
## 2 food_waste~   161     0  6.55   2.1   1.1   2.95  1.85 1.33  2.19  1.40  0.111
## 3 liquid_was~   161     0  4.5    1.5   0.65  2.05  1.4  1.04  1.48  0.995 0.078
## 4 solid_wast~   161     0  2.95   0.65  0.35  0.95  0.6  0.445 0.708 0.499 0.039
## # i 1 more variable: ci <dbl>
```

```r
library(stargazer)
df_fw <- as.data.frame(subset(df, is_closed==FALSE,
                  select = c("food_loss_kg","food_waste_kg",
                             "liquid_waste_kg","solid_waste_kg")))
stargazer(df_fw, type = "latex")
```

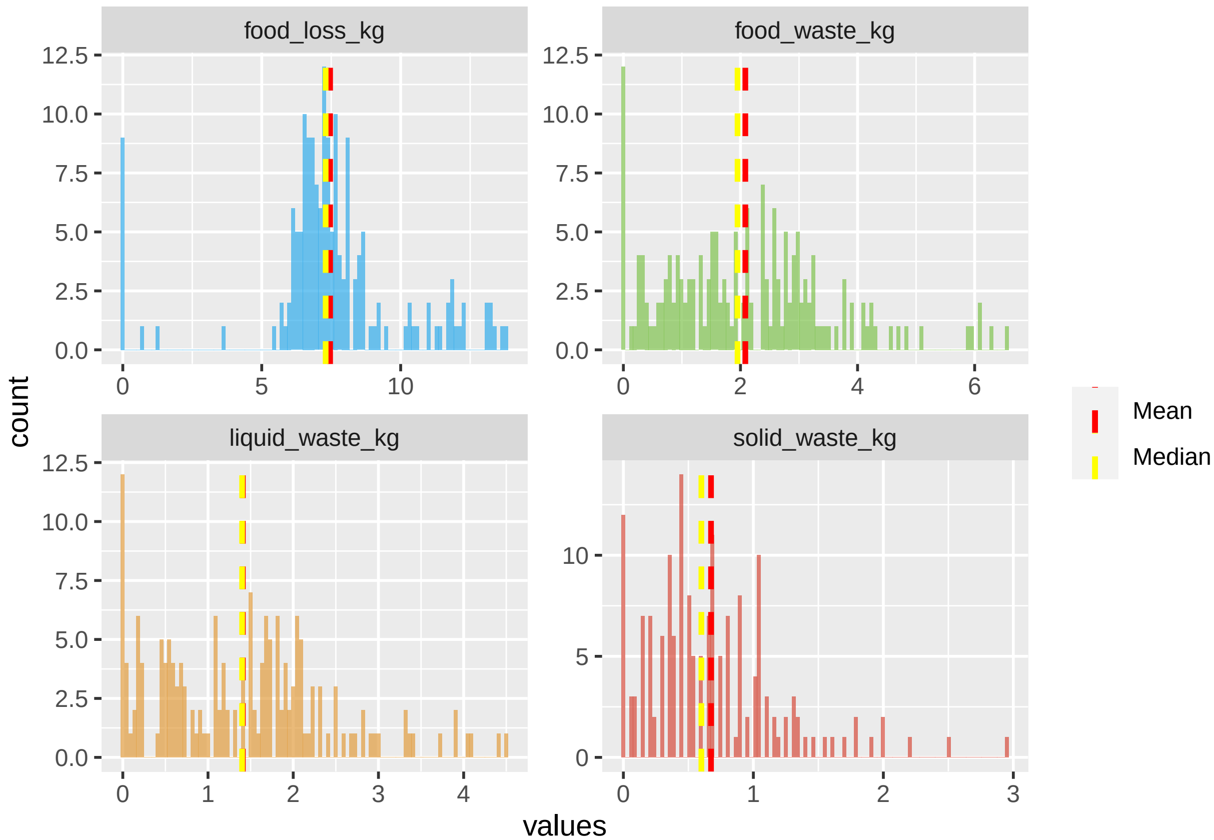
```
##
## % Table created by stargazer v.5.2.3 by Marek Hlavac, Social Policy Institute. E-mail: marek.hlavac a
## % Date and time: Mon, Jan 15, 2024 - 10:49:23
## \begin{table}[!htbp] \centering
##   \caption{}
##   \label{}
## \begin{tabular}{@{\extracolsep{5pt}}lccccc}
## \\[-1.8ex]\hline
## \hline \\[-1.8ex]
## Statistic & \multicolumn{1}{c}{N} & \multicolumn{1}{c}{Mean} & \multicolumn{1}{c}{St. Dev.} & \multic
## \hline \\[-1.8ex]
## food\_loss\_kg & 161 & 7.831 & 2.167 & 0.000 & 13.800 \\
## food\_waste\_kg & 161 & 2.187 & 1.403 & 0.000 & 6.550 \\
## liquid\_waste\_kg & 161 & 1.479 & 0.995 & 0.000 & 4.500 \\
## solid\_waste\_kg & 161 & 0.708 & 0.499 & 0.000 & 2.950 \\
## \hline \\[-1.8ex]
## \end{tabular}
## \end{table}
```

```r
# summary of--------------------
# 1. number of observations
# 2. Averages
# 3. standard deviations
# 4. Min values
# 4. Max values
# stargazer(subset(df[4:7], df$is_closed == FALSE),  flip=TRUE,
#           type = "text",digits=2, out="deps1.txt")
#
# # Excluding the restaurant closed ---------------
# stargazer(subset(df[4:7], df$is_closed == FALSE),  flip=TRUE,
#           type = "text",digits=2, out="deps2.txt")
```

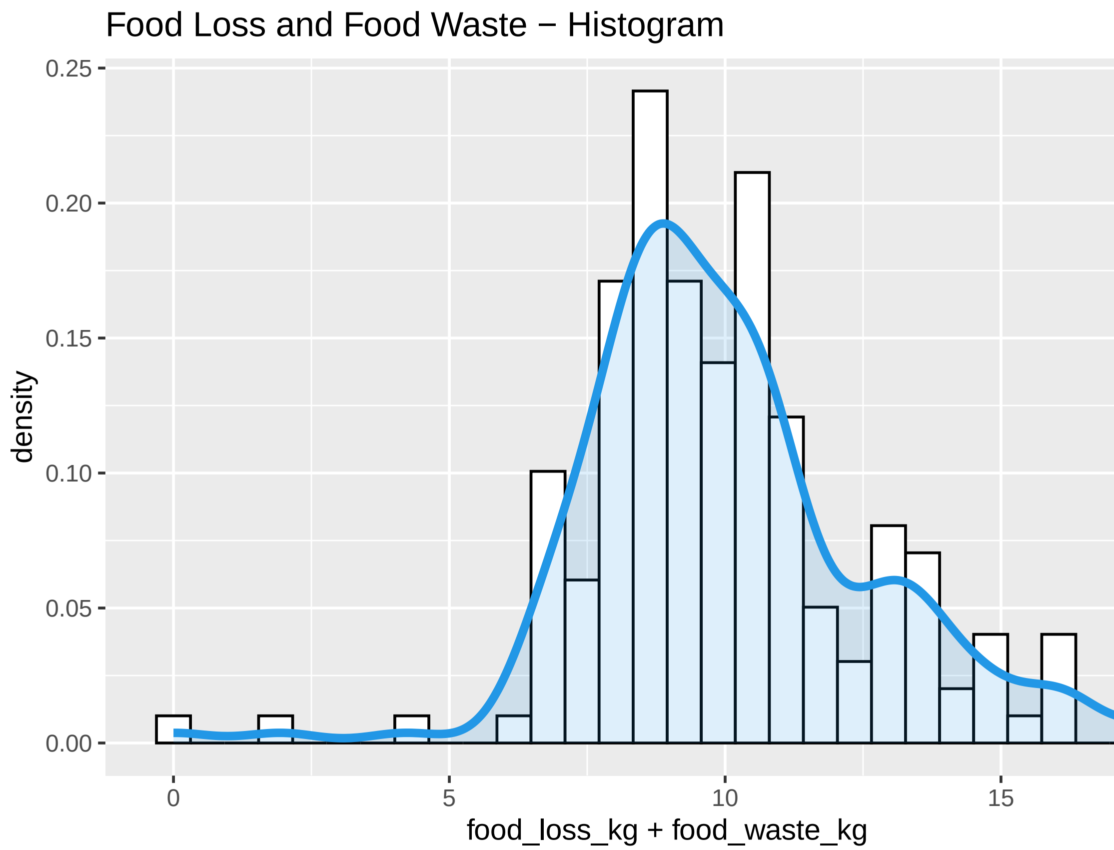
## Histograms —

### Normal histogram

```r
# Create a data frame of numeric features & label
dep_features <- df %>%
  select(c(is_closed, food_loss_kg, food_waste_kg,
           solid_waste_kg, liquid_waste_kg))

# Pivot data to a long format
dep_features <- dep_features %>%
  pivot_longer(!is_closed, names_to = "features",
               values_to = "values") %>%
  group_by(features) %>%
  mutate(Mean = mean(values),
         Median = median(values))


# Plot a histogram for each feature
dep_features %>%
  ggplot() +
  geom_histogram(aes(x = values, fill = features),
                 bins = 100, alpha = 0.7, show.legend = F) +
  facet_wrap(~ features, scales = 'free')+
  paletteer::scale_fill_paletteer_d("ggthemes::excel_Parallax") +
  # Add lines for mean and median
  geom_vline(aes(xintercept = Mean, color = "Mean"),
             linetype = "dashed", linewidth = 1 ) +
  geom_vline(aes(xintercept = Median, color = "Median"),
             linetype = "dashed", linewidth = 1 ) +
  scale_color_manual(name = "",
                     values = c(Mean = "red", Median = "yellow"))
```

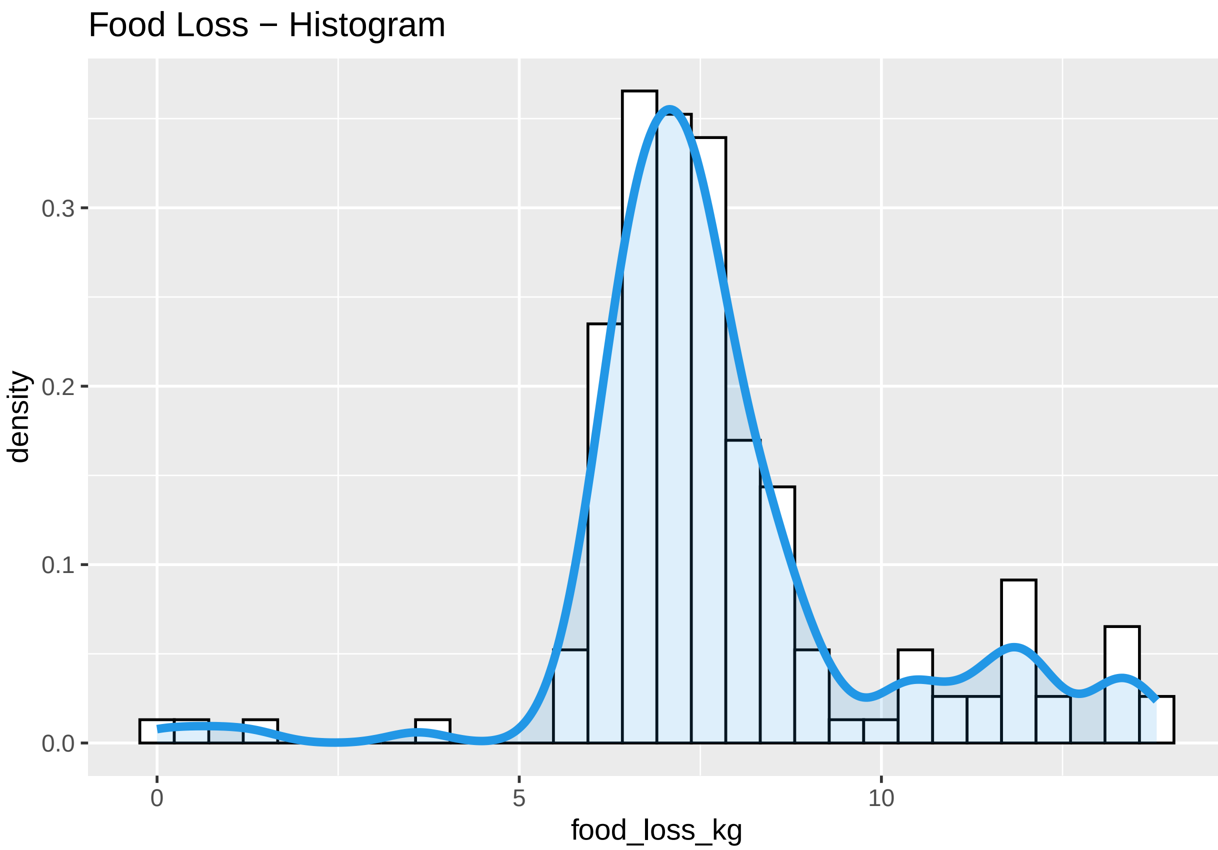



```
# binwidth = bw
# bw <- 2 * IQR(df$food_loss_kg) / length(df$food_loss_kg)^(1/3)

# Histogram on food loss + food waste ----------------------------------------
hist_loss_waste <-
  ggplot(data = subset(df, is_closed %in% FALSE),
         aes(x = food_loss_kg + food_waste_kg)) +
  geom_histogram(aes(y = after_stat(density)), bins = 30, colour = 1, fill = "white") +
  geom_density(linewidth = 1.5, colour = 4, fill = 4, alpha = 0.15) +
  labs(title = "Food Loss and Food Waste - Histogram")
hist_loss_waste
```

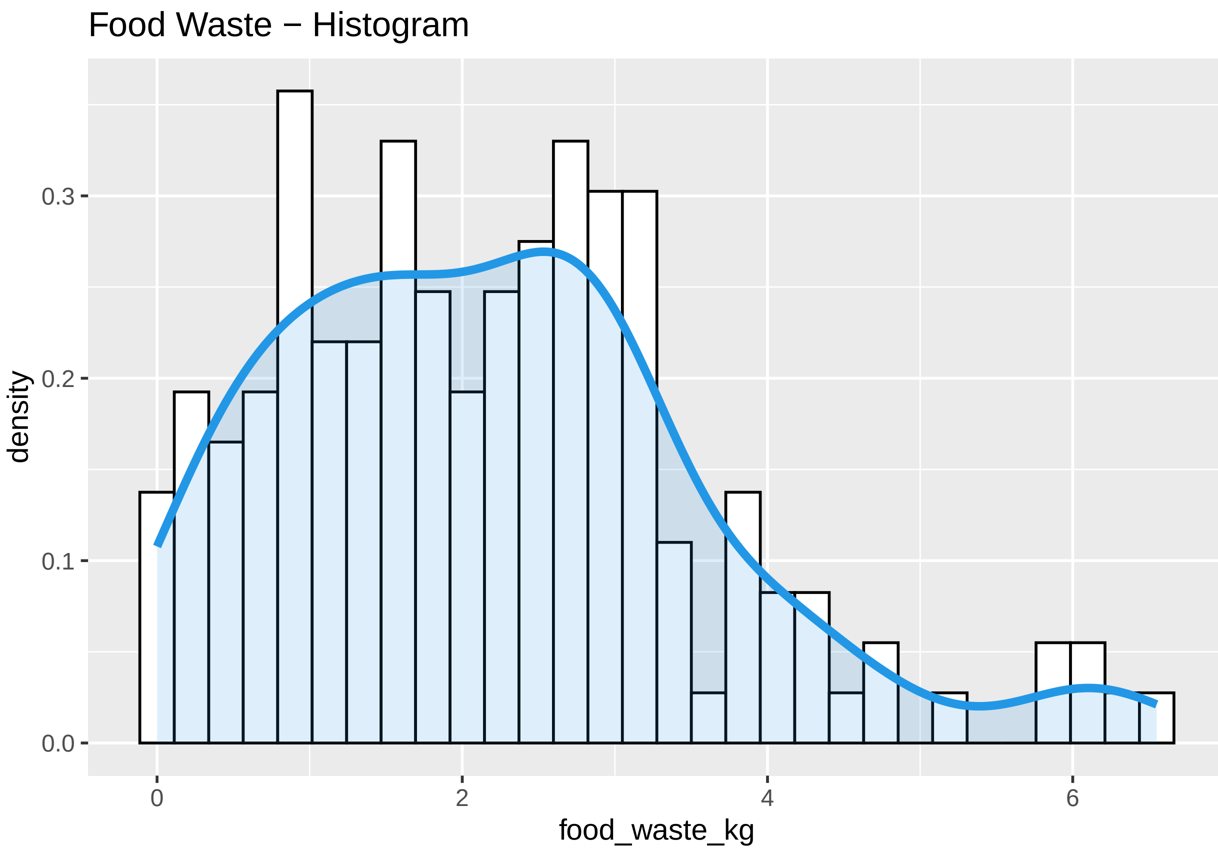



**Histogram with density**

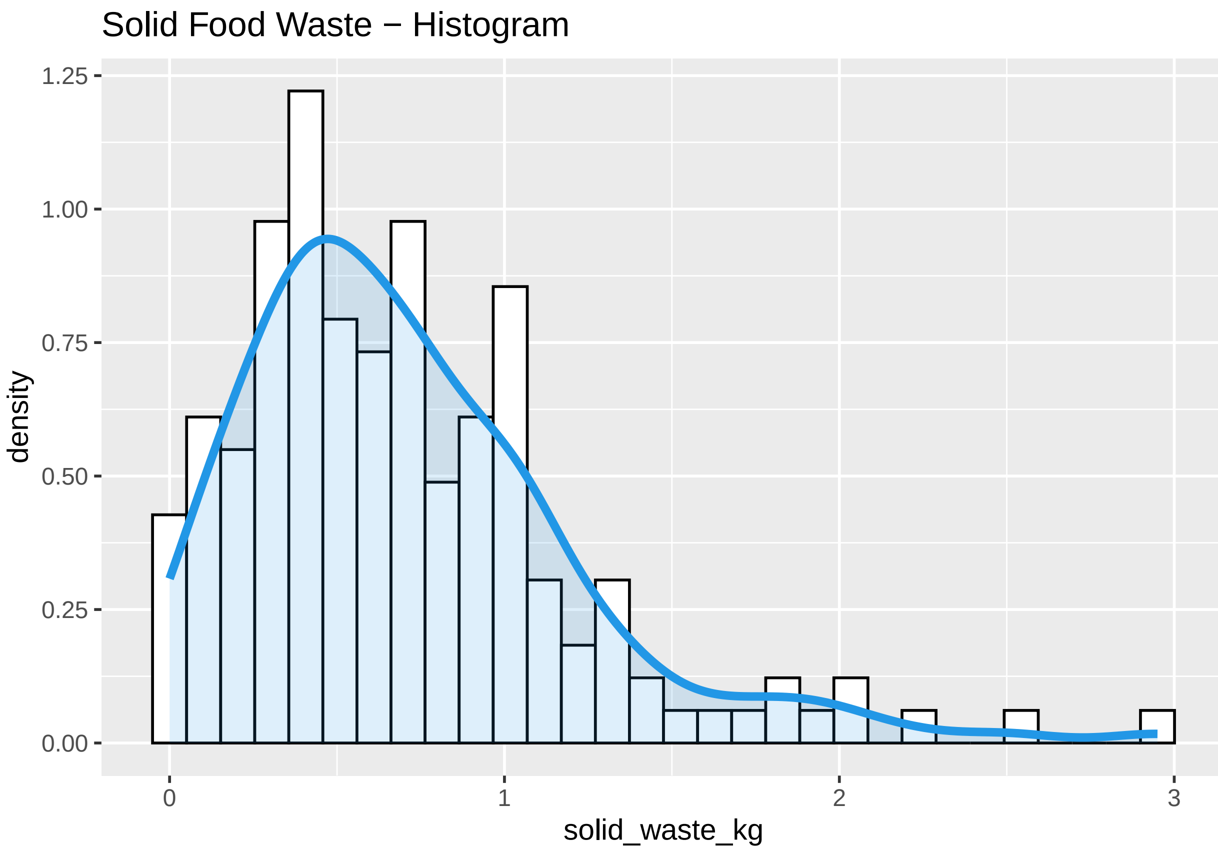
```
# Histogram on food loss----------------------------------------------------
hist_loss <-
  ggplot(data = subset(df, is_closed %in% FALSE), aes(x = food_loss_kg)) +
  geom_histogram(aes(y = after_stat(density)), bins = 30, colour = 1, fill = "white") +
  geom_density(linewidth = 1.5, colour = 4, fill = 4, alpha = 0.15) +
  labs(title = "Food Loss - Histogram")
hist_loss
```



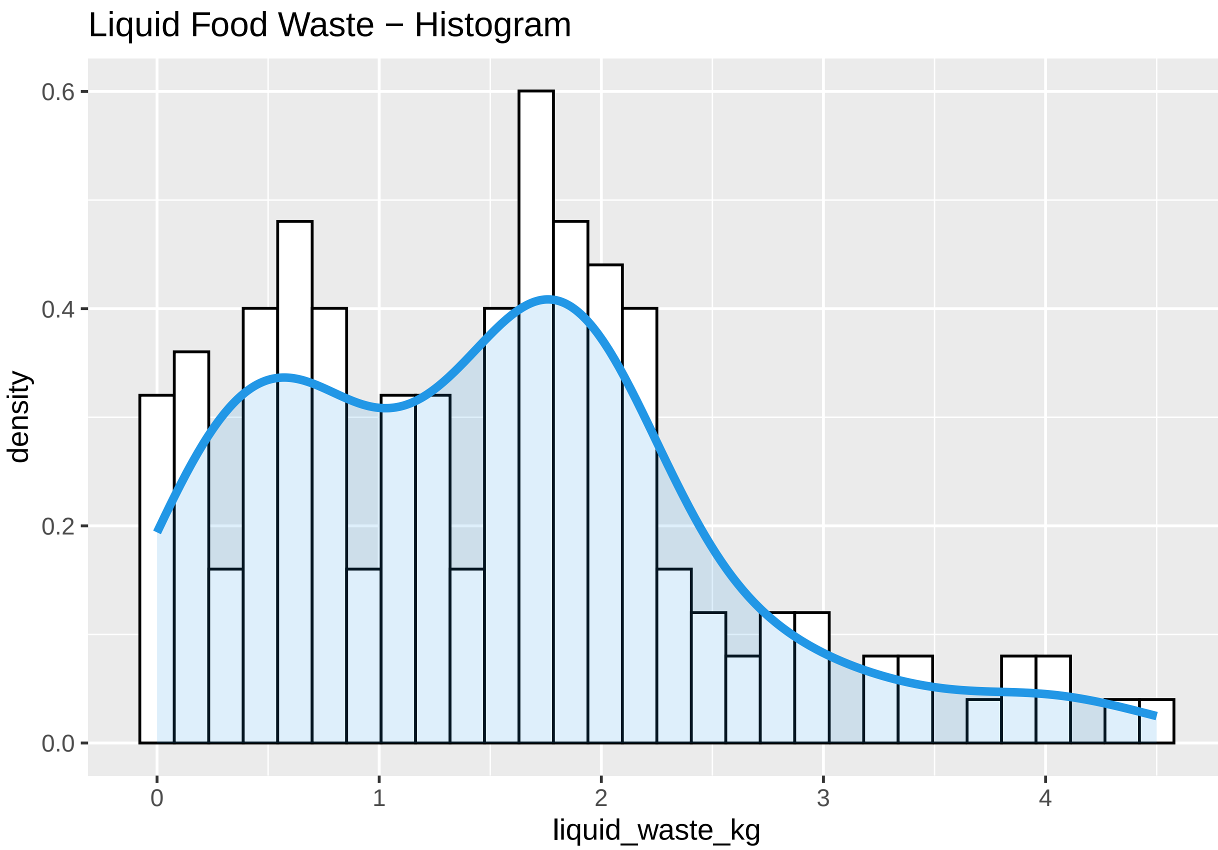

```r
# Histogram of food waste ----------------------------------------------------
hist_food_waste <-
  ggplot(data = subset(df, is_closed %in% FALSE), aes(x = food_waste_kg)) +
  geom_histogram(aes(y = after_stat(density)), bins = 30,colour = 1, fill = "white") +
  geom_density(linewidth = 1.5, colour = 4, fill = 4, alpha = 0.15) +
  labs(title = "Food Waste - Histogram")
hist_food_waste
```



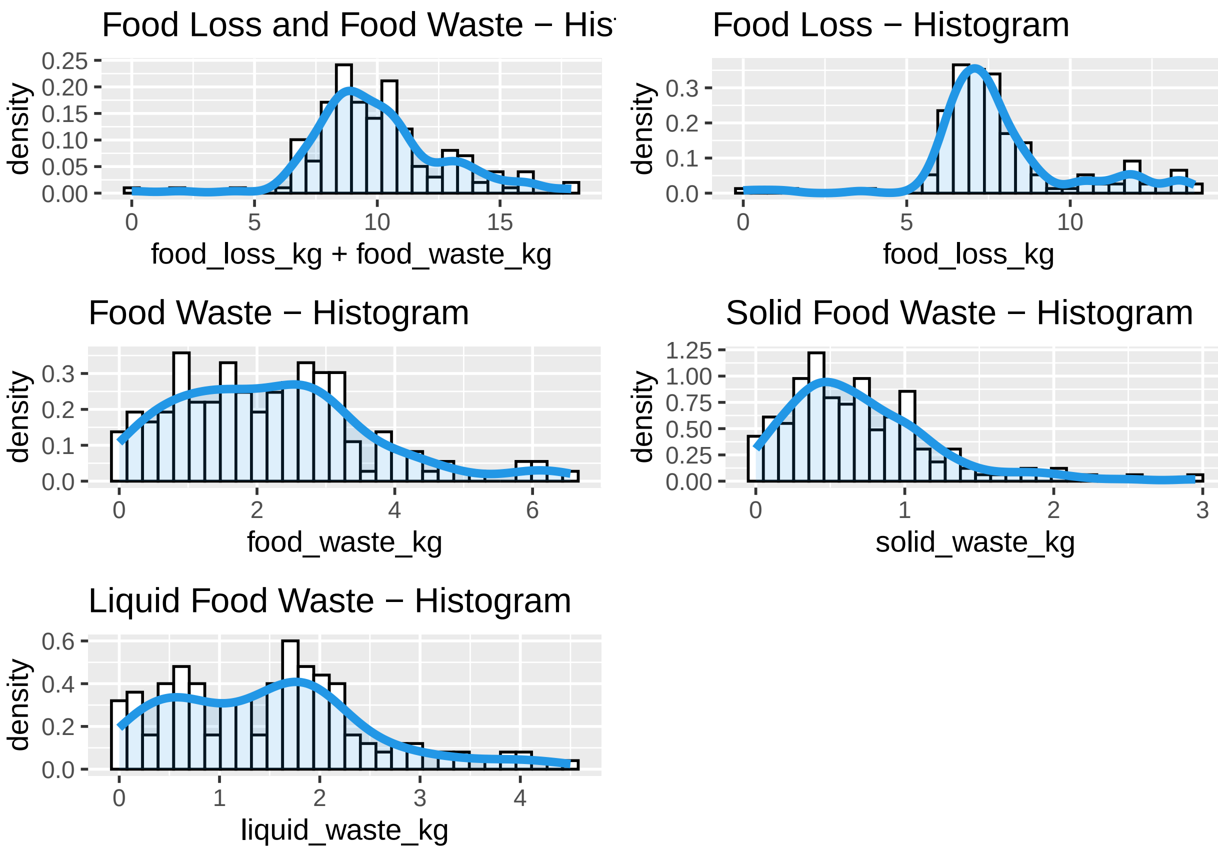

```r
# Histogram of solid waste ----------------------------------------------------
hist_solid_waste <-
  ggplot(data = subset(df, is_closed %in% FALSE), aes(x = solid_waste_kg)) +
  geom_histogram(aes(y = after_stat(density)), bins = 30,colour = 1, fill = "white") +
  geom_density(linewidth = 1.5, colour = 4, fill = 4, alpha = 0.15) +
  labs(title = "Solid Food Waste - Histogram")
hist_solid_waste
```




```
# Histogram of liquid waste -----------------------------------------------------
hist_liquid_waste <-
  ggplot(data = subset(df, is_closed %in% FALSE), aes(x = liquid_waste_kg)) +
  geom_histogram(aes(y = after_stat(density)), bins = 30,colour = 1, fill = "white") +
  geom_density(linewidth = 1.5, colour = 4, fill = 4, alpha = 0.15) +
  labs(title = "Liquid Food Waste - Histogram")
hist_liquid_waste
```

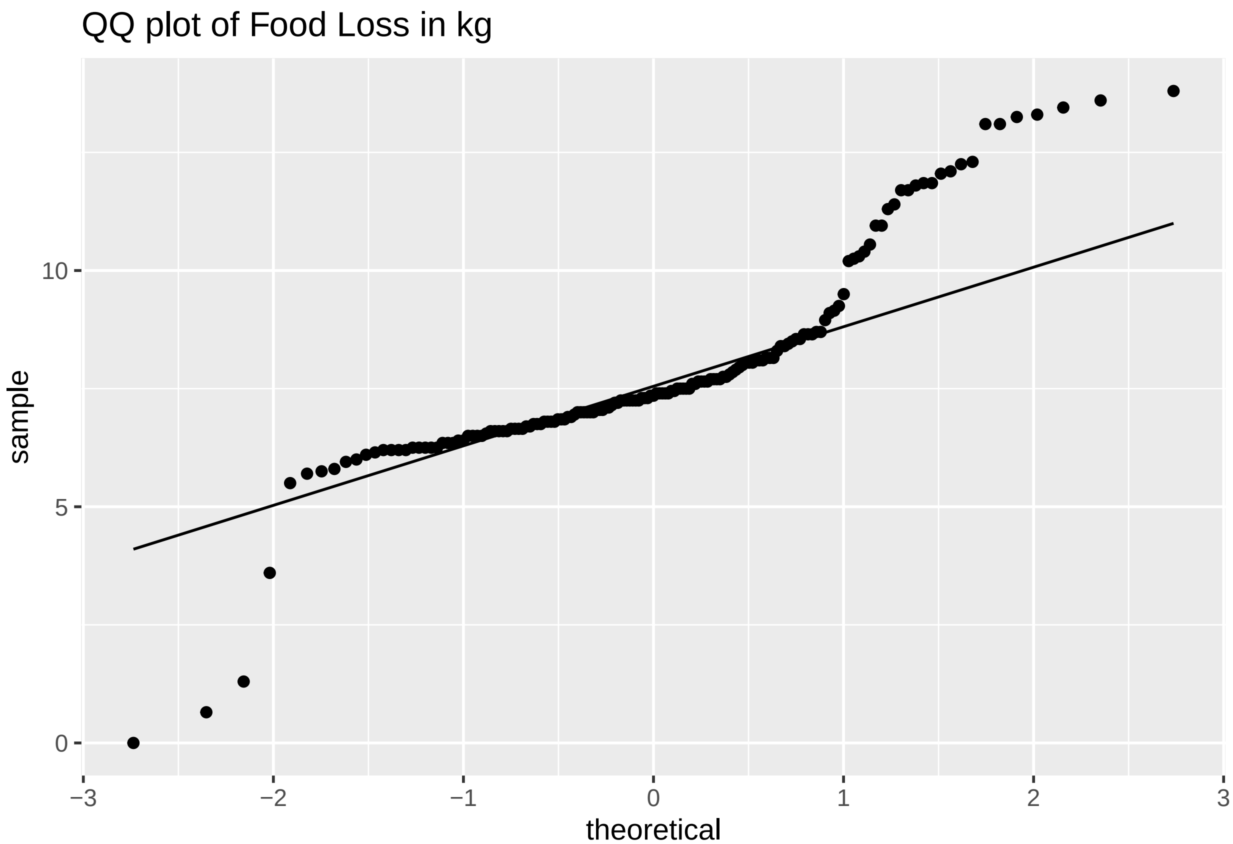



```
grid.arrange(hist_loss_waste,hist_loss,
             hist_food_waste,hist_solid_waste,hist_liquid_waste)
```

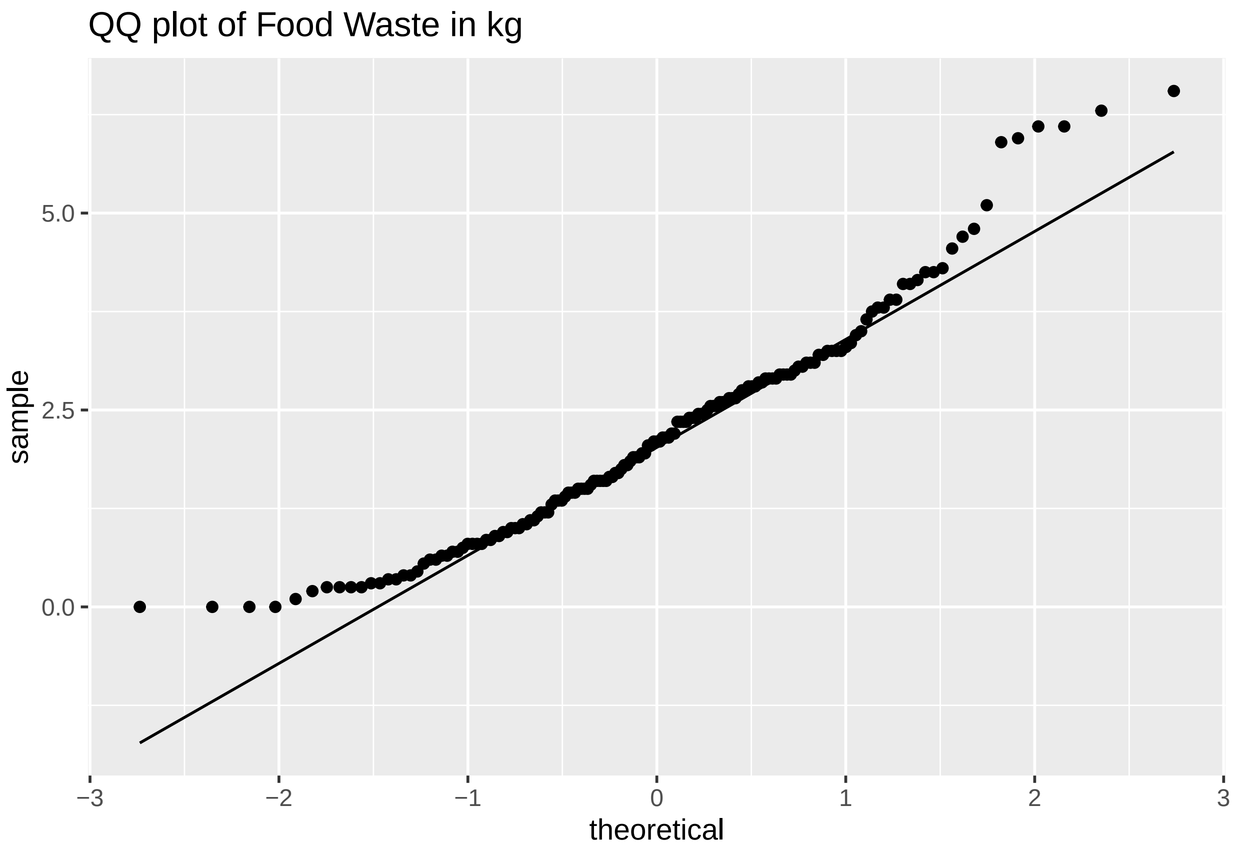



#### Q-Q plot

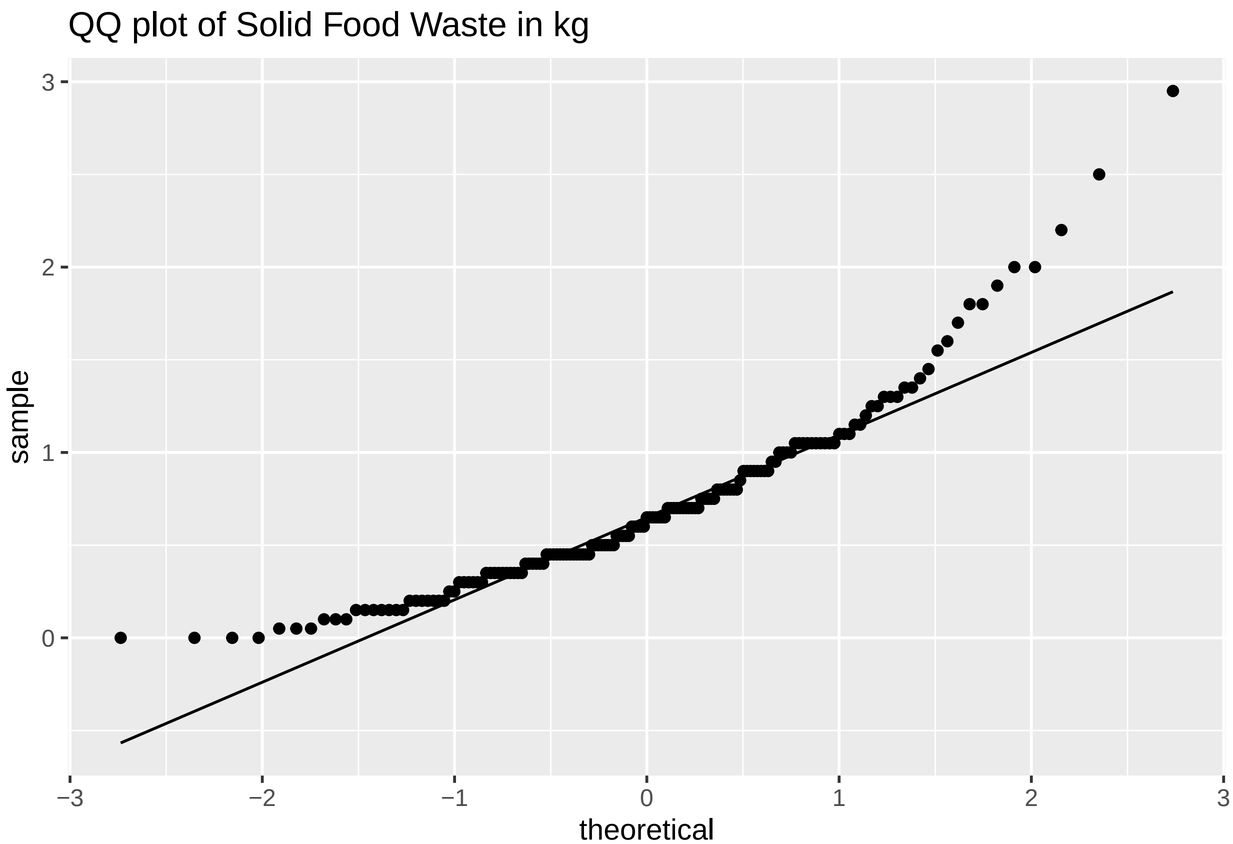
```
# Food loss ------------
ggplot(subset(df, is_closed %in% FALSE),
       aes(sample=food_loss_kg)) +
  stat_qq() + stat_qq_line() +
  xlab("theoretical") + ylab("sample") +
  ggtitle("QQ plot of Food Loss in kg")
```



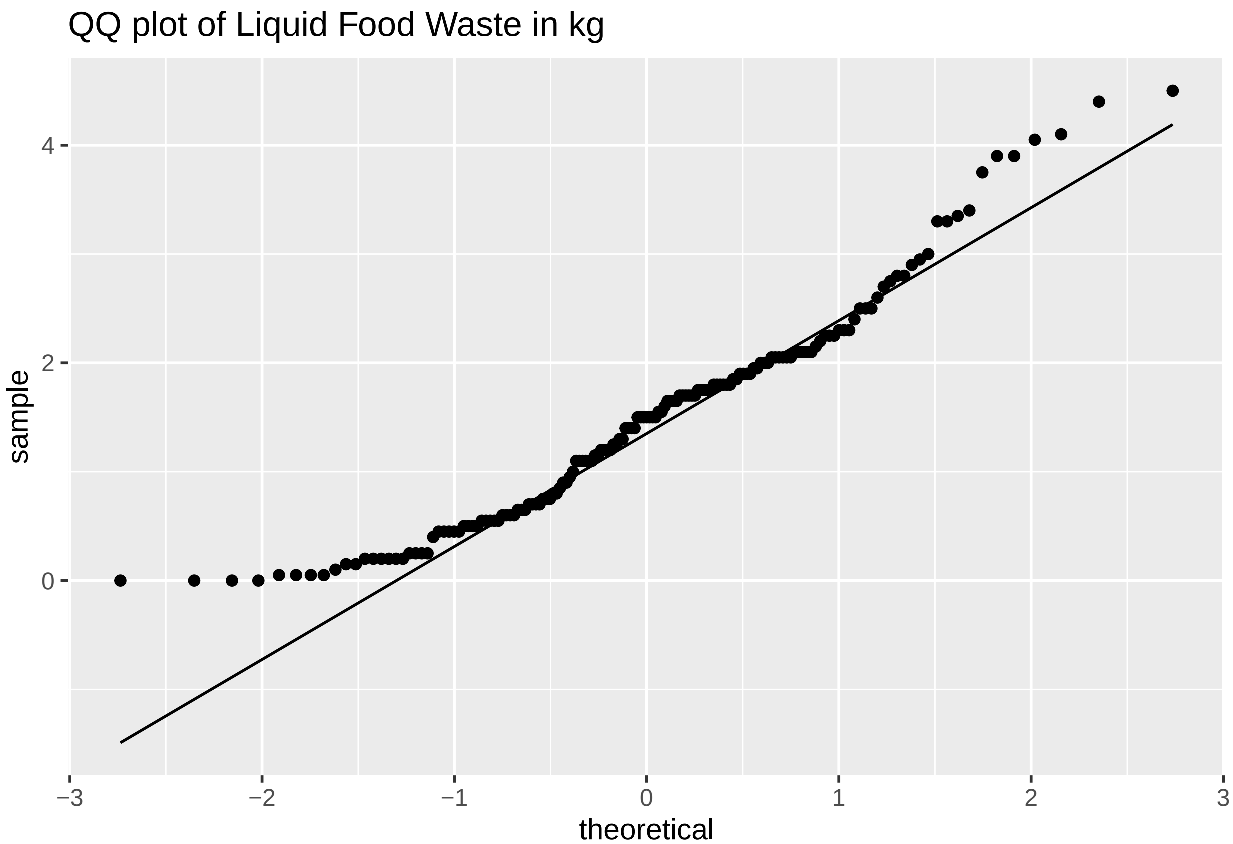

```
# Food waste ------------
ggplot(subset(df, is_closed %in% FALSE),
       aes(sample=food_waste_kg)) +
  stat_qq() + stat_qq_line() +
  xlab("theoretical") + ylab("sample") +
  ggtitle("QQ plot of Food Waste in kg")
```




```
# Solid Food waste ------------
ggplot(subset(df, is_closed %in% FALSE),
       aes(sample=solid_waste_kg)) +
  stat_qq() + stat_qq_line() +
  xlab("theoretical") + ylab("sample") +
  ggtitle("QQ plot of Solid Food Waste in kg")
```




```
# Liquid Food waste ------------
ggplot(subset(df, is_closed %in% FALSE),
       aes(sample=liquid_waste_kg)) +
  stat_qq() + stat_qq_line() +
  xlab("theoretical") + ylab("sample") +
  ggtitle("QQ plot of Liquid Food Waste in kg")
```




```
# Food waste ------------
df %>%
  filter(is_closed == FALSE) %>%
  shapiro_test(food_waste_kg, solid_waste_kg, liquid_waste_kg)
```

**shapiro test**

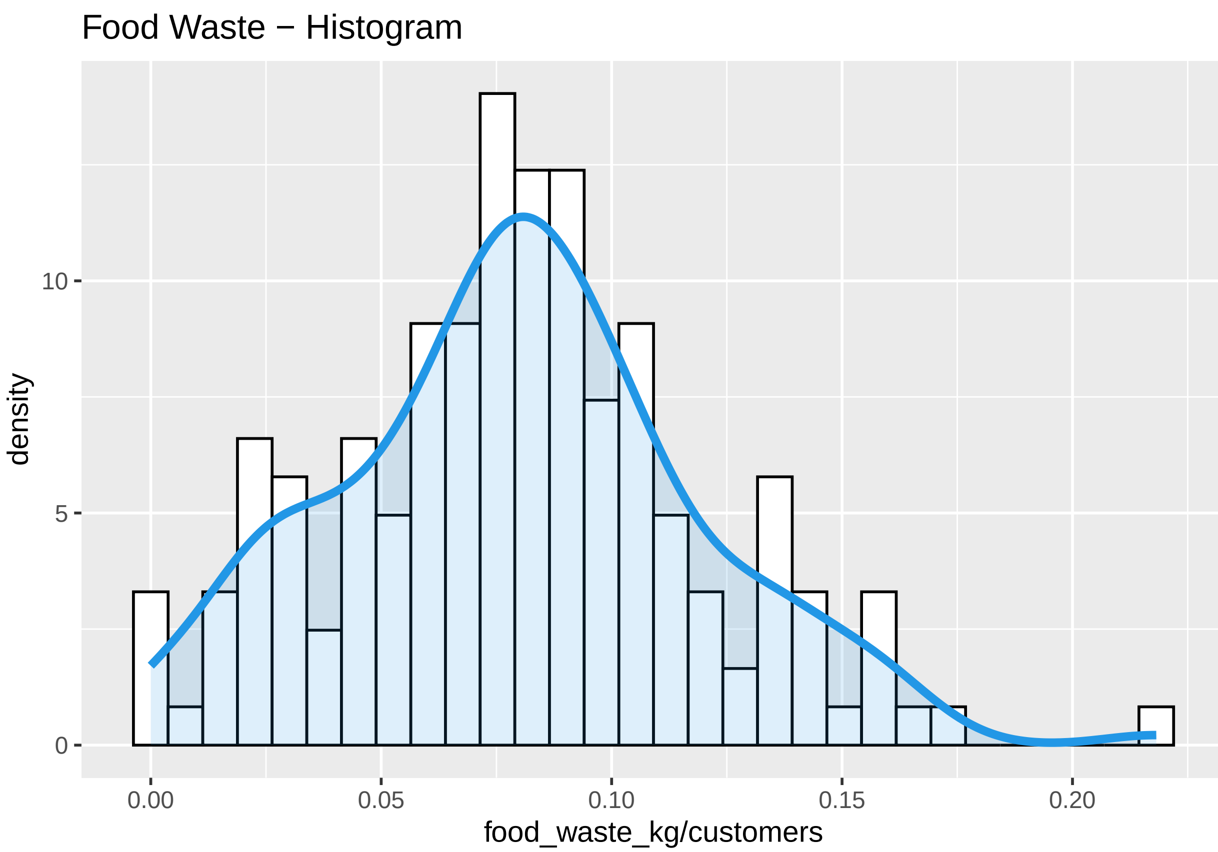
```
## # A tibble: 3 x 3
##   variable        statistic             p
##   <chr>               <dbl>         <dbl>
## 1 food_waste_kg       0.952 0.0000260
## 2 liquid_waste_kg     0.951 0.0000192
## 3 solid_waste_kg      0.903 0.00000000783
```

From the output, all the p-value is far less than 0.05; so implying that the distribution of the data are significantly different from normal distribution. In other words, we can not assume the normality.

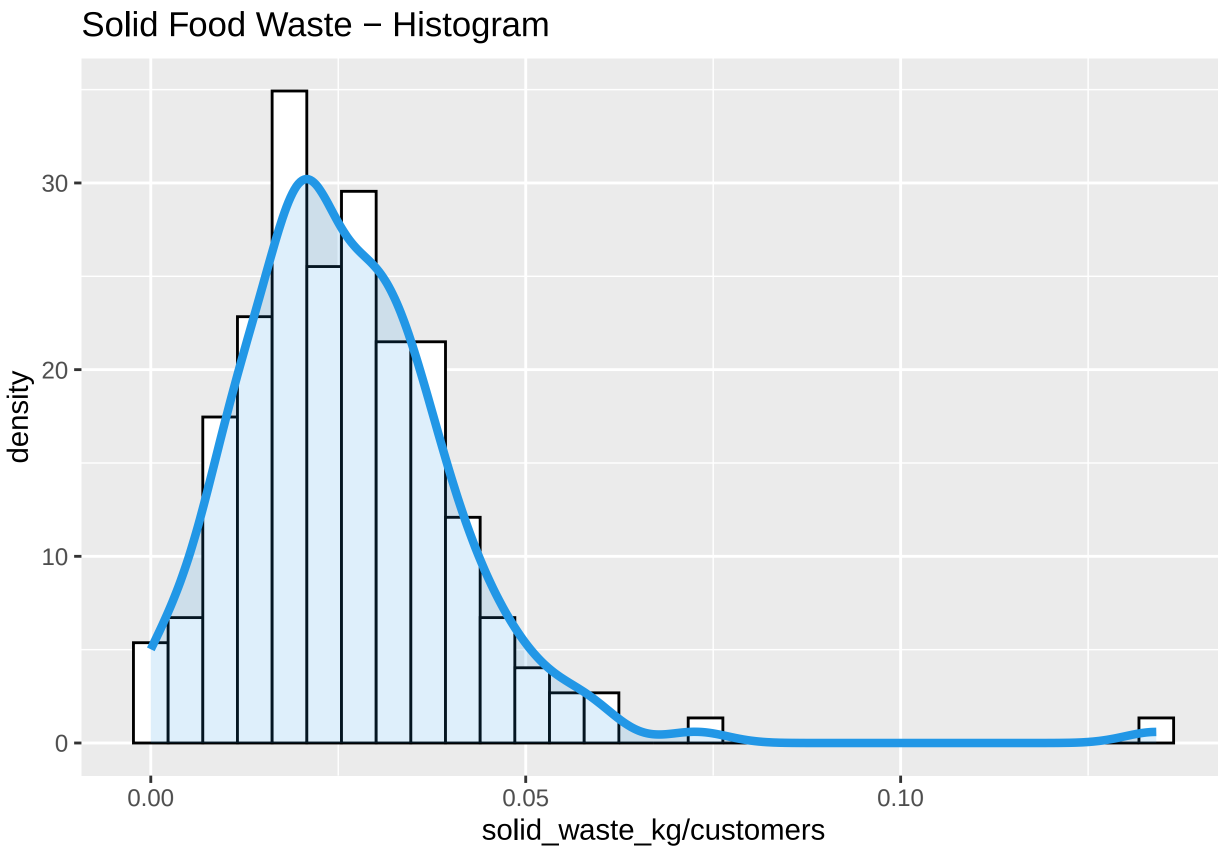
**Histogram per capita**

```
# Histogram of food waste ------------------------------------------
hist_food_waste <-
```

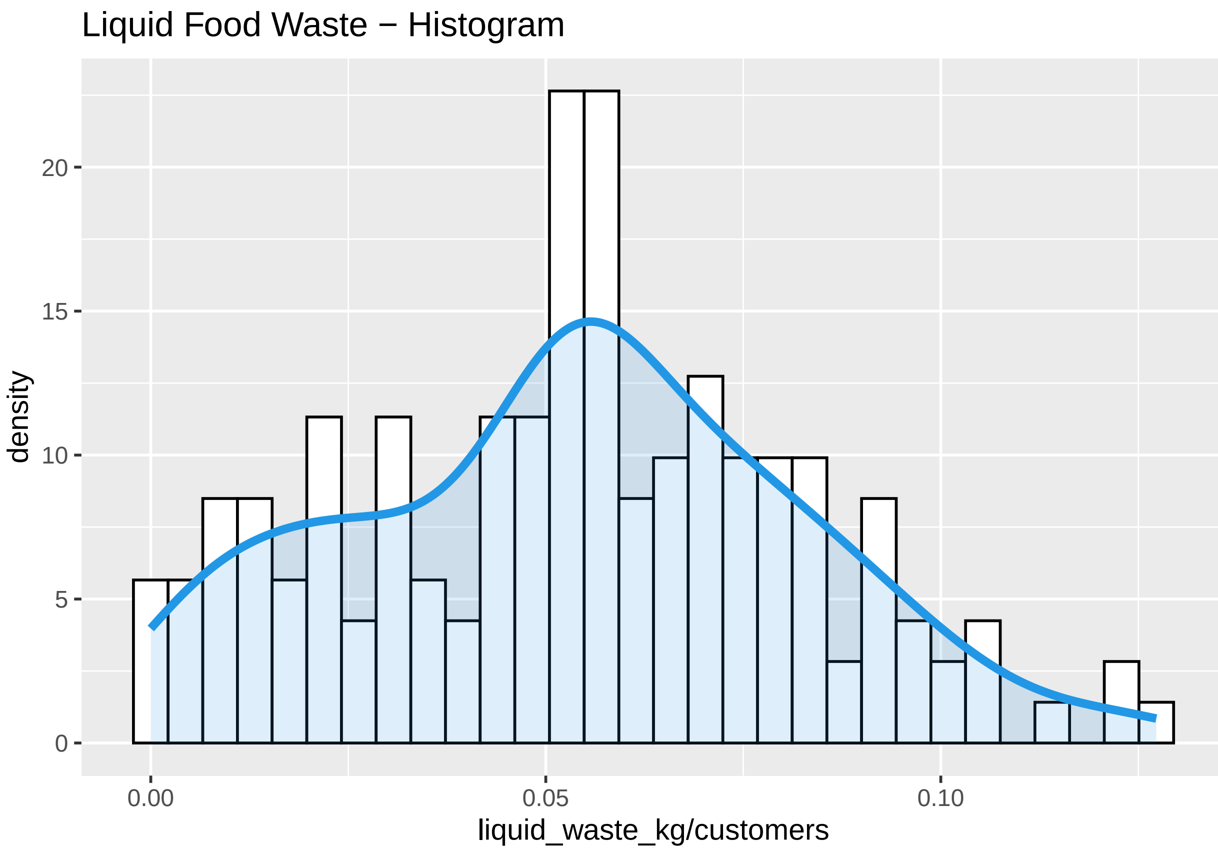
```
  ggplot(data = subset(df, is_closed %in% FALSE),
         aes(x = food_waste_kg/customers)) +
  geom_histogram(aes(y = after_stat(density)),
                 bins = 30,colour = 1, fill = "white") +
  geom_density(linewidth = 1.5, colour = 4, fill = 4, alpha = 0.15) +
  labs(title = "Food Waste - Histogram")
hist_food_waste
```



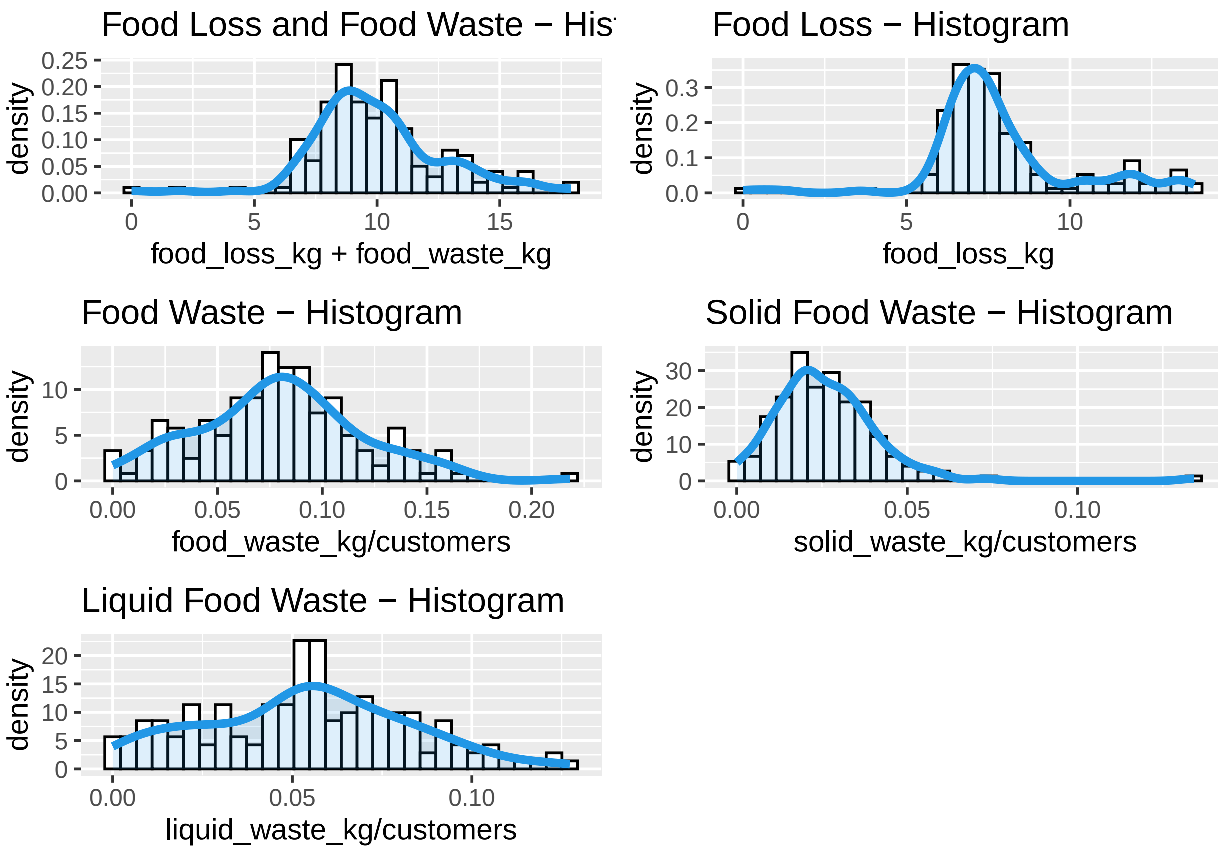

```
# Histogram of solid waste -------------------------------------------
hist_solid_waste <-
  ggplot(data = subset(df, is_closed %in% FALSE),
         aes(x = solid_waste_kg/customers)) +
  geom_histogram(aes(y = after_stat(density)),
                 bins = 30,colour = 1, fill = "white") +
  geom_density(linewidth = 1.5, colour = 4, fill = 4, alpha = 0.15) +
  labs(title = "Solid Food Waste - Histogram")
hist_solid_waste
```




```r
# Histogram of liquid waste ----------------------------------------
hist_liquid_waste <-
  ggplot(data = subset(df, is_closed %in% FALSE),
         aes(x = liquid_waste_kg/customers)) +
  geom_histogram(aes(y = after_stat(density)),
                 bins = 30,colour = 1, fill = "white") +
  geom_density(linewidth = 1.5, colour = 4, fill = 4, alpha = 0.15) +
  labs(title = "Liquid Food Waste - Histogram")
hist_liquid_waste
```




```
grid.arrange(hist_loss_waste,hist_loss,
             hist_food_waste,hist_solid_waste,hist_liquid_waste)
```



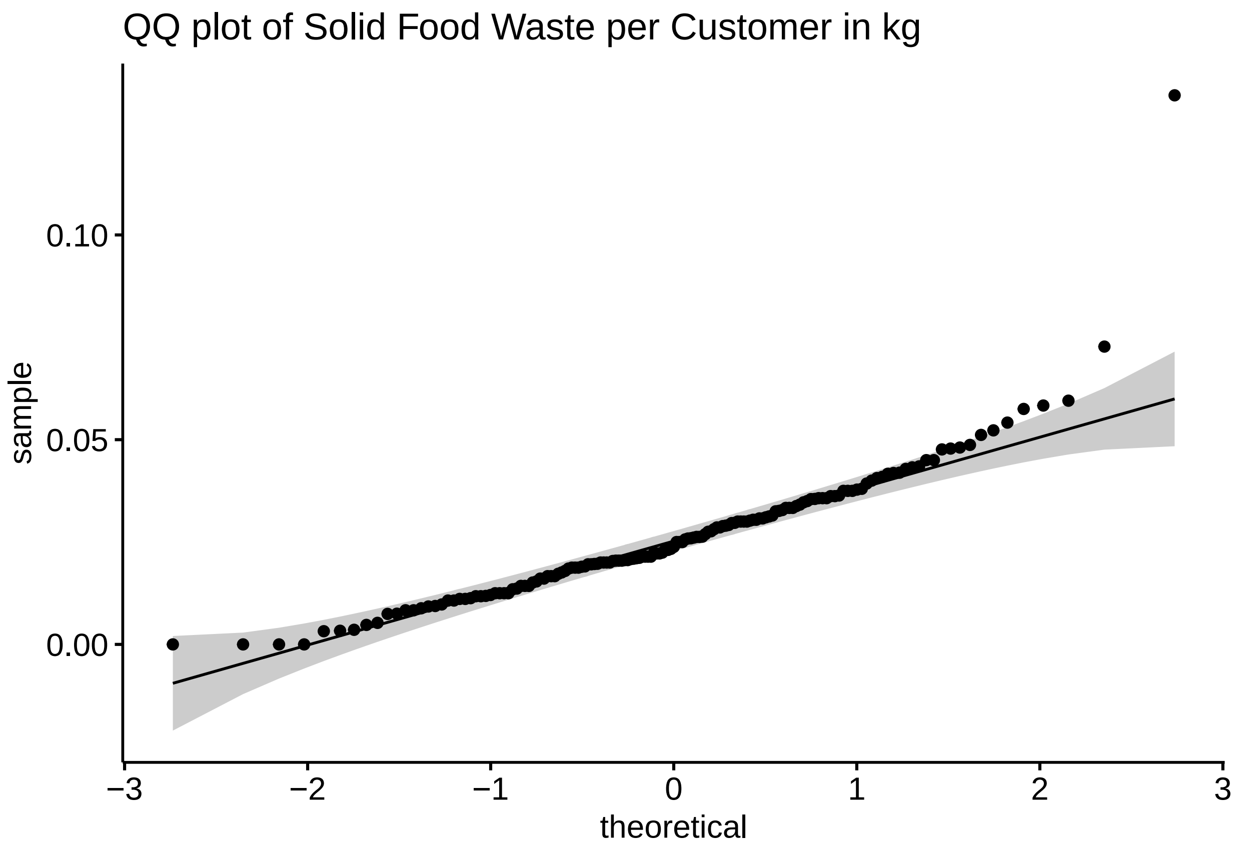

```
library(ggpubr)
```

**Q-Q plot per capita**

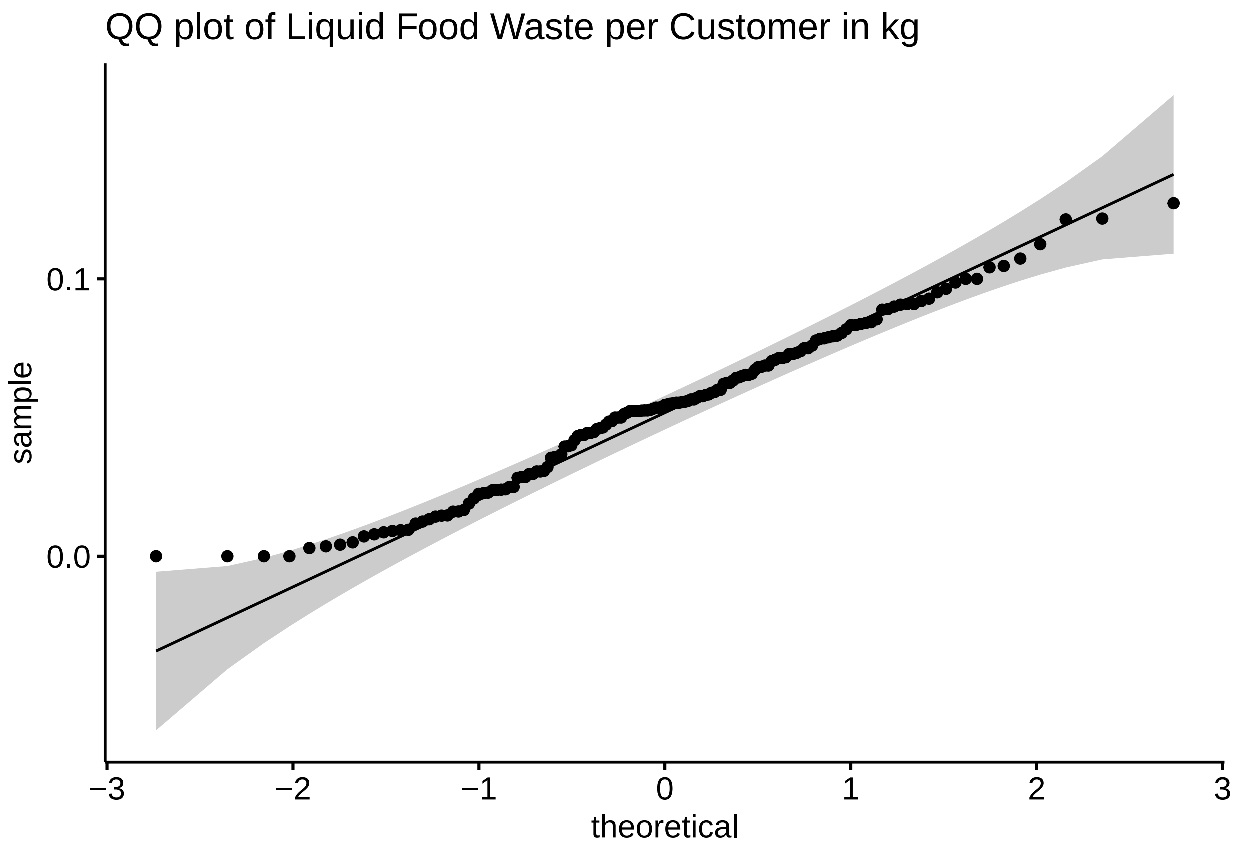
```
##
## Attaching package: 'ggpubr'

## The following object is masked from 'package:forecast':
##
##     gghistogram
```

```
# Food waste ------------
ggqqplot(subset(df$food_waste_kg/df$customers,
                df$is_closed %in% FALSE)) +
  xlab("theoretical") + ylab("sample") +
  ggtitle("QQ plot of Food Waste per Customer in kg")
```


```
# Solid Food waste ------------
ggqqplot(subset(df$solid_waste_kg/df$customers,
                df$is_closed %in% FALSE)) +
  xlab("theoretical") + ylab("sample") +
  ggtitle("QQ plot of Solid Food Waste per Customer in kg")
```




```r
# Liquid Food waste ------------
ggqqplot(subset(df$liquid_waste_kg/df$customers,
                df$is_closed %in% FALSE)) +
  xlab("theoretical") + ylab("sample") +
  ggtitle("QQ plot of Liquid Food Waste per Customer in kg")
```




```r
# Food waste ------------
df %>%
  filter(is_closed == FALSE) %>%
  mutate(food_waste_p_kg   = food_waste_kg/customers,
         solid_waste_p_kg  = solid_waste_kg/customers,
         liquid_waste_p_kg = liquid_waste_kg/customers) %>%
  shapiro_test(food_waste_p_kg, solid_waste_p_kg, liquid_waste_p_kg)
```

**shapiro test for per capita**

```
## # A tibble: 3 x 3
##   variable          statistic        p
##   <chr>                 <dbl>    <dbl>
## 1 food_waste_p_kg       0.987 1.38e- 1
## 2 liquid_waste_p_kg     0.984 6.10e- 2
## 3 solid_waste_p_kg      0.863 6.24e-11
```

From the output, the p-value of solid food waste per customer is far less that the significant level of 0.05; but the others are not. So it imply that the distribution of the data for solid food waste per customer is significantly different from normal distribution. In other words, we can assume the normality for food waste and liquid food waste per customer but not for solid food waste.

**Histogram per customer w/o outlier**

```
# find outliers ----
# food waste -----
which(df$food_waste_kg/df$customers > 0.2)  # => 46
```

```
## [1] 46
```

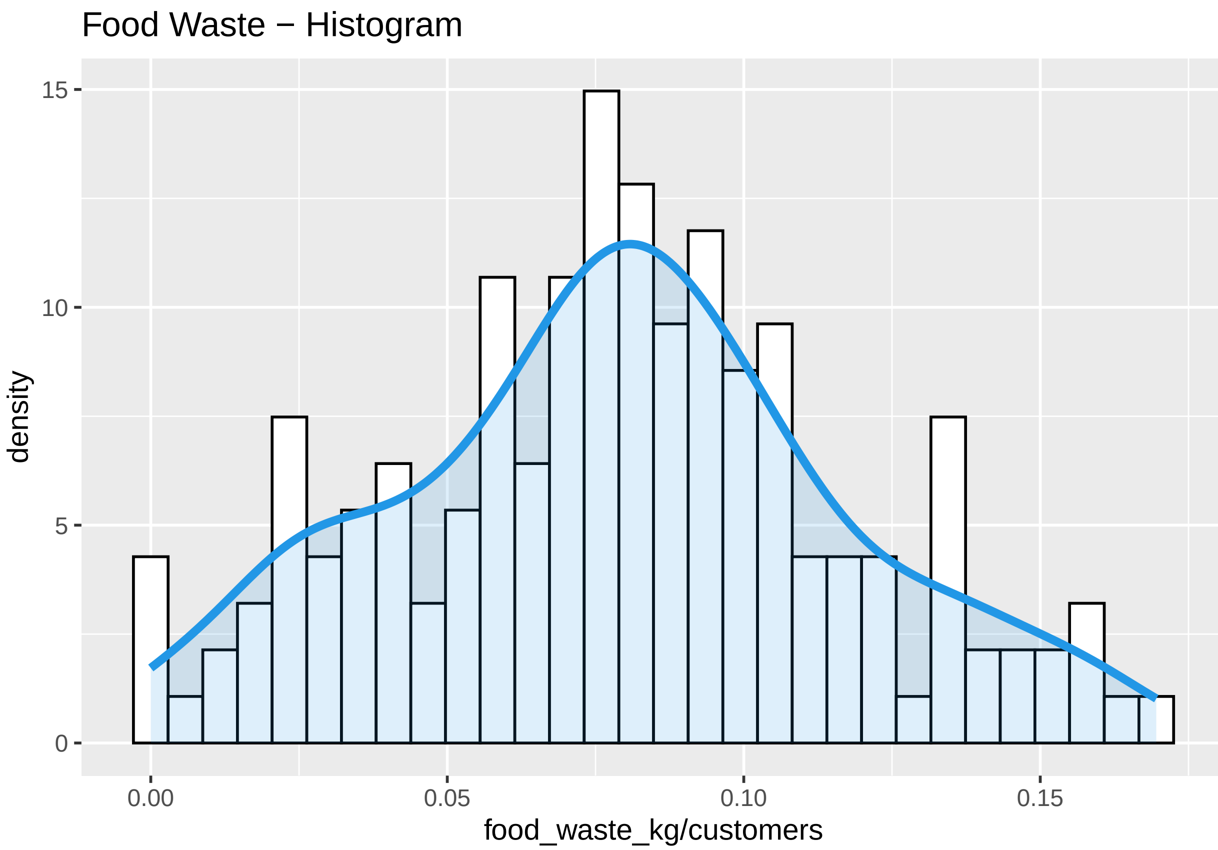
```
which(df$solid_waste_kg/df$customers > 0.1) # => 46
```

```
## [1] 46
```

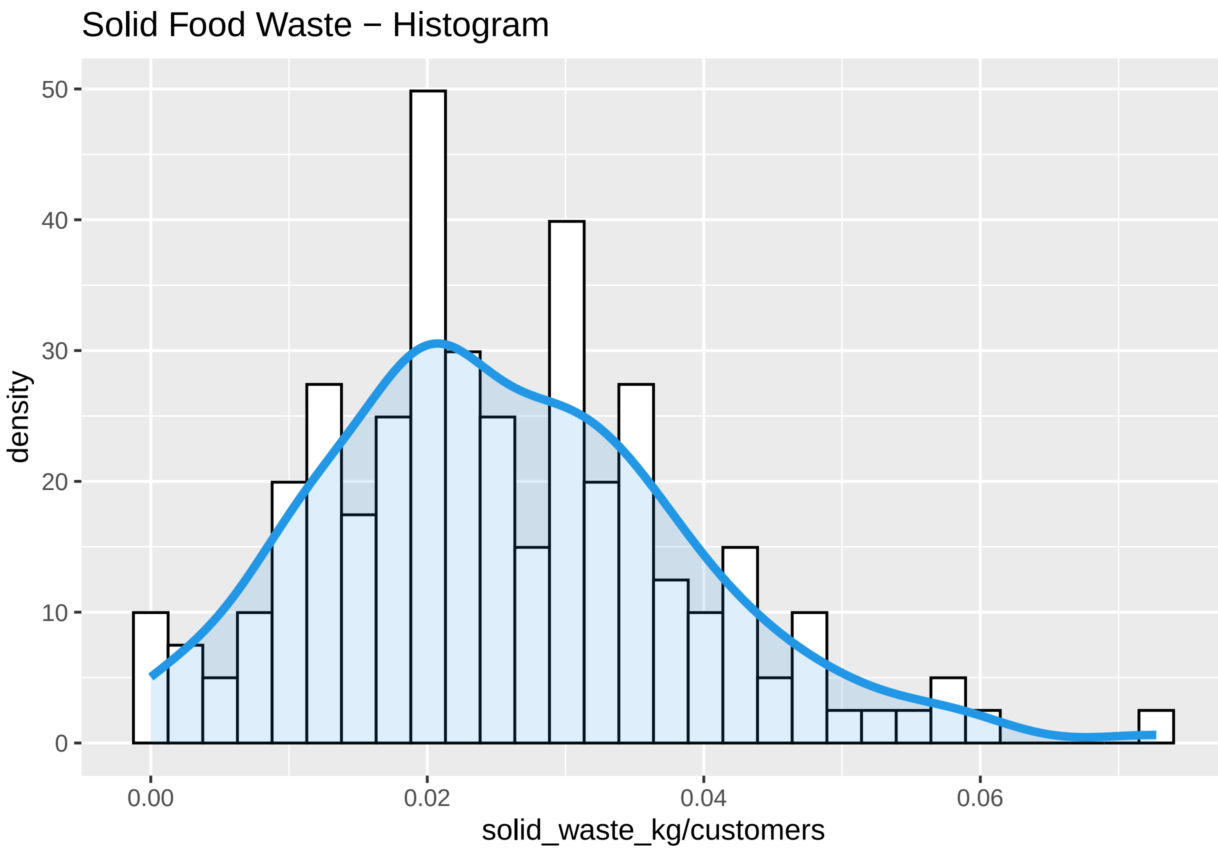
```
df[46,]$date
```

```
## [1] "2022-11-08"
```

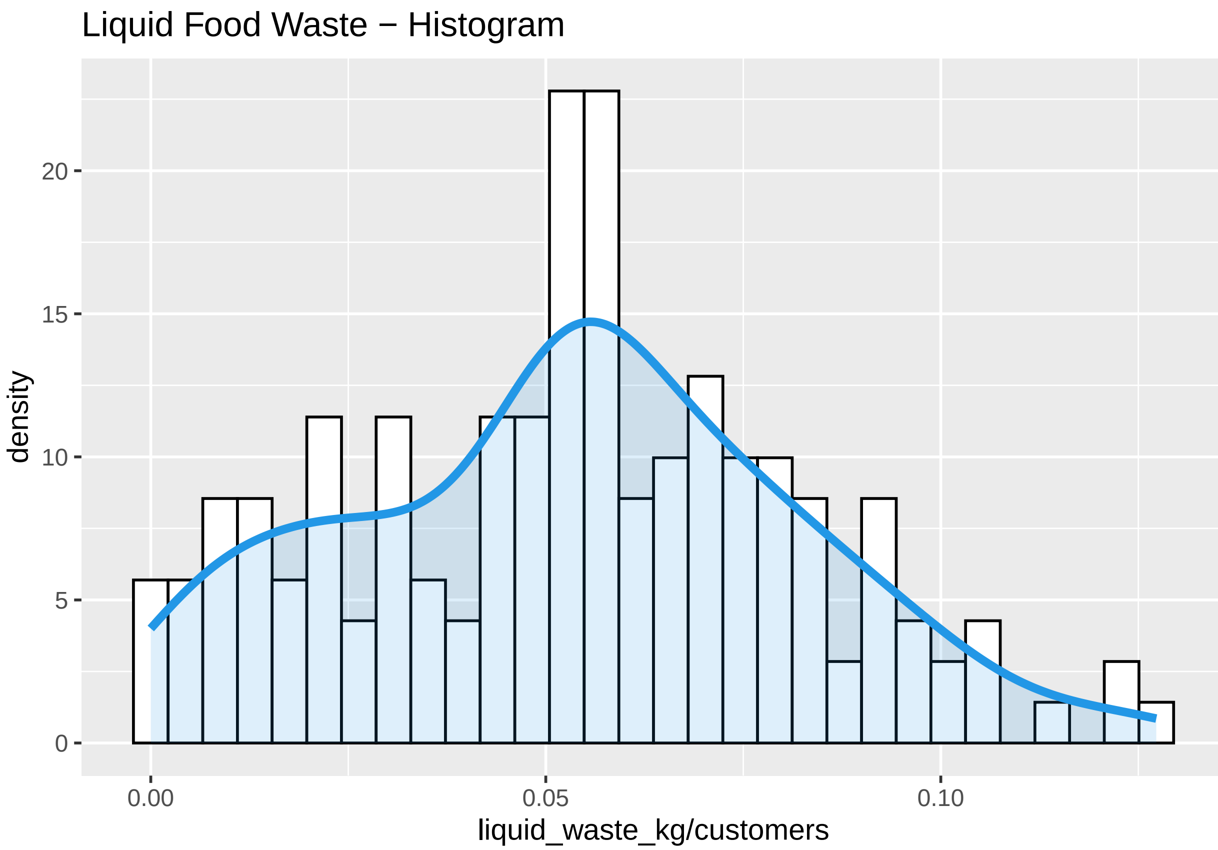
```
# outlier is 46; 2022-11-08

# Histogram of food waste -------------------------------------------
hist_food_waste <-
  df %>%
  filter(is_closed %in% FALSE) %>%
  filter(!row_number() %in% c(45)) %>%
  ggplot(aes(x = food_waste_kg/customers)) +
  geom_histogram(aes(y = after_stat(density)),
                 bins = 30,colour = 1, fill = "white") +
  geom_density(linewidth = 1.5, colour = 4, fill = 4, alpha = 0.15) +
  labs(title = "Food Waste - Histogram")
hist_food_waste
```



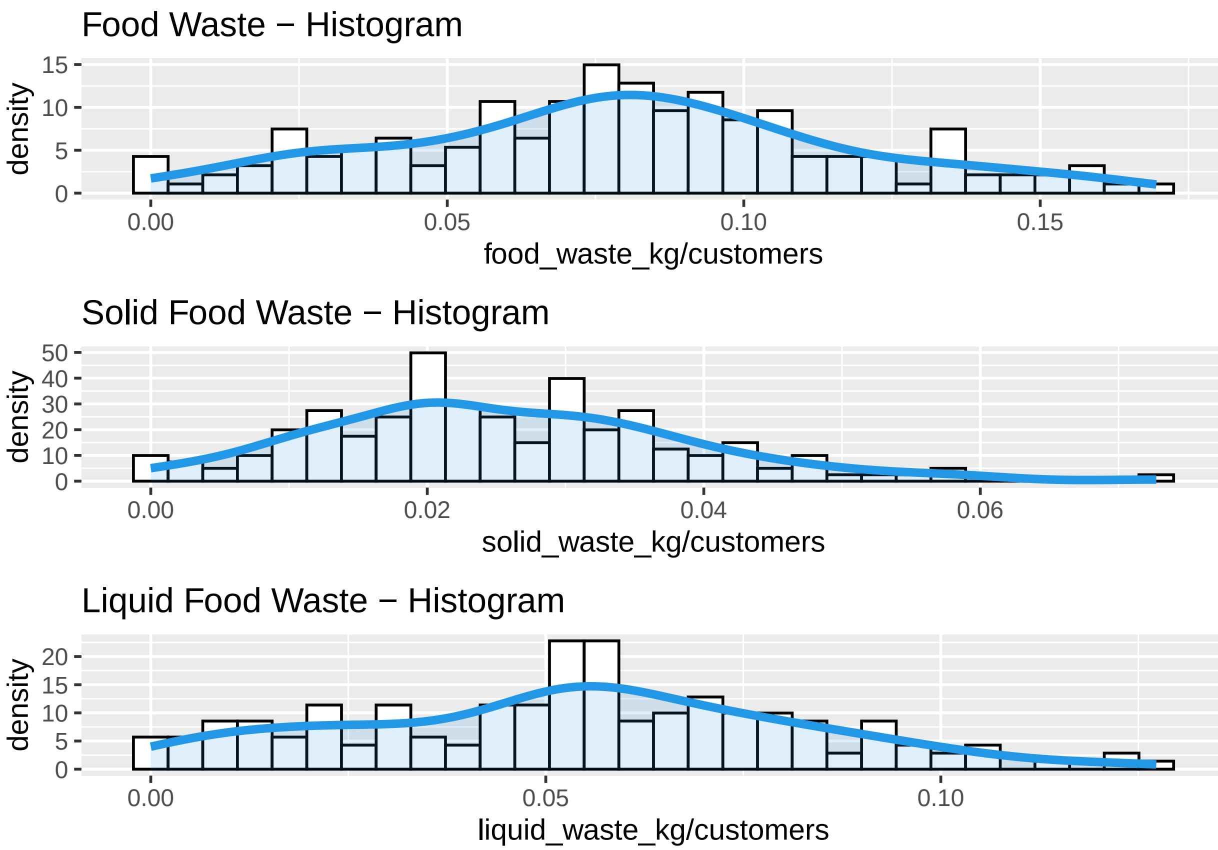

```
# Histogram of solid waste ------------------------------------------
hist_solid_waste <-
  df %>%
  filter(is_closed %in% FALSE) %>%
  filter(!row_number() %in% c(45)) %>%
  ggplot(aes(x = solid_waste_kg/customers)) +
  geom_histogram(aes(y = after_stat(density)),
                 bins = 30,colour = 1, fill = "white") +
  geom_density(linewidth = 1.5, colour = 4, fill = 4, alpha = 0.15) +
  labs(title = "Solid Food Waste - Histogram")
hist_solid_waste
```




```r
# Histogram of liquid waste ----------------------------------------
hist_liquid_waste <-
  df %>%
  filter(is_closed %in% FALSE) %>%
  filter(!row_number() %in% c(45)) %>%
  ggplot(aes(x = liquid_waste_kg/customers)) +
  geom_histogram(aes(y = after_stat(density)),
                 bins = 30,colour = 1, fill = "white") +
  geom_density(linewidth = 1.5, colour = 4, fill = 4, alpha = 0.15) +
  labs(title = "Liquid Food Waste - Histogram")
hist_liquid_waste
```




```
grid.arrange(hist_food_waste,hist_solid_waste,
             hist_liquid_waste)
```



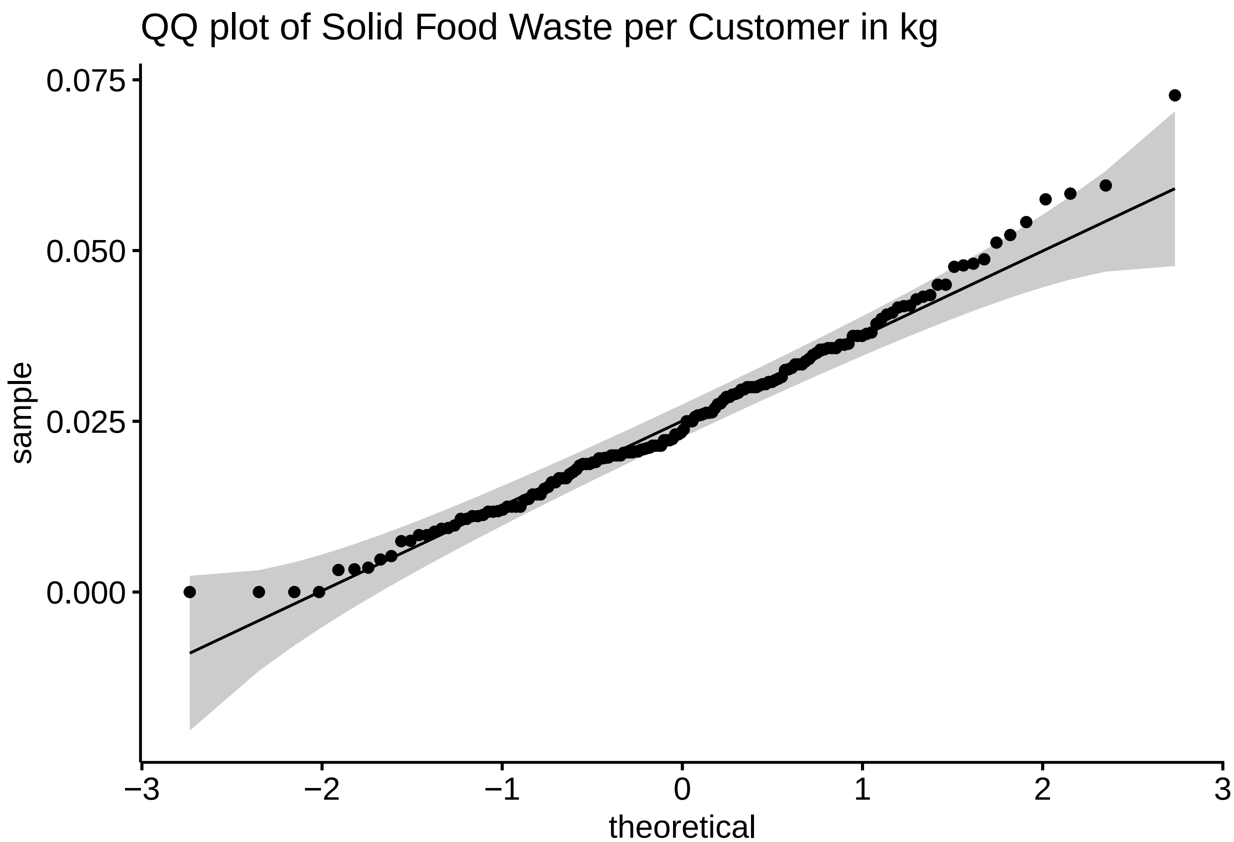

```
library(qqplotr)
```

**Q-Q plot per capita w/o outlier**

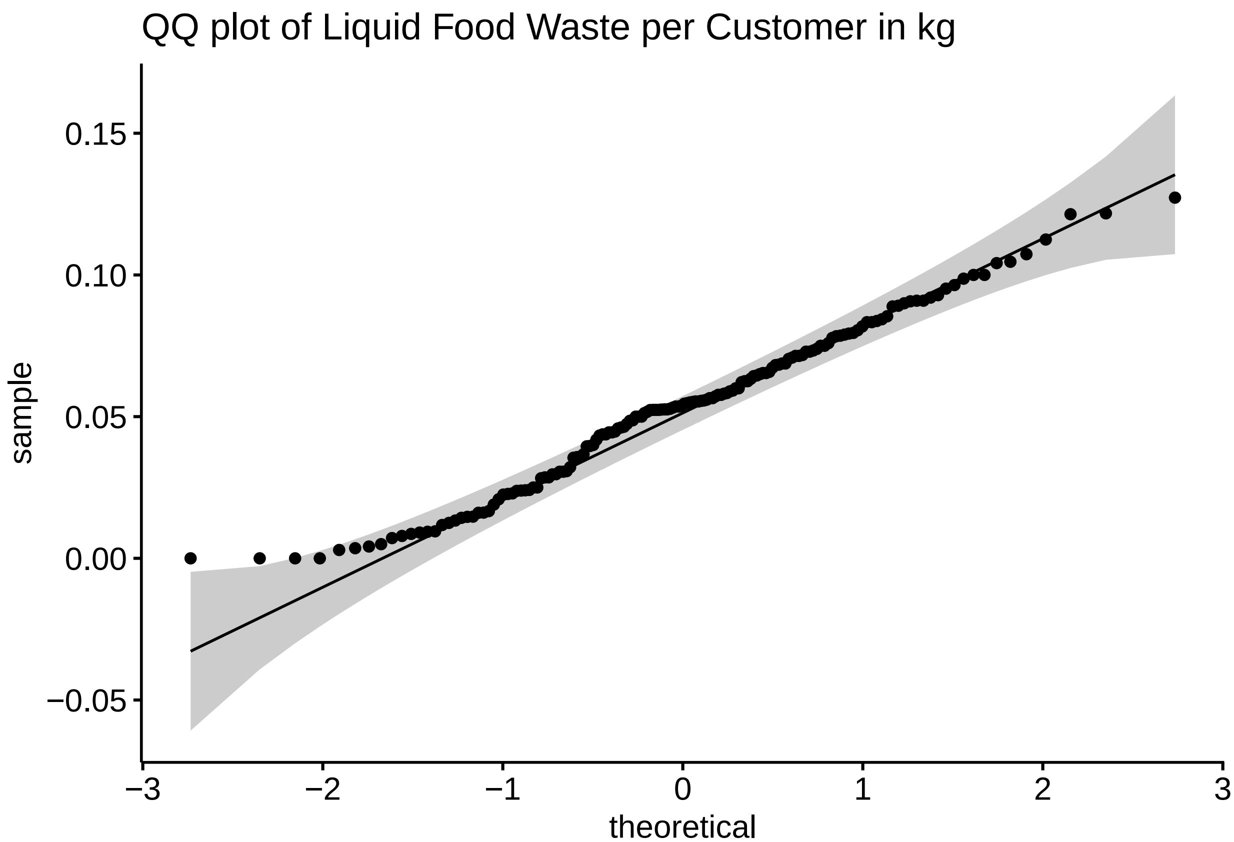
```
##
## Attaching package: 'qqplotr'

## The following objects are masked from 'package:ggplot2':
##
##     stat_qq_line, StatQqLine
```

```
# Food waste ------------
# df %>%
#   filter(is_closed == FALSE) %>%
#   filter(!row_number() %in% c(45)) %>%
#   ggplot(aes(sample = liquid_waste_kg/customers)) +
#   stat_qq() + stat_qq_line() +
ggqqplot(subset(df[-46,]$food_waste_kg/df[-46,]$customers,
                df[-46,]$is_closed %in% FALSE)) +
  xlab("theoretical") + ylab("sample") +
  ggtitle("QQ plot of Food Waste per Customer in kg")
```


```
# Solid Food waste ------------
ggqqplot(subset(df[-46,]$solid_waste_kg/df[-46,]$customers,
                df[-46,]$is_closed %in% FALSE)) +
  xlab("theoretical") + ylab("sample") +
  ggtitle("QQ plot of Solid Food Waste per Customer in kg")
```




```
# Liquid Food waste ------------
ggqqplot(subset(df[-46,]$liquid_waste_kg/df[-46,]$customers,
                df[-46,]$is_closed %in% FALSE)) +
  xlab("theoretical") + ylab("sample") +
  ggtitle("QQ plot of Liquid Food Waste per Customer in kg")
```




```
# Food waste ------------
df %>%
  filter(is_closed %in% FALSE) %>%
  filter(!row_number() == 45) %>%
  mutate(food_waste_p_kg   = food_waste_kg/customers,
         solid_waste_p_kg  = solid_waste_kg/customers,
         liquid_waste_p_kg = liquid_waste_kg/customers) %>%
  shapiro_test(food_waste_p_kg, solid_waste_p_kg, liquid_waste_p_kg)
```

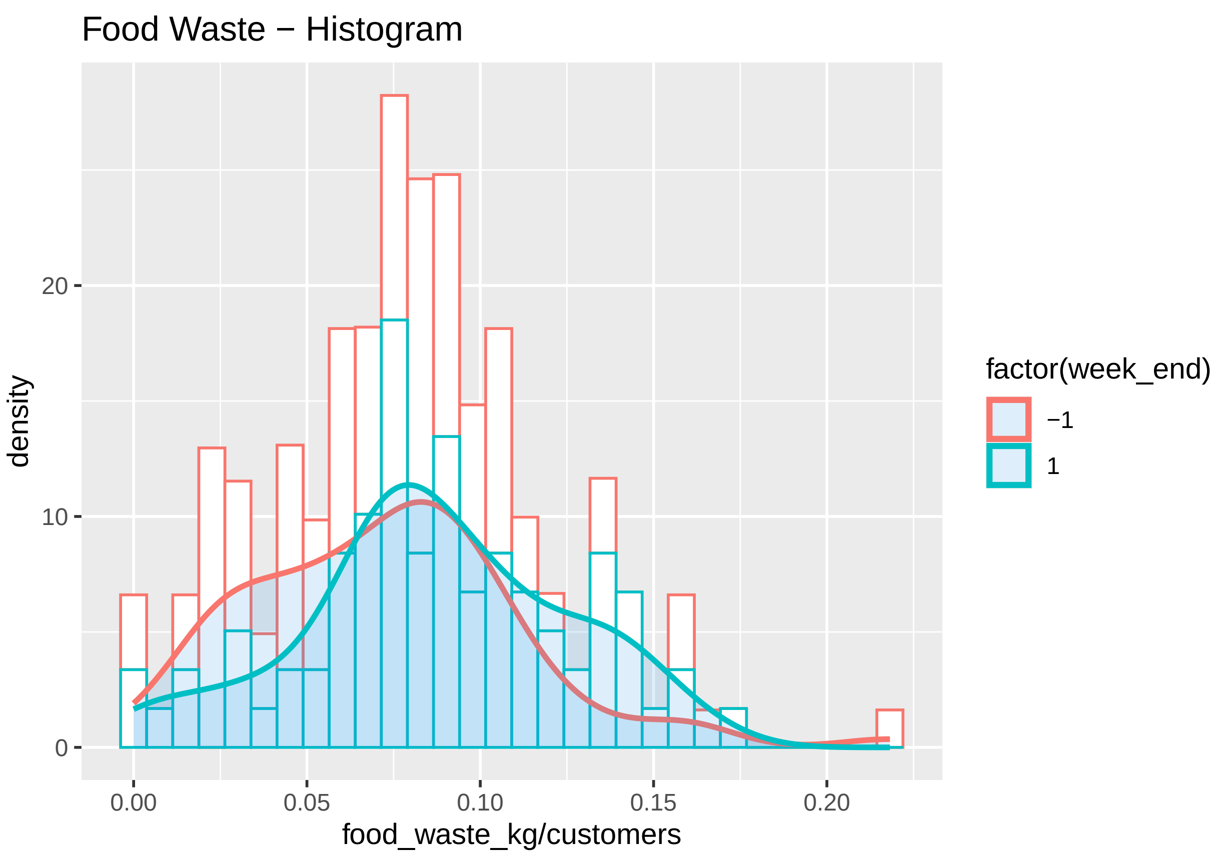
**shapiro test for per capita w/o outlier**

```
## # A tibble: 3 x 3
##   variable          statistic      p
##   <chr>                 <dbl>  <dbl>
## 1 food_waste_p_kg       0.988 0.210
## 2 liquid_waste_p_kg     0.984 0.0601
## 3 solid_waste_p_kg      0.980 0.0222
```

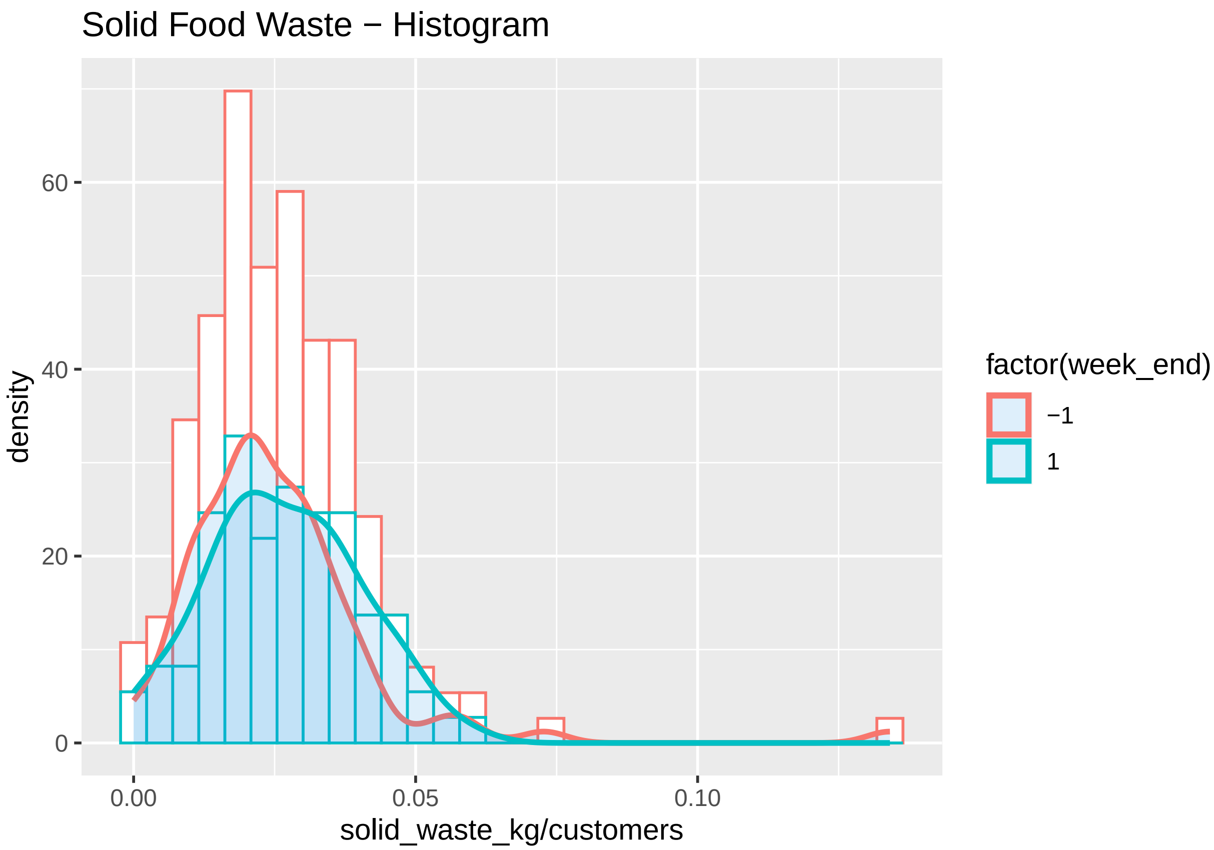
From the output, the p-value of solid food waste per customer is far less that the significant level of 0.05; but the others are not. So it imply that the distribution of the data for solid food waste per customer is significantly different from normal distribution. In other words, we can assume the normality for food waste and liquid food waste per customer but not for solid food waste.

**Histogram weekdays_ends**

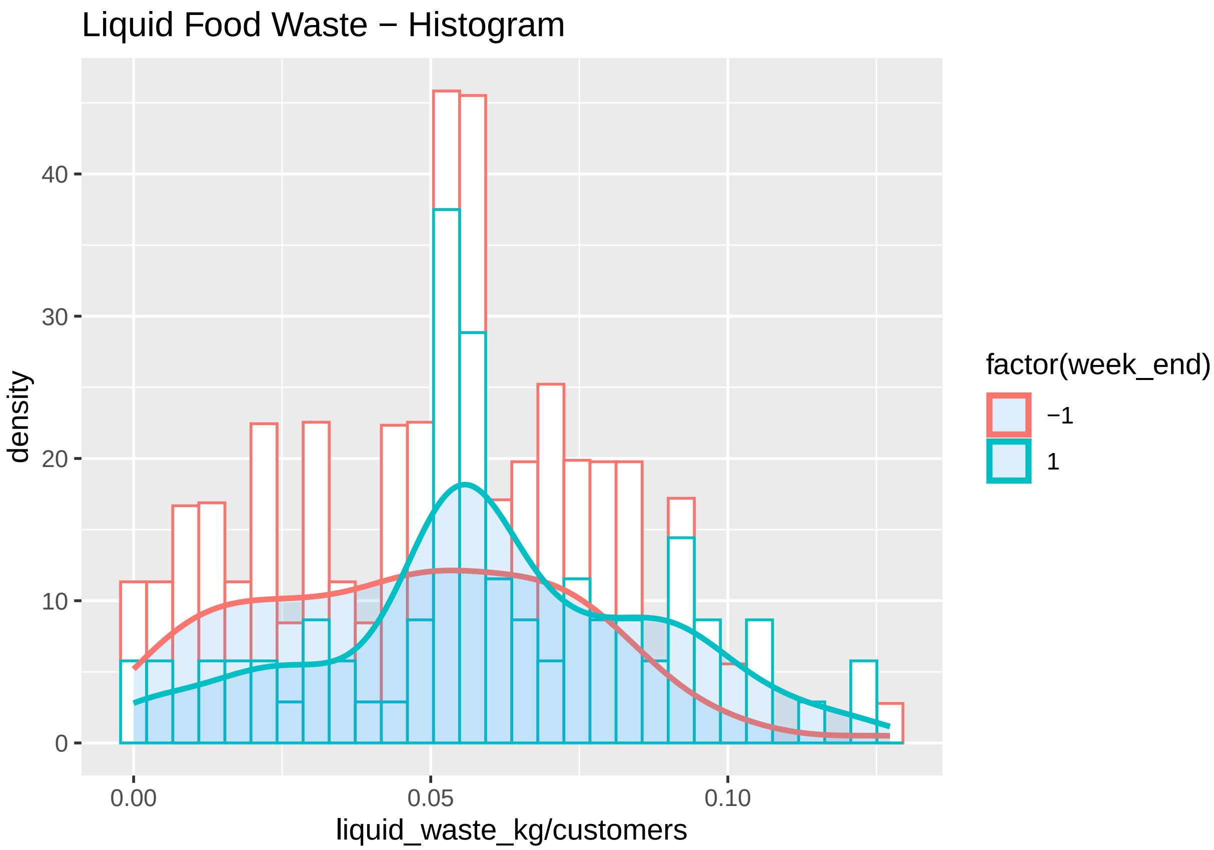
```
# Histogram of food waste ------------------------------------------
hist_food_waste_wk <-
  ggplot(data = subset(df, is_closed %in% FALSE),
         aes(x = food_waste_kg/customers,
             colour = factor(week_end))) +
  geom_histogram(aes(y = after_stat(density) ),
                 fill = "white", bins = 30) +
  geom_density(linewidth = 1, fill = 4, alpha = 0.15) +
  labs(title = "Food Waste - Histogram")
hist_food_waste_wk
```



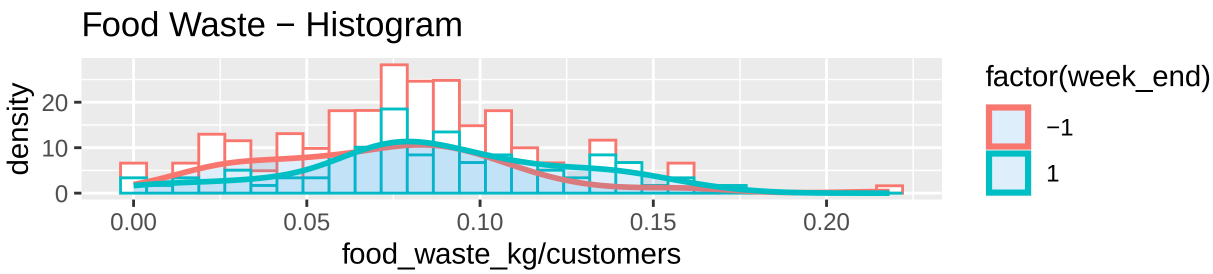

```
# Histogram of solid waste -------------------------------------------
hist_solid_waste_wk <-
  ggplot(data = subset(df, is_closed %in% FALSE),
         aes(x = solid_waste_kg/customers,
             colour = factor(week_end))) +
  geom_histogram(aes(y = after_stat(density)),
                 bins = 30, fill = "white") +
  geom_density(linewidth = 1, fill = 4, alpha = 0.15) +
  labs(title = "Solid Food Waste - Histogram")
hist_solid_waste_wk
```

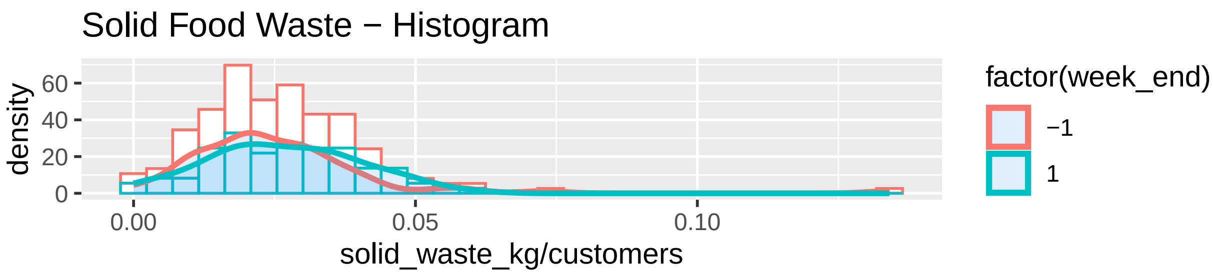


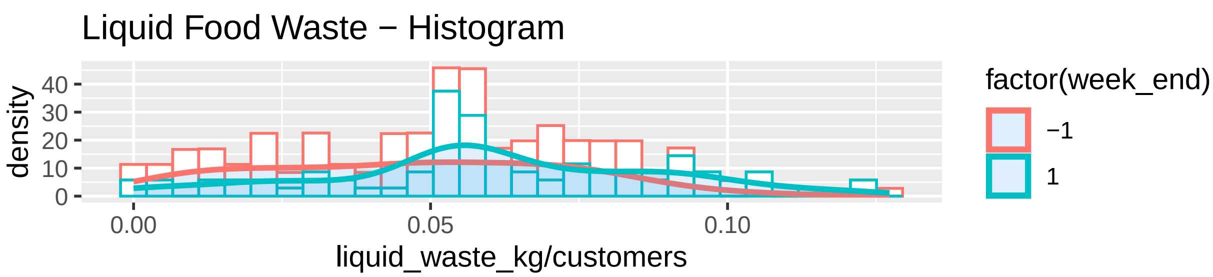

```
# Histogram of liquid waste ----------------------------------------
hist_liquid_waste_wk <-
  ggplot(data = subset(df, is_closed %in% FALSE),
         aes(x = liquid_waste_kg/customers,
             colour = factor(week_end))) +
  geom_histogram(aes(y = after_stat(density)),
                 bins = 30, fill = "white") +
  geom_density(linewidth = 1, fill = 4, alpha = 0.15) +
  labs(title = "Liquid Food Waste - Histogram")
hist_liquid_waste_wk
```




```
grid.arrange(hist_food_waste_wk,
             hist_solid_waste_wk,
             hist_liquid_waste_wk)
```



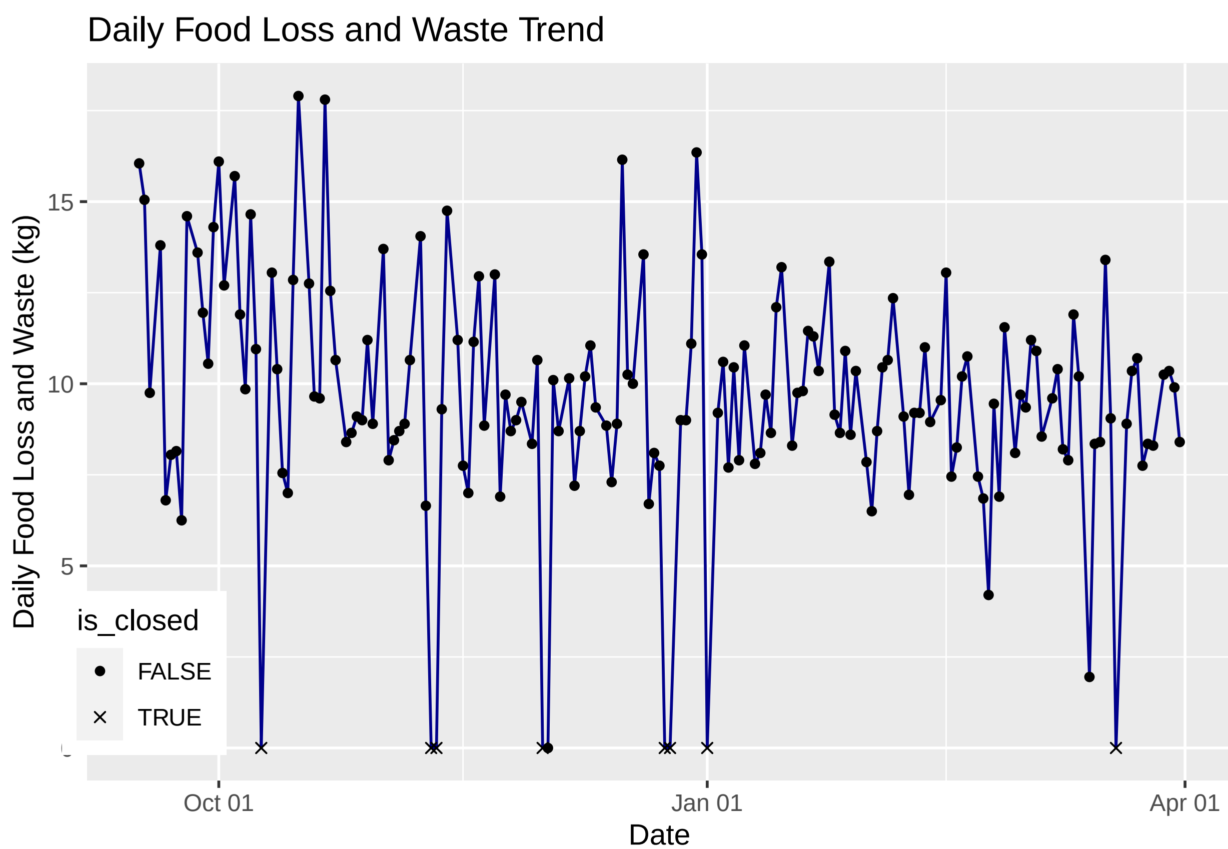



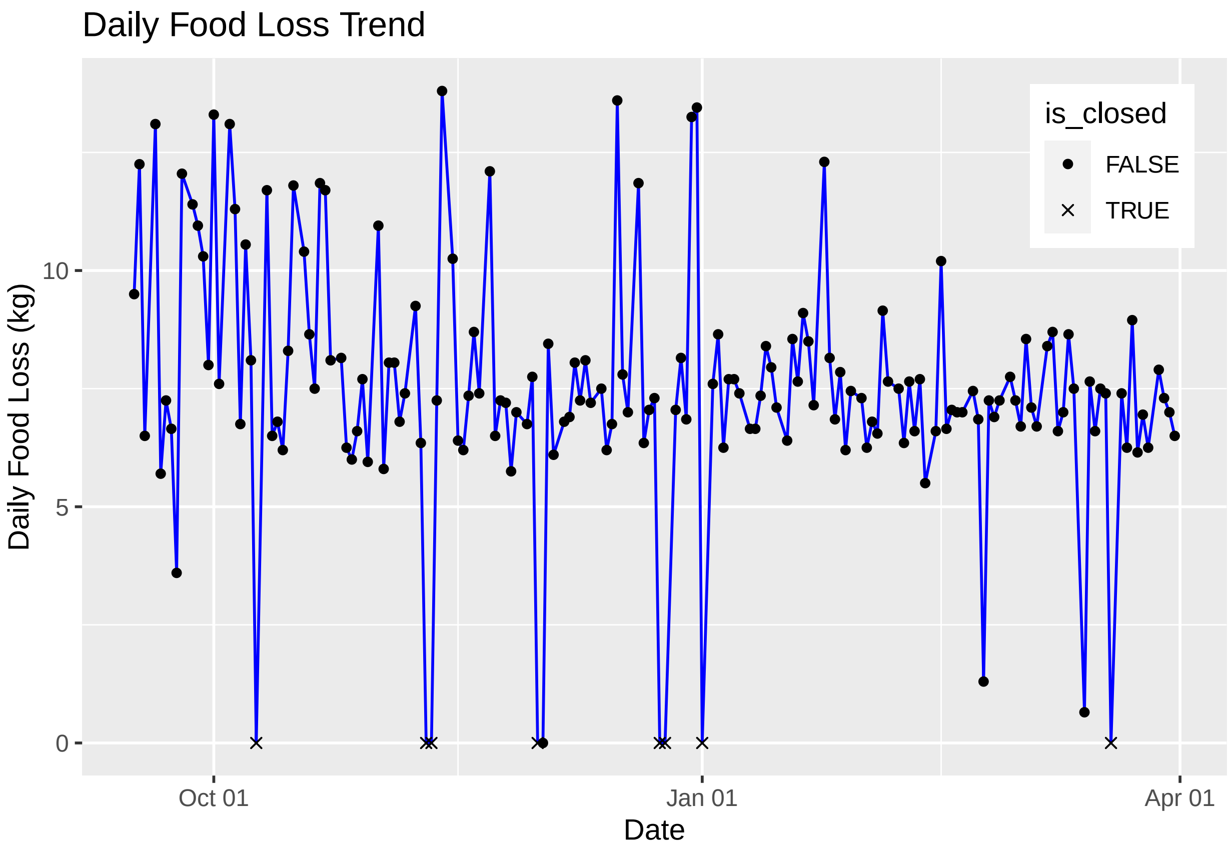

## Time Series Plots —

### Daily Time Series

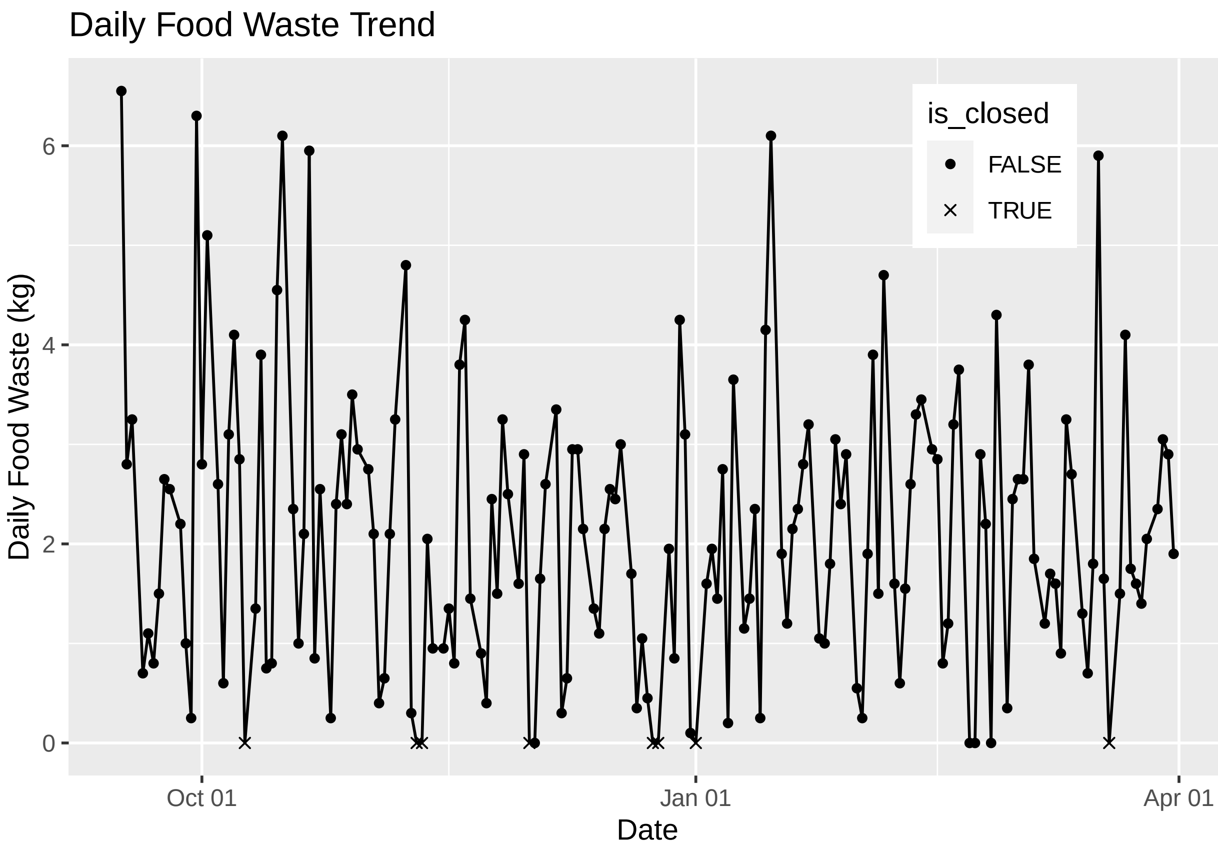
```
# Daily Plot on food loss + food waste ---------------------------------
daily_loss_waste <-
  ggplot(data = df, aes(x = as.Date(date),
                        y = food_loss_kg + food_waste_kg)) +
  geom_line(aes(group = 1), color="dark blue") +
  geom_point(aes(shape = is_closed)) +
  scale_shape_manual(values=c(16, 4)) +
  scale_x_date(date_labels = "%b %d") +
  theme(legend.position = c(0.05,0.15)) +
  xlab("Date") + ylab("Daily Food Loss and Waste (kg)") +
  ggtitle("Daily Food Loss and Waste Trend")
daily_loss_waste
```



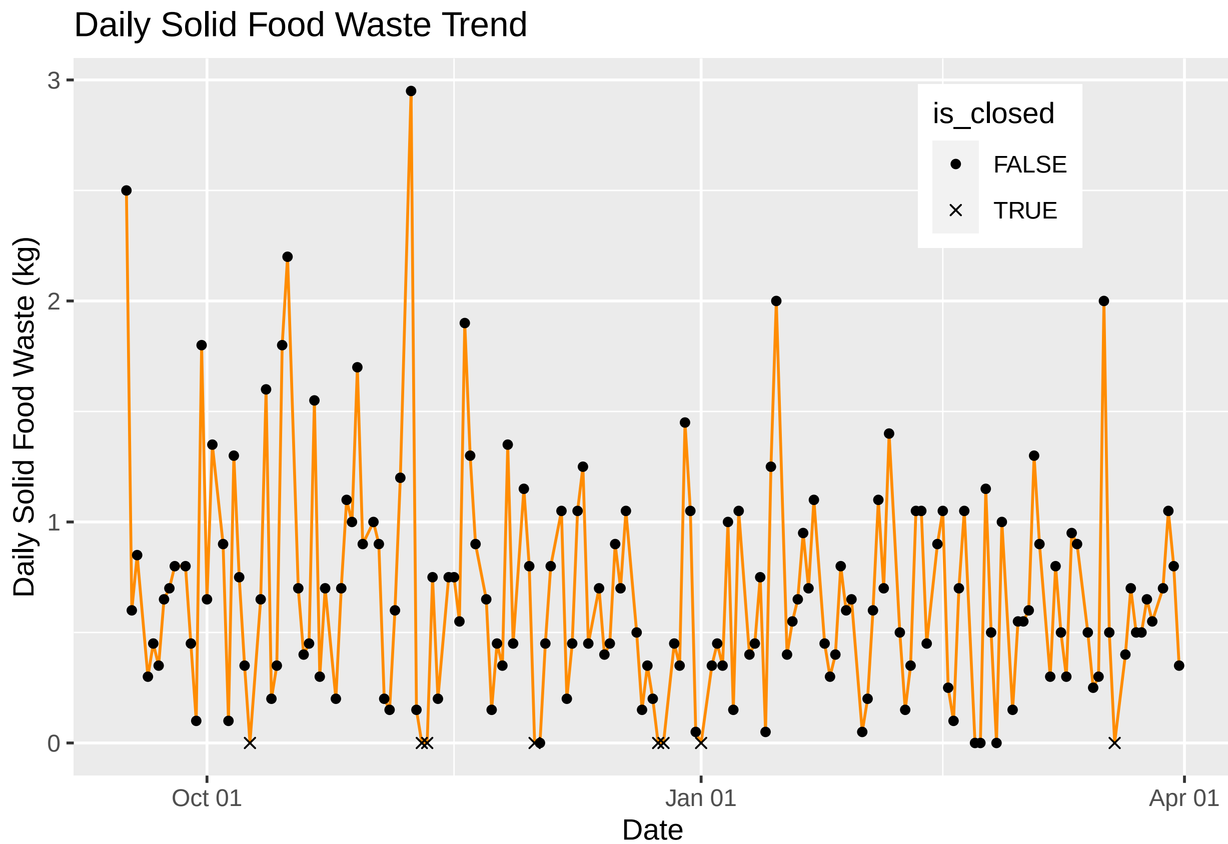

```
# Daily Plot on food loss ------------------------------------------------
daily_loss <-
  ggplot(data = df, aes(x = as.Date(date), y = food_loss_kg)) +
  geom_line(color="blue") +
  geom_point(aes(shape = is_closed)) +
  scale_x_date(date_labels = "%b %d") +
  scale_shape_manual(values=c(16, 4))+
  theme(legend.position = c(0.9,0.85)) +
  xlab("Date") + ylab("Daily Food Loss (kg)") +
  ggtitle("Daily Food Loss Trend")
daily_loss
```



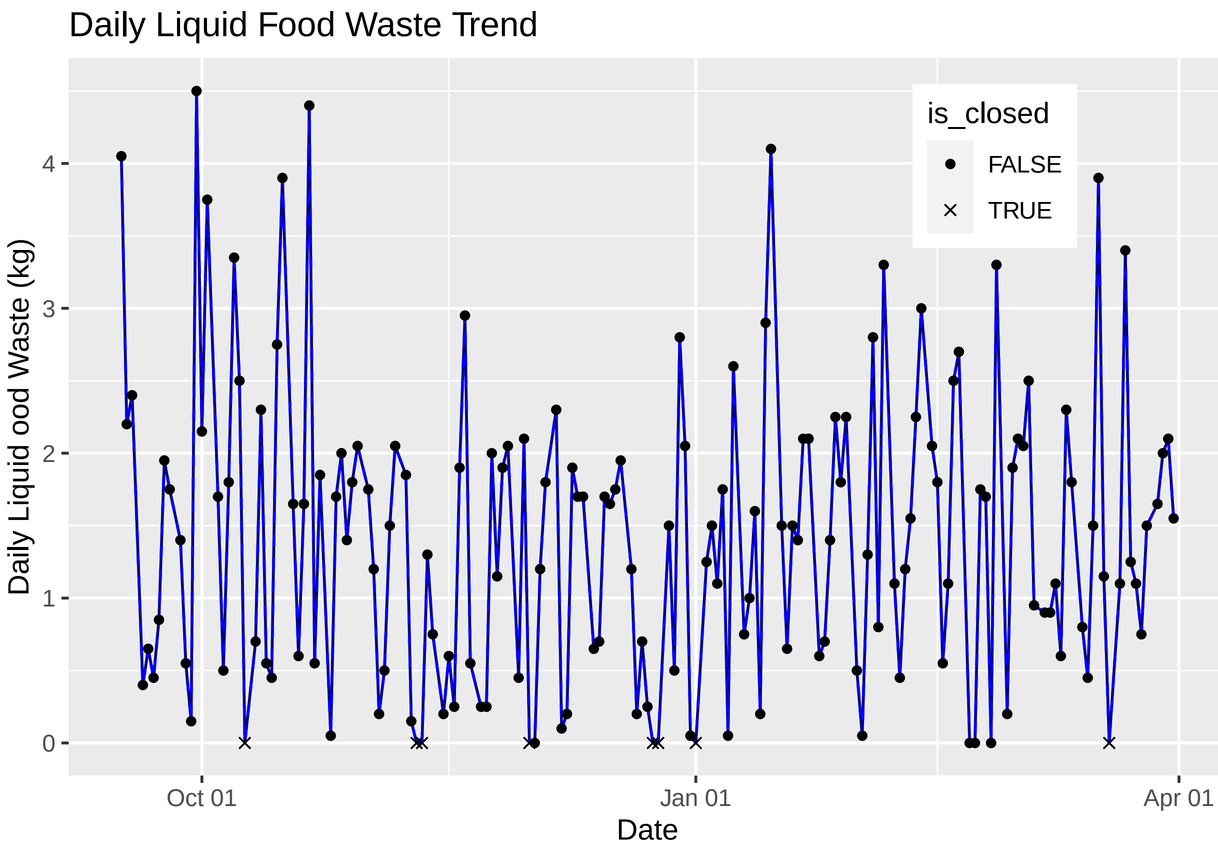

```r
# Daily Plot on food waste ----------------------------------------------
daily_waste <-
  ggplot(data = df, aes(x = as.Date(date), y = food_waste_kg)) +
  geom_line(color="black") +
  geom_point(aes(shape = is_closed)) +
  scale_x_date(date_labels = "%b %d") +
  scale_shape_manual(values=c(16, 4))+
  theme(legend.position = c(0.8,0.85)) +
  xlab("Date") + ylab("Daily Food Waste (kg)") +
  ggtitle("Daily Food Waste Trend")
daily_waste
```



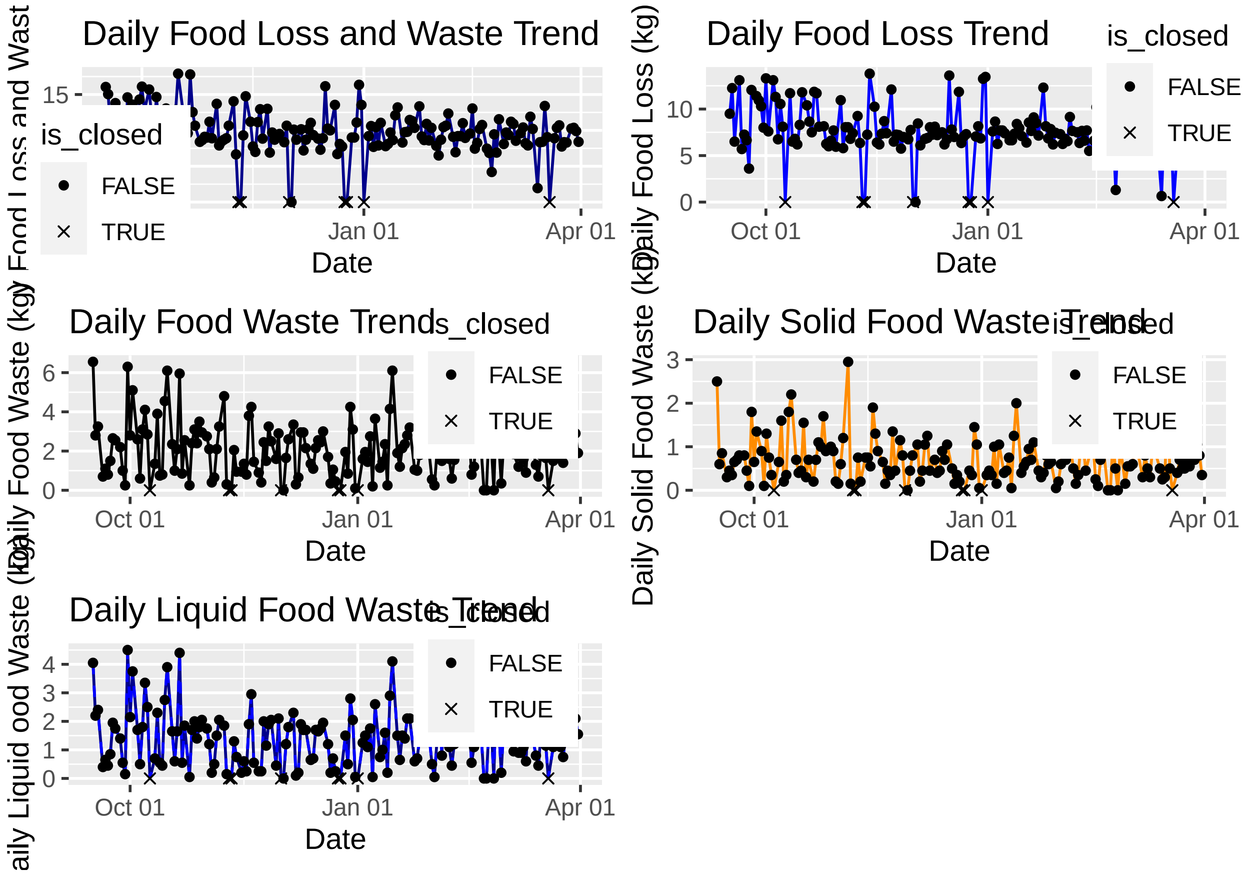

```r
# Daily Plot on solid food waste ----------------------------------------
daily_solid_waste <-
  ggplot(data = df, aes(x = as.Date(date), y = solid_waste_kg)) +
  geom_line(color="dark orange") +
  geom_point(aes(shape = is_closed)) +
  scale_x_date(date_labels = "%b %d") +
  scale_shape_manual(values=c(16, 4))+
  theme(legend.position = c(0.8,0.85)) +
  xlab("Date") + ylab("Daily Solid Food Waste (kg)") +
  ggtitle("Daily Solid Food Waste Trend")
daily_solid_waste
```




```
# Daily Plot on liquid food waste ----------------------------------------
daily_liquid_waste <-
  ggplot(data = df, aes(x = as.Date(date), y = liquid_waste_kg)) +
  geom_line(color="dark blue") +
  geom_line(color="blue", linetype = "dashed") +
  geom_point(aes(shape = is_closed)) +
  scale_x_date(date_labels = "%b %d") +
  scale_shape_manual(values=c(16, 4))+
  theme(legend.position = c(0.8,0.85)) +
  xlab("Date") + ylab("Daily Liquid ood Waste (kg)") +
  ggtitle("Daily Liquid Food Waste Trend")
daily_liquid_waste
```




```
grid.arrange(daily_loss_waste,daily_loss, daily_waste,
             daily_solid_waste,daily_liquid_waste)
```

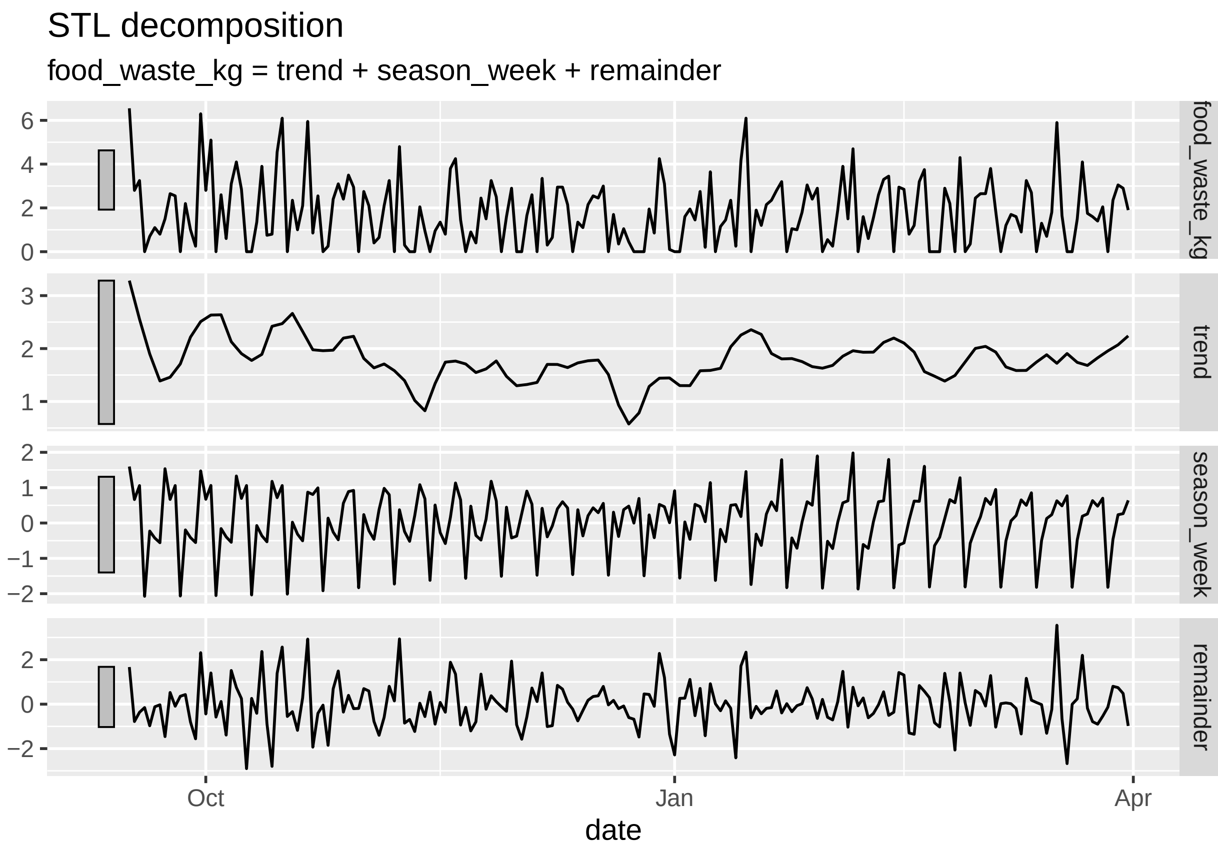



**Decompsiotion**

```
library(fpp3, seasonal)
```

```
## -- Attaching packages --------------------------------------------- fpp3 0.5 --

## v tsibble     1.1.3     v fable       0.3.3
## v tsibbledata 0.4.1     v fabletools  0.3.4
## v feasts      0.3.1

## -- Conflicts -------------------------------------------------- fpp3_conflicts --
## x dplyr::combine()        masks gridExtra::combine()
## x lubridate::date()       masks base::date()
## x rstatix::filter()       masks dplyr::filter(), stats::filter()
## x tsibble::intersect()    masks base::intersect()
## x tsibble::interval()     masks lubridate::interval()
## x dplyr::lag()            masks stats::lag()
## x fabletools::model()     masks bayesforecast::model()
## x tsibble::setdiff()      masks base::setdiff()
## x qqplotr::stat_qq_line() masks ggplot2::stat_qq_line()
## x tsibble::union()        masks base::union()
## x summarytools::view()    masks tibble::view()
```

```
df %>%
  as_tsibble(index = date) %>%
  select(food_waste_kg) %>%
  fill_gaps(food_waste_kg = 0) %>%
  model(STL(food_waste_kg)) |>
  components() |>
  autoplot()
```




```
auto.arima(df$food_waste_kg, trace = TRUE)
```

```
##
##  Fitting models using approximations to speed things up...
##
##  ARIMA(2,0,2) with non-zero mean : 595.2761
##  ARIMA(0,0,0) with non-zero mean : 607.2775
##  ARIMA(1,0,0) with non-zero mean : 598.3493
##  ARIMA(0,0,1) with non-zero mean : 606.2906
##  ARIMA(0,0,0) with zero mean     : 795.7987
##  ARIMA(1,0,2) with non-zero mean : 593.7226
##  ARIMA(0,0,2) with non-zero mean : 603.5818
##  ARIMA(1,0,1) with non-zero mean : 598.3892
##  ARIMA(1,0,3) with non-zero mean : 594.7845
##  ARIMA(0,0,3) with non-zero mean : 602.7266
##  ARIMA(2,0,1) with non-zero mean : 593.1346
##  ARIMA(2,0,0) with non-zero mean : 593.03
##  ARIMA(3,0,0) with non-zero mean : 591.0829
```

```
##  ARIMA(4,0,0) with non-zero mean : 593.9004
##  ARIMA(3,0,1) with non-zero mean : 593.1032
##  ARIMA(4,0,1) with non-zero mean : 594.6705
##  ARIMA(3,0,0) with zero mean     : 655.5828
##
##  Now re-fitting the best model(s) without approximations...
##
##  ARIMA(3,0,0) with non-zero mean : 600.6932
##
##  Best model: ARIMA(3,0,0) with non-zero mean
```

```
## Series: df$food_waste_kg
## ARIMA(3,0,0) with non-zero mean
##
## Coefficients:
##          ar1      ar2      ar3    mean
##       0.1053  -0.2083  -0.1262  2.0746
## s.e.  0.0788   0.0769   0.0786  0.0871
##
## sigma^2 = 1.97:  log likelihood = -295.16
## AIC=600.33   AICc=600.69   BIC=615.97
```

```
auto.arima(df$solid_waste_kg, trace = TRUE)
```

```
##
##  Fitting models using approximations to speed things up...
##
##  ARIMA(2,0,2) with non-zero mean : 242.2204
##  ARIMA(0,0,0) with non-zero mean : 254.9591
##  ARIMA(1,0,0) with non-zero mean : 242.9804
##  ARIMA(0,0,1) with non-zero mean : 254.9337
##  ARIMA(0,0,0) with zero mean     : 424.4576
##  ARIMA(1,0,2) with non-zero mean : 240.5345
##  ARIMA(0,0,2) with non-zero mean : 253.0456
##  ARIMA(1,0,1) with non-zero mean : 242.4608
##  ARIMA(1,0,3) with non-zero mean : 241.1252
##  ARIMA(0,0,3) with non-zero mean : 252.9766
##  ARIMA(2,0,1) with non-zero mean : 240.7382
##  ARIMA(2,0,3) with non-zero mean : 243.1306
##  ARIMA(1,0,2) with zero mean     : 290.294
##
##  Now re-fitting the best model(s) without approximations...
##
##  ARIMA(1,0,2) with non-zero mean : 252.8433
##
##  Best model: ARIMA(1,0,2) with non-zero mean
```

```
## Series: df$solid_waste_kg
## ARIMA(1,0,2) with non-zero mean
##
## Coefficients:
##          ar1      ma1      ma2    mean
##       0.3933  -0.3011  -0.2195  0.6723
```

```
## s.e.  0.2334   0.2269   0.0728  0.0303
##
## sigma^2 = 0.2516:  log likelihood = -121.24
## AIC=252.48   AICc=252.84   BIC=268.12
```

```r
auto.arima(df$liquid_waste_kg, trace = TRUE)
```

```
##
##  Fitting models using approximations to speed things up...
##
##  ARIMA(2,0,2) with non-zero mean : 481.848
##  ARIMA(0,0,0) with non-zero mean : 489.7931
##  ARIMA(1,0,0) with non-zero mean : 483.6428
##  ARIMA(0,0,1) with non-zero mean : 488.6056
##  ARIMA(0,0,0) with zero mean     : 668.5145
##  ARIMA(1,0,2) with non-zero mean : 481.4292
##  ARIMA(0,0,2) with non-zero mean : 487.558
##  ARIMA(1,0,1) with non-zero mean : 484.5832
##  ARIMA(1,0,3) with non-zero mean : 482.8695
##  ARIMA(0,0,3) with non-zero mean : 487.0004
##  ARIMA(2,0,1) with non-zero mean : 480.5155
##  ARIMA(2,0,0) with non-zero mean : 480.0232
##  ARIMA(3,0,0) with non-zero mean : 478.3711
##  ARIMA(4,0,0) with non-zero mean : 480.7297
##  ARIMA(3,0,1) with non-zero mean : 480.1401
##  ARIMA(4,0,1) with non-zero mean : 479.0072
##  ARIMA(3,0,0) with zero mean     : 539.5893
##
##  Now re-fitting the best model(s) without approximations...
##
##  ARIMA(3,0,0) with non-zero mean : 484.9027
##
##  Best model: ARIMA(3,0,0) with non-zero mean
```

```
## Series: df$liquid_waste_kg
## ARIMA(3,0,0) with non-zero mean
##
## Coefficients:
##          ar1      ar2     ar3    mean
##       0.1128  -0.1804  -0.124  1.4030
## s.e.  0.0780   0.0767   0.078  0.0638
##
## sigma^2 = 0.9932:  log likelihood = -237.27
## AIC=484.53   AICc=484.9   BIC=500.18
```

```r
auto.arima(df[1:92,]$food_waste_kg, trace = TRUE)
```

```
##
##  ARIMA(2,1,2) with drift         : Inf
##  ARIMA(0,1,0) with drift         : 382.2608
##  ARIMA(1,1,0) with drift         : 376.6995
##  ARIMA(0,1,1) with drift         : Inf
```

```
##  ARIMA(0,1,0)                    : 380.2918
##  ARIMA(2,1,0) with drift         : 371.6764
##  ARIMA(3,1,0) with drift         : 361.5494
##  ARIMA(4,1,0) with drift         : 358.102
##  ARIMA(5,1,0) with drift         : 360.2444
##  ARIMA(4,1,1) with drift         : Inf
##  ARIMA(3,1,1) with drift         : Inf
##  ARIMA(5,1,1) with drift         : Inf
##  ARIMA(4,1,0)                    : 355.9381
##  ARIMA(3,1,0)                    : 359.4474
##  ARIMA(5,1,0)                    : 358.0249
##  ARIMA(4,1,1)                    : 344.9549
##  ARIMA(3,1,1)                    : 342.6938
##  ARIMA(2,1,1)                    : 343.3855
##  ARIMA(3,1,2)                    : 344.9619
##  ARIMA(2,1,0)                    : 369.616
##  ARIMA(2,1,2)                    : 342.9415
##  ARIMA(4,1,2)                    : 347.2447
##
##  Best model: ARIMA(3,1,1)
```

```
## Series: df[1:92, ]$food_waste_kg
## ARIMA(3,1,1)
##
## Coefficients:
##          ar1      ar2      ar3      ma1
##       0.1433  -0.1843  -0.1961  -0.9352
## s.e.  0.1118   0.1076   0.1129   0.0380
##
## sigma^2 = 2.284:  log likelihood = -165.99
## AIC=341.99   AICc=342.69   BIC=354.54
```

```
auto.arima(df[1:92,]$solid_waste_kg, trace = TRUE)
```

```
##
##  ARIMA(2,0,2) with non-zero mean : 165.4809
##  ARIMA(0,0,0) with non-zero mean : 162.51
##  ARIMA(1,0,0) with non-zero mean : 163.1611
##  ARIMA(0,0,1) with non-zero mean : 162.7369
##  ARIMA(0,0,0) with zero mean     : 247.8297
##  ARIMA(1,0,1) with non-zero mean : 164.7709
##
##  Best model: ARIMA(0,0,0) with non-zero mean
```

```
## Series: df[1:92, ]$solid_waste_kg
## ARIMA(0,0,0) with non-zero mean
##
## Coefficients:
##         mean
##       0.7207
## s.e.  0.0597
##
## sigma^2 = 0.3311:  log likelihood = -79.19
## AIC=162.38   AICc=162.51   BIC=167.42
```

```r
auto.arima(df[1:92,]$liquid_waste_kg, trace = TRUE)
```

```
##
##  ARIMA(2,1,2) with drift         : Inf
##  ARIMA(0,1,0) with drift         : 315.6767
##  ARIMA(1,1,0) with drift         : 309.1532
##  ARIMA(0,1,1) with drift         : Inf
##  ARIMA(0,1,0)                    : 313.6831
##  ARIMA(2,1,0) with drift         : 303.7267
##  ARIMA(3,1,0) with drift         : 292.6036
##  ARIMA(4,1,0) with drift         : 287.7742
##  ARIMA(5,1,0) with drift         : 289.7147
##  ARIMA(4,1,1) with drift         : Inf
##  ARIMA(3,1,1) with drift         : Inf
##  ARIMA(5,1,1) with drift         : Inf
##  ARIMA(4,1,0)                    : 285.6019
##  ARIMA(3,1,0)                    : 290.4933
##  ARIMA(5,1,0)                    : 287.4865
##  ARIMA(4,1,1)                    : 278.0896
##  ARIMA(3,1,1)                    : 275.979
##  ARIMA(2,1,1)                    : 276.3443
##  ARIMA(3,1,2)                    : 278.1815
##  ARIMA(2,1,0)                    : 301.653
##  ARIMA(2,1,2)                    : 277.205
##  ARIMA(4,1,2)                    : 280.5544
##
##  Best model: ARIMA(3,1,1)
```

```
## Series: df[1:92, ]$liquid_waste_kg
## ARIMA(3,1,1)
##
## Coefficients:
##          ar1      ar2      ar3      ma1
##       0.1304  -0.1809  -0.1865  -0.9185
## s.e.  0.1141   0.1076   0.1145   0.0510
##
## sigma^2 = 1.101:  log likelihood = -132.64
## AIC=275.27   AICc=275.98   BIC=287.83
```

```r
auto.arima(df[93:169,]$food_waste_kg, trace = TRUE)
```

```
##
##  ARIMA(2,0,2) with non-zero mean : Inf
##  ARIMA(0,0,0) with non-zero mean : 264.1095
##  ARIMA(1,0,0) with non-zero mean : 266.2064
##  ARIMA(0,0,1) with non-zero mean : 266.0714
##  ARIMA(0,0,0) with zero mean     : 360.2653
##  ARIMA(1,0,1) with non-zero mean : Inf
##
##  Best model: ARIMA(0,0,0) with non-zero mean
```

```
## Series: df[93:169, ]$food_waste_kg
```

```
## ARIMA(0,0,0) with non-zero mean
##
## Coefficients:
##         mean
##       2.1032
## s.e.  0.1491
##
## sigma^2 = 1.735:  log likelihood = -129.97
## AIC=263.95   AICc=264.11   BIC=268.63
```

```
auto.arima(df[93:169,]$solid_waste_kg, trace = TRUE)
```

```
##
##  ARIMA(2,0,2) with non-zero mean : 86.42921
##  ARIMA(0,0,0) with non-zero mean : 86.32735
##  ARIMA(1,0,0) with non-zero mean : 88.43897
##  ARIMA(0,0,1) with non-zero mean : 88.33825
##  ARIMA(0,0,0) with zero mean     : 174.9761
##  ARIMA(1,0,1) with non-zero mean : Inf
##
##  Best model: ARIMA(0,0,0) with non-zero mean
```

```
## Series: df[93:169, ]$solid_waste_kg
## ARIMA(0,0,0) with non-zero mean
##
## Coefficients:
##         mean
##       0.6188
## s.e.  0.0470
##
## sigma^2 = 0.1724:  log likelihood = -41.08
## AIC=86.17   AICc=86.33   BIC=90.85
```

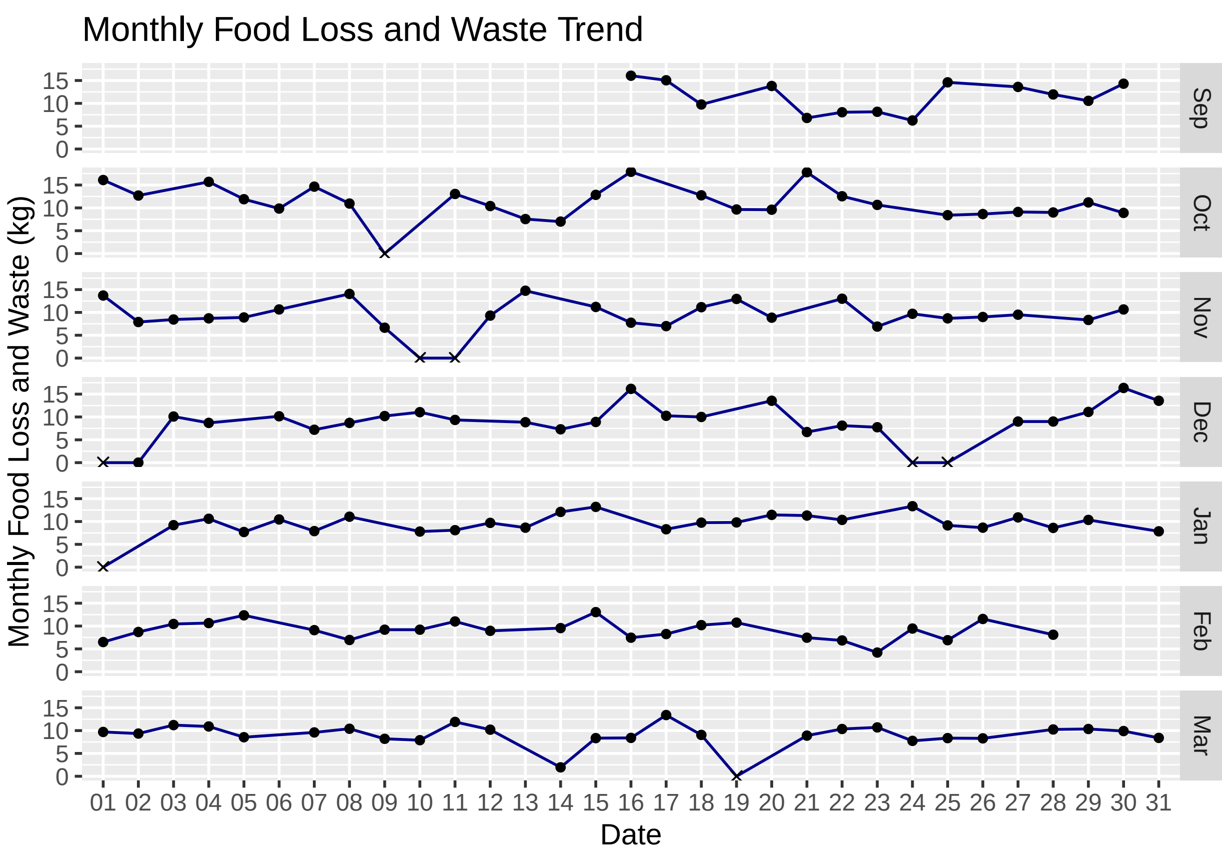
```
auto.arima(df[93:169,]$liquid_waste_kg, trace = TRUE)
```

```
##
##  ARIMA(2,0,2) with non-zero mean : Inf
##  ARIMA(0,0,0) with non-zero mean : 214.2947
##  ARIMA(1,0,0) with non-zero mean : 216.4005
##  ARIMA(0,0,1) with non-zero mean : 216.3053
##  ARIMA(0,0,0) with zero mean     : 307.6959
##  ARIMA(1,0,1) with non-zero mean : Inf
##
##  Best model: ARIMA(0,0,0) with non-zero mean
```

```
## Series: df[93:169, ]$liquid_waste_kg
## ARIMA(0,0,0) with non-zero mean
##
## Coefficients:
##         mean
##       1.4844
## s.e.  0.1079
```

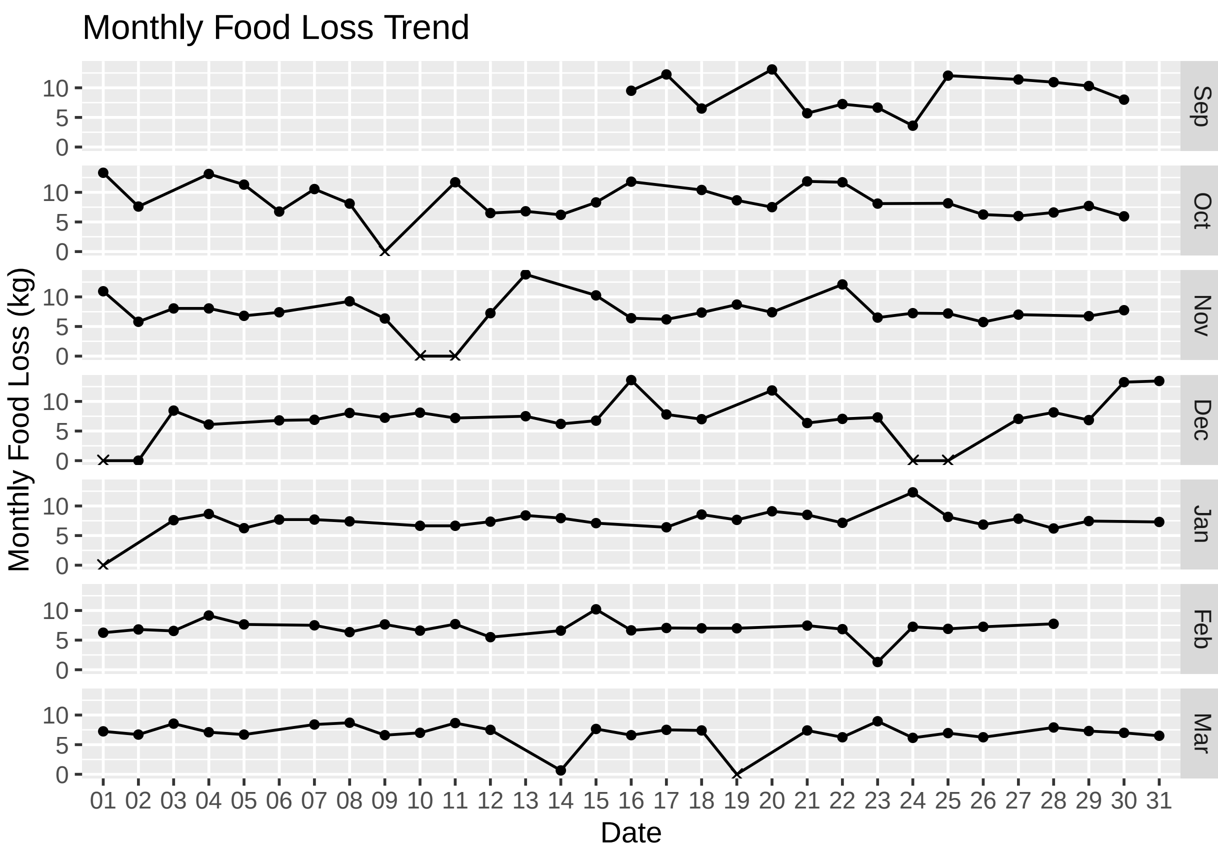
```
##
## sigma^2 = 0.9086:  log likelihood = -105.07
## AIC=214.13   AICc=214.29   BIC=218.82
```

```r
# Monthly Plot on food loss and food waste --------------------------------
monthly_loss_waste <-
  ggplot(data = df, aes(x = day_name,
                        y = food_loss_kg + food_waste_kg, group=1)) +
  geom_line(color="dark blue") +
  geom_point(aes(shape = is_closed)) +
  scale_shape_manual(values=c(16, 4)) +
  theme(legend.position = "none") +
  # geom_rect(data = df, aes(xmin = date, xmax = dplyr::lead(date),
  #                          ymin = -Inf, ymax = Inf,
  #                          fill = factor(!is_closed)),  alpha = .3) +
  facet_grid(month_name~.) +
  xlab("Date") + ylab("Monthly Food Loss and Waste (kg)") +
  ggtitle("Monthly Food Loss and Waste Trend")
monthly_loss_waste
```

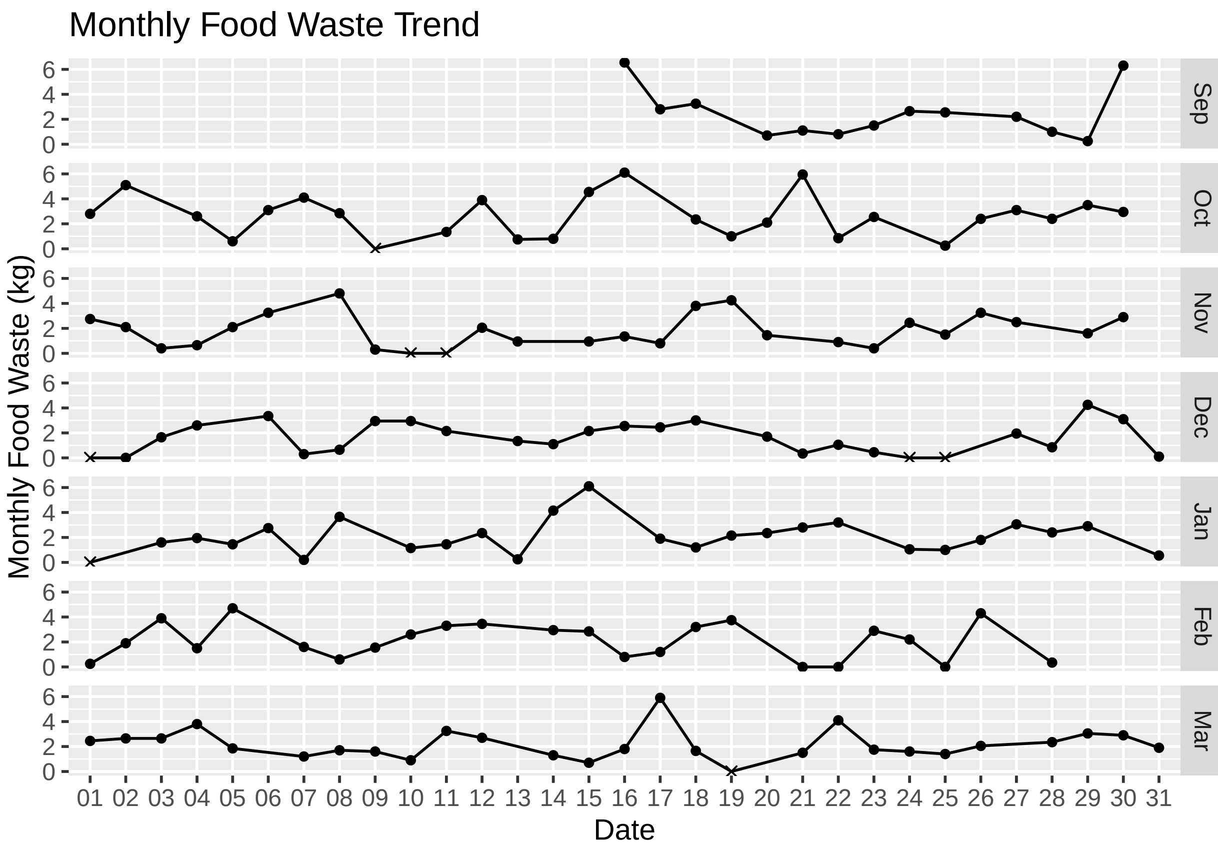



```r
# Monthly Plot on food loss -----------------------------------------------
monthly_loss <-
  ggplot(data = df, aes(x = day_name, y = food_loss_kg, group=1)) +
  geom_line(color="black") +
  geom_point(aes(shape = is_closed)) +
  facet_grid(month_name~.) +
```

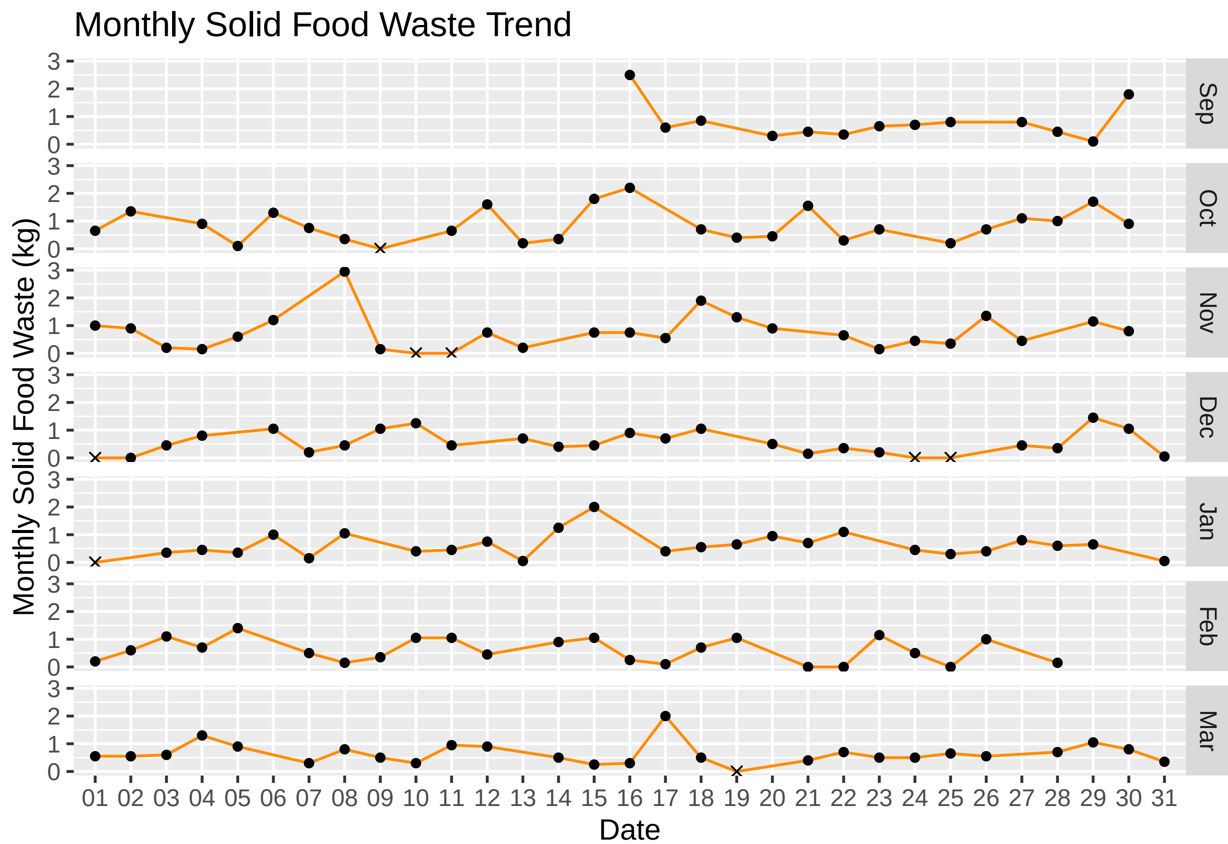
```
  scale_shape_manual(values=c(16, 4))+
  theme(legend.position = "none") +
  xlab("Date") + ylab("Monthly Food Loss (kg)") +
  ggtitle("Monthly Food Loss Trend")
monthly_loss
```



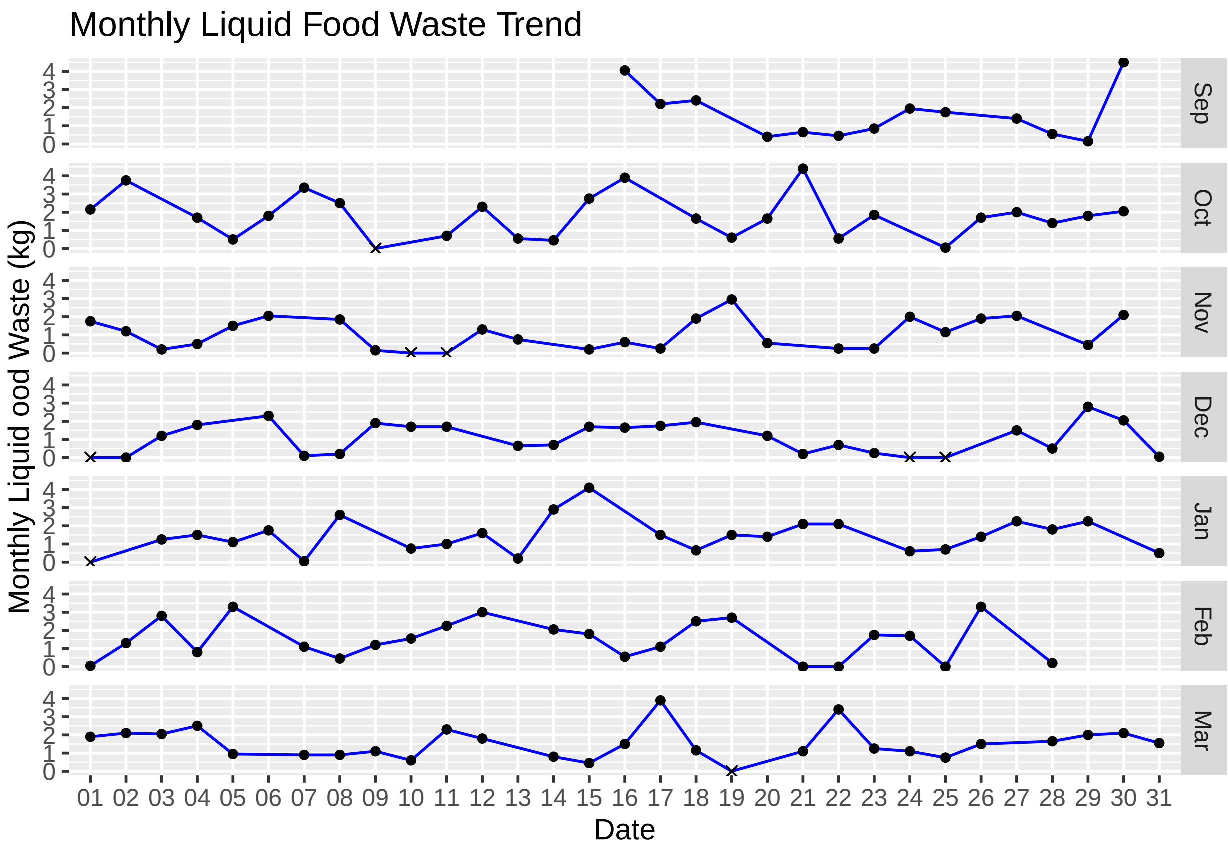

```
# Monthly Plot on food waste ------------------------------------------------
monthly_waste <-
  ggplot(data = df, aes(x = day_name, y = food_waste_kg, group=1)) +
  geom_line(color="black") +
  geom_point(aes(shape = is_closed)) +
  facet_grid(month_name~.) +
  scale_shape_manual(values=c(16, 4))+
  theme(legend.position = "none") +
  xlab("Date") + ylab("Monthly Food Waste (kg)") +
  ggtitle("Monthly Food Waste Trend")
monthly_waste
```




```
# Monthly Plot on solid food waste ----------------------------------------
monthly_solid_waste <-
  ggplot(data = df, aes(x = day_name, y = solid_waste_kg, group=1)) +
  geom_line(color="dark orange") +
  geom_point(aes(shape = is_closed)) +
  facet_grid(month_name~.) +
  scale_shape_manual(values=c(16, 4))+
  theme(legend.position = "none") +
  xlab("Date") + ylab("Monthly Solid Food Waste (kg)") +
  ggtitle("Monthly Solid Food Waste Trend")
monthly_solid_waste
```




```r
# Monthly Plot on liquid food waste ----------------------------------------
monthly_liquid_waste <-
  ggplot(data = df, aes(x = day_name, y = liquid_waste_kg, group=1)) +
  geom_line(color="blue") +
  geom_point(aes(shape = is_closed)) +
  facet_grid(month_name~.) +
  scale_shape_manual(values=c(16, 4))+
  theme(legend.position = "none") +
  xlab("Date") + ylab("Monthly Liquid ood Waste (kg)") +
  ggtitle("Monthly Liquid Food Waste Trend")
monthly_liquid_waste
```

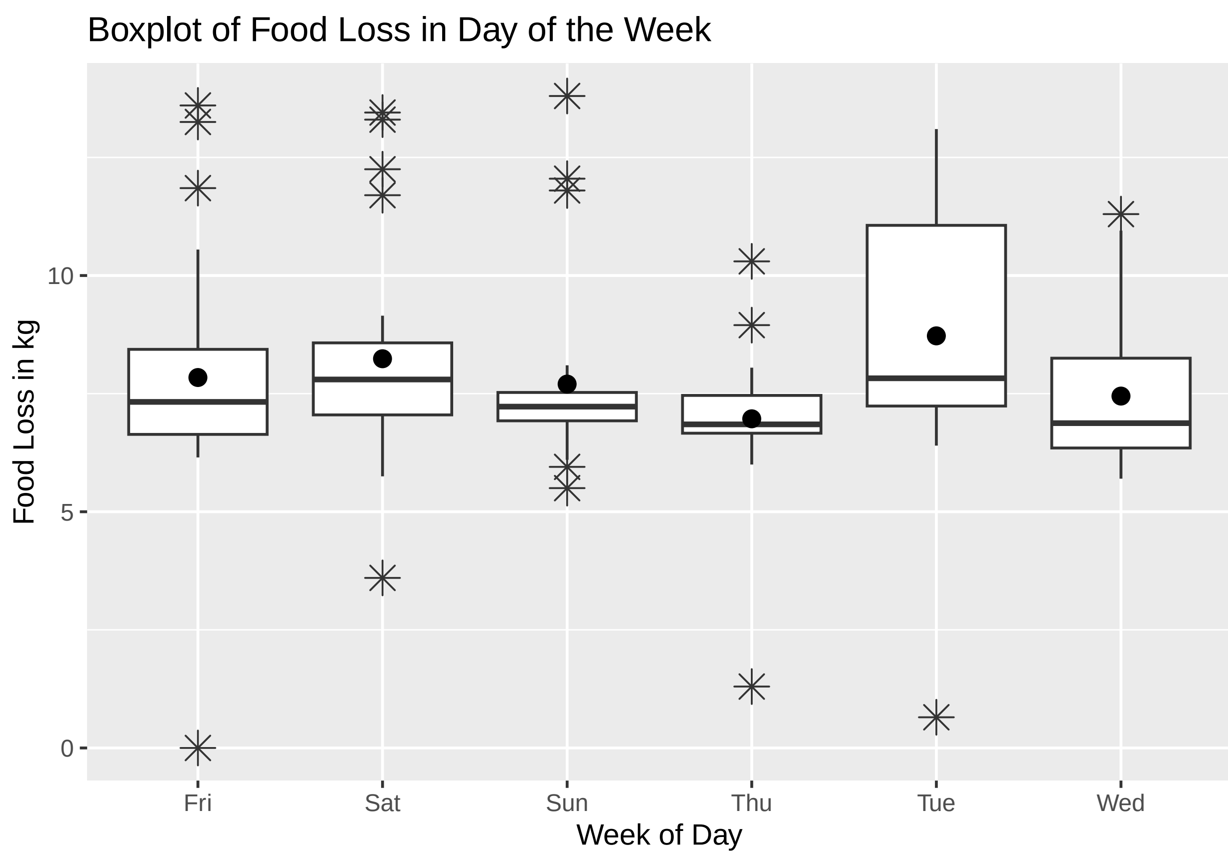



```
# grid.arrange(monthly_loss_waste,monthly_loss, monthly_waste,
#              monthly_solid_waste,monthly_liquid_waste)
```

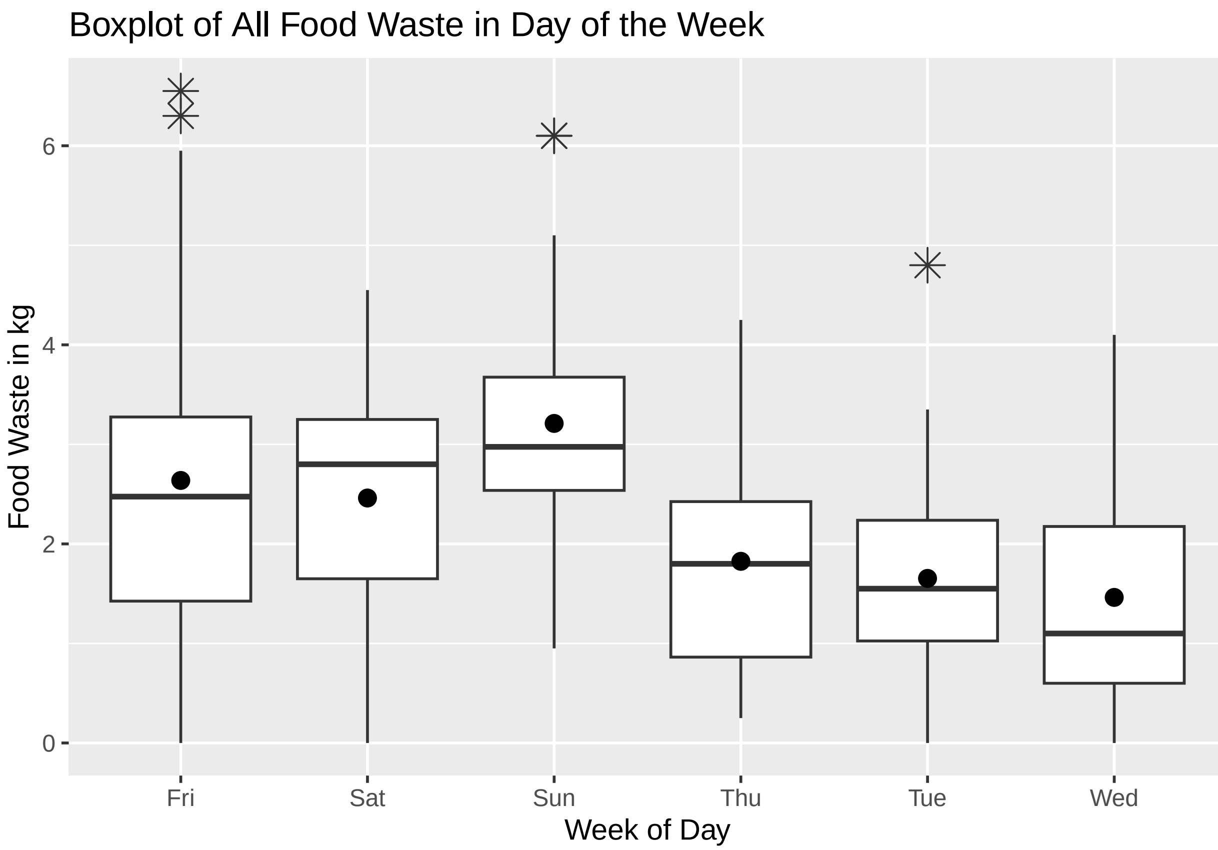
#### Boxplots

```
# weekly boxplot on food loss + food waste ----------------------------
boxplot_week_loss_waste <-
  ggplot(data = subset(df, is_closed %in% FALSE),
         aes(x=day, y=food_loss_kg + food_waste_kg)) +
  geom_boxplot(outlier.shape=8, outlier.size=4) +
  stat_summary(fun=mean, geom="point", shape=16, size=3) +
  labs(title = "Boxplot of Food Loss and Food Waste in Day of the Week",
       x = "Week of Day", y = "Food Loss and Food Waste in kg")
boxplot_week_loss_waste
```


```r
# weekly boxplot on food loss ---------------------------------
boxplot_week_food_loss <-
  ggplot(data = subset(df, is_closed %in% FALSE),
         aes(x=day, y=food_loss_kg)) +
  geom_boxplot(outlier.shape=8, outlier.size=4) +
  stat_summary(fun=mean, geom="point", shape=16, size=3) +
  labs(title = "Boxplot of Food Loss in Day of the Week",
       x = "Week of Day", y = "Food Loss in kg")
boxplot_week_food_loss
```

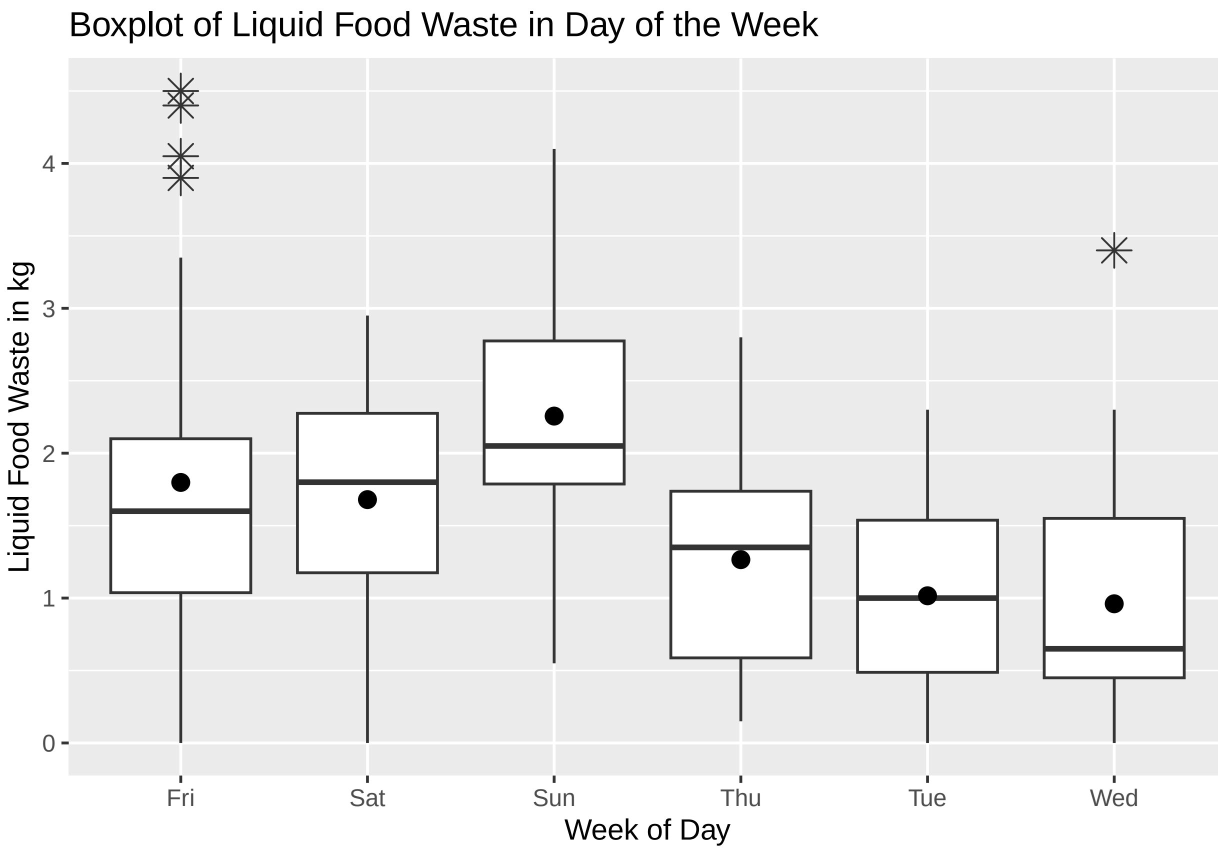



```r
# weekly boxplot on food waste ------------------------------------
boxplot_week_food_waste <-
  ggplot(data = subset(df, is_closed %in% FALSE),
         aes(x=day, y=food_waste_kg)) +
  geom_boxplot(outlier.shape=8, outlier.size=4) +
  stat_summary(fun=mean, geom="point", shape=16, size=3) +
  labs(title = "Boxplot of All Food Waste in Day of the Week",
       x = "Week of Day", y = "Food Waste in kg")
boxplot_week_food_waste
```

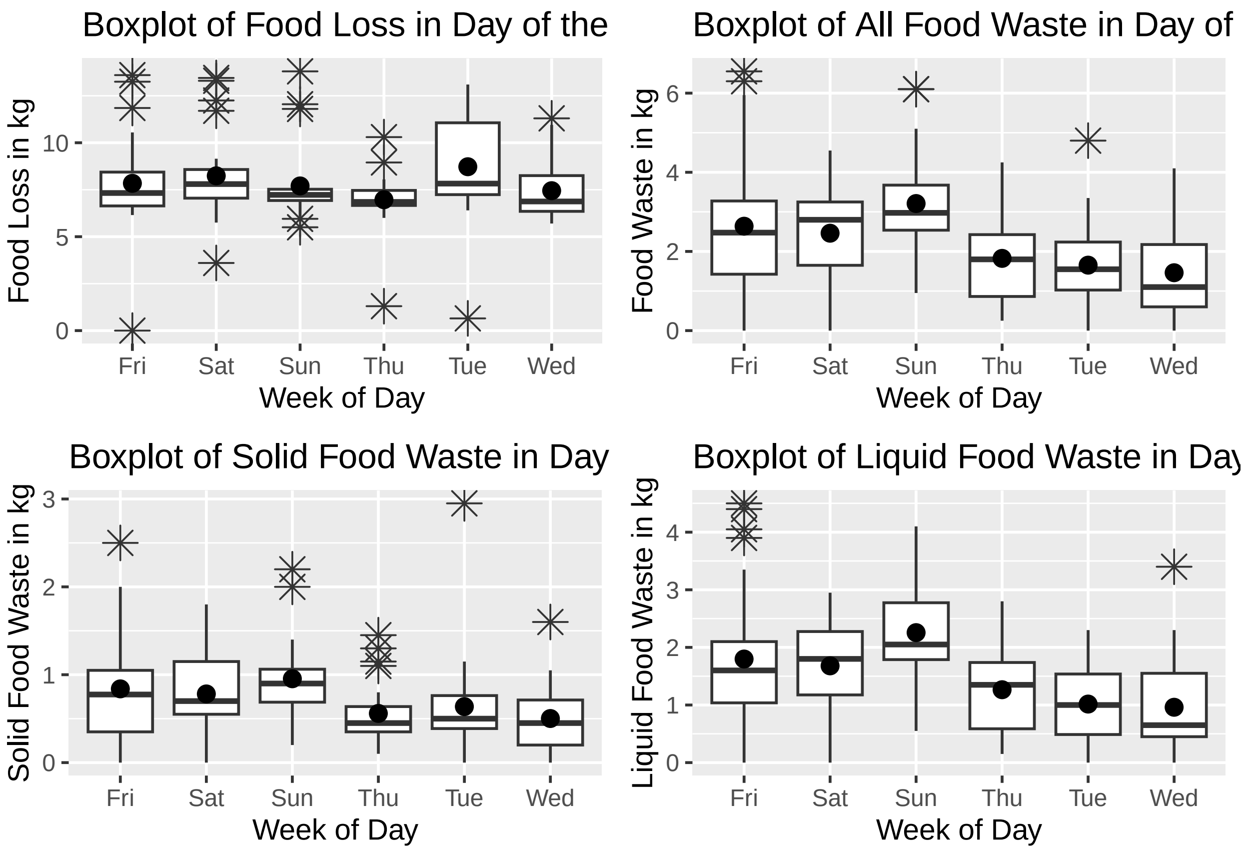



```r
# weekly boxplot on solid food waste -----------------------------------
boxplot_week_solidWaste <-
  ggplot(data = subset(df, is_closed %in% FALSE),
         aes(x=day, y=solid_waste_kg)) +
  geom_boxplot(outlier.shape=8, outlier.size=4) +
  stat_summary(fun=mean, geom="point", shape=16, size=3) +
  labs(title = "Boxplot of Solid Food Waste in Day of the Week",
       x = "Week of Day", y = "Solid Food Waste in kg")
boxplot_week_solidWaste
```

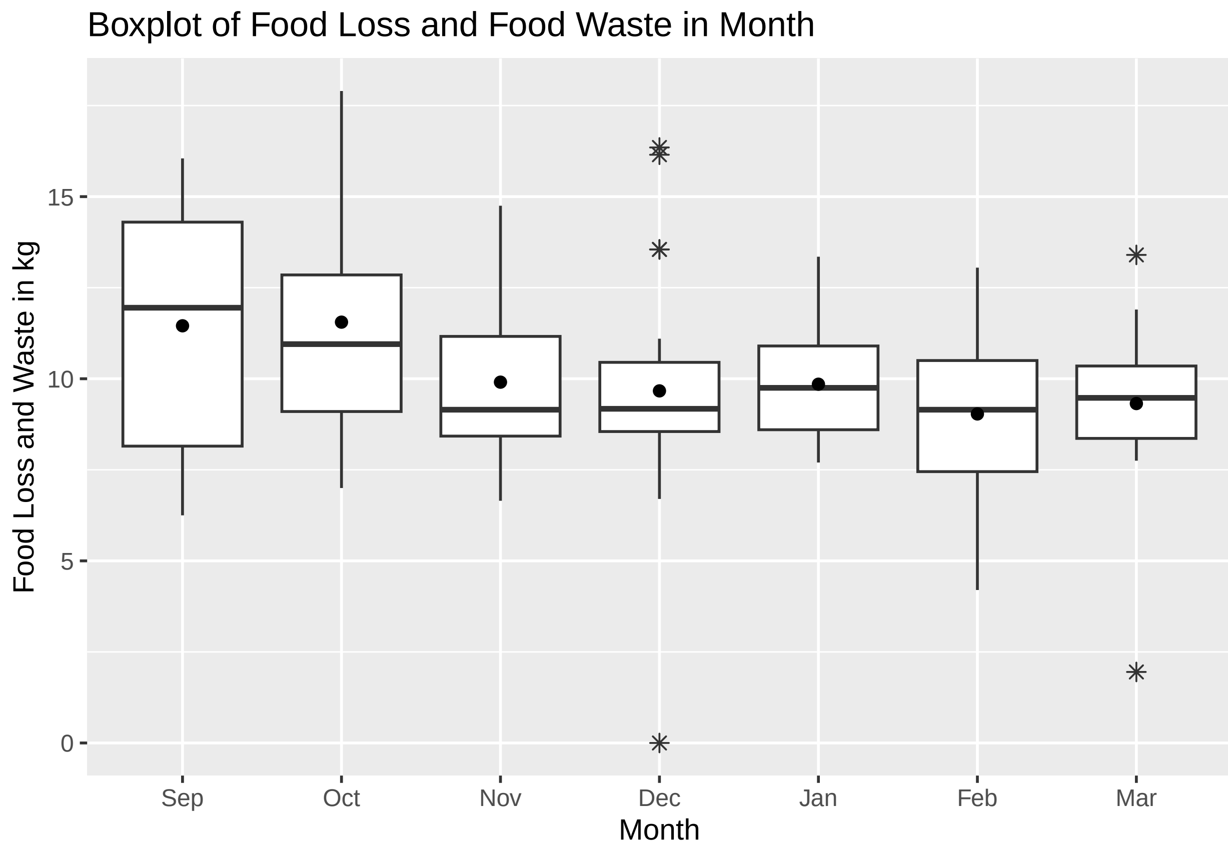

```r
# weekly boxplot on liquid food waste ------------------------------------
boxplot_week_liquidWaste <-
  ggplot(data = subset(df, is_closed %in% FALSE),
         aes(x=day, y=liquid_waste_kg)) +
  geom_boxplot(outlier.shape=8, outlier.size=4) +
  stat_summary(fun=mean, geom="point", shape=16, size=3) +
  labs(title = "Boxplot of Liquid Food Waste in Day of the Week",
       x = "Week of Day", y = "Liquid Food Waste in kg")
boxplot_week_liquidWaste
```



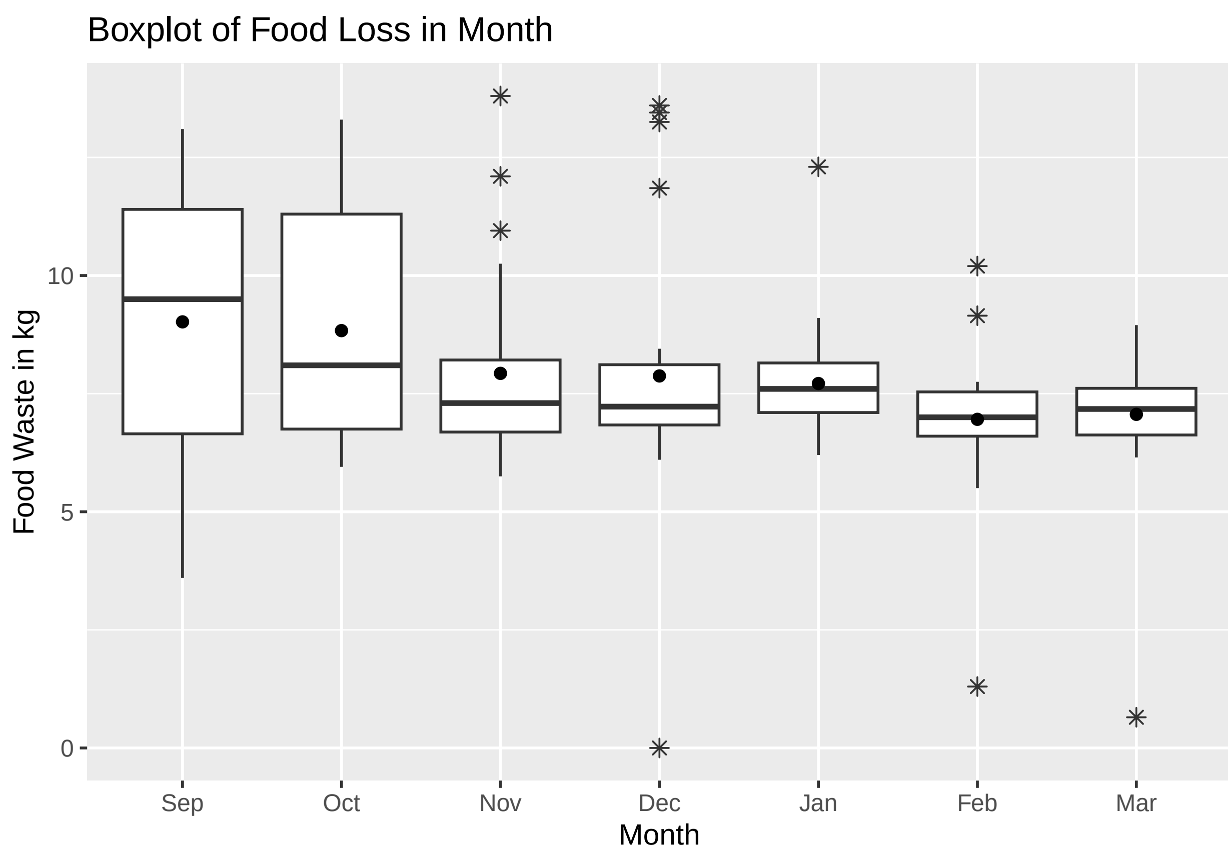

```
grid.arrange(boxplot_week_food_loss,boxplot_week_food_waste,
             boxplot_week_solidWaste,boxplot_week_liquidWaste)
```



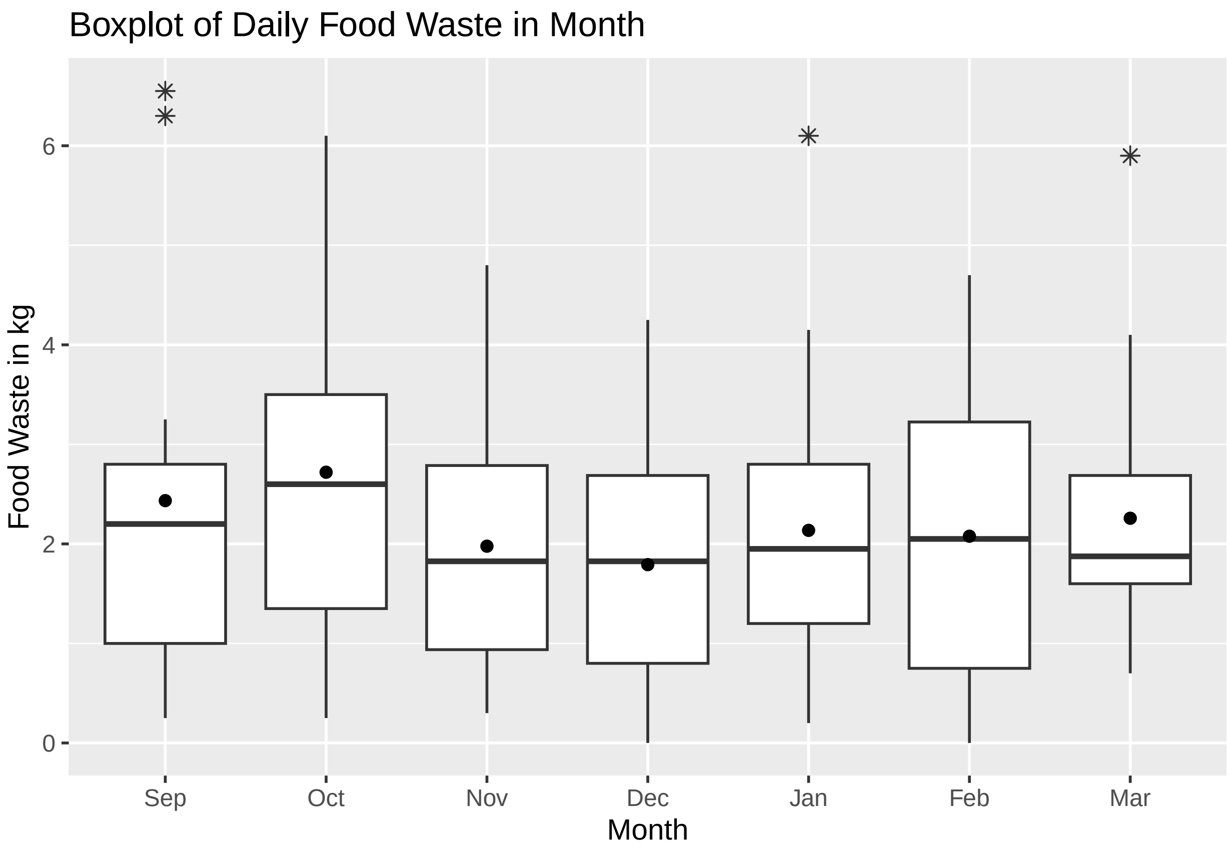

```
# monthly boxplot on food loss + food waste ------------------------------------
boxplot_month_loss_waste <-
  ggplot(data = subset(df, is_closed %in% FALSE),
         aes(x=month_name, y=food_loss_kg+food_waste_kg)) +
  geom_boxplot(outlier.shape=8, outlier.size=2) +
  stat_summary(fun=mean, geom="point", shape=16, size=2) +
  labs(title = "Boxplot of Food Loss and Food Waste in Month",
       x = "Month", y = "Food Loss and Waste in kg")
boxplot_month_loss_waste
```



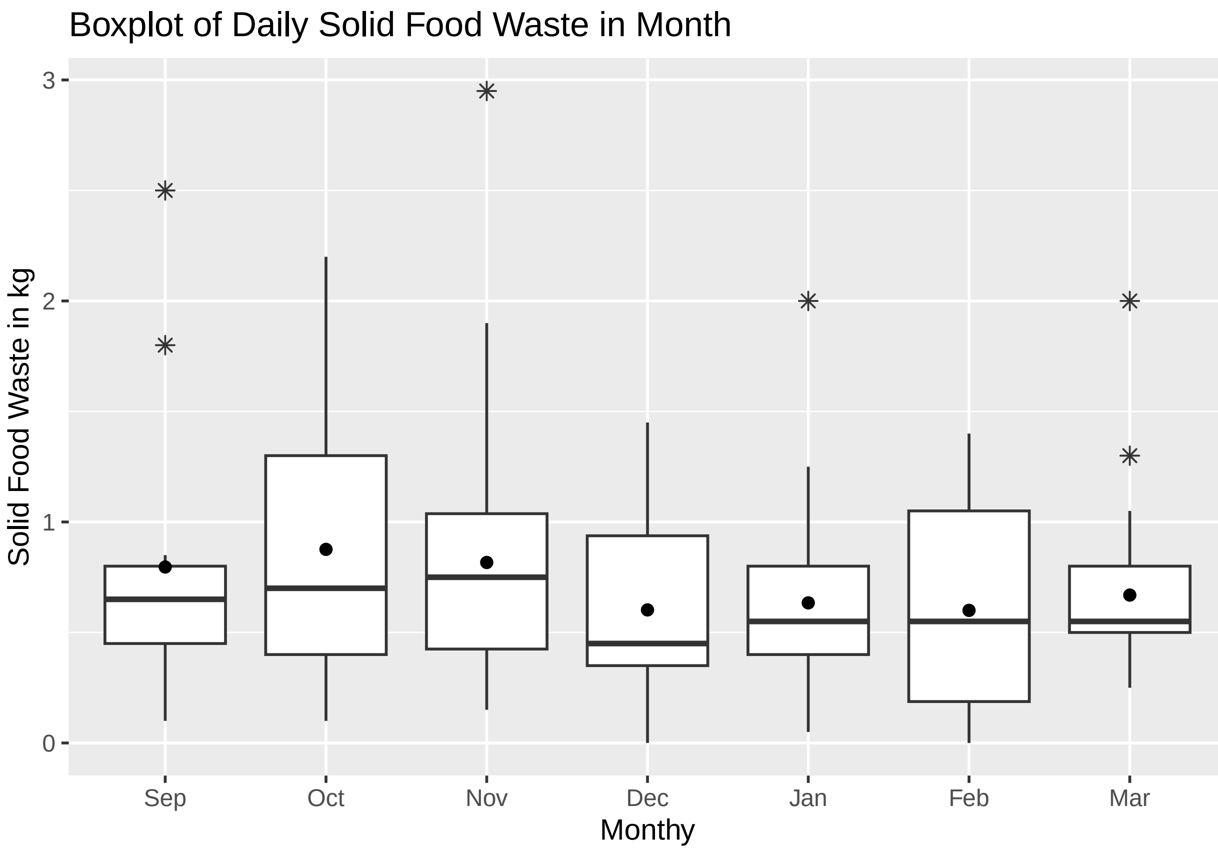

```r
# monthly boxplot on food loss ------------------------------------
boxplot_month_loss <-
  ggplot(data = subset(df, is_closed %in% FALSE),
         aes(x=month_name, y=food_loss_kg)) +
  geom_boxplot(outlier.shape=8, outlier.size=2) +
  stat_summary(fun=mean, geom="point", shape=16, size=2) +
  labs(title = "Boxplot of Food Loss in Month",
       x = "Month", y = "Food Waste in kg")
boxplot_month_loss
```



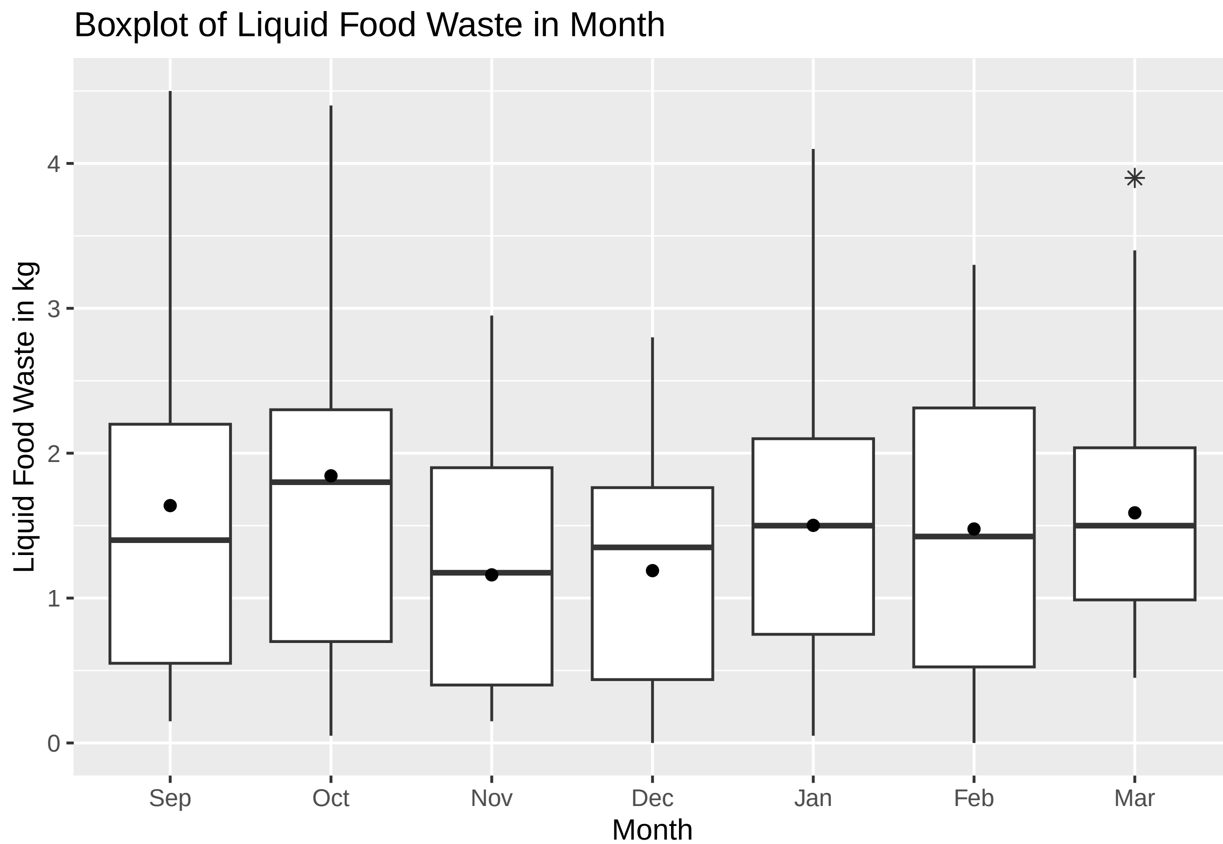

```r
# monthly boxplot on food waste ------------------------------------
boxplot_month_waste <-
  ggplot(data = subset(df, is_closed %in% FALSE),
         aes(x=month_name, y=food_waste_kg)) +
  geom_boxplot(outlier.shape=8, outlier.size=2) +
  stat_summary(fun=mean, geom="point", shape=16, size=2) +
  labs(title = "Boxplot of Daily Food Waste in Month",
       x = "Month", y = "Food Waste in kg")
boxplot_month_waste
```



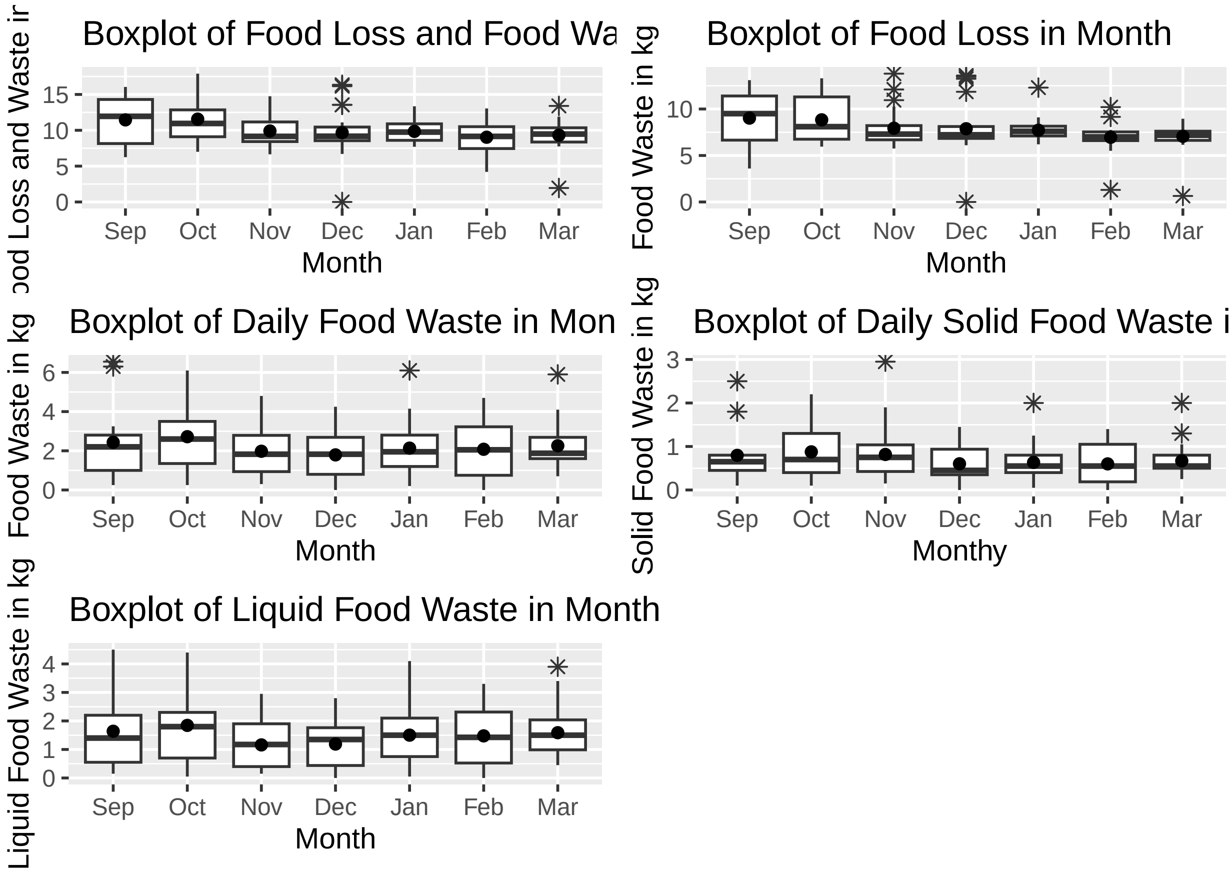

```r
# monthly boxplot on solid food waste -----------------------------------
boxplot_month_solidWaste <-
  ggplot(data = subset(df, is_closed %in% FALSE),
         aes(x=month_name, y=solid_waste_kg)) +
  geom_boxplot(outlier.shape=8, outlier.size=2) +
  stat_summary(fun=mean, geom="point", shape=16, size=2) +
  labs(title = "Boxplot of Daily Solid Food Waste in Month",
       x = "Monthy", y = "Solid Food Waste in kg")
boxplot_month_solidWaste
```




```r
# boxplot of week of day for solid food waste -----------------------------------
boxplot_month_liquidWaste <-
  ggplot(data = subset(df, is_closed %in% FALSE),
         aes(x=month_name, y=liquid_waste_kg)) +
  geom_boxplot(outlier.shape=8, outlier.size=2) +
  stat_summary(fun=mean, geom="point", shape=16, size=2) +
  labs(title = "Boxplot of Liquid Food Waste in Month",
       x = "Month", y = "Liquid Food Waste in kg")
boxplot_month_liquidWaste
```



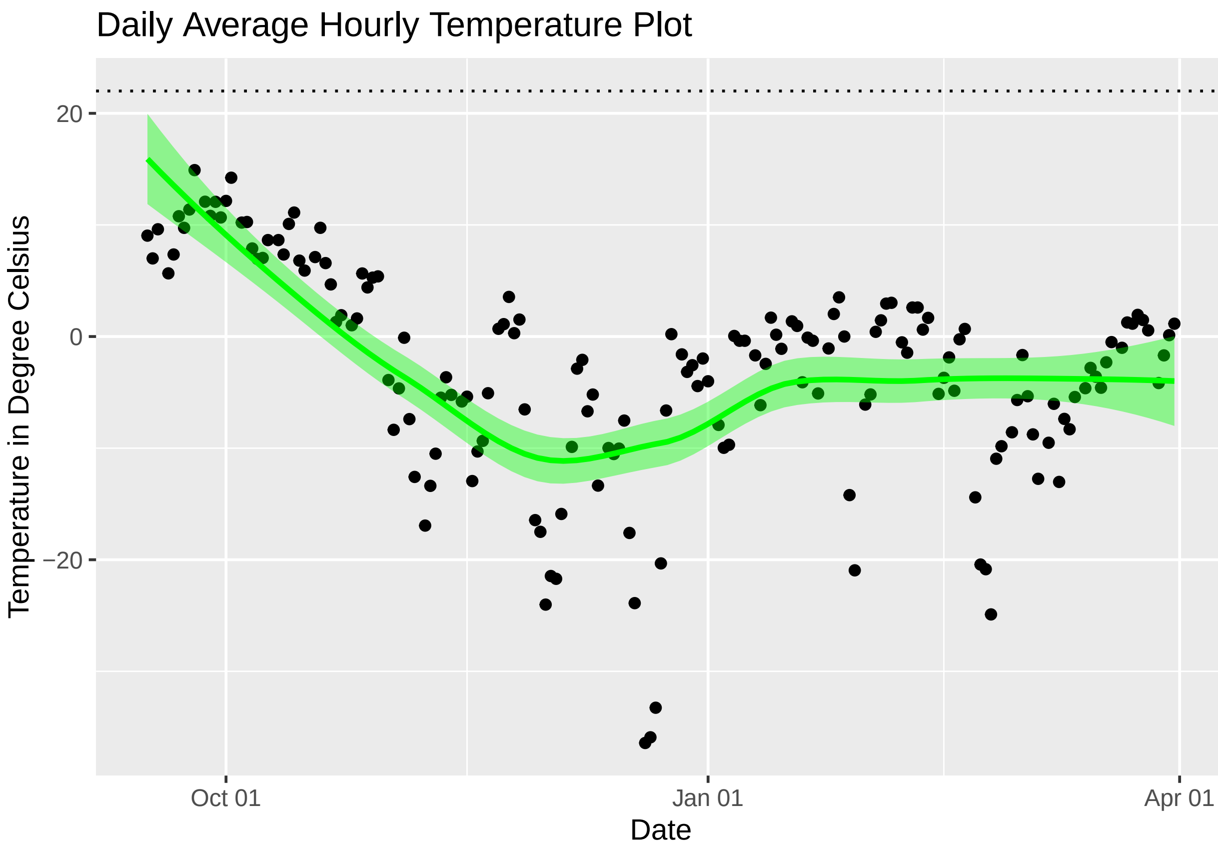

```
grid.arrange(boxplot_month_loss_waste, boxplot_month_loss,boxplot_month_waste,
             boxplot_month_solidWaste,boxplot_month_liquidWaste)
```




### Time Series Plots for Independents

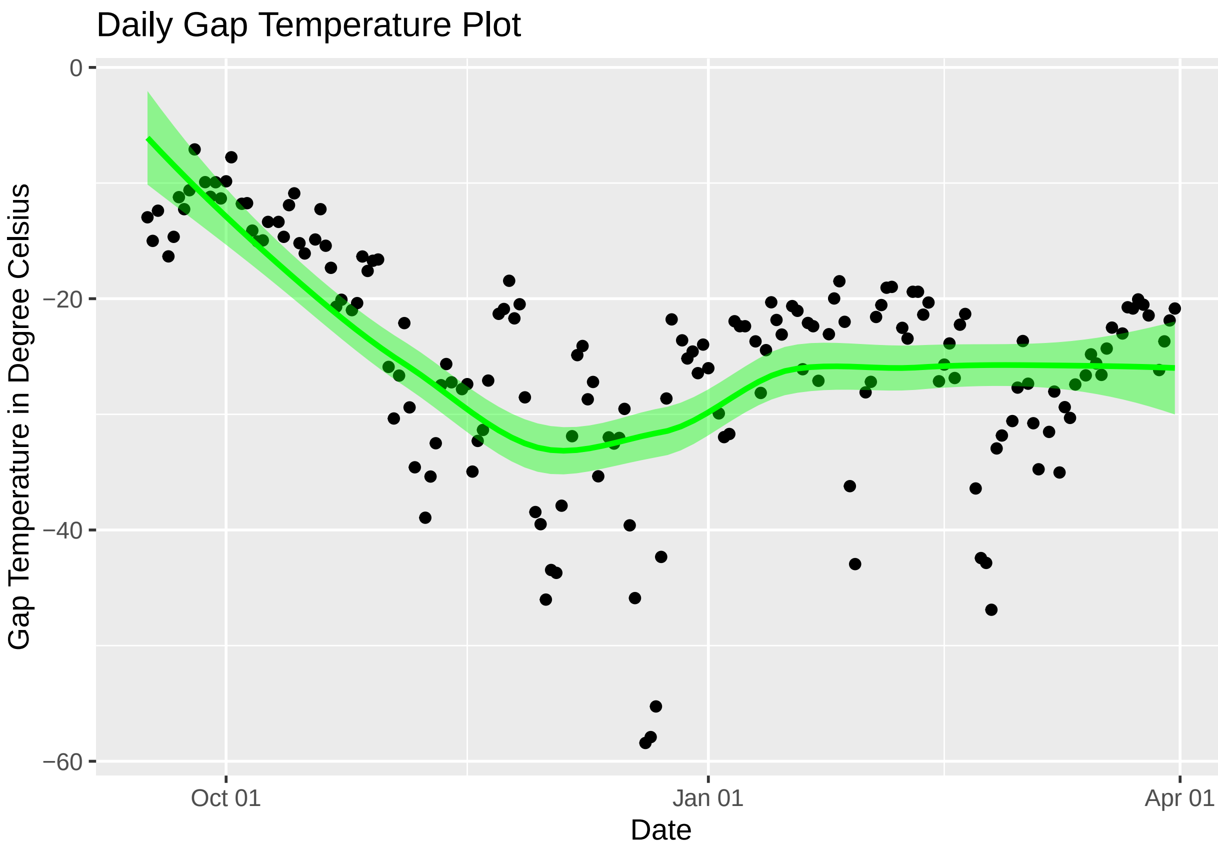
```
## Time Series plots of:
# 1. weather conditions: temperature, humidity, precipitation
# 2. # orders + dine in + size + liquor + daily sales (confident)

# Time Series Plot on temperature ---------------------------------
tsPlot_temp <-
  ggplot(data = df, aes(x = as.Date(date), y = temp_c)) +
  geom_point() +
  stat_smooth(method = "loess", color = "green", fill = "green") +
  # geom_line(aes(group = 1), color="orange") +
  geom_hline(aes(yintercept = 22), linetype='dotted') +
  scale_x_date(date_labels = "%b %d") +
  xlab("Date") + ylab("Temperature in Degree Celsius") +
  ggtitle("Daily Average Hourly Temperature Plot")
tsPlot_temp
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



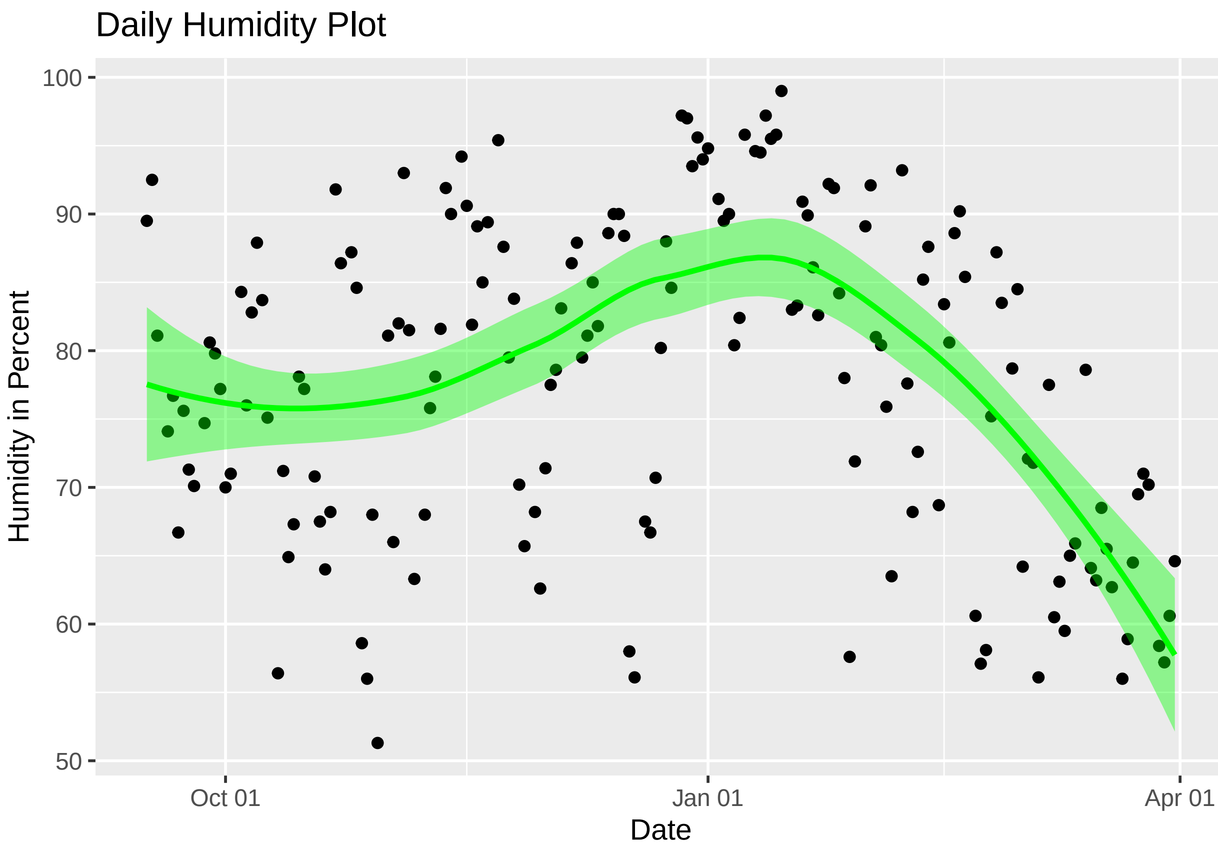

```r
# Time Series Plot on gap temperature with 22C--------------------------------
tsPlot_temp_gap <-
  ggplot(data = df, aes(x = as.Date(date), y = temp_c-22)) +
  geom_point() +
  stat_smooth(method = "loess", color = "green", fill = "green") +
  # geom_line(color="green") +
  scale_x_date(date_labels = "%b %d") +
  xlab("Date") + ylab("Gap Temperature in Degree Celsius") +
  ggtitle("Daily Gap Temperature Plot")
tsPlot_temp_gap
```

```
## 'geom_smooth()' using formula = 'y ~ x'
```



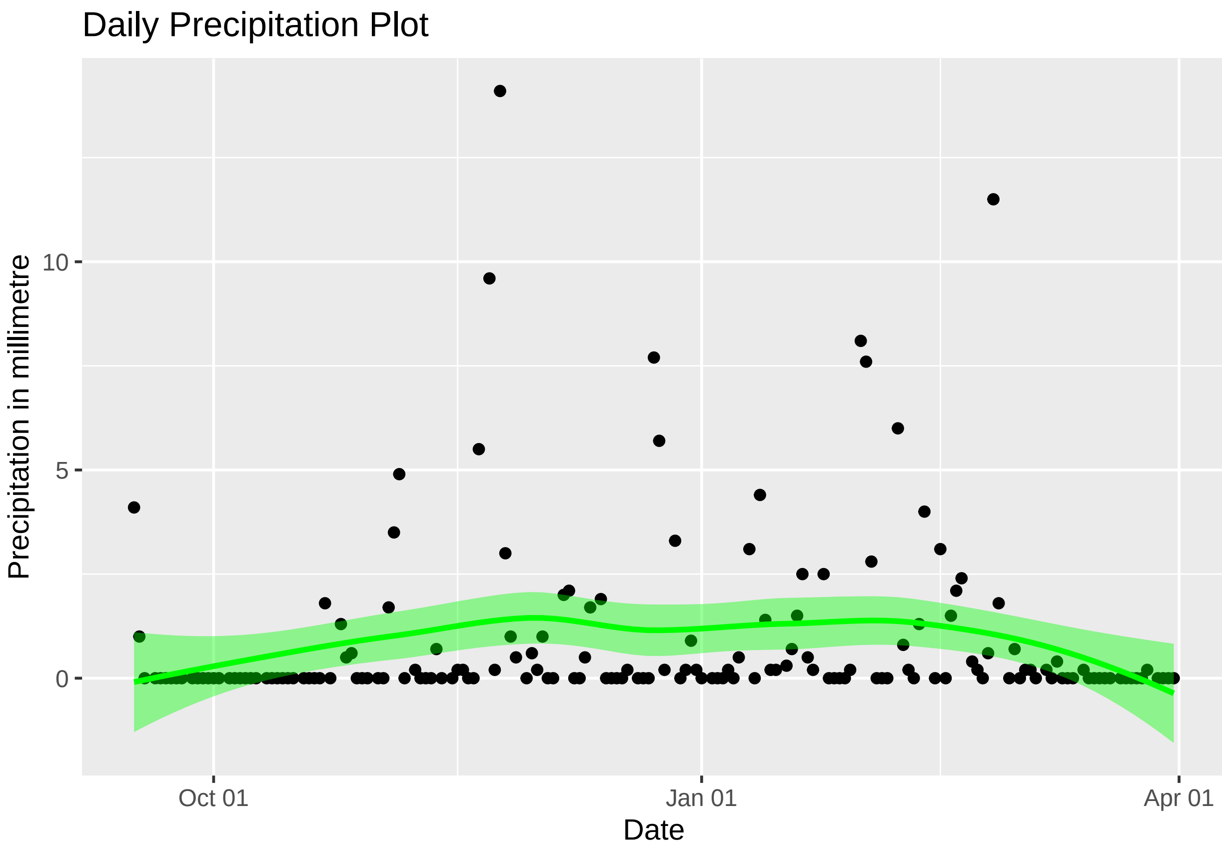

```
# Time Series Plot on humidity ---------------------------------
tsPlot_humidity <-
  ggplot(data = df, aes(x = as.Date(date), y = humi_p)) +
  geom_point() +
  stat_smooth(method = "loess", color = "green", fill = "green") +
  # geom_line(color="red") +
  scale_x_date(date_labels = "%b %d") +
  xlab("Date") + ylab("Humidity in Percent") +
  ggtitle("Daily Humidity Plot")
tsPlot_humidity
```

```
## 'geom_smooth()' using formula = 'y ~ x'
```



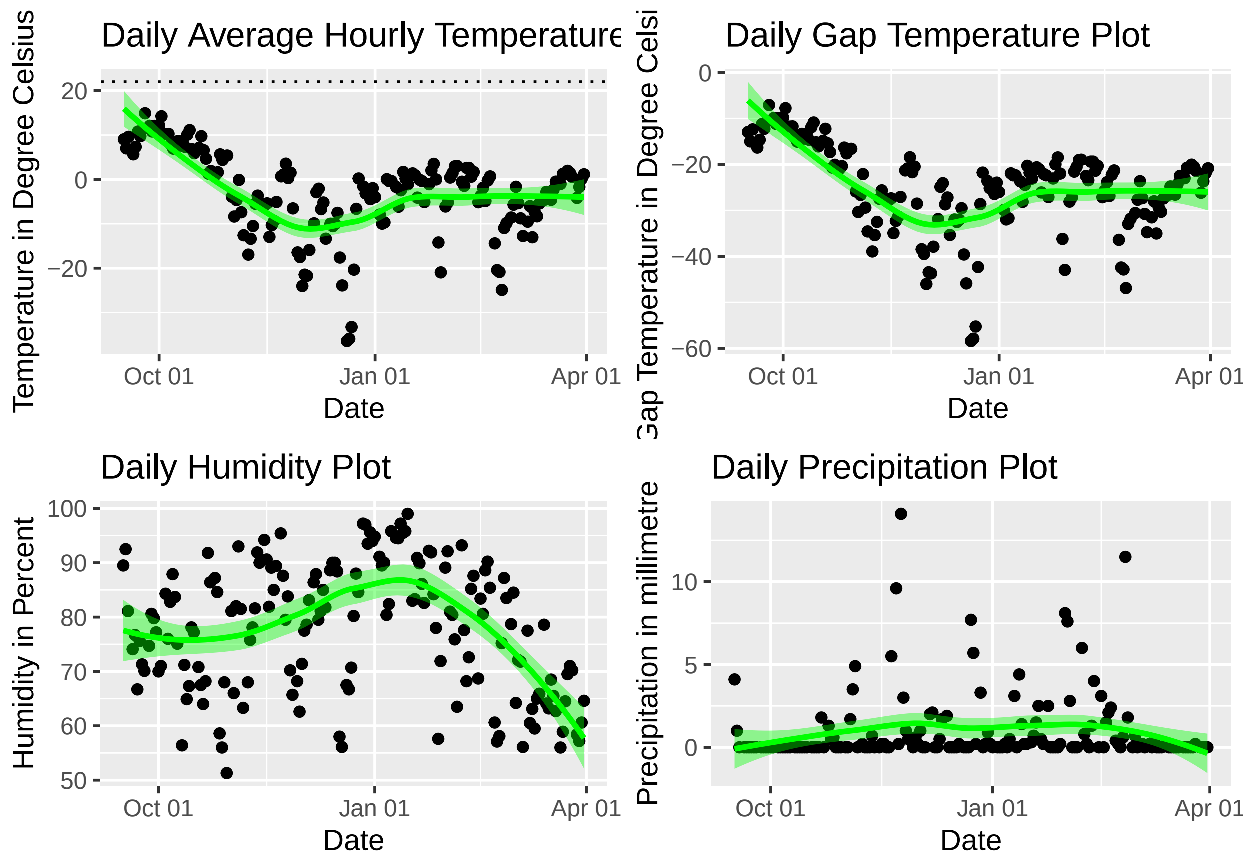

```
# Time Series Plot on precipitation --------------------------------
tsPlot_precip <-
  ggplot(data = df, aes(x = as.Date(date), y = prcp_mm)) +
  geom_point() +
  stat_smooth(method = "loess", color = "green", fill = "green") +
  # geom_line(color="blue") +
  scale_x_date(date_labels = "%b %d") +
  xlab("Date") + ylab("Precipitation in millimetre") +
  ggtitle("Daily Precipitation Plot")
tsPlot_precip
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

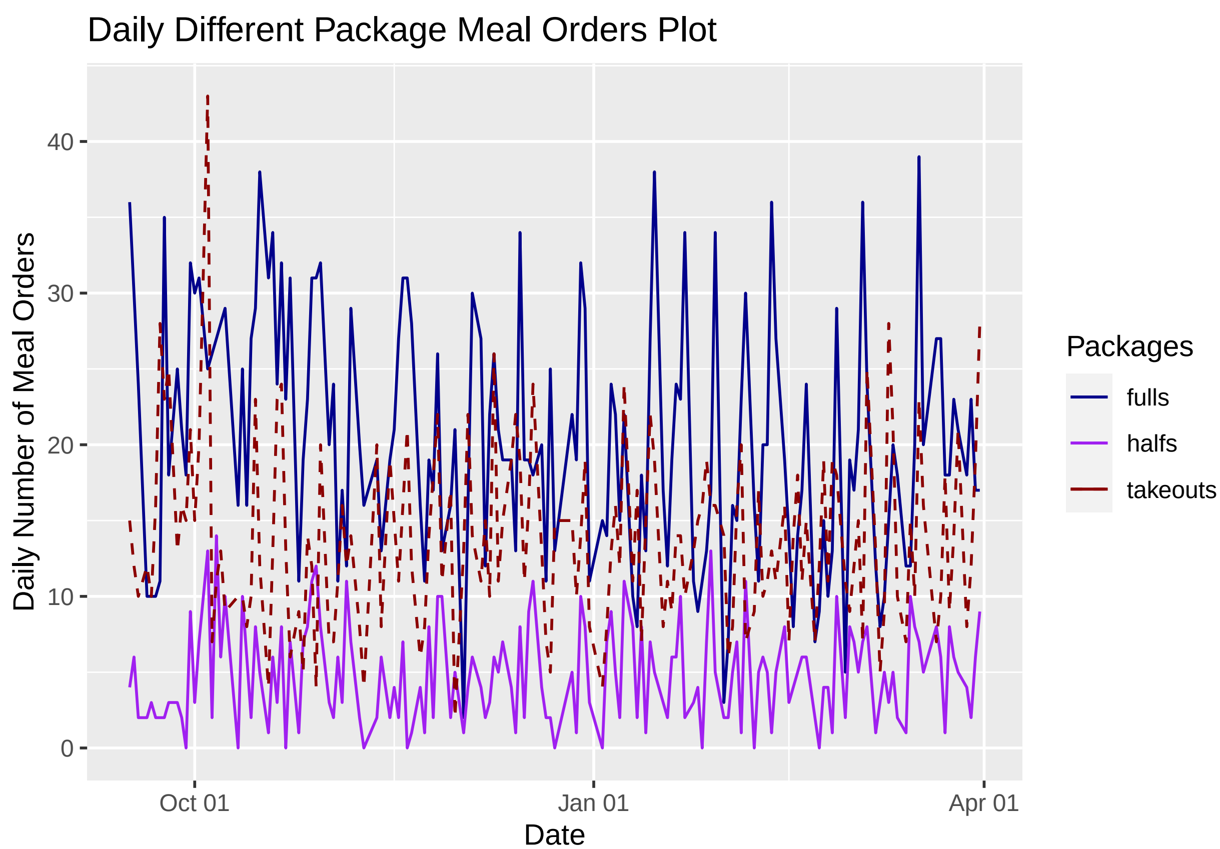



```
grid.arrange(tsPlot_temp,tsPlot_temp_gap,tsPlot_humidity, tsPlot_precip)
```

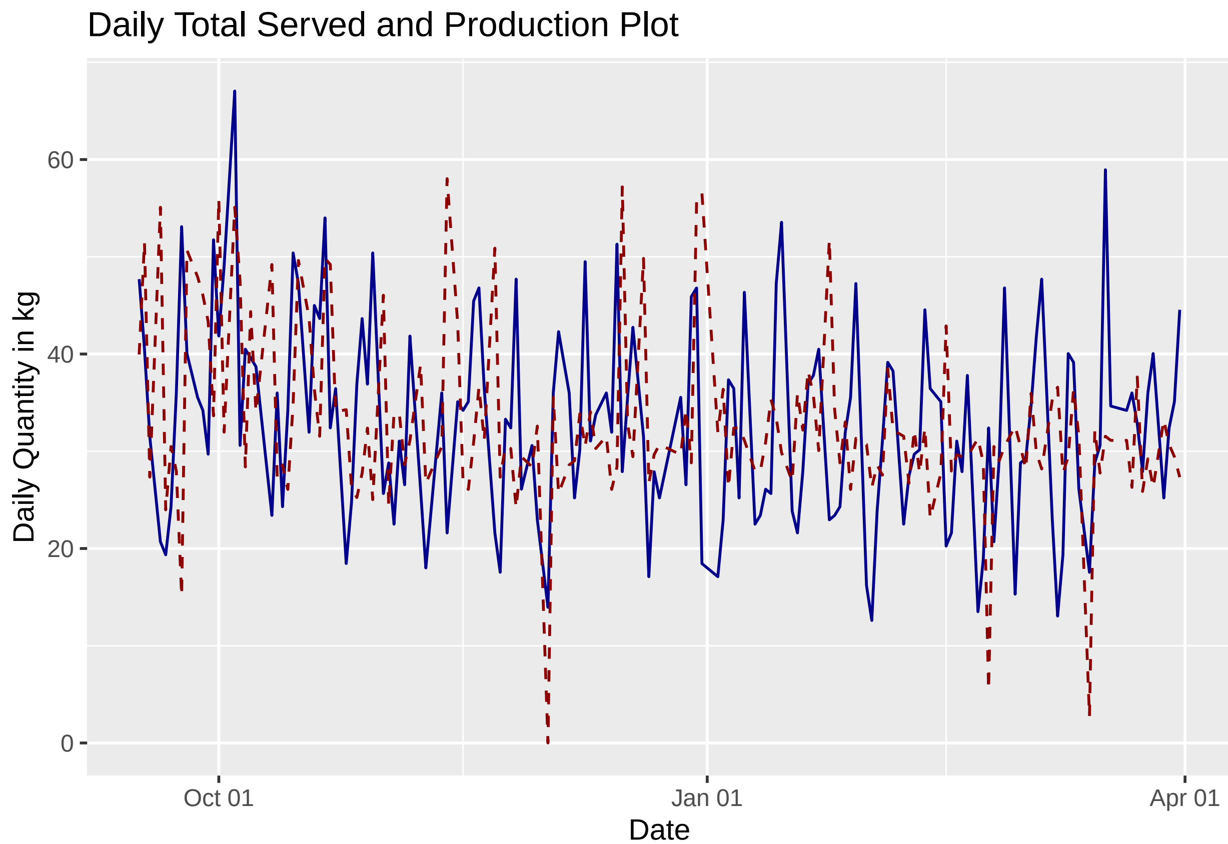
```
## 'geom_smooth()' using formula = 'y ~ x'
## 'geom_smooth()' using formula = 'y ~ x'
## 'geom_smooth()' using formula = 'y ~ x'
## 'geom_smooth()' using formula = 'y ~ x'
```




```
## Time Series plots of:
# 1. # orders (full, half, takeouts)
# 2. daily dine in served (kg)
# 4. liquor
# 5. daily sales (confident)

# Time Series Plot on Meal Orders ---------------------------------
tsPlot_total_orders <-
  ggplot(data = subset(df, is_closed %in% FALSE), aes(x=as.Date(date))) +
  geom_line(aes(y = fulls, color="fulls")) +
  geom_line(aes(y = halfs, color="halfs")) +
  scale_x_date(date_labels = "%b %d") +
  geom_line(aes(y = takeouts, color="takeouts"), linetype = "dashed") +
  xlab("Date") + ylab("Daily Number of Meal Orders") +
  ggtitle("Daily Different Package Meal Orders Plot")+
  scale_color_manual(name='Packages',
                     breaks=c('fulls', 'halfs', 'takeouts'),
                     values=c('fulls' = 'dark blue',
                              'halfs' = 'purple',
                              'takeouts'='dark red')) +
  theme(legend.position = "right")
tsPlot_total_orders
```

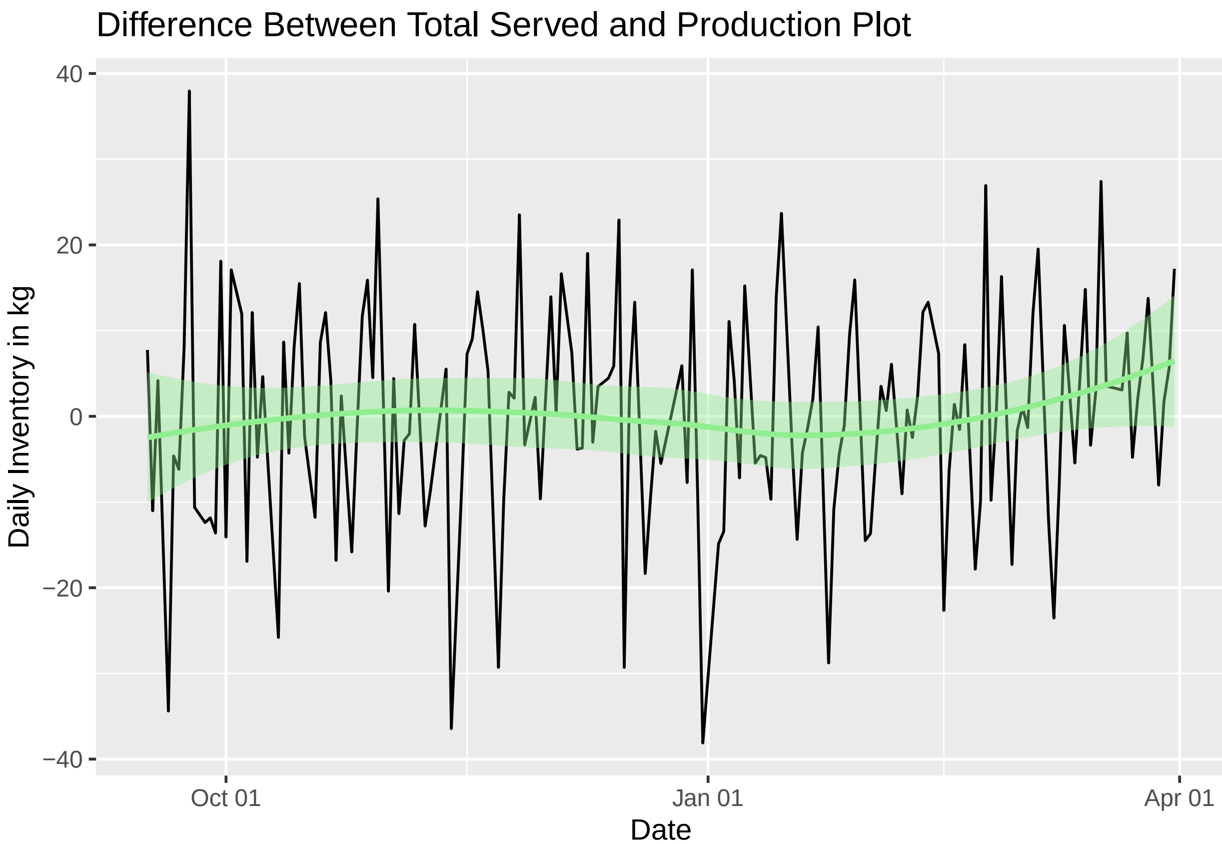



```r
# Time Series Plot on demand and production --------------------------------
tsPlot_D_S <-
  ggplot(data = subset(df, is_closed %in% FALSE), aes(x = as.Date(date))) +
  geom_line(aes(y = daily_total_served), color="dark blue") +
  geom_line(aes(y = FL_FP_kg), color="dark red", linetype = "dashed") +
  scale_x_date(date_labels = "%b %d") +
  xlab("Date") + ylab("Daily Quantity in kg") +
  ggtitle("Daily Total Served and Production Plot") +
  scale_color_manual(name='Served',
                     breaks=c('daily_total_served', 'FL_FP_kg'),
                     values=c('daily_total_served' = 'dark blue',
                              'FL_FP_kg' = 'dark red')) +
  theme(legend.position = "right")
tsPlot_D_S
```



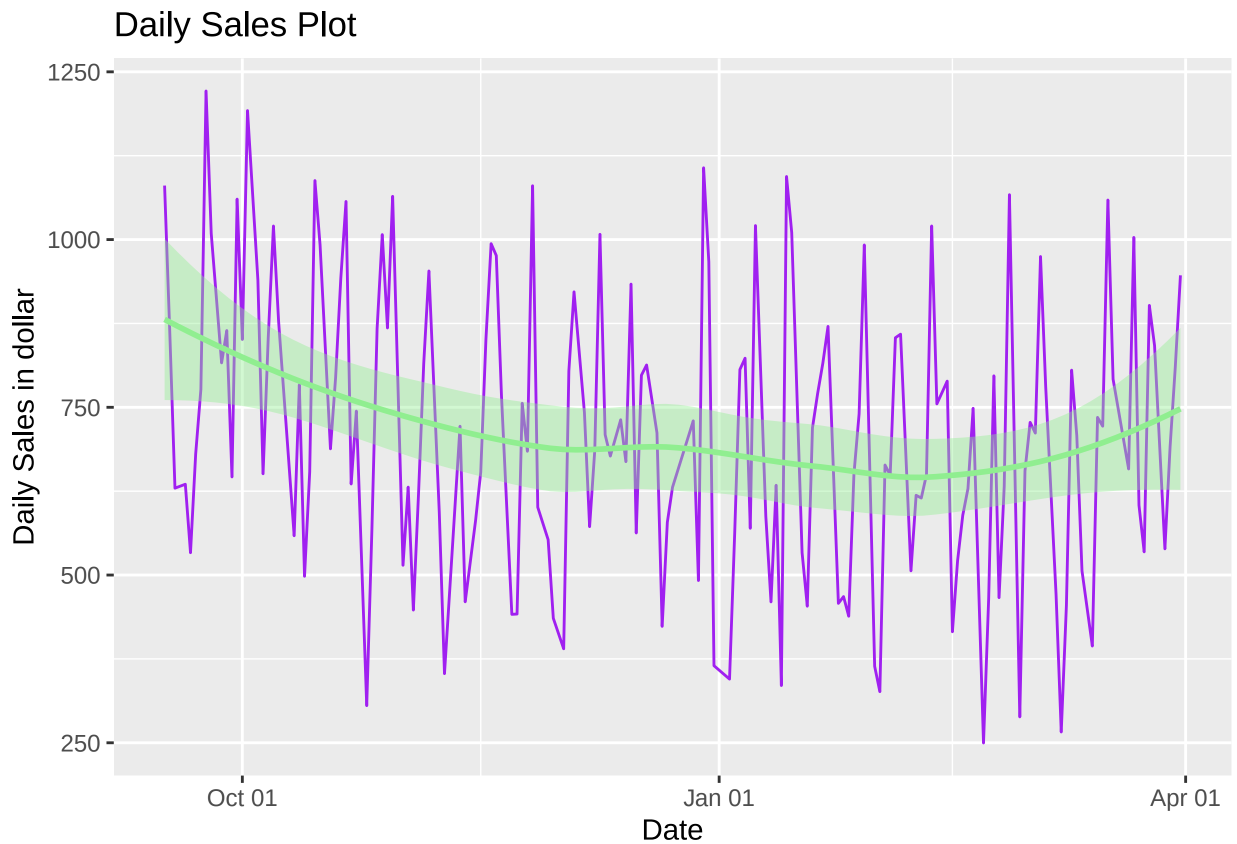

```r
tsPlot_diff_D_S <-
  ggplot(data = subset(df, is_closed %in% FALSE), aes(x = as.Date(date))) +
  geom_line(aes(y = daily_total_served - FL_FP_kg), color="black") +
  stat_smooth(aes(y = daily_total_served - FL_FP_kg), method = "loess",
              color = "light green", fill = "light green") +
  scale_x_date(date_labels = "%b %d") +
  xlab("Date") + ylab("Daily Inventory in kg") +
  ggtitle("Difference Between Total Served and Production Plot")
tsPlot_diff_D_S
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```r
# Time Series Plot on precipitation --------------------------------
tsPlot_sales <-
  ggplot(data = subset(df, is_closed %in% FALSE),aes(x = as.Date(date))) +
  geom_line(aes(y = sales), color="purple") +
  scale_x_date(date_labels = "%b %d") +
  stat_smooth(aes(y = sales), method = "loess",
              color = "light green", fill = "light green") +
  xlab("Date") + ylab("Daily Sales in dollar") +
  ggtitle("Daily Sales Plot")
tsPlot_sales
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

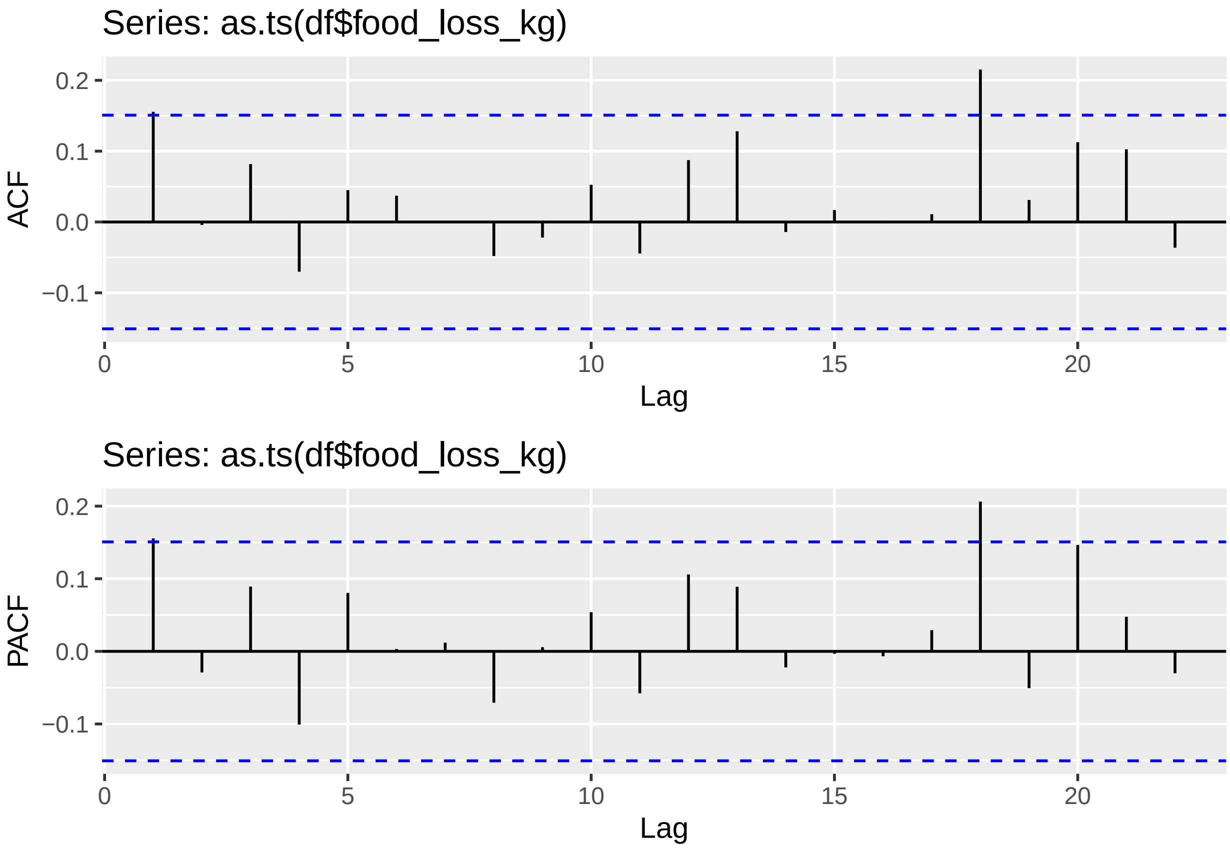



```
grid.arrange(tsPlot_total_orders,tsPlot_D_S, tsPlot_diff_D_S,tsPlot_sales)
```

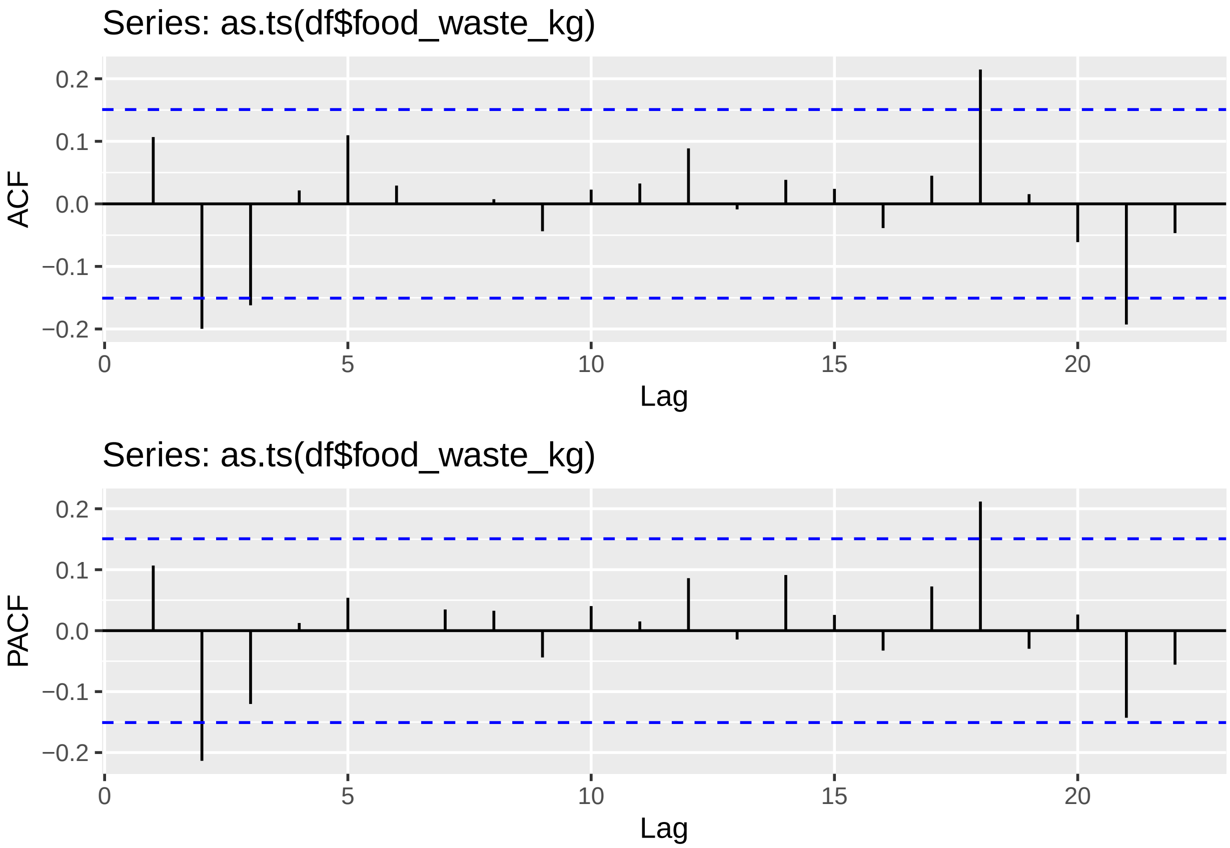
```
## ‘geom_smooth()‘ using formula = ’y ~ x’
## ‘geom_smooth()‘ using formula = ’y ~ x’
```


### (Partial and) Autocorrelation Function

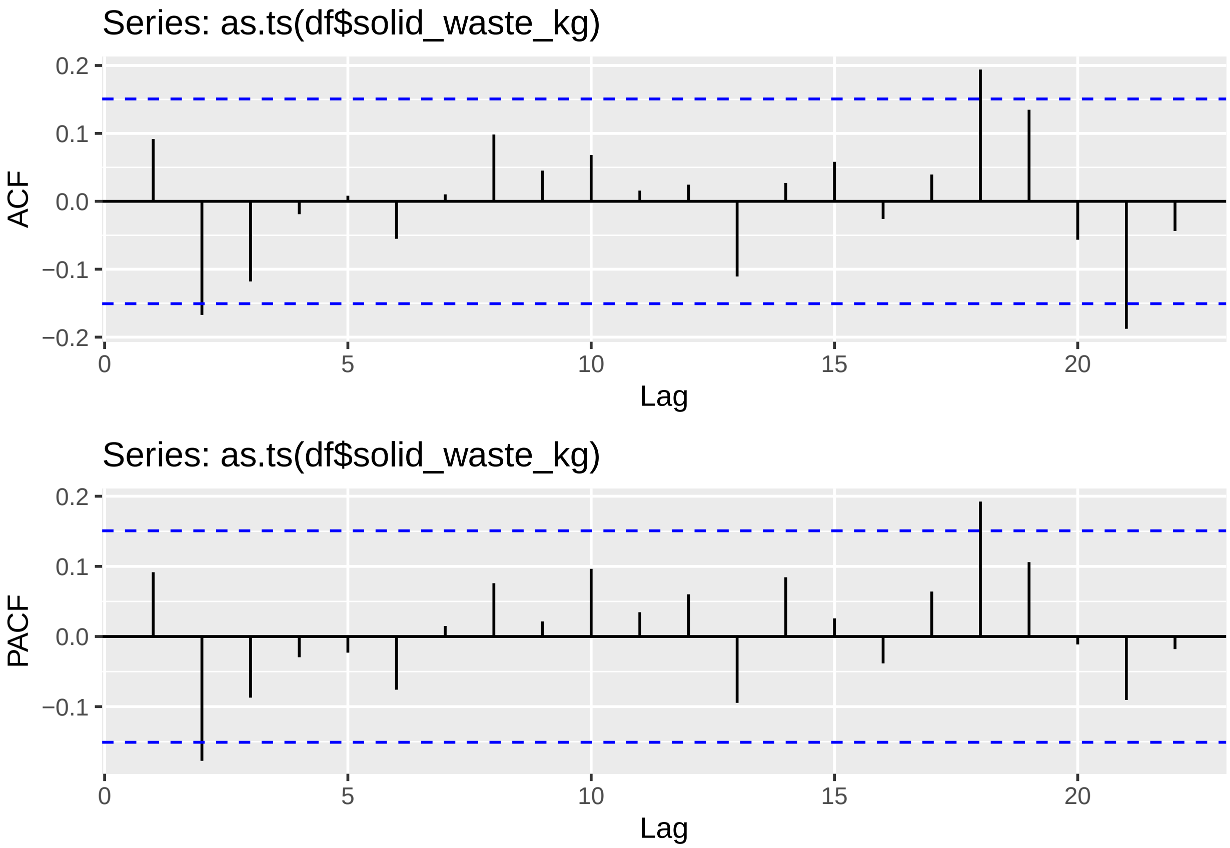
```
## acf and pacf -------------------------------------------------------------
# acf and pacf for food loss -------------------------------------------------
acf_fl  <- ggAcf(as.ts(df$food_loss_kg))
pacf_fl <- ggPacf(as.ts(df$food_loss_kg))
# acf and pacf for all food waste --------------------------------------------
acf_fw  <- ggAcf(as.ts(df$food_waste_kg))
pacf_fw <- ggPacf(as.ts(df$food_waste_kg))
# acf and pacf for solid food waste ------------------------------------------
acf_sfw  <- ggAcf(as.ts(df$solid_waste_kg))
pacf_sfw <- ggPacf(as.ts(df$solid_waste_kg))
# acf and pacf for liquid food waste -----------------------------------------
acf_lfw  <- ggAcf(as.ts(df$liquid_waste_kg))
pacf_lfw <- ggPacf(as.ts(df$liquid_waste_kg))


grid.arrange(acf_fl,pacf_fl)
```



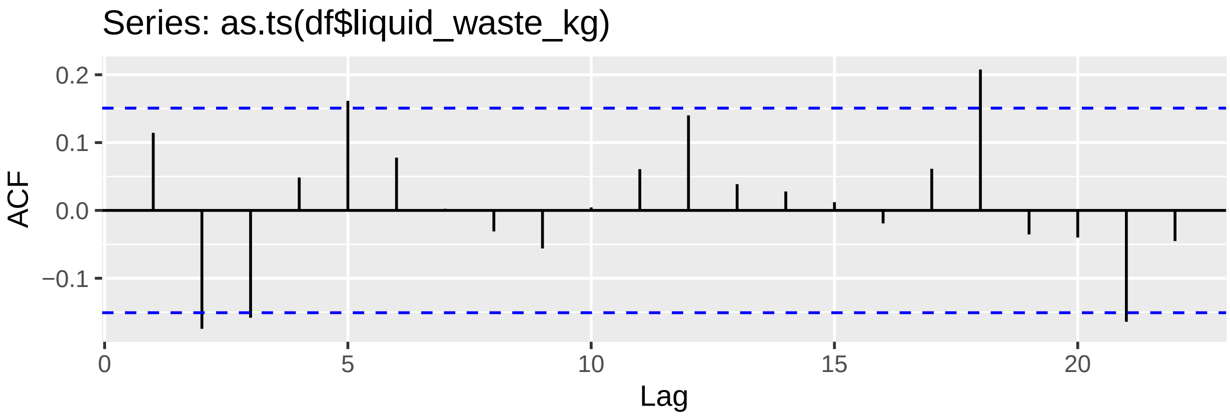

```
grid.arrange(acf_fw,pacf_fw)
```

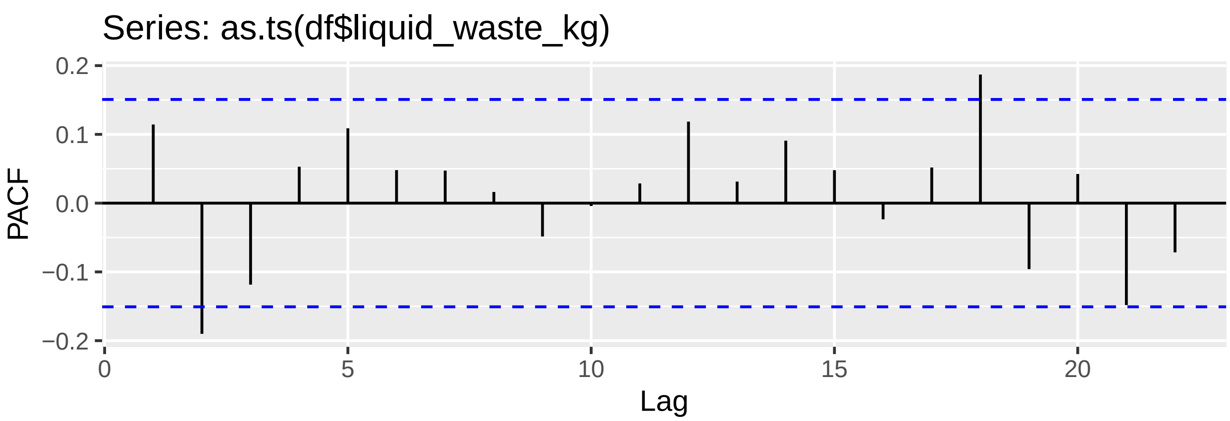



```
grid.arrange(acf_sfw,pacf_sfw)
```



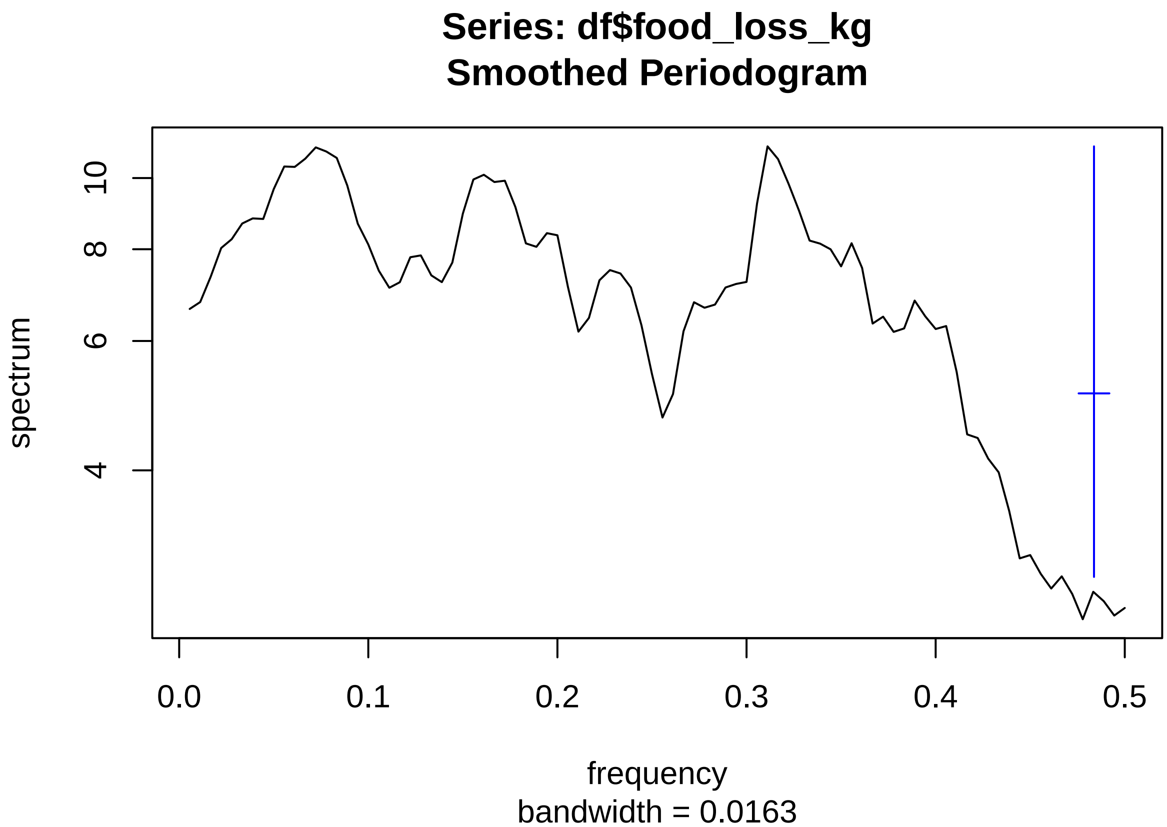

```
grid.arrange(acf_lfw,pacf_lfw)
```




### Spectral Analysis

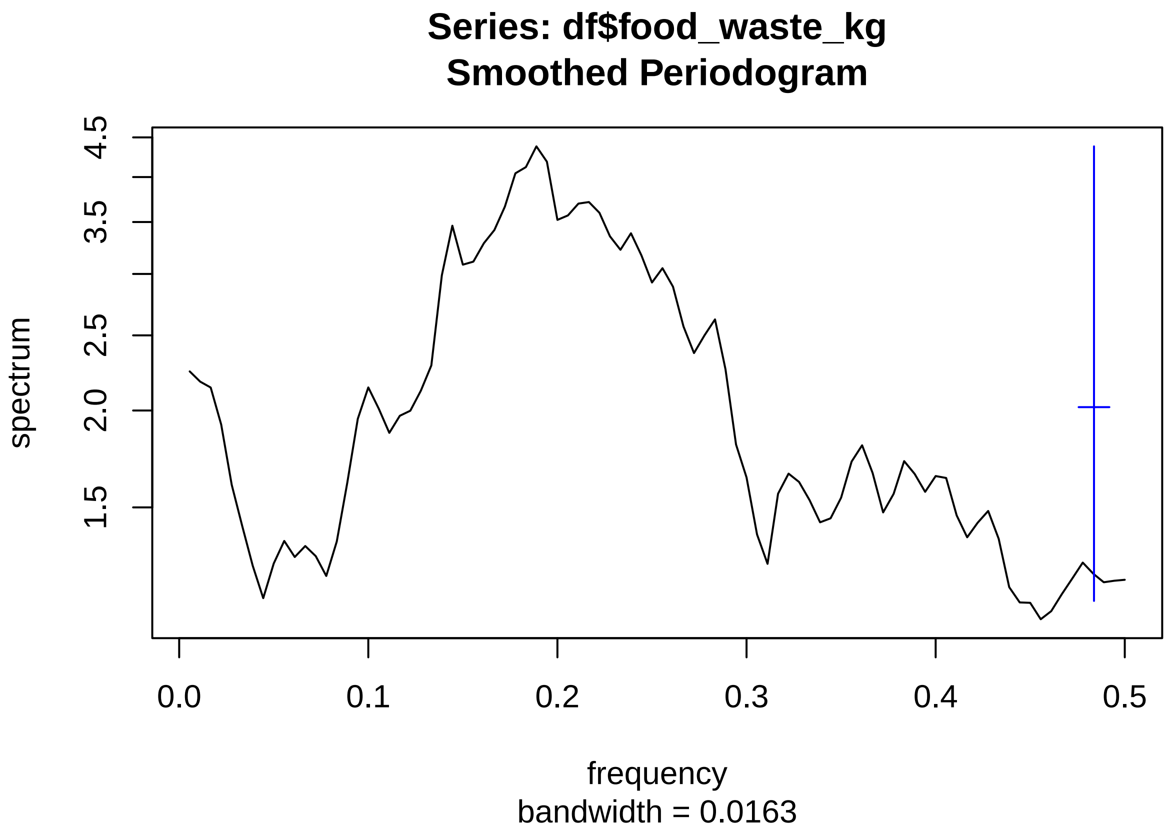
```
# spectrum analysis for food loss ---------------------------------------------
# plot.spectrum(dt$allWasteKg)
raw.spec_fl <- list(spec.pgram(df$food_loss_kg, spans = 10))
```




```
1/raw.spec_fl[[1]]$freq[which.max(raw.spec_fl[[1]]$spec)]
```

```
## [1] 3.214286
```

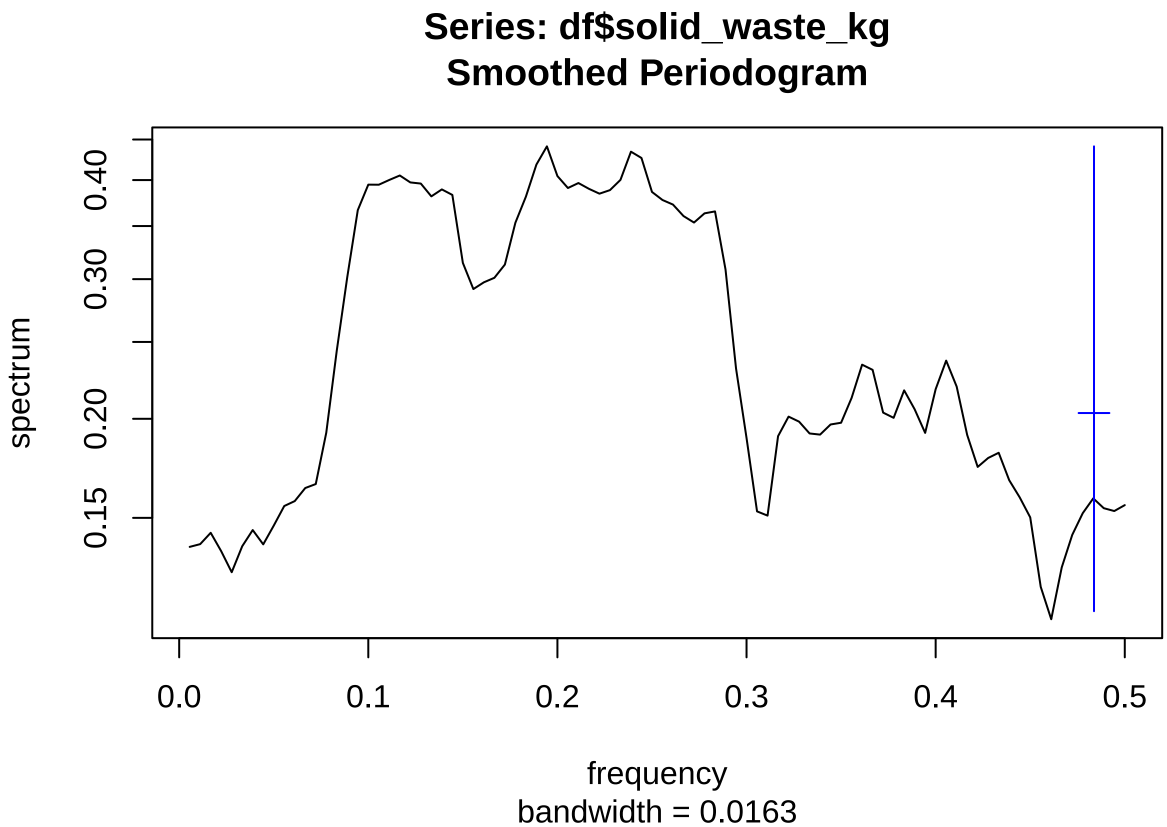
```
# spectrum analysis for food waste ---------------------------------------------
# plot.spectrum(dt$allWasteKg)
raw.spec_fw<- list(spec.pgram(df$food_waste_kg, spans = 10))
```




```
1/raw.spec_fw[[1]]$freq[which.max(raw.spec_fw[[1]]$spec)]
```

```
## [1] 5.294118
```

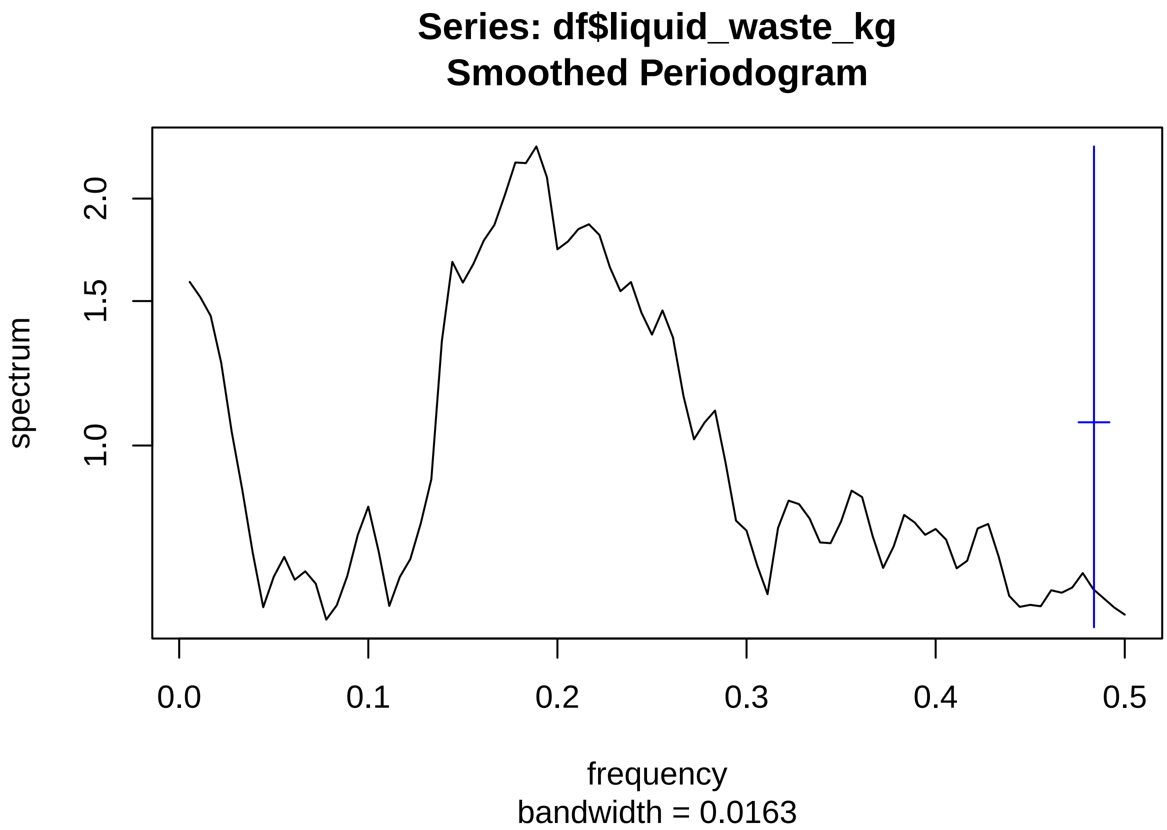
```
# spectrum analysis for food waste ---------------------------------------------
# plot.spectrum(dt$allWasteKg)
raw.spec_sfw<- list(spec.pgram(df$solid_waste_kg, spans = 10))
```




```
1/raw.spec_sfw[[1]]$freq[which.max(raw.spec_sfw[[1]]$spec)]
```

```
## [1] 5.142857
```

```
# spectrum analysis for food waste ---------------------------------------------
# plot.spectrum(dt$allWasteKg)
raw.spec_lfw<- list(spec.pgram(df$liquid_waste_kg, spans = 10))
```




```
1/raw.spec_lfw[[1]]$freq[which.max(raw.spec_lfw[[1]]$spec)]
```

```
## [1] 5.294118
```

roughly 6 (days) period for food waste, but food loss is approx. 3 days or 20 days cycle.

**Erase states**

```
rm(list = ls()[! ls() %in% c("df", "AdjMat")])
```