

caf_diet

Yanireth Jimenez

2022-06-06

Load libraries

```
pacman::p_load(  
  ggplot2,  
  tidyverse,  
  googlesheets4,  
  ggpubr,  
  ggrepel  
)
```

Load data

```
# 2h daily exposure to cafeteria diet  
caf_diet_intake_raw <- read_sheet("https://docs.google.com/spreadsheets/d/1yN0byHvCX7\_lhpqhMryK8NE8271o")  
  
## ! Using an auto-discovered, cached token.  
  
## To suppress this message, modify your code or options to clearly consent to  
## the use of a cached token.  
  
## See gargle's "Non-interactive auth" vignette for more details:  
## <https://gargle.r-lib.org/articles/non-interactive-auth.html>  
  
## i The googlesheets4 package is using a cached token for 'l.luarte@uc.cl'.  
  
## v Reading from "ingesta_caf".  
## v Range 'ingesta_caf'.  
  
# 22 hour chow (standard diet) intake  
chow_diet_intake_raw <- read_sheet("https://docs.google.com/spreadsheets/d/1LhdjUG9xIxbZ3kmRWPWzSAUHyMc")  
  
## v Reading from "ingesta_chow_22hr".  
## v Range 'Hoja 1'.  
  
# cafeteria diet macronutrient information  
macronutrients_info <- read_sheet("https://docs.google.com/spreadsheets/u/1/d/1a7jy9vIB0iPlJYsKFF4vUyEt")  
  
## v Reading from "macronutrientes".  
## v Range 'macronutrientes'.
```

Data preproc

```
# daily cafeteria dieta intake
caf_diet_intake <- caf_diet_intake_raw %>%
  # fix hour in and hour out format
  mutate(
    hour_in = hms::as_hms(hour_in),
    hour_out = hms::as_hms(hour_out),
    # corrected intake
    # criollitas
    criollitas_intake = criollitas_in_gr - (criollitas_out_gr + criollitas_sp_gr),
    # trencito
    trencito_intake = trencito_in_gr - (trencito_out_gr + trencito_sp_gr),
    # cheetos
    cheetos_intake = cheetos_in_gr - (cheetos_out_gr + cheetos_sp_gr),
    # papas
    papas_intake = papas_in_gr - (papas_out_gr + papas_sp_gr),
    # chow
    chow_intake = chow_in_gr - (chow_out_gr + chow_sp_gr)
  ) %>%
  pivot_longer(
    cols = contains("intake"),
    names_to = "food_name",
    values_to = "corrected_intake_gr"
  ) %>%
  mutate(
    food_name = str_extract(food_name, "[^_]*")
  )

# isolate caja control to obtain percentage increase/decrease of food
# without a mice
caja_control <- caf_diet_intake_raw %>%
  filter(
    exp_tag == "caja_control"
  ) %>%
  mutate(
    # fix hour format
    hour_in = hms::as_hms(hour_in),
    hour_out = hms::as_hms(hour_out),
    # criollitas
    criollitas_percent_diff = ((criollitas_out_gr + criollitas_sp_gr) - criollitas_in_gr) /
      criollitas_in_gr,
    # cheetos
    cheetos_percent_diff = ((cheetos_out_gr + cheetos_sp_gr) - cheetos_in_gr) /
      cheetos_in_gr,
    # trencito
    trencito_percent_diff = ((trencito_out_gr + trencito_sp_gr) - trencito_in_gr) /
      trencito_in_gr,
    # papas
    papas_percent_diff = ((papas_out_gr + papas_sp_gr) - papas_in_gr) /
      papas_in_gr,
    # chow
    chow_percent_diff = ((chow_out_gr + chow_sp_gr) - chow_in_gr) /
      chow_in_gr
```

```

) %>%
select(ID:hour_out, criollitas_percent_diff:chow_percent_diff) %>%
pivot_longer(
  cols = contains("percent_diff"),
  names_to = "food_name",
  values_to = "control_percent_diff"
) %>%
mutate(
  food_name = str_extract(food_name, "[^_]*")
)

# add back caja control percent difference per food into caf_diet_intake
caf_diet_intake_cr <- caf_diet_intake %>%
left_join(
  caja_control %>% select(
    date,
    food_name,
    control_percent_diff
  ),
  by = c(
    "date",
    "food_name"
  )
) %>%
group_by(
  date,
  food_name,
  ID
) %>%
summarise(
  # corrected intake by spillage AND control cage
  # criollitas
  criollitas_intake_cr = (criollitas_in_gr + (criollitas_in_gr * control_percent_diff))
  - (criollitas_out_gr + criollitas_sp_gr),
  # trencito
  trencito_intake_cr = (trencito_in_gr + (trencito_in_gr * control_percent_diff))
  - (trencito_out_gr + trencito_sp_gr),
  # papas
  papas_intake_cr = (papas_in_gr + (papas_in_gr * control_percent_diff))
  - (papas_out_gr + papas_sp_gr),
  # cheetos
  cheetos_intake_cr = (cheetos_in_gr + (cheetos_in_gr * control_percent_diff))
  - (cheetos_out_gr + cheetos_sp_gr),
  # chow
  chow_intake_cr = (chow_in_gr + (chow_in_gr * control_percent_diff))
  - (chow_out_gr + chow_sp_gr)
) %>%
pivot_longer(
  cols = contains("intake_cr"),
  names_to = "food_name_cr",
  values_to = "corrected_intake_gr_cr"
) %>%
mutate(

```

```

    food_name = str_extract(food_name_cr, "[^_]*"),
    # if its less than 0.01 this cannot be measured by the scale
    corrected_intake_gr_cr = if_else(corrected_intake_gr_cr < 0.01, 0, corrected_intake_gr_cr)
  )

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

# macronutrient information
macronutrients_info_long <- macronutrients_info %>%
  pivot_longer(
    cols = !macronutriente,
    names_to = "food_name",
    values_to = "kcal_per_gr"
  )

# 22 hour chow intake
chow_diet_intake <- chow_diet_intake_raw %>%
  mutate(
    corrected_intake_gr = chow_22_in_gr - (chow_22_out_gr + chow_22_sp_gr),
    hour_in = hms::as_hms(hour_in),
    hour_out = hms::as_hms(hour_out)
  ) %>%
  mutate(food_name = "chow", time = "22_hr")

# TODO correct by percent difference
chow_22_control <- chow_diet_intake_raw %>%
  filter(
    exp_tag == "caja_control"
  ) %>%
  mutate(
    hour_in = hms::as_hms(hour_in),
    hour_out = hms::as_hms(hour_out),
    # percent diff in control chow
    chow_22_percent_diff = ((chow_22_out_gr + chow_22_sp_gr) - chow_22_in_gr) /
      chow_22_in_gr,
  ) %>%
  select(ID:hour_out, chow_22_percent_diff) %>%
  pivot_longer(
    cols = contains("percent_diff"),
    names_to = "food_name",
    values_to = "control_percent_diff"
  ) %>%
  mutate(
    food_name = str_extract(food_name, "[^_]*")
  )

chow_diet_intake_cr <- chow_diet_intake %>%
  left_join(
    chow_22_control %>% select(
      date,
      food_name,
      control_percent_diff
    ),

```

```

    by = c(
      "date",
      "food_name"
    )
  ) %>%
  group_by(
    date,
    food_name,
    ID
  ) %>%
  summarise(
    # corrected intake by spillage AND control cage
    # chow
    chow_22_intake_cr = (chow_22_in_gr + (chow_22_in_gr * control_percent_diff))
    - (chow_22_out_gr + chow_22_sp_gr)
  ) %>%
  pivot_longer(
    cols = contains("intake_cr"),
    names_to = "food_name_cr",
    values_to = "corrected_intake_gr_cr"
  ) %>%
  mutate(
    food_name = str_extract(food_name_cr, "[^_]*"),
    # if its less than 0.01 this cannot be measured by the scale
    corrected_intake_gr_cr = if_else(corrected_intake_gr_cr < 0.01, 0, corrected_intake_gr_cr)
  )

```

`summarise()` has grouped output by 'date', 'food_name'. You can override using
the `.groups` argument.

merge tables

```

# this adds macro nutrients to each corrected food intake
caf_diet_intake_macros <- caf_diet_intake_cr %>%
  left_join(
    macronutrients_info_long,
    by = c("food_name")
  ) %>%
  mutate(time = "2_hr") %>%
# next we compute the real kcal eaten for each food
  mutate(
    kcal_intake = kcal_per_gr * corrected_intake_gr_cr
  )

chow_22_intake_macros <- chow_diet_intake_cr %>%
  left_join(
    macronutrients_info_long,
    by = c("food_name")
  ) %>%
  mutate(
    kcal_intake = kcal_per_gr * corrected_intake_gr_cr
  )

```

```

# we bind both data sets to add up normal chow intake to the caf diet
# chow intake
caf_diet_intake_macros <- bind_rows(caf_diet_intake_macros, chow_22_intake_macros)

```

summary tables

```

# add back session info
session <- caf_diet_intake_raw %>%
  select(
    ID,
    date,
    session,
    exp_tag
  )
caf_diet_intake_macros <- caf_diet_intake_macros %>%
  left_join(
    session,
    by = c("ID", "date")
  )
# this tables considers total chow intake
# normal chow + chow consumed during cafeteria diet
caf_chow_kcal <- caf_diet_intake_macros %>%
  group_by(
    ID,
    session,
    food_name,
    macronutriente,
    exp_tag
  ) %>%
  summarise(
    kcal_intake = sum(kcal_intake)
  )

## `summarise()` has grouped output by 'ID', 'session', 'food_name',
## 'macronutriente'. You can override using the `.groups` argument.

# normal + chow consumed during cafeteria diet vs caf diet in % of daily intake kcal
caf_chow_kcal_daily_ind <- caf_chow_kcal %>%
  group_by(
    ID,
    session,
    exp_tag,
    macronutriente
  ) %>%
  summarise(
    total_intake_sum_kcal = sum(kcal_intake)
  ) %>%
  right_join(
    caf_chow_kcal,
    by = c("ID", "session", "exp_tag", "macronutriente")
  ) %>%
  mutate(
    percent_of_daily_intake = (kcal_intake / total_intake_sum_kcal) * 100,
    type_of_food = if_else(food_name == "chow", "chow", "caf")
  )

```

```

) %>%
group_by(
  ID,
  session,
  macronutriente,
  exp_tag,
  type_of_food
) %>%
summarise(
  percent_of_daily_intake_total = sum(percent_of_daily_intake)
) %>%
ungroup()

```

`summarise()` has grouped output by 'ID', 'session', 'exp_tag'. You can
 ## override using the `.groups` argument. `summarise()` has grouped output by 'ID',
 ## 'session', 'macronutriente', 'exp_tag'. You can override using the `.groups`
 ## argument.

```

# this is the same data but at a group level
caf_chow_kcal_daily_group <- caf_chow_kcal_daily_ind %>%
  filter(exp_tag != "caja_control", macronutriente == "total") %>%
  drop_na() %>%
  group_by(
    session,
    exp_tag,
    type_of_food
  ) %>%
  summarise(
    percent_of_daily_intake_total_group = mean(percent_of_daily_intake_total),
    err = sd(percent_of_daily_intake_total) / sqrt(n())
  ) %>%
  ungroup()

```

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

```

# intake by macro nutrient in kcal
# this considers only the 2hr intake
macro_intake_kcal_ind <- caf_diet_intake_macros %>%
  drop_na() %>%
  filter(time == "2_hr") %>%
  group_by(
    ID,
    session,
    macronutriente,
    exp_tag
  ) %>%
  summarise(
    total_intake_kcal = sum(kcal_intake)
  )

```

`summarise()` has grouped output by 'ID', 'session', 'macronutriente'. You can
 ## override using the `.groups` argument.

```

macro_intake_kcal_group <- macro_intake_kcal_ind %>%
  # at this point we have the total intake per session

```

```

# now we get the mean and std error
ungroup() %>%
group_by(
  session,
  macronutriente,
  exp_tag
) %>%
summarise(
  mean_total_intake_kcal = mean(total_intake_kcal),
  err = sd(total_intake_kcal) / sqrt(n())
)

```

`summarise()` has grouped output by 'session', 'macronutriente'. You can
override using the `.groups` argument.

```

# intake in gr
# intake by macro nutrient in kcal
macro_intake_gr <- caf_diet_intake_macros %>%
drop_na() %>%
filter(time == "2_hr", macronutriente == "total") %>%
group_by(
  ID,
  session,
  exp_tag,
  food_name
) %>%
summarise(
  total_intake_gr = sum(corrected_intake_gr_cr)
) %>%
# at this point we have the total intake per session
# now we get the mean and std error
ungroup() %>%
group_by(
  session,
  exp_tag,
  food_name
) %>%
summarise(
  mean_total_intake_gr = mean(total_intake_gr),
  err = sd(total_intake_gr) / sqrt(n())
)

```

`summarise()` has grouped output by 'ID', 'session', 'exp_tag'. You can override
using the `.groups` argument. `summarise()` has grouped output by 'session',
'exp_tag'. You can override using the `.groups` argument.

Plots

2 hours kcal

x axis = experimental sessions y axis = kcal or grams

```

# macro intake in kcal
p1 <- macro_intake_kcal_ind %>%
mutate(ID = as.factor(ID)) %>%

```



```

filter(
  macronutriente == "total",
  exp_tag != "caja_control"
) %>%
ggplot(aes(
  session,
  total_intake_kcal,
  group = ID,
  color = exp_tag
)) +
geom_line(alpha = 0.2) +
geom_point(alpha = 0.2) +
geom_label_repel(aes(label = ID)) +
geom_line(
  data = macro_intake_kcal_group %>%
    filter(
      macronutriente == "total",
      exp_tag != "caja_control"
    ),
  inherit.aes = FALSE,
  aes(
    session,
    mean_total_intake_kcal,
    color = exp_tag
  )
) +
geom_point(
  data = macro_intake_kcal_group %>%
    filter(
      macronutriente == "total",
      exp_tag != "caja_control"
    ),
  inherit.aes = FALSE,
  aes(
    session,
    mean_total_intake_kcal,
    color = exp_tag
  )
) +
geom_errorbar(
  data = macro_intake_kcal_group %>%
    filter(
      macronutriente == "total",
      exp_tag != "caja_control"
    ),
  inherit.aes = FALSE,
  aes(
    session,
    mean_total_intake_kcal,
    color = exp_tag,
    ymin = mean_total_intake_kcal - err,
    ymax = mean_total_intake_kcal + err
  ),

```

```

    width = 0.01
  ) +
  theme_pubr() +
  labs(color = "Experimental condition") +
  ylab("Mean intake (kcal)") +
  xlab("Sessions") +
  scale_x_continuous(breaks = seq(1, max(macro_intake_kcal_ind$session), 1))

```

2 hours gr per food type

```

# macro intake in gr
p2 <- macro_intake_gr %>%
  filter(
    exp_tag != "caja_control"
  ) %>%
  ggplot(aes(
    as.factor(session),
    group = food_name,
    mean_total_intake_gr,
    color = food_name,
    ymin = mean_total_intake_gr - err,
    ymax = mean_total_intake_gr + err
  )) +
  geom_point() +
  geom_line() +
  geom_errorbar(width = 0.1) +
  theme_pubr() +
  labs(color = "Experimental condition") +
  ylab("Mean intake (gr)") +
  xlab("Sessions") +
  facet_wrap(~exp_tag)

```

percent intake in 24 hours of cafeteria dieta vs chow

x axis = experimental sessions y axis = % of intake in kcal caf vs chow

```

p3 <- caf_chow_kcal_daily_group %>%
  filter(exp_tag != "caja_control") %>%
  ggplot(aes(
    session,
    percent_of_daily_intake_total_group,
    color = type_of_food
  )) +
  geom_hline(yintercept = 50, color = "gray") +
  geom_point() +
  geom_line() +
  geom_errorbar(aes(
    ymin = percent_of_daily_intake_total_group - err,
    ymax = percent_of_daily_intake_total_group + err
  ), width = 0.01) +
  geom_point(
    inherit.aes = FALSE,
    data = caf_chow_kcal_daily_ind %>%

```

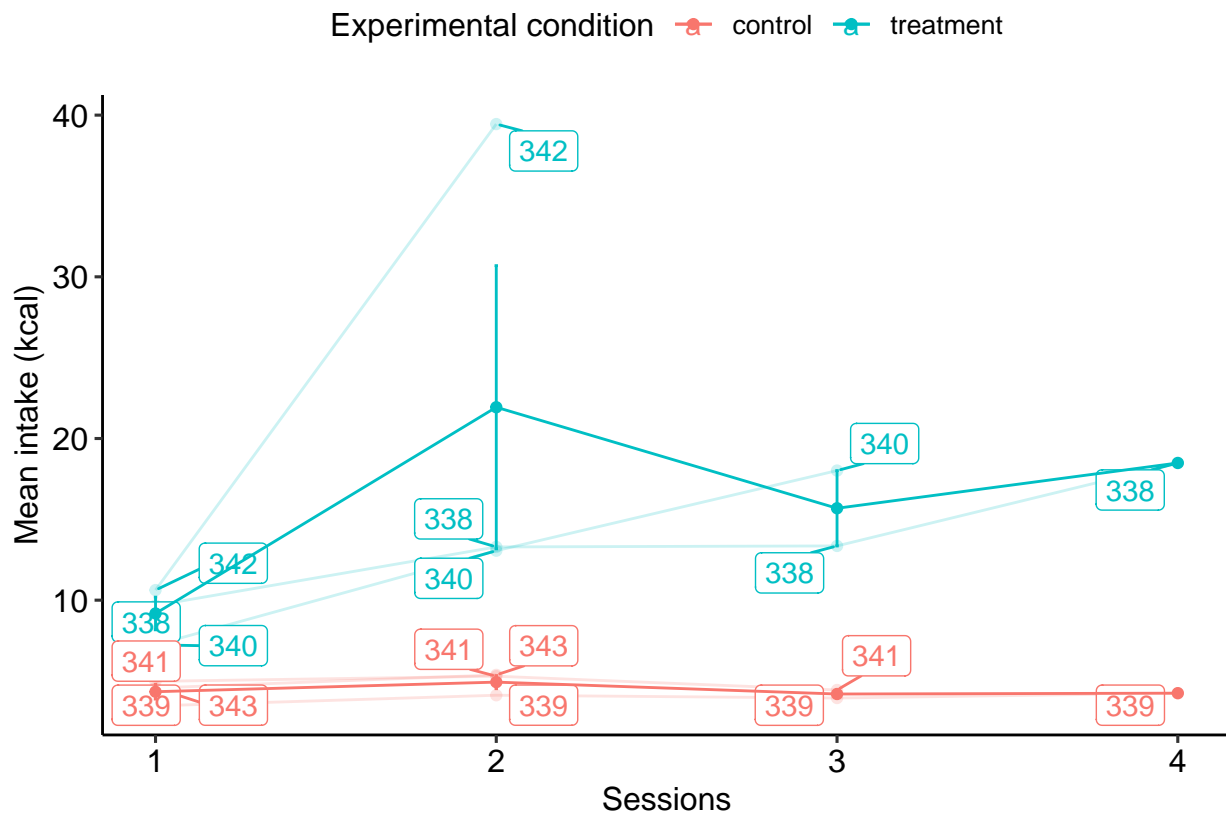
```

    filter(macronutriente == "total", exp_tag != "caja_control"),
    aes(
      session,
      percent_of_daily_intake_total,
      color = type_of_food
    ), alpha = 0.3
  ) +
  geom_label_repel(
    inherit.aes = FALSE,
    data = caf_chow_kcal_daily_ind %>%
      filter(macronutriente == "total", exp_tag != "caja_control"),
    aes(
      session,
      percent_of_daily_intake_total,
      color = type_of_food,
      label = ID
    ), size = 2
  ) +
  facet_wrap(~exp_tag) +
  ylab("Mean percent of daily intake") +
  scale_x_continuous(breaks = seq(1, max(caf_chow_kcal_daily_group$session), 1)) +
  theme_pubr()

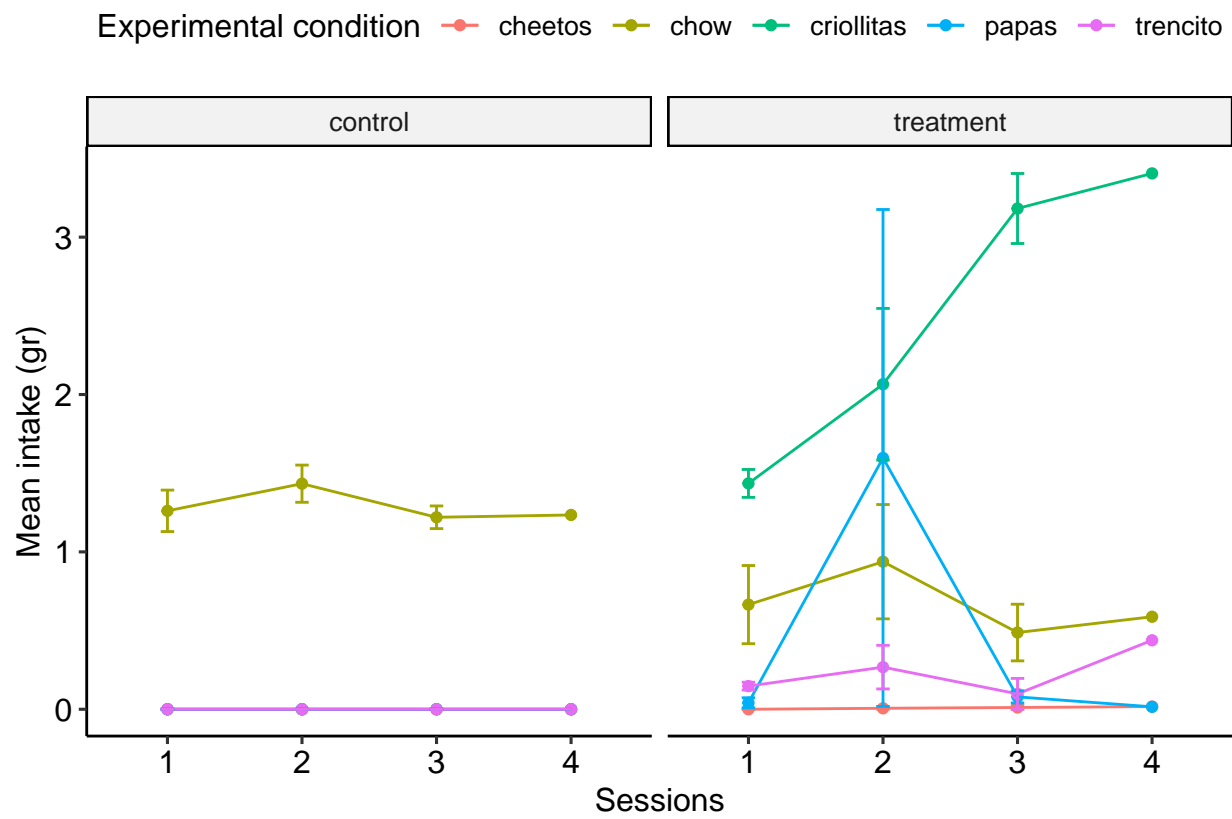
```

save plots

p1



p2



p3

