```r
#0. you might have to install missing packages, comment in the following 2 lines to do so
#install.packages("RColorBrewer")
#install.packages("tidyverse")
library(RColorBrewer)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.5.0     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

Importing the current dataset + setup

```r
##if you want to run the code, you have to change the path to your folder
#you can run main.py with you configuration, which will create a new folder with a timestamp
#in the datafolder
#1. add the path to the data folder in PATH
#2. add the name of the folder you want to analyse e.g.  "2024-04-10at09-45/"
PATH = "E:/UF/ifwaste/data/"
INPUT_FOLDER = "2024-04-10at09-45/"

#3. if you want to save all generated plots, set save_plots to TRUE, otherwise set it to FALSE
save_plots = TRUE

#4. optional: change the design settings:
PALETTE = "Spectral"
WIDTH = 1200
HEIGHT = 900


OUTPUT_FOLDER = "plots/"

hh_foodwasted <- read.csv(paste0(PATH, INPUT_FOLDER, "/wasted.csv"))
hh_foodeaten <- read.csv(paste0(PATH, INPUT_FOLDER, "/eaten.csv"))
hh_foodremaining <- read.csv(paste0(PATH, INPUT_FOLDER, "/still_have.csv"))
hh_foodbought <- read.csv(paste0(PATH, INPUT_FOLDER, "/bought.csv"))
hh_daily <- read.csv(paste0(PATH, INPUT_FOLDER, "/daily.csv"))
hh_config <- read.csv(paste0(PATH, INPUT_FOLDER, "/config.csv"))
```

Label generator for plots

```r
# Create named vector with labels for each house
label_house_num_people <- as.vector(
  with(
    hh_config,
    sprintf("House: %s Concern: %.2f", House, LvlOfConcern)
  )
)
```

```
house_labels <- function(x) paste0("House ", x)
```

Biomass check:

1. in kg

```
sum(hh_foodbought$Kg)
```

```
## [1] 3065.748
```

```
sum(hh_foodwasted$Kg)
```

```
## [1] 1653.786
```

```
sum(hh_foodeaten$Kg)
```

```
## [1] 1255.832
```

```
sum(hh_foodremaining$Kg)
```

```
## [1] 78.3095
```

```
total <- sum(hh_foodwasted$Kg) + sum(hh_foodeaten$Kg) + sum(hh_foodremaining$Kg)
total
```

```
## [1] 2987.928
```

```
missing <- sum(hh_foodbought$Kg) - total
missing
```

```
## [1] 77.82034
```

2. in servings

```
sum(hh_foodbought$Servings)
```

```
## [1] 32318
```

```
sum(hh_foodwasted$Servings)
```

```
## [1] 17344.05
```

```
sum(hh_foodeaten$Servings)
```

```
## [1] 13344.83
```

```
sum(hh_foodremaining$Servings)
```

```
## [1] 810.9833
```

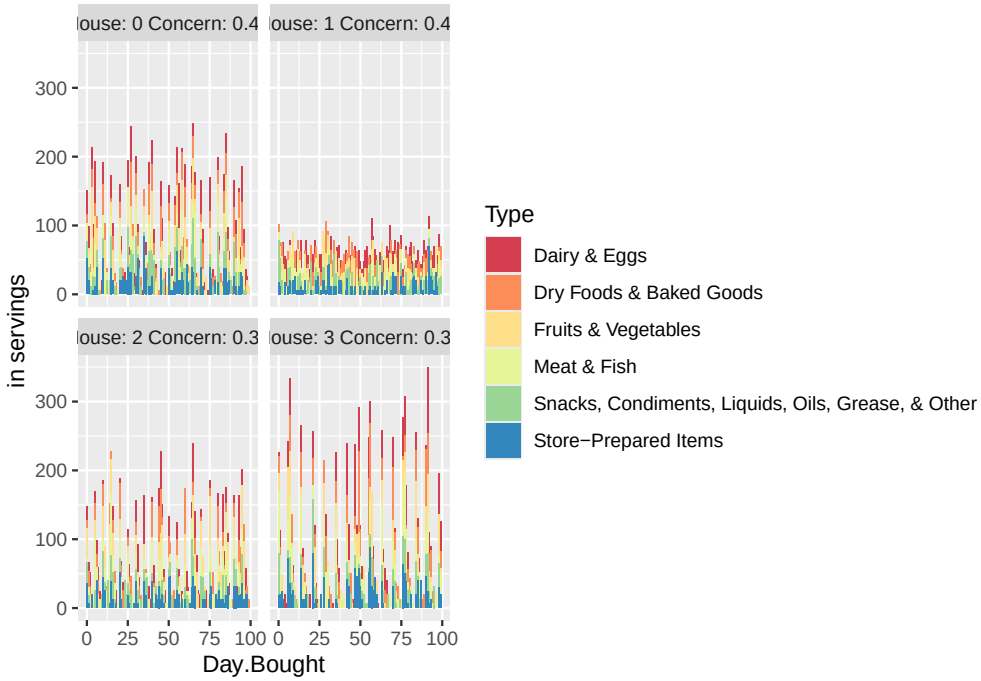
```
total <- sum(hh_foodwasted$Servings) + sum(hh_foodeaten$Servings) + sum(hh_foodremaining$Servings)
total
```

```
## [1] 31499.86
```

```
missing <- sum(hh_foodbought$Servings) - total
missing
```

```
## [1] 818.1435
```

Analysis of food purchase ### 1. Purchased food by household divided by type, in servings
“‘r # Plotting purchased_by_household <- ggplot(data = hh_foodbought) + geom_bar(mapping = aes(x = Day.Bought, fill = Type, weight = Servings), position = “stack”) + facet_wrap(~House, labeller = labeller(House = as_labeller(setNames(label_house_num_people, hh_config$House)))) + ylab(“in servings”) + scale_fill_brewer(palette = PALETTE)
Display the plot purchased_by_household “‘



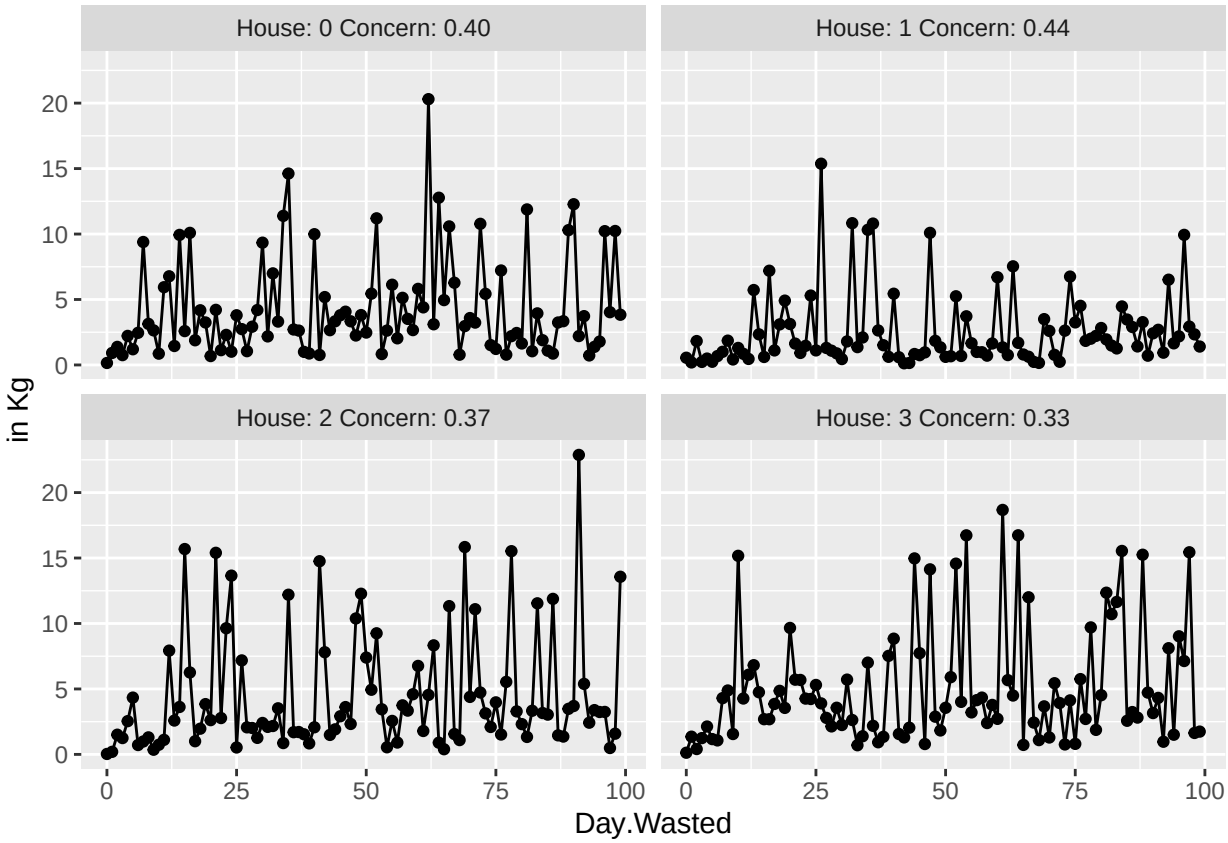

```
r if (save_plots) { png(file = paste0(OUTPUT_FOLDER,
"bought_hh_servings.png"), width = WIDTH, height = HEIGHT)  # Adjust
width and height as needed print(purchased_by_household) dev.off() }
## pdf ##   2
```

Analysis of food waste

1. Food waste by household by kg

```
# Plotting
wasted_by_household <- ggplot(data = hh_foodwasted) +
  stat_summary(aes(x = Day.Wasted, y = Kg), fun="sum", geom="point") +
  stat_summary(aes(x = Day.Wasted, y = Kg), fun="sum", geom="line") +
  facet_wrap(~House, labeller = labeller(House = as_labeller(setNames(label_house_num_people, hh_config$
  ylab("in Kg") +
  scale_fill_brewer(palette = PALETTE)

wasted_by_household
```




```
if (save_plots) {
   png(file = paste0(OUTPUT_FOLDER, "waste_hh_kg.png"), width = WIDTH, height = HEIGHT)  # Adjust width
   print(wasted_by_household)
   dev.off()
}
```

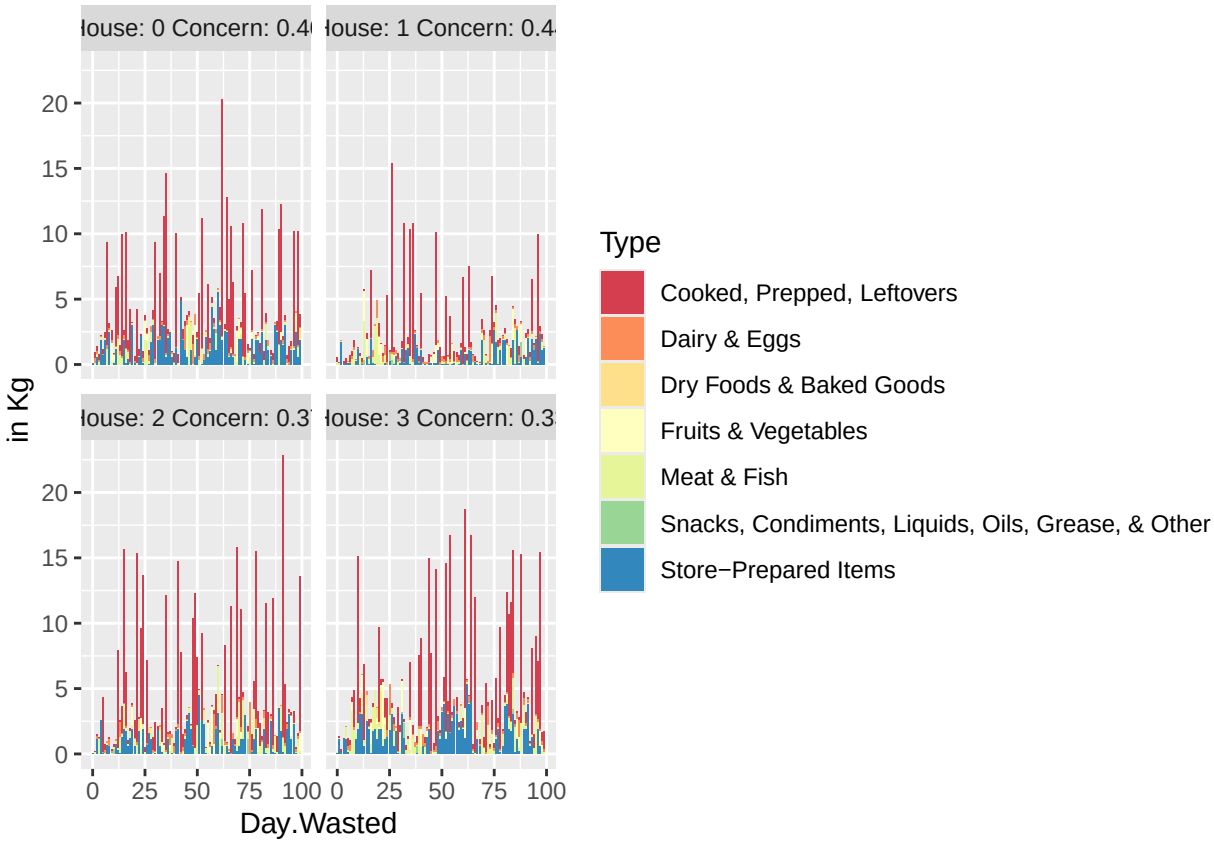
```
## pdf
##   2
```

Food waste per household and food type

2. Food waste by household divided by type in kg

```
# Plotting
wasted_by_household_kg <- ggplot(data = hh_foodwasted) +
  geom_bar(mapping = aes(x = Day.Wasted, fill = Type, weight = Kg), position = "stack") +
  facet_wrap(~House, labeller = labeller(House = as_labeller(setNames(label_house_num_people, hh_config$
  ylab("in Kg") +
  scale_fill_brewer(palette = PALETTE)

# Display the plot
wasted_by_household_kg
```



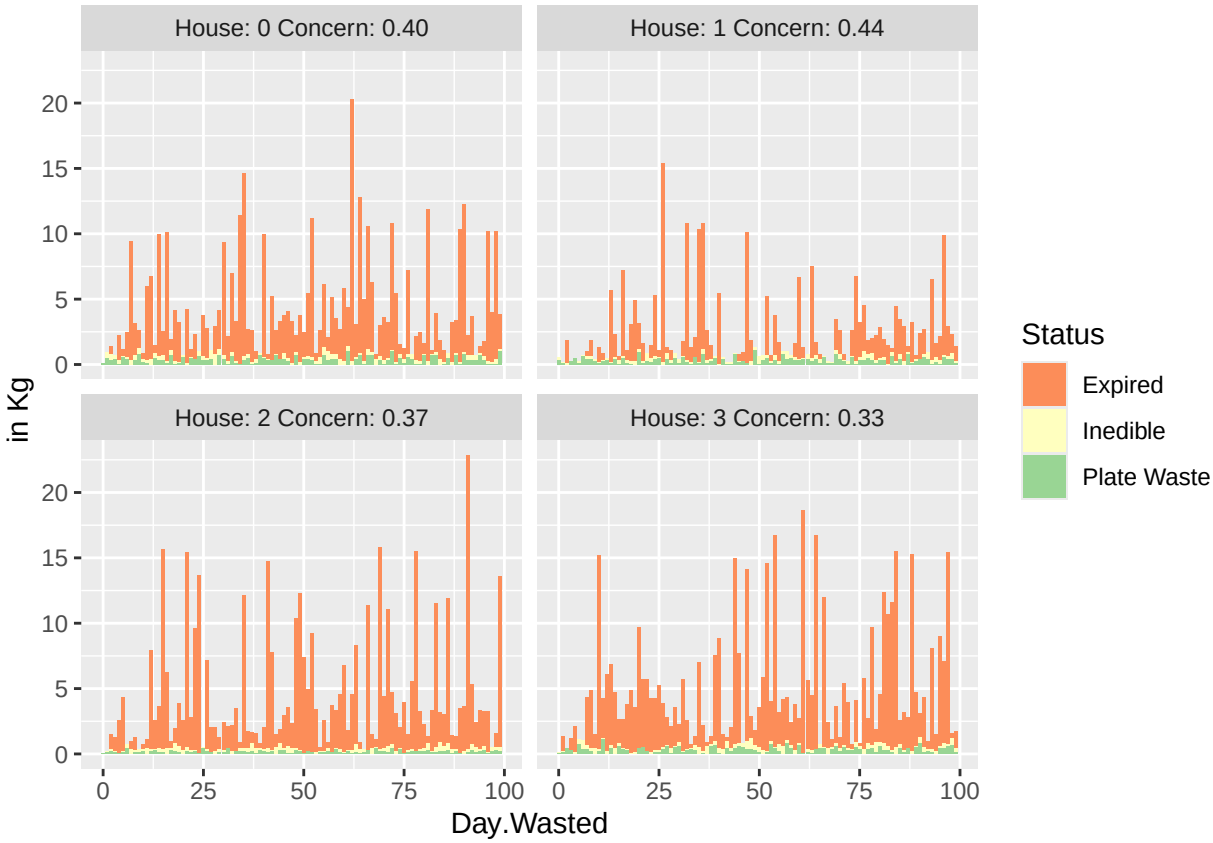

```
if (save_plots) {
   png(file = paste0(OUTPUT_FOLDER, "waste_hh_kg_type.png"), width = WIDTH, height = HEIGHT)  # Adjust
   print(wasted_by_household_kg)
   dev.off()
}
```

```
## pdf
##   2
```

3. Food waste by household divided by food waste type in kg

```
# Plotting
wasted_by_household_fw_type <- ggplot(data = hh_foodwasted) +
  geom_bar(mapping = aes(x = Day.Wasted, fill = Status, weight = Kg), position = "stack") +
  facet_wrap(~House, labeller = labeller(House = as_labeller(setNames(label_house_num_people, hh_config$
  ylab("in Kg") +
  scale_fill_brewer(palette = PALETTE)

# Display the plot
wasted_by_household_fw_type
```



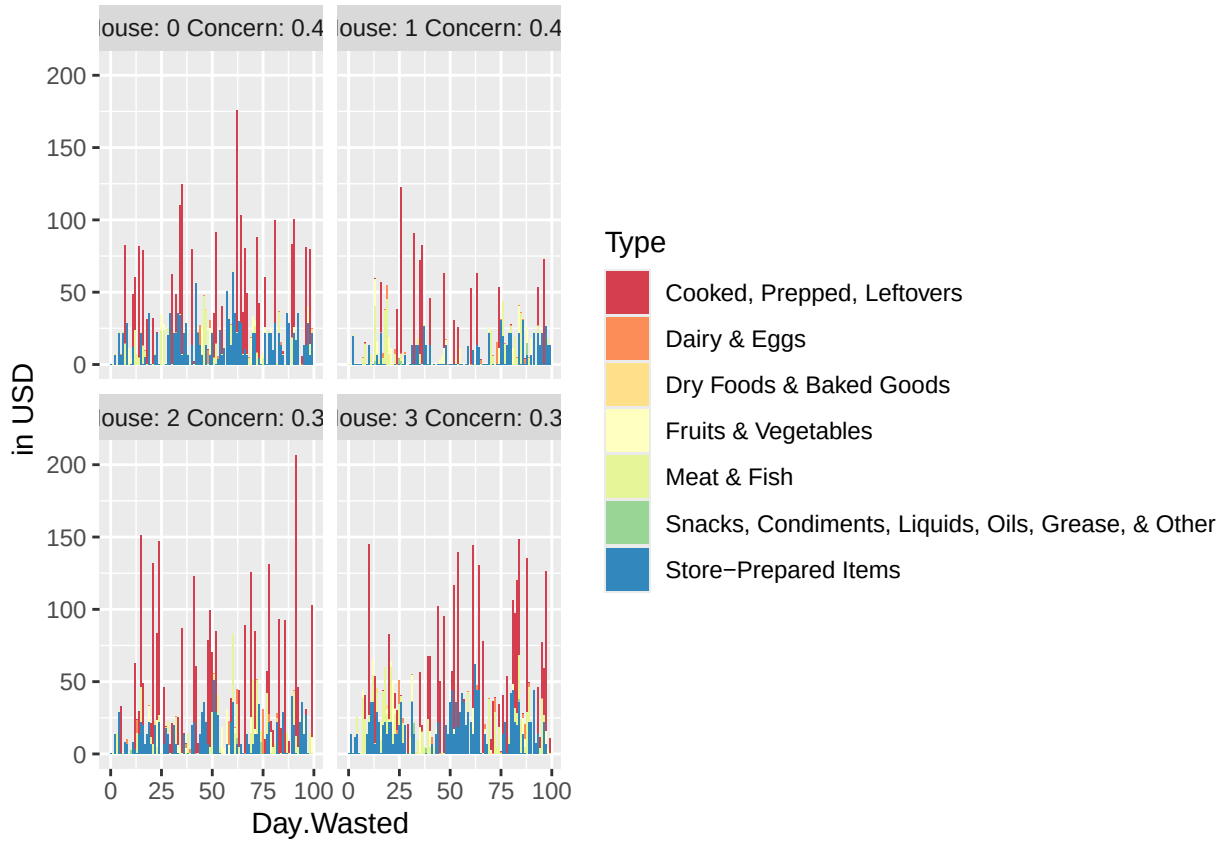

```
if (save_plots) {
   png(file = paste0(OUTPUT_FOLDER, "waste_hh_fwtype_kg.png"), width = WIDTH, height = HEIGHT)  # Adjus
   print(wasted_by_household_fw_type)
   dev.off()
}
```

```
## pdf
##   2
```

4. Food waste by household divided by food type in USD

```
# Plotting
wasted_by_household_price <- ggplot(data = hh_foodwasted) +
  geom_bar(mapping = aes(x = Day.Wasted, fill = Type, weight = Price), position = "stack") +
  facet_wrap(~House, labeller = labeller(House = as_labeller(setNames(label_house_num_people, hh_config
  ylab("in USD") +
  scale_fill_brewer(palette = PALETTE)

# Display the plot
wasted_by_household_price
```




```
if (save_plots) {
   png(file = paste0(OUTPUT_FOLDER, "waste_hh_fwtype_usd.png"), width = WIDTH, height = HEIGHT)  # Adju
   print(wasted_by_household_price)
   dev.off()
}
```

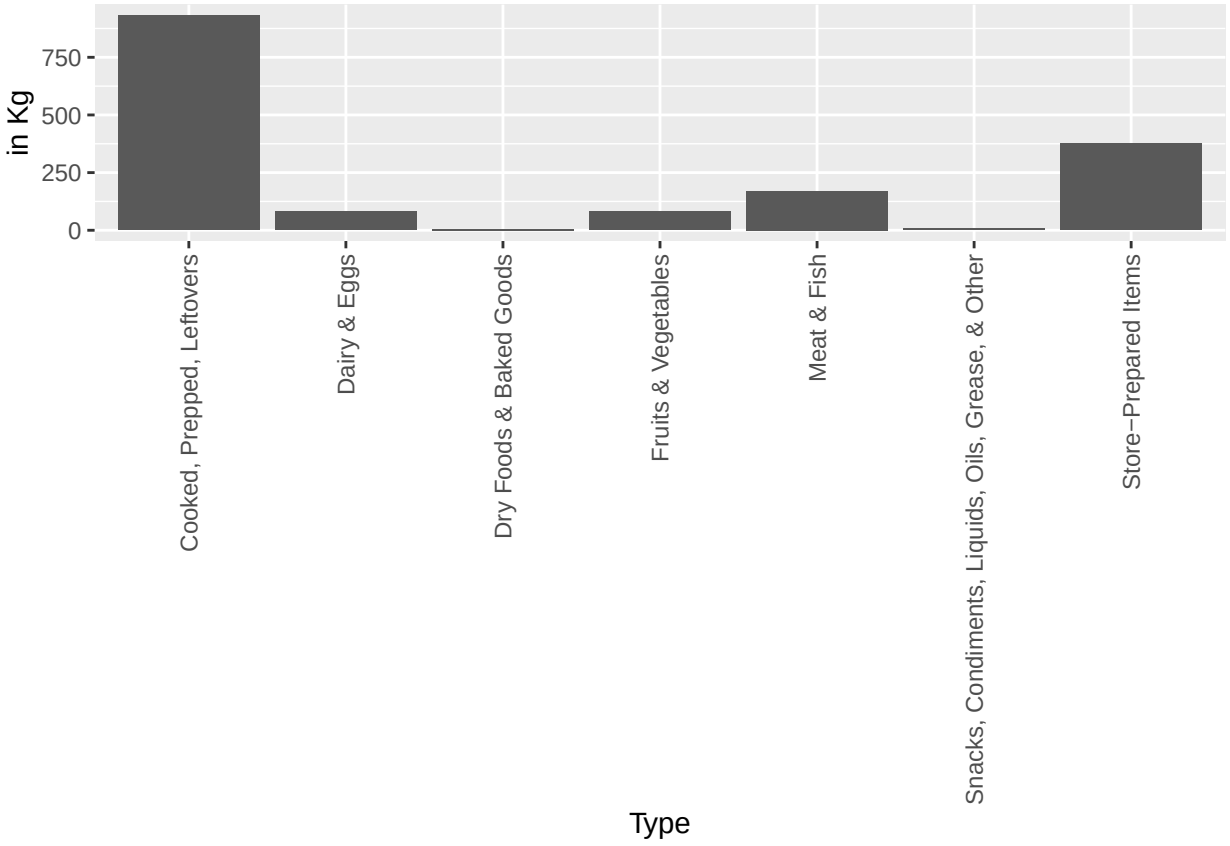
```
## pdf
##   2
```

5. Food waste type in kg

```
wasted_by_type = ggplot(data=hh_foodwasted) +
    geom_bar(mapping = aes(x=Type, fill=House, weight=Kg), position="stack") +
    ylab("in Kg") +
    theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))

wasted_by_type
```

```
## Warning: The following aesthetics were dropped during statistical transformation: fill.
## i This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## i Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
```




```
if (save_plots) {
   png(file = paste0(OUTPUT_FOLDER, "waste_type_kg.png"), width = WIDTH, height = HEIGHT)  # Adjust wid
   print(wasted_by_type)
   dev.off()
}
```

```
## Warning: The following aesthetics were dropped during statistical transformation: fill.
## i This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## i Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
```

```
## pdf
##   2
```

6. Total food wasted in Kg by food waste type:

```
sum_by_status <- aggregate(Kg ~ Status, data=hh_foodwasted, FUN = sum)
print(sum_by_status)
```

```
##        Status         Kg
## 1     Expired 1450.88539
## 2    Inedible   93.23266
## 3 Plate Waste  109.66767
```

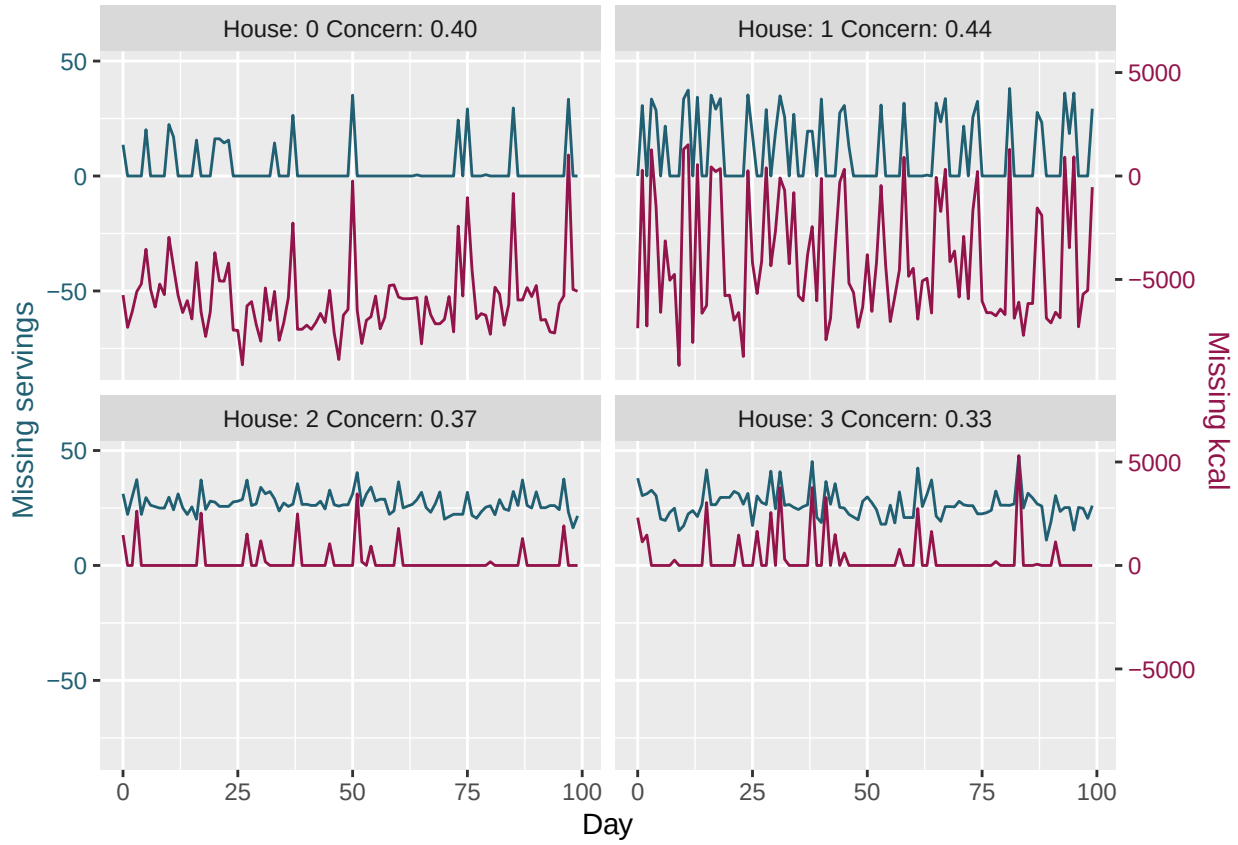
Nutrients and Behavior

1. Missing servings and kcals per household

```
scaleFactor <- max(hh_daily$Serving)/ max(hh_daily$Kcal)

consumption_serv_kcal = ggplot(data=hh_daily) +
    geom_line(mapping = aes(x=Day, y=Servings, group=House), color="#216073") +
    geom_line(mapping = aes(x=Day, y=Kcal*scaleFactor, group=House), color="#93144b") +
    scale_y_continuous(name="Missing servings", sec.axis=sec_axis(~./scaleFactor, name="Missing kcal")) +
    facet_wrap(~House, labeller = labeller(House = as_labeller(setNames(label_house_num_people, hh_confi
    theme(
        axis.title.y.left=element_text(color="#216073"),
        axis.text.y.left=element_text(color="#216073"),
        axis.title.y.right=element_text(color="#93144b"),
        axis.text.y.right=element_text(color="#93144b")
    )

consumption_serv_kcal
```




```
if (save_plots) {
   png(file = paste0(OUTPUT_FOLDER, "eaten_serv_kcal.png"), width = WIDTH, height = HEIGHT)  # Adjust w
   print(consumption_serv_kcal)
   dev.off()
}
```

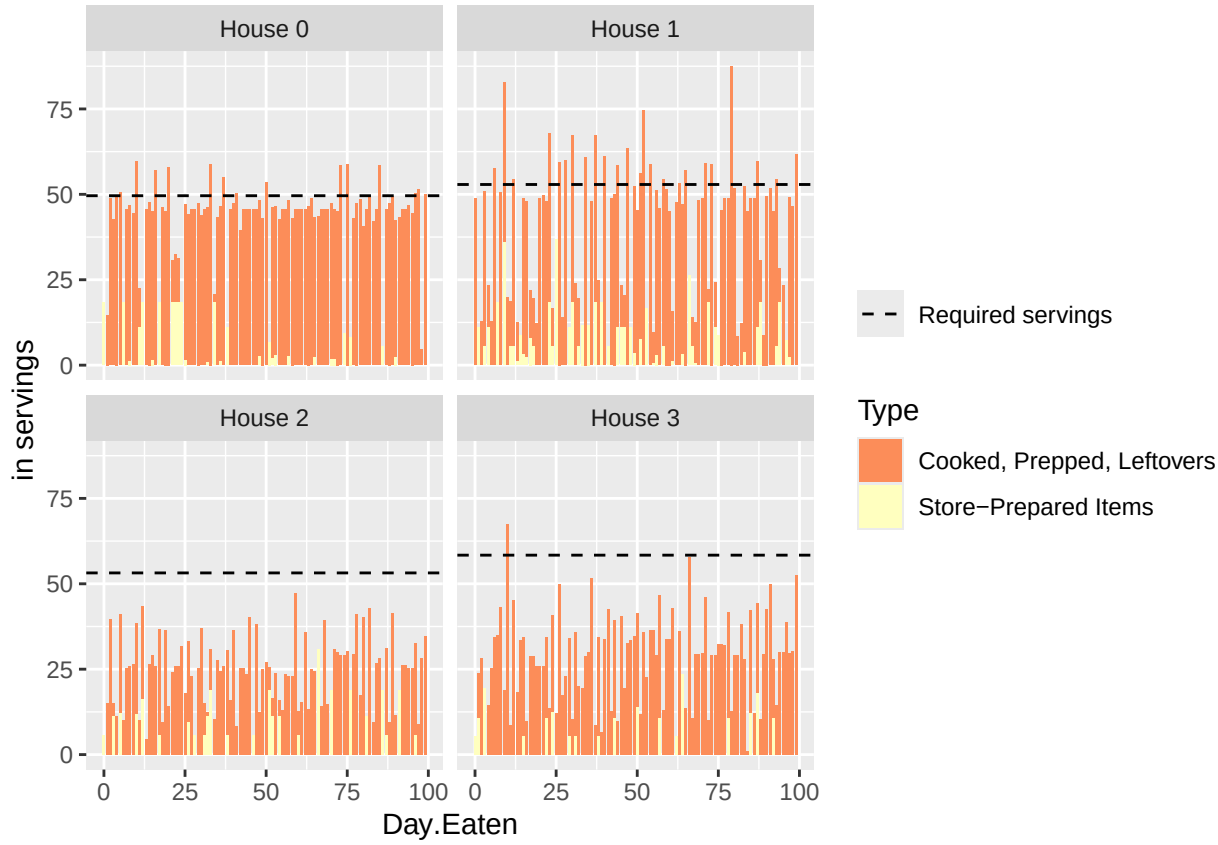
```
## pdf
```

```
##   2
```

2. Eaten servings in relation to what is required

```
consumed_by_household = ggplot(data=NULL) +
    geom_bar(data=hh_foodeaten, mapping = aes(x=Day.Eaten, fill=Type, weight=Servings), position="stack
    geom_hline(data=hh_config, mapping= aes(yintercept=RequiredServings, linetype="Threshold")) +
    facet_wrap(~House, labeller = as_labeller(house_labels)) +
    ylab("in servings") +
    scale_fill_brewer(palette = PALETTE) +
    scale_linetype_manual(values = c("dashed"), labels = c("Required servings"), name = NULL)

consumed_by_household
```



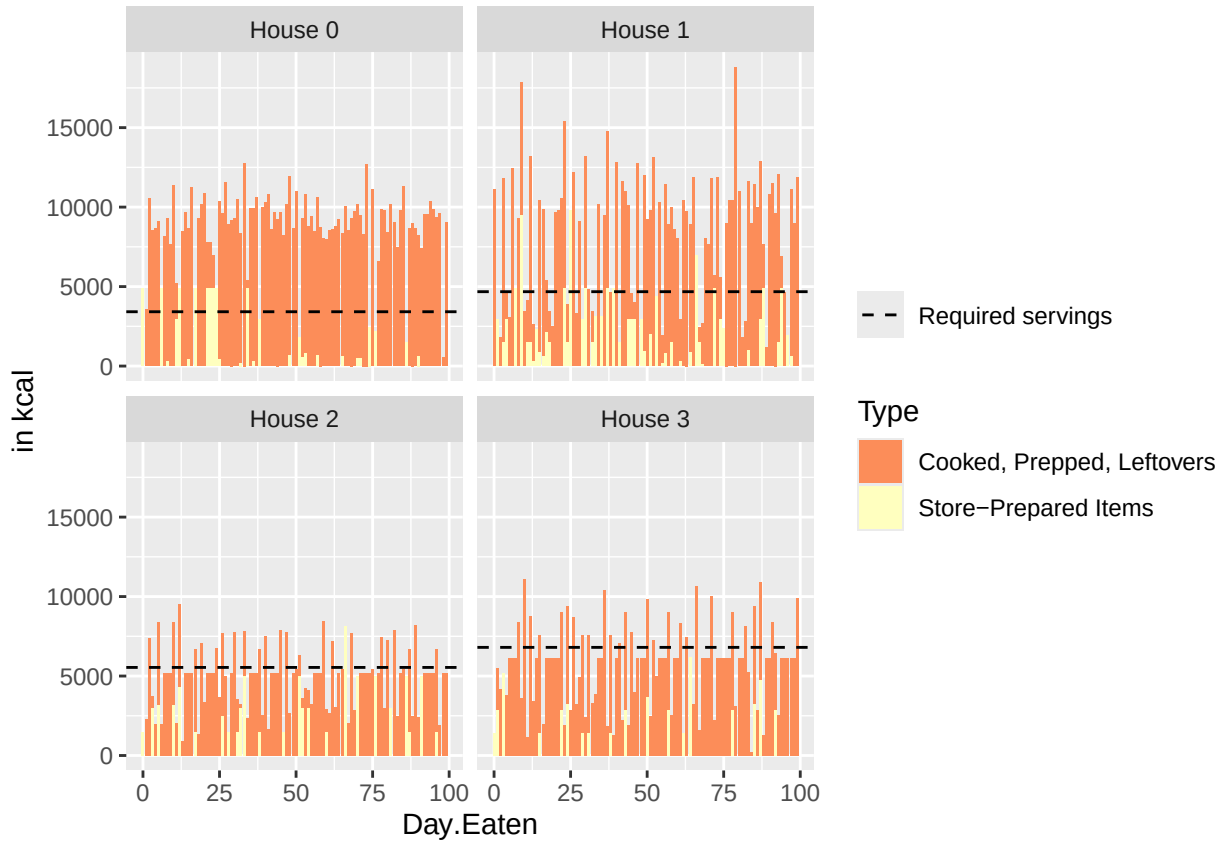

```
if (save_plots) {
   png(file = paste0(OUTPUT_FOLDER, "eaten_serv_reqserv.png"), width = WIDTH, height = HEIGHT)  # Adjus
   print(consumed_by_household)
   dev.off()
}
```

```
## pdf
##   2
```

3. Eating kcal in relation to what is required

```
consumed_by_household = ggplot(data=NULL) +
    geom_bar(data=hh_foodeaten, mapping = aes(x=Day.Eaten, fill=Type, weight=Kcal), position="stack") +
    geom_hline(data=hh_config, mapping= aes(yintercept=RequiredKcal, linetype="Threshold")) +
    facet_wrap(~House, labeller = as_labeller(house_labels)) +
    ylab("in kcal") +
    scale_fill_brewer(palette = PALETTE) +
    scale_linetype_manual(values = c("dashed"), labels = c("Required servings"), name = NULL)

consumed_by_household
```




```
if (save_plots) {
   png(file = paste0(OUTPUT_FOLDER, "eaten_kcal_reqkcal.png"), width = WIDTH, height = HEIGHT)  # Adjus
   print(consumed_by_household)
   dev.off()
}
```

```
## pdf
##   2
```

3. Meal preparation choice distribution

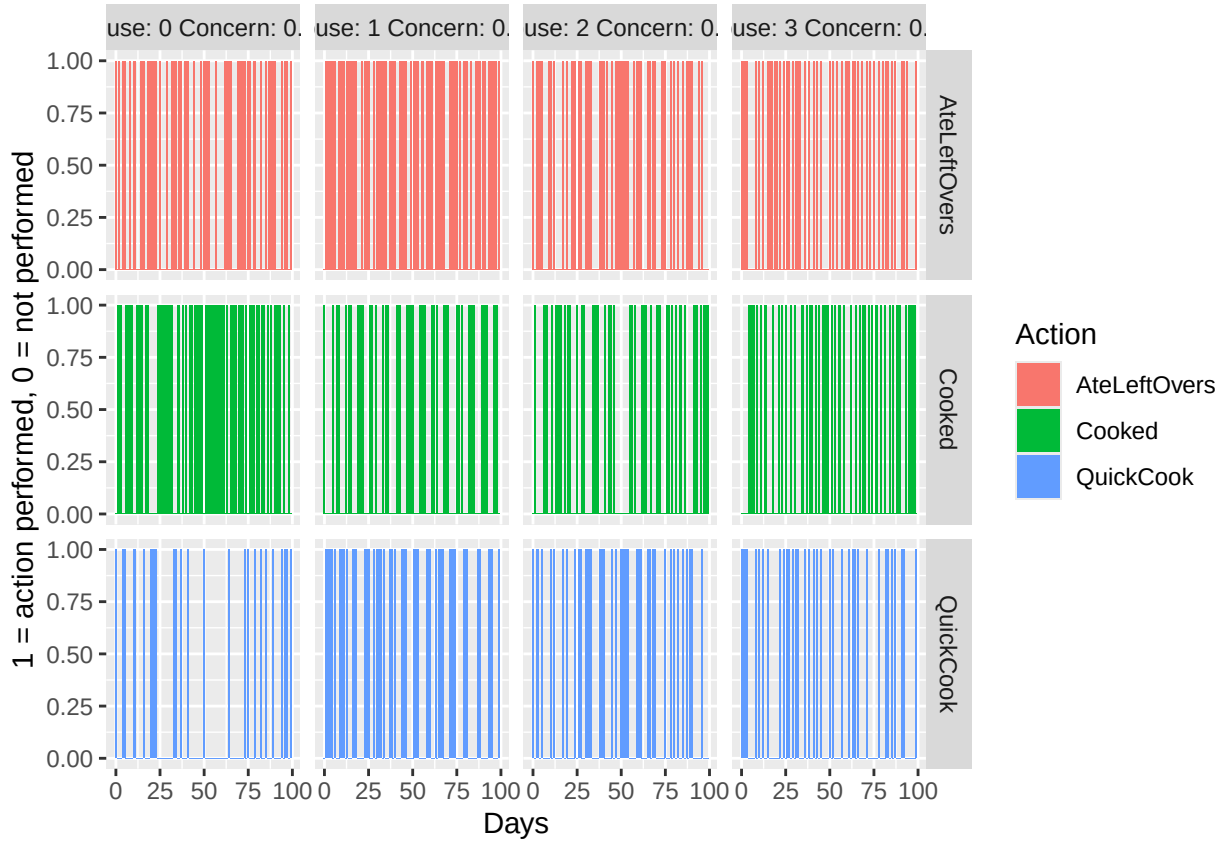
```
# Reshape data to long format
hh_daily_long <- pivot_longer(hh_daily, cols = c("Cooked", "AteLeftOvers", "QuickCook"), names_to = "Ac


label_house_concern <- as.vector(
```

```
  with(
    hh_config,
    sprintf("House: %s Concern: %.2f", House, LvlOfConcern)
))

# Create the plot
plot <- ggplot(data = hh_daily_long, aes(x = Day, y = Value, fill = Action)) +
  geom_bar(stat = "identity", position = "dodge") +
  facet_grid(rows = vars(Action), cols = vars(House),
      scales = "free",
      labeller = labeller(House = as_labeller(setNames(label_house_concern, hh_config$House)))) +
  labs(x = "Days",
       y = "1 = action performed, 0 = not performed")

plot
```




```
if (save_plots) {
   png(file = paste0(OUTPUT_FOLDER, "meal_prep_behavior.png"), width = WIDTH, height = HEIGHT)  # Adjus
   print(plot)
   dev.off()
}
```

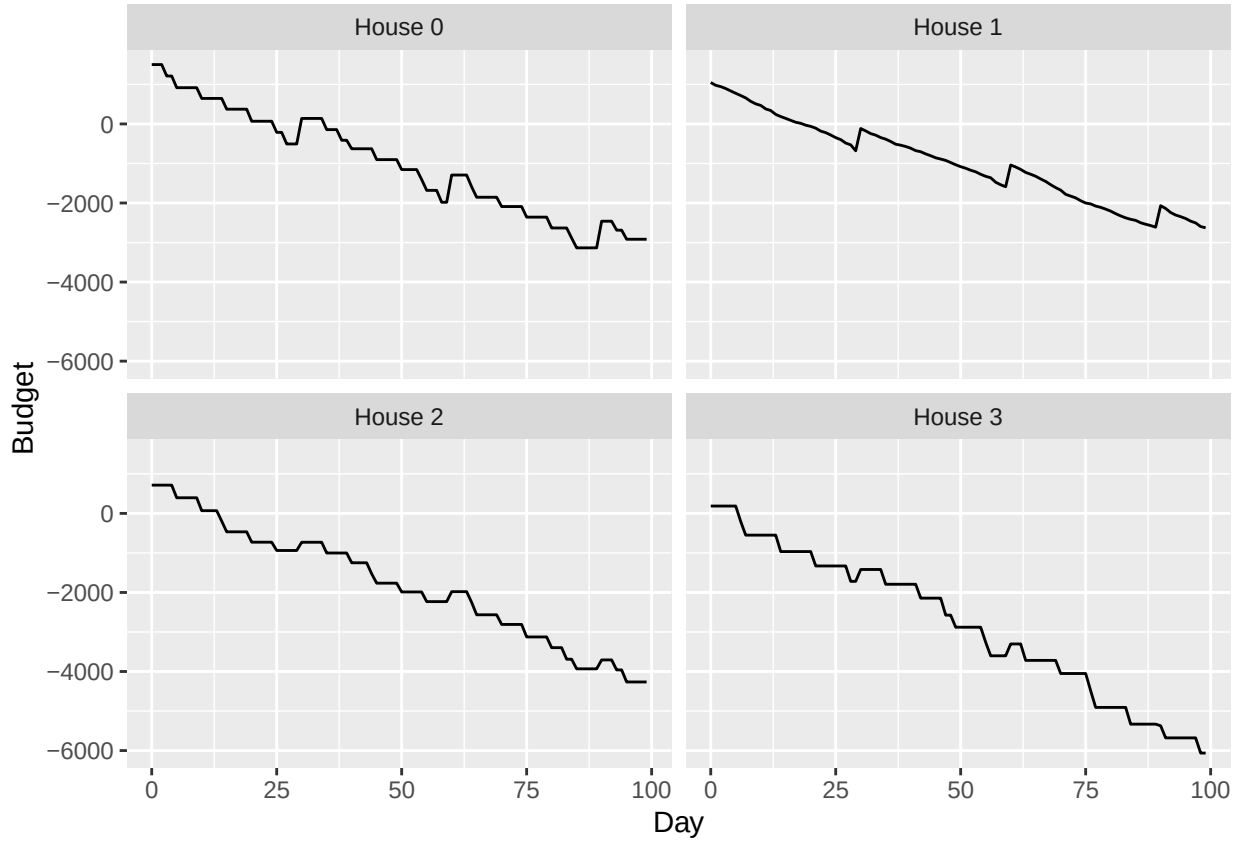
```
## pdf
##   2
```

Budget

1. Budget per household

```
budget = ggplot(data=hh_daily) +
    geom_line(mapping = aes(x=Day, y=Budget, group=House)) +
    facet_wrap(~House, nrow=2,labeller = as_labeller(house_labels))

budget
```




```
if (save_plots) {
   png(file = paste0(OUTPUT_FOLDER, "usd.png"), width = WIDTH, height = HEIGHT)  # Adjust width and hei
   print(budget)
   dev.off()
}
```

```
## pdf
##   2
```

2. Money spend on food vs. money wasted due to food waste

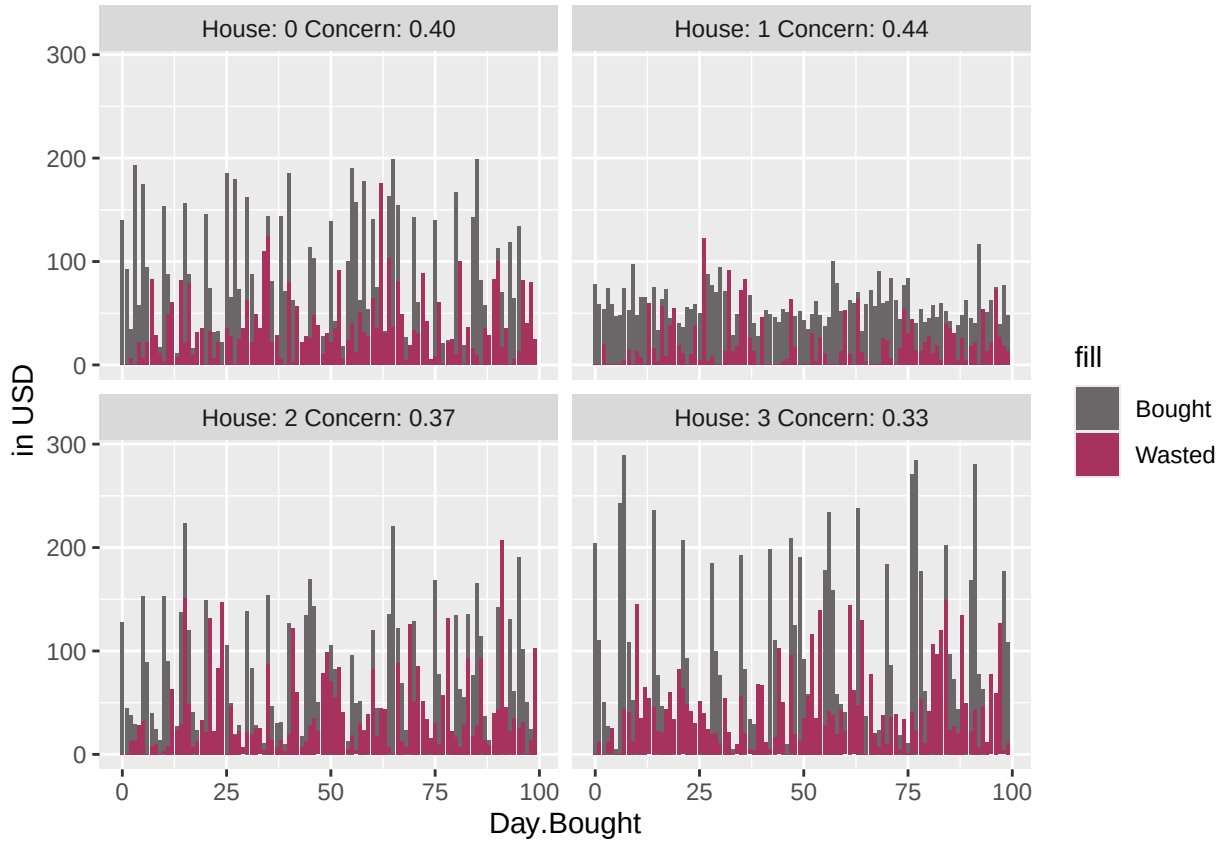
```
sum_by_status <- aggregate(Price ~ Type, data=hh_foodbought, FUN = sum)

# Plotting
bought_vs_wasted <- ggplot(data =NULL) +
  geom_bar(data=hh_foodbought, mapping = aes(x = Day.Bought, weight = Price, fill="Bought")) +
  geom_bar(data=hh_foodwasted, mapping = aes(x = Day.Wasted, weight = Price, fill="Wasted")) +
```

```
  facet_wrap(~House, labeller = labeller(House = as_labeller(setNames(label_house_num_people, hh_config$
  ylab("in USD") +
  scale_fill_brewer(palette = PALETTE) +
  scale_fill_manual(values = c( "Bought" = "#6b6668", "Wasted" = "#a8325e"))
```

```
## Scale for fill is already present.
## Adding another scale for fill, which will replace the existing scale.
```

```
# Display the plot
bought_vs_wasted
```




```
if (save_plots) {
   png(file = paste0(OUTPUT_FOLDER, "bought_waste_usd.png"), width = WIDTH, height = HEIGHT)  # Adjust
   print(bought_vs_wasted)
   dev.off()
}
```

```
## pdf
##   2
```

```
#spendings_buying <- aggregate(Price ~ Type, data=hh_foodbought, FUN = sum)
#spendings_wasting <- aggregate(Price ~ Type, data=hh_foodwasted, FUN = sum)
#difference <- spendings_buying - spendings_wasting
#time <- hh_foodbought$Day.Bought
# Plotting
#wasted_by_household <- ggplot(data = NULL) +
#    geom_col(data = difference)
#  geom_bar(data=hh_foodbought, mapping = aes(x = Day.Bought, weight = Price, fill="Bought")) +
```

```
#  geom_bar(data=hh_foodwasted, mapping = aes(x = Day.Wasted, weight = Price, fill="Wasted")) +
#  facet_wrap(~House, labeller = labeller(House = as_labeller(setNames(label_house_num_people, hh_confi
#  ylab("in USD") +
#  scale_fill_brewer(palette = PALETTE) +
#  scale_fill_manual(values = c( "Bought" = "#6b6668", "Wasted" = "#a8325e")) +
#  #ggtitle("Spendings for groceries vs lost value due to waste")

# Display the plot
#print(wasted_by_household)
```

```
### TODO: split by household and sum


#spendings_buying <- aggregate(Price ~ Day.Bought , data=hh_foodbought, FUN = sum)
#spendings_wasting <- aggregate(Price ~ Day.Wasted , data=hh_foodwasted, FUN = sum)

#difference <- spendings_buying - spendings_wasting
#time <- unique(hh_foodbought$Day.Bought)

#print(time)
#print(difference)

#df <- data.frame(
#    "difference" = c(difference),
#    "time" = c(time)
#)

#df$direction <- ifelse(df$difference >= 0, "Positive", "Negative")

# Plot the differences
#ggplot(df, aes(x = time, y = difference, fill = direction)) +
#  geom_bar(stat = "identity", position = "identity") +
#  scale_fill_manual(values = c("Positive" = "blue", "Negative" = "red")) +
#  labs(x = "Category", y = "Difference", title = "Difference between Values") +
#  theme_minimal()
```