Lira_combined_Statistical

Asger_Wretlind

2022-07-27

```r
library(here)
library(tidyverse)
library(vroom)
library(tableone)

#Color Palette inspired by Kandinsky - Circles in a circle
color_palette <- c("yellow" = "#F2D479",
                   "pink" = "#BF4E83",
                   "green" = "#36BFA6",
                   "red" = "#C50B42",
                   "blue" = "#1262B2",
                   "sand" = "#F9E2C0",
                   "white" = "#E7E1D1")
```

```r
#Import LirAlbu preprocessed data
data_albu <- vroom(here("data/0097_liralbu_4cer_data_preprocessed.csv"))

#Convert factors
data_albu <- data_albu %>%
    mutate(across(which(
        apply(., 2, function(x) {
            tmp <- unique(x)
            length(tmp) - sum(is.na(tmp)) == 2L})),
        ~ factor(.))) %>%
    mutate_at(c("PrePost", "Time_point"), ~ factor(.))

#Import LiraFlame preprocessed data
data_flame <- vroom(here("data/0083_liraflame_4cer_data_preprocessed.csv"))

#Convert factors
data_flame <- data_flame %>%
    mutate(across(which(
        apply(., 2, function(x) {
            tmp <- unique(x)
            length(tmp) - sum(is.na(tmp)) == 2L})),
        ~ factor(.)))
```

```r
#Vector of clinical characteristics to for CCtable_albu
CCtable_vars_albu <- c("Age", "Sex", "Weight", "Diabetes_Duration", "HbA1C_mmol_mol",
                       "HbA1c_percent", "Total_Cholesterol", "LDL", "Triglyceride",
                       "OfficeSBP", "OfficeDBP",
                       "Plasma_Creatinine", "eGFR_CKDEPI", "Log10MeanUAER")
```

```r
#Clinical characteristics table albu at baseline
CCtable_albu <- data_albu %>%
    filter(Time_point == 1) %>%
    mutate(Sex = factor(Sex, levels = c(1, 0), labels = c("Man", "Woman"))) %>%
    rename(HbA1C_mmol_mol = HbA1c) %>%
    mutate(HbA1c_percent = (HbA1C_mmol_mol/10.929)+2.15) %>%
    CreateTableOne(data = ., vars = CCtable_vars_albu)

#Clinical characteristics table albu before each treatment
CCtable_albu_stratified <- data_albu %>%
    filter(Visit == 1) %>%
    mutate(Sex = factor(Sex, levels = c(1, 0), labels = c("Man", "Woman"))) %>%
    rename(HbA1C_mmol_mol = HbA1c) %>%
    mutate(HbA1c_percent = (HbA1C_mmol_mol/10.929)+2.15) %>%
    CreateTableOne(data = ., vars = CCtable_vars_albu, strata = "Treatment")

# #Save CCtable_albu
# write.csv(print(CCtable_albu, printToggle = FALSE), here("data/CCtable_albu.csv"))
# write.csv(print(CCtable_albu_stratified, printToggle = FALSE), here("data/CCtable_albu_stratified.csv

#UAER median and IQR
#Start of trial total
quantile(exp(data_albu$Log10MeanUAER)[data_albu$Visit == 1], na.rm = TRUE)
```

```
##        0%       25%       50%       75%      100% 
##  3.506440  6.858487  8.467922 12.782060 32.186905
```

```r
#Before Liraglutide
quantile(exp(data_albu$Log10MeanUAER[data_albu$Visit == 1 & data_albu$Treatment == "Liraglutide"]),
         na.rm = TRUE)
```

```
##        0%       25%       50%       75%      100% 
##  3.506440  6.718327  8.749661 14.453193 30.569415
```

```r
#Before Placebo
quantile(exp(data_albu$Log10MeanUAER[data_albu$Visit == 1 & data_albu$Treatment == "Placebo"]),
         na.rm = TRUE)
```

```
##        0%       25%       50%       75%      100% 
##  4.660147  6.901838  8.115836 12.176272 32.186905
```

```r
##Now for LiraFlame

#Vector of clinical characteristics to for CCtable_flame
CCtable_vars_flame <- c("age", "Sex", "V2_B_VAEGT", "dm_var", "HbA1C_mmol_mol",
                        "HbA1c_percent", "CHOL_v2", "LDL_v2", "TRIG_v2",
                        "Sys_gen_v2_beregnet", "Dia_gen_v2_beregnet",
                        "V2_B_CREA", "GFRepi_V2", "V2_U_LogMeanUAER")

#Clinical characteristics table flame, stratified treatment type
CCtable_flame <- data_flame %>%
    filter(Time_point == 1) %>%
    mutate(Sex = factor(Sex, levels = c(1, 2), labels = c("Man", "Woman"))) %>%
    rename(HbA1C_mmol_mol = V2_B_HBA1C) %>%
    mutate(HbA1c_percent = (HbA1C_mmol_mol/10.929)+2.15) %>%
    mutate(Treatment = factor(Treatment,
                              levels = c(1, 2),
                              labels = c("Liraglutide", "Placebo"))) %>%
    CreateTableOne(data = ., vars = CCtable_vars_flame, strata = "Treatment", addOverall = TRUE)

# #Save CCtable_flame
# write.csv(print(CCtable_flame, printToggle = FALSE), here("data/CCtable_flame.csv"))

#UAER median and IQR
#Total
quantile(exp(data_flame$V2_U_LogMeanUAER[data_flame$Time_point == 1]), na.rm = TRUE)
```

```
##        0%       25%       50%       75%      100%
##  1.351250  1.723002  2.328206  3.460542 26.144427
```

```
#Liraglutide
quantile(exp(data_flame$V2_U_LogMeanUAER[data_flame$Time_point == 1 & data_flame$Treatment == 1]),  na.
```

```
##        0%       25%       50%       75%      100%
##  1.351250  1.849833  2.533037  3.693273 26.144427
```

```
#Placebo
quantile(exp(data_flame$V2_U_LogMeanUAER[data_flame$Time_point == 1 & data_flame$Treatment == 2]),  na.
```

```
##       0%      25%      50%      75%     100%
## 1.351250 1.611429 1.921704 3.194300 7.547296
```

```
rm(CCtable_vars_albu, CCtable_vars_flame)
rm(CCtable_albu, CCtable_albu_stratified, CCtable_flame)
```

```
#Table of ceramide means at different stages and treatment
CerTable_means_albu <- data_albu %>%
    select(PrePost, Sample_ID, starts_with("Cer")) %>%
    pivot_wider(names_from = c(PrePost, Sample_ID), values_from = starts_with("Cer")) %>%
    pivot_longer(starts_with("Cer")) %>%
    separate(name, sep = "_", c("Ceramide", "PrePost", "Sample_ID")) %>%
    pivot_wider(names_from = c(Ceramide, PrePost), values_from = value) %>%
    summarise(across(starts_with("Cer"), list(Mean = ~ mean(., na.rm = TRUE),
                                              SD = ~ sd(., na.rm = TRUE)))) %>%
    pivot_longer(everything()) %>%
    separate(name, sep = "_", c("Ceramide", "PrePost", "Measure")) %>%
    pivot_wider(names_from = c(PrePost, Measure)) %>%
    relocate(starts_with("PlaceboPost"), .after = PlaceboPre_SD)


#t-test change between post and pre.
CerTable_ttestChange_albu <- data_albu %>%
    select(PrePost, Sample_ID, starts_with("Cer")) %>%
    pivot_wider(names_from = c(PrePost, Sample_ID), values_from = starts_with("Cer")) %>%
    pivot_longer(starts_with("Cer")) %>%
    separate(name, sep = "_", c("Ceramide", "PrePost", "Sample_ID")) %>%
    pivot_wider(names_from = c(Ceramide, PrePost), values_from = value) %>%
    #filter(!if_any(everything(), ~ is.na(.x))) %>%
    summarise(across(
        ends_with("Post"),
        list(meandif = ~ t.test(x = .,
                                y = get(glue::glue(str_replace(
                                    cur_column(), "Post", "Pre"))),
                                paired = TRUE)$estimate[[1]],
             CIlower = ~ t.test(x = .,
                                y = get(glue::glue(str_replace(
                                    cur_column(), "Post", "Pre"))),
                                paired = TRUE)$conf.int[[1]],
             CIupper = ~ t.test(x = .,
                                y = get(glue::glue(str_replace(
                                    cur_column(), "Post", "Pre"))),
                                paired = TRUE)$conf.int[[2]],
             pval =  ~ t.test(x = .,
                              y = get(glue::glue(str_replace(
                                  cur_column(), "Post", "Pre"))),
                              paired = TRUE)$p.value))) %>%
    rename_with(~str_replace(., "Post", ""), everything()) %>%
    pivot_longer(everything()) %>%
    separate(name, sep = "_", c("Ceramide", "Treatment", "ttest_output")) %>%
    pivot_wider(names_from = c(Treatment, ttest_output), values_from = value) %>%
    rename_with(~ paste0("Change_", .), !contains("Ceramide"))

# t.test(data_albu$`Cer24:1`[data_albu$PrePost == "PlaceboPost"],
#        data_albu$`Cer24:1`[data_albu$PrePost == "PlaceboPre"], paired = TRUE)


#Paired sample t-test between post placebo and post liraglutide from full data
CerTable_ttestEndpoints_albu <- data_albu %>%
    select(PrePost, Sample_ID, starts_with("Cer")) %>%
    pivot_wider(names_from = c(PrePost, Sample_ID), values_from = starts_with("Cer")) %>%
    pivot_longer(starts_with("Cer")) %>%
    separate(name, sep = "_", c("Ceramide", "PrePost", "Sample_ID")) %>%
    pivot_wider(names_from = c(Ceramide, PrePost), values_from = value) %>%
    summarise(across(
        contains("LiraPost"),
        list(meandif = ~ t.test(x = .,
                                y = get(glue::glue(
                                    str_replace(cur_column(), "Lira", "Placebo"))),
                                paired = TRUE)$estimate[[1]],
             CIlower = ~ t.test(x = .,
                                y = get(glue::glue(
                                    str_replace(cur_column(), "Lira", "Placebo"))),
                                paired = TRUE)$conf.int[[1]],
             CIupper = ~ t.test(x = .,
                                y = get(glue::glue(
                                    str_replace(cur_column(), "Lira", "Placebo"))),
                                paired = TRUE)$conf.int[[2]],
             pval =  ~ t.test(x = .,
                              y = get(glue::glue(
                                  str_replace(cur_column(), "Lira", "Placebo"))),
                              paired = TRUE)$p.value))) %>%
    rename_with(~str_replace(., "_LiraPost", ""), everything()) %>%
    pivot_longer(everything()) %>%
    separate(name, sep = "_", c("Ceramide", "ttest_output")) %>%
    pivot_wider(names_from = ttest_output, values_from = value) %>%
    rename_with(~ paste0("EndVsEnd_", .), !contains("Ceramide"))


#Combine CerTable_albu (Similar style to table 2 in von Scholten (2017))
CerTable_albu <- CerTable_means_albu %>%
    left_join(., CerTable_ttestChange_albu) %>%
    left_join(., CerTable_ttestEndpoints_albu) %>%
    relocate(starts_with("Change_Lira_"), .after = LiraPost_SD)

CerTable_albu
```

```
## # A tibble: 6 x 21
##   Ceramide LiraPre_Mean LiraP~1 LiraP~2 LiraP~3 Change~4 Chang~5 Chang~6 Chang~7
##   <chr>           <dbl>   <dbl>   <dbl>   <dbl>    <dbl>   <dbl>   <dbl>   <dbl>
## 1 Cer16           0.748   0.162   0.732   0.149 -0.00975 -0.0558  0.0364   0.665
## 2 Cer18           0.546   0.164   0.534   0.153 -0.0121  -0.0680  0.0437   0.657
## 3 Cer20           0.644   0.230   0.620   0.227 -0.0215  -0.106   0.0630   0.604
## 4 Cer22           2.62    1.09    2.48    1.18  -0.194   -0.607   0.219    0.341
## 5 Cer24:0         8.77    4.82    7.98    4.33  -1.15    -3.10    0.794    0.233
## 6 Cer24:1         4.16    1.36    4.37    1.83   0.0762  -0.488   0.640    0.782
## # ... with 12 more variables: PlaceboPre_Mean <dbl>, PlaceboPre_SD <dbl>,
## #   PlaceboPost_Mean <dbl>, PlaceboPost_SD <dbl>, Change_Placebo_meandif <dbl>,
## #   Change_Placebo_CIlower <dbl>, Change_Placebo_CIupper <dbl>,
## #   Change_Placebo_pval <dbl>, EndVsEnd_meandif <dbl>, EndVsEnd_CIlower <dbl>,
## #   EndVsEnd_CIupper <dbl>, EndVsEnd_pval <dbl>, and abbreviated variable names
## #   1: LiraPre_SD, 2: LiraPost_Mean, 3: LiraPost_SD, 4: Change_Lira_meandif,
## #   5: Change_Lira_CIlower, 6: Change_Lira_CIupper, 7: Change_Lira_pval
```

```
# #Save CerTable_albu
# write.csv(CerTable_albu, here("data/CerTable_albu.csv"))

# #Combine all CerTable_albu
# CerTable_albu_full <- CerTable_albu %>%
#     left_join(CerTable_quartiles_albu, .) %>%
#     left_join(., CerTable_PercentChange_albu) %>%
#     relocate(Percent_change_Lira, .after = Change_Lira_pval) %>%
#     relocate(Percent_change_Placebo, .after = Change_Placebo_pval) %>%
#     left_join(., CerTable_EndVsEndDifference_albu) %>%
#     left_join(., CerTable_EffectSize_albu) %>%
#     pivot_longer(!starts_with("Cer")) %>%
#     pivot_wider(names_from = Ceramide, values_from = value)
#
# CerTable_albu_full
```

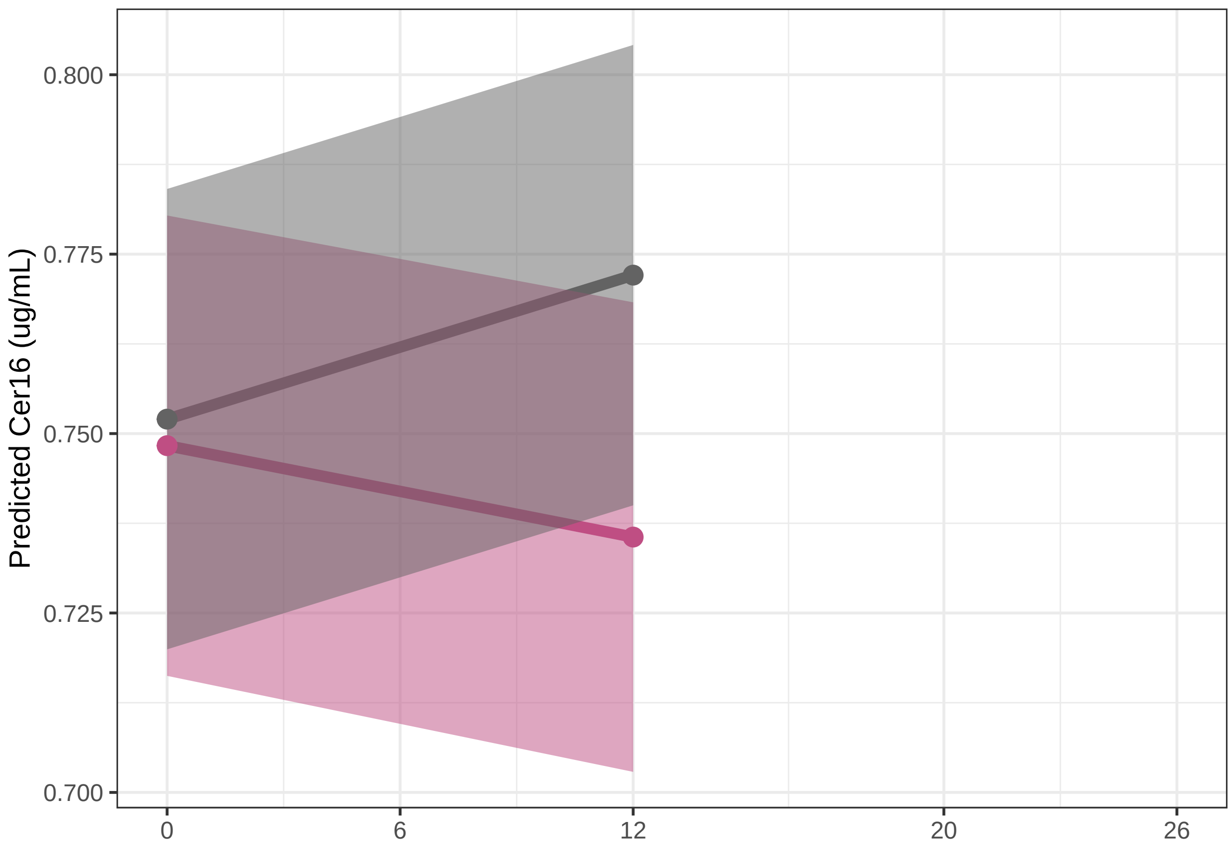
```r
# #Save CerTable_albu_full
# write.csv(CerTable_albu_full, here("data/CerTable_albu_full.csv"))

rm(CerTable_means_albu, CerTable_ttestChange_albu, CerTable_ttestEndpoints_albu,
   CerTable_quartiles_albu, CerTable_PercentChange_albu, CerTable_EffectSize_albu,
   CerTable_EndVsEndDifference_albu)
rm(CerTable_albu, CerTable_albu_full)
```

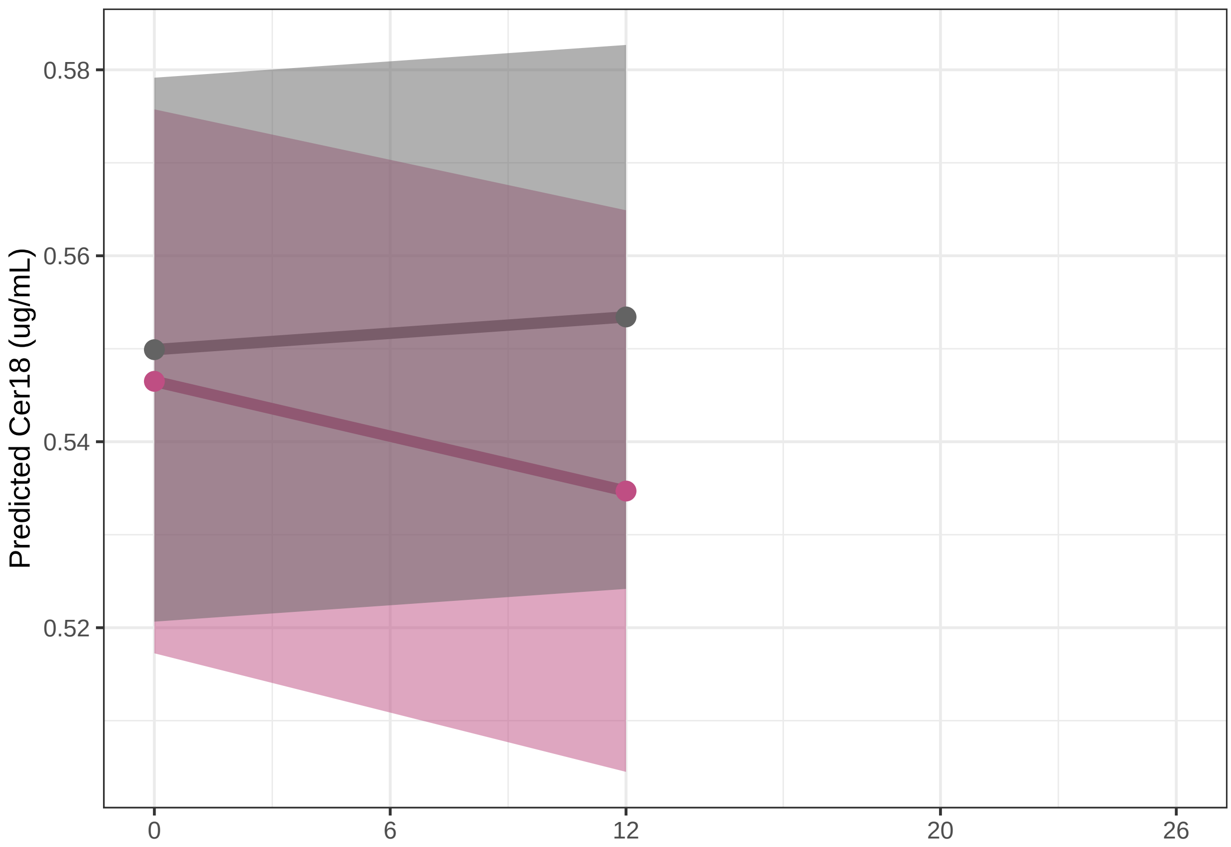
```r
library(lme4)
library(lmerTest)

#Function for extracting Coefficient (estimate), Standard error and p-value from LMM
LMM_extract <- function(Data, Formula){

    #Extract response variable
    response_var <- word(as.character(c(Formula)), 1)

    #Fit Linear mixed model
    tmp_LMM <- lmer(formula = as.formula(Formula), data = Data)

    #Summary object
    tmp_LMM_sum <- summary(tmp_LMM)

    #Extract explanetory variable(s)
    explanetory_var <- rownames(tmp_LMM_sum$coefficients)[nrow(tmp_LMM_sum$coefficients)]

    #Output
    out_df <- tibble(Resp_var = response_var,
                     Expl_var = explanetory_var,
                     LMM_Coeff = tmp_LMM_sum$coefficients[explanetory_var, "Estimate"],
                     Std_error = tmp_LMM_sum$coefficients[explanetory_var, "Std. Error"],
                     pval = tmp_LMM_sum$coefficients[explanetory_var, "Pr(>|t|)"])

    return(out_df)

}

# #Example
# data_albu %>%
#     lmer(formula =  Cer16 ~ Treatment+Visit + Treatment:Visit + (1|Sample_ID)) %>%
#     summary()
#
# LMM_extract(Data = data_albu,
#             Formula = Cer16 ~ Treatment+Visit + Treatment:Visit + (1|Sample_ID) )
```

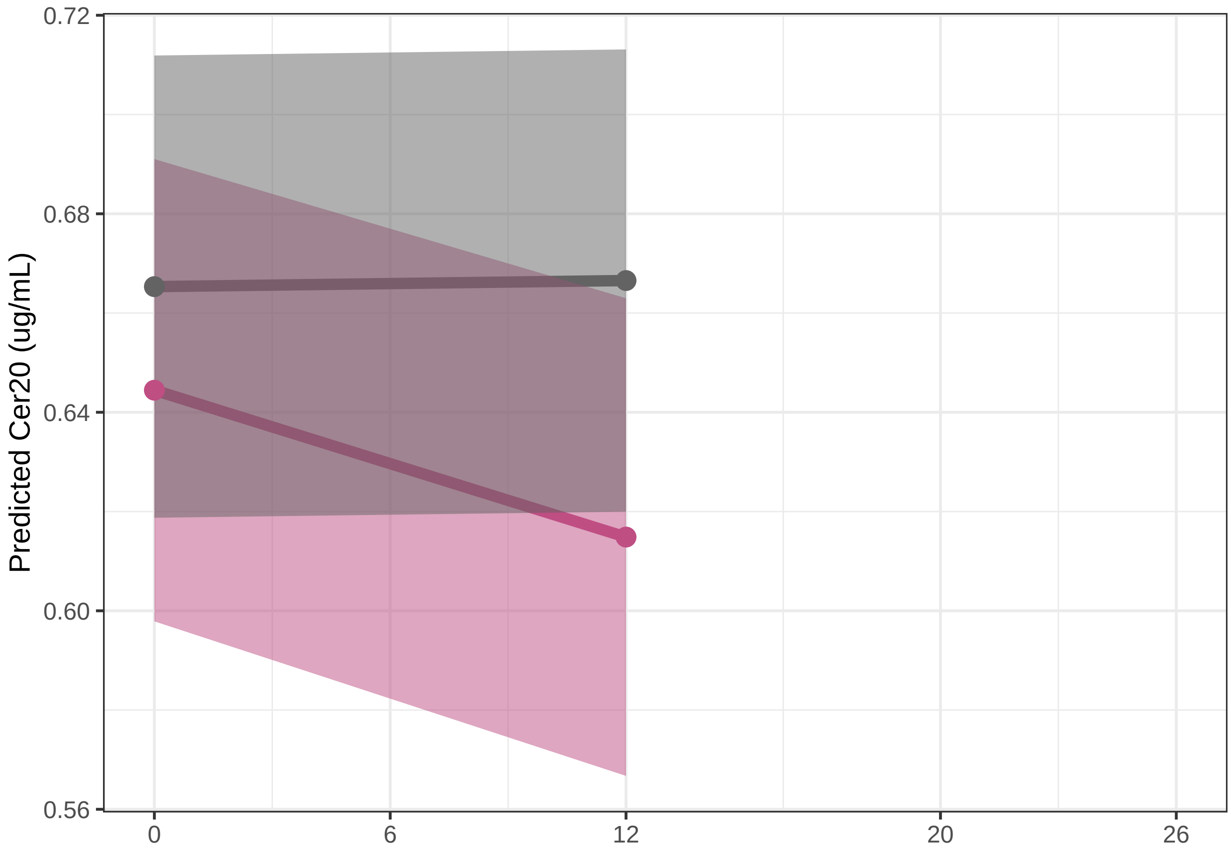
```r
#Change the factor order
data_albu <- data_albu %>%
    mutate(Treatment = factor(Treatment,levels = c("Placebo", "Liraglutide")))

#Linear mixed models for each lipid in LirAlbu
LMM_overview_albu <- data_albu %>%
    summarise(across(starts_with("Cer"),
                  ~ LMM_extract(Data = data_albu,
                      Formula = . ~ Treatment+Visit + Treatment:Visit + (1|Sample_ID)))) %>%
    pivot_longer(cols = everything()) %>%
    unnest(cols = everything()) %>%
    mutate("Project" = "LirAlbu")

#Change the factor order back
data_albu <- data_albu %>%
    mutate(Treatment = factor(Treatment,levels = c("Liraglutide", "Placebo")))

#Change the factor order
data_flame <- data_flame %>%
    mutate(Treatment = factor(Treatment,levels = c(2, 1)))

#Linear mixed models for each lipid in LiraFlame
LMM_overview_flame <- data_flame %>%
    rename(Visit = Time_point) %>%
    summarise(across(starts_with("Cer"),
                  ~ LMM_extract(Data = data_flame,
                      Formula = . ~ Treatment+Visit + Treatment:Visit + (1|Sample_ID)))) %>%
    pivot_longer(cols = everything()) %>%
    unnest(cols = everything()) %>%
    mutate("Project" = "LiraFlame")

#Change the factor order
data_flame <- data_flame %>%
    mutate(Treatment = factor(Treatment,levels = c(1, 2)))

#Combine tables
LMM_overview <- rbind(LMM_overview_albu, LMM_overview_flame) %>%
    select(-c(Resp_var, Expl_var)) %>%
    pivot_wider(names_from = Project,
                values_from = c(LMM_Coeff, Std_error, pval)) %>%
    rename(Ceramide = name) %>%
    relocate(contains("LirAlbu"), .after = Ceramide)

# #Save LMM_overview as CSV
# write.csv(LMM_overview, here("data/LMM_overview.csv"))

rm(LMM_overview_albu, LMM_overview_flame, LMM_extract)
rm(LMM_overview)
```

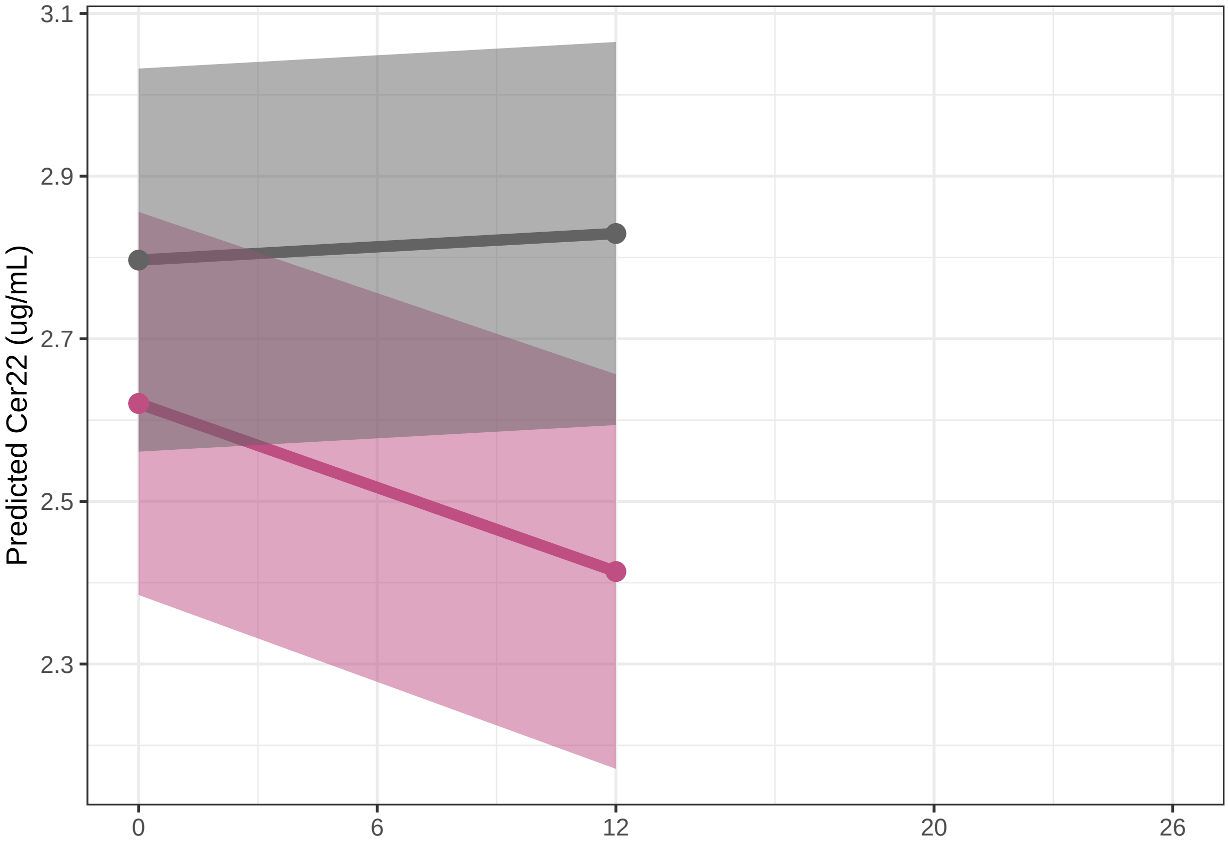
```
library(lme4)
library(ggeffects)

#Dirty fix on for error in names with ":"
colnames(data_albu) <- gsub("Cer24:", "Cer24_", colnames(data_albu))

#Ceramides to plot
tmp_cer <- data_albu %>%
    select(starts_with("Cer")) %>%
    colnames()

##LirAlbu
```

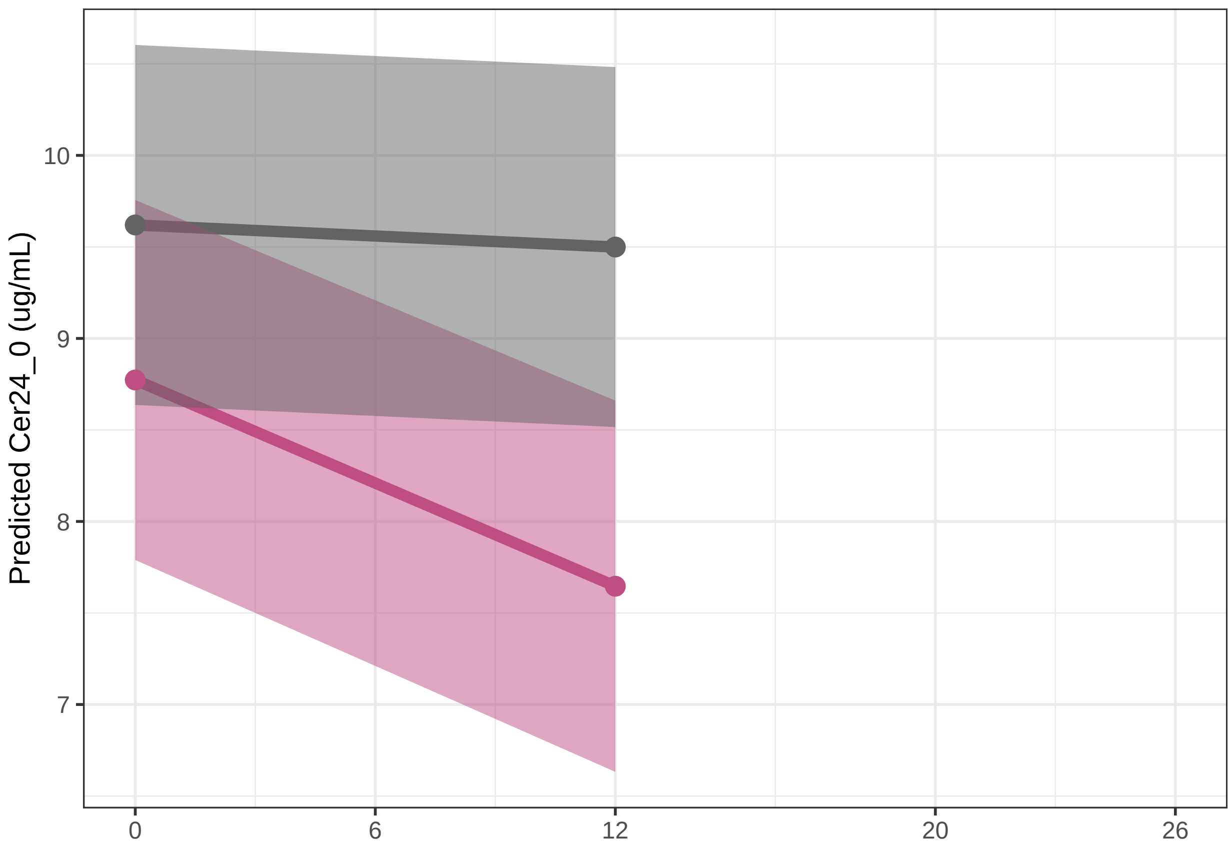
```r
#Empty list to populate in loop
LMM_albu_list <- list()

#Loop through each ceramide and save LMM model, prediction and plot into LMM_albu_list
for(i in 1:length(tmp_cer)){

    #LMM model
    model <- data_albu %>%
        #mutate(Visit = recode(Visit, `1` = 0, `2` = 12)) %>%
        lmer(formula =  paste0("`", tmp_cer[i], "`",
                               " ~ Treatment+Visit + Treatment:Visit + (1|Sample_ID)"))

    #LMM prediction
    prediction <- ggpredict(model, terms = c("Visit", "Treatment"))
    prediction$x <- c(0, 0, 12, 12)

    #LMM plot
    plot <- ggplot(prediction)+
        geom_line(aes(x = x, y = predicted, color = group), lwd = 2, alpha = 1) +
        geom_ribbon(aes(x = x, ymin = predicted - std.error,
                               ymax = predicted + std.error,fill = group),
                    alpha = 0.5) +
        geom_point(aes(x = x, y = predicted, color = group), size = 3)+

        scale_colour_manual(values = c("#BF4E83", "#636363"))+
        scale_fill_manual(values = c("#BF4E83", "#636363"))+
        scale_x_continuous(breaks = c(0, 6, 12, 20, 26), limits = c(0, 26))+
        #ylim(-2.65, -2.55)+
        ylab(paste0("Predicted ", tmp_cer[i], " (ug/mL)"))+
        xlab("Weeks") +
        #ggtitle("Trial 1") +
        theme_bw()+
        theme(legend.position= "none",#c(0.15,0.1),
              legend.background = element_rect(fill = NA),
              axis.title.x = element_blank(),
              rect = element_rect(fill = "transparent"),
              panel.background = element_rect(fill = "transparent"))

    #Populate LMM_albu_list with more lists
    LMM_albu_list[tmp_cer[i]] <- list(c(Model = list(model),
                                   Prediction = list(prediction),
                                   Plot = list(print(plot))))

    rm(model, prediction, plot)
}
```

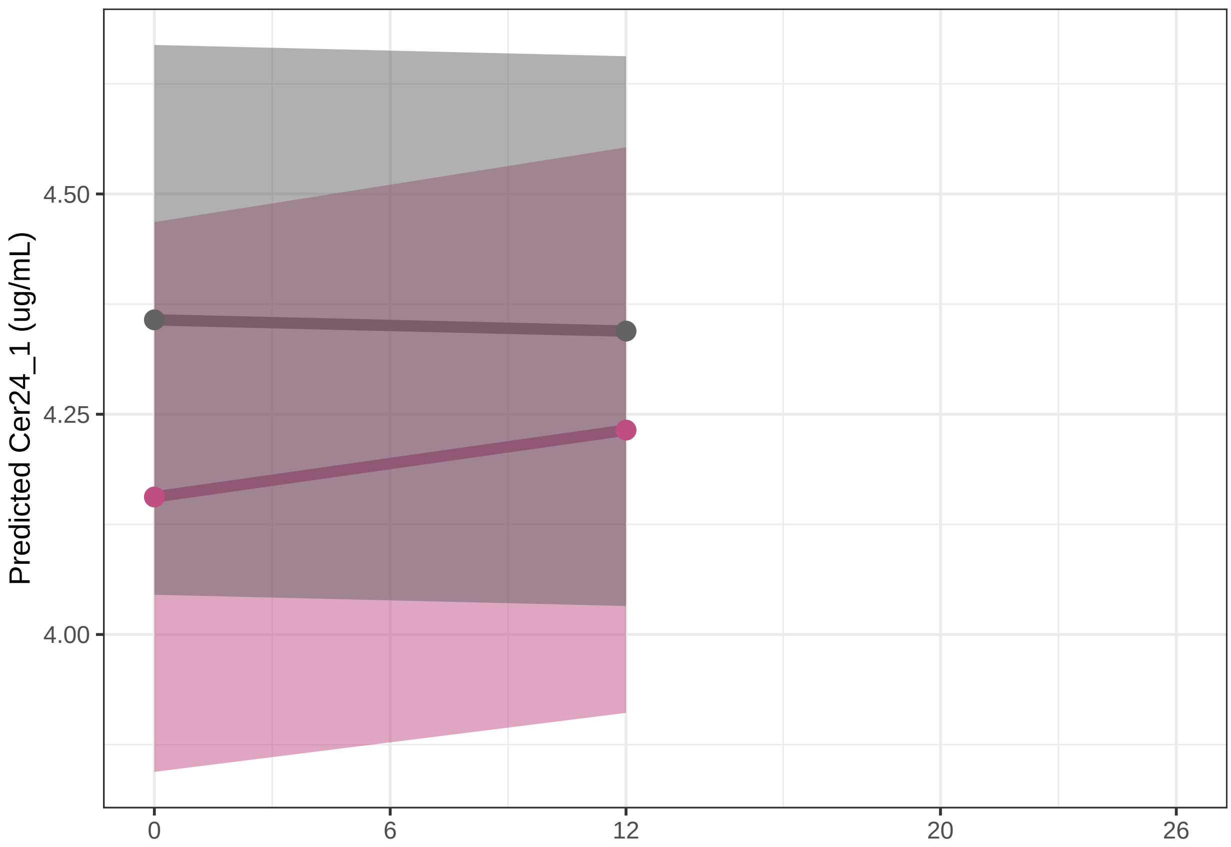

Predicted Cer16 (ug/mL)
0.800
0.775
0.750
0.725
0.700
0
6
12
20
26


Predicted Cer18 (ug/mL)
0.58
0.56
0.54
0.52
0
6
12
20
26

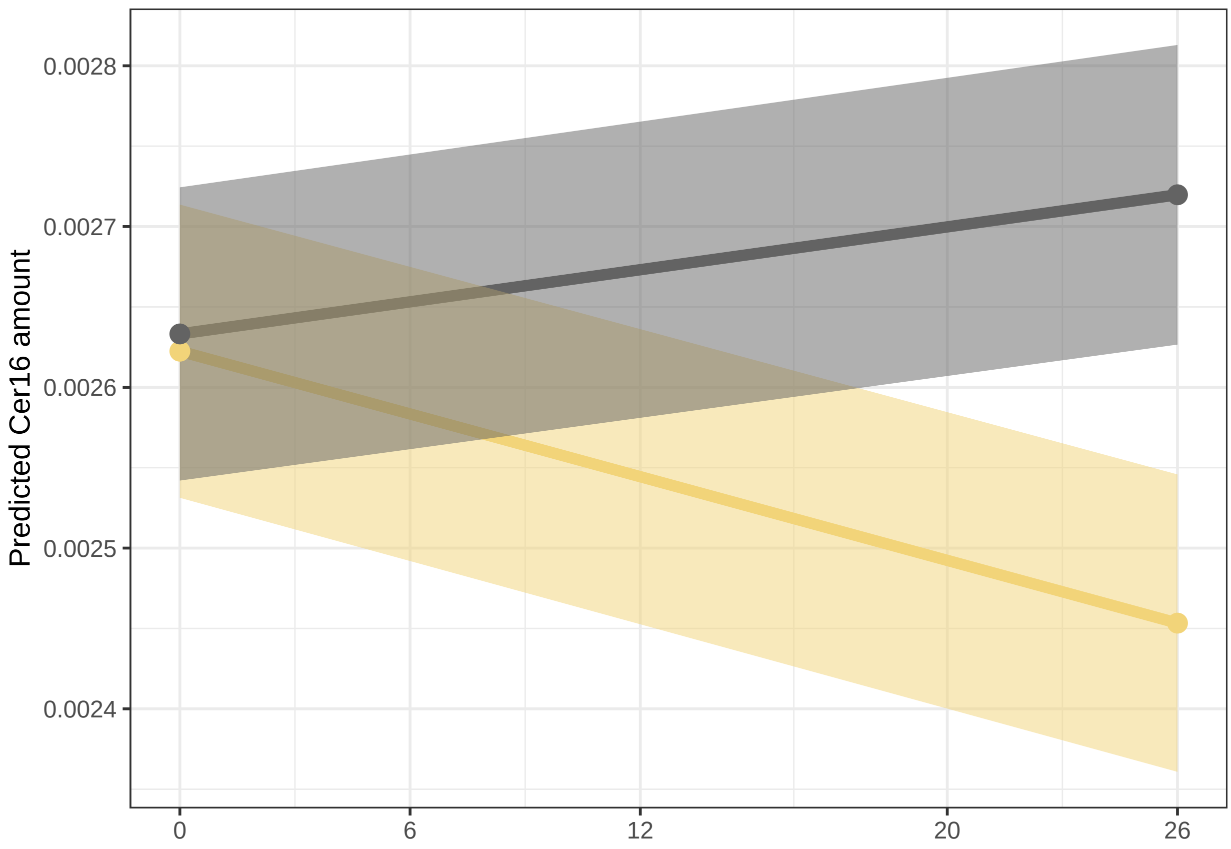

Predicted Cer20 (ug/mL)
0.72
0.68
0.64
0.60
0.56
0
6
12
20
26

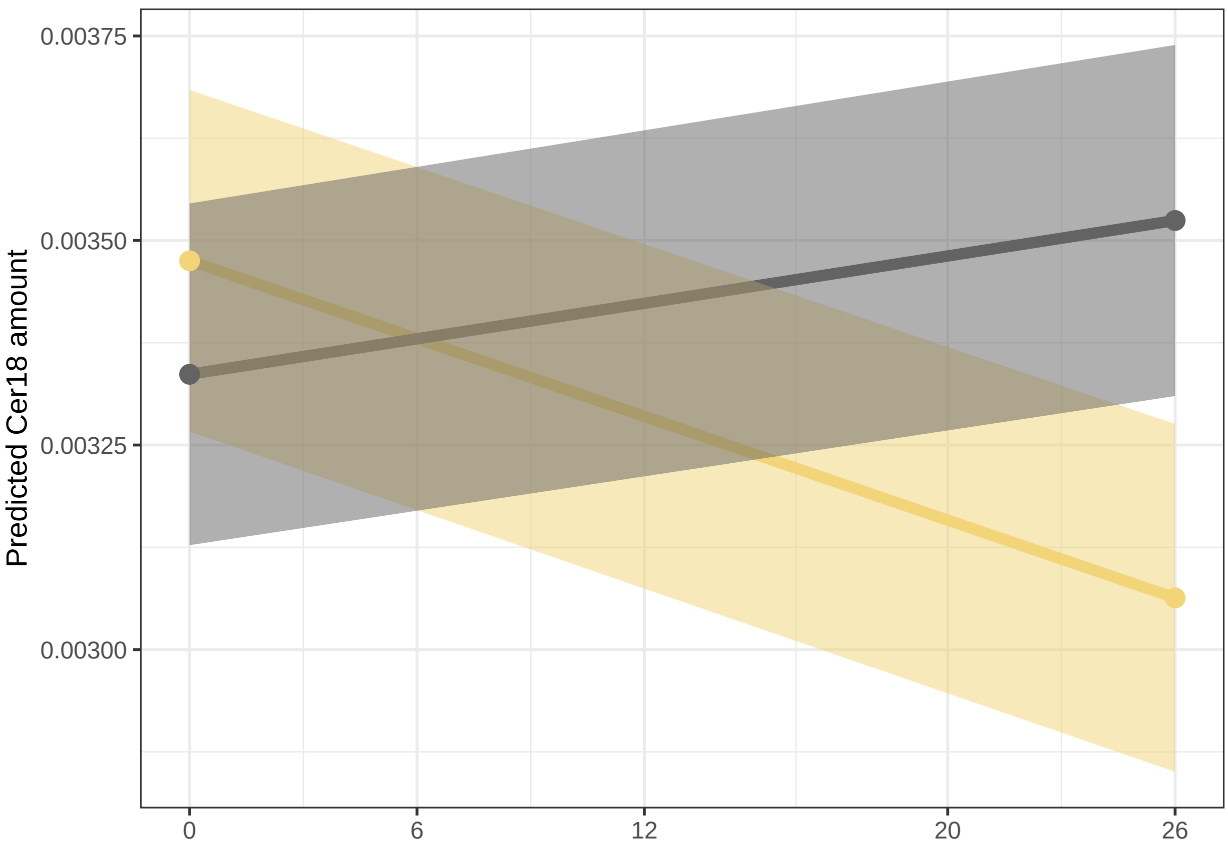

Predicted Cer22 (ug/mL)
3.1
2.9
2.7
2.5
2.3
0
6
12
20
26

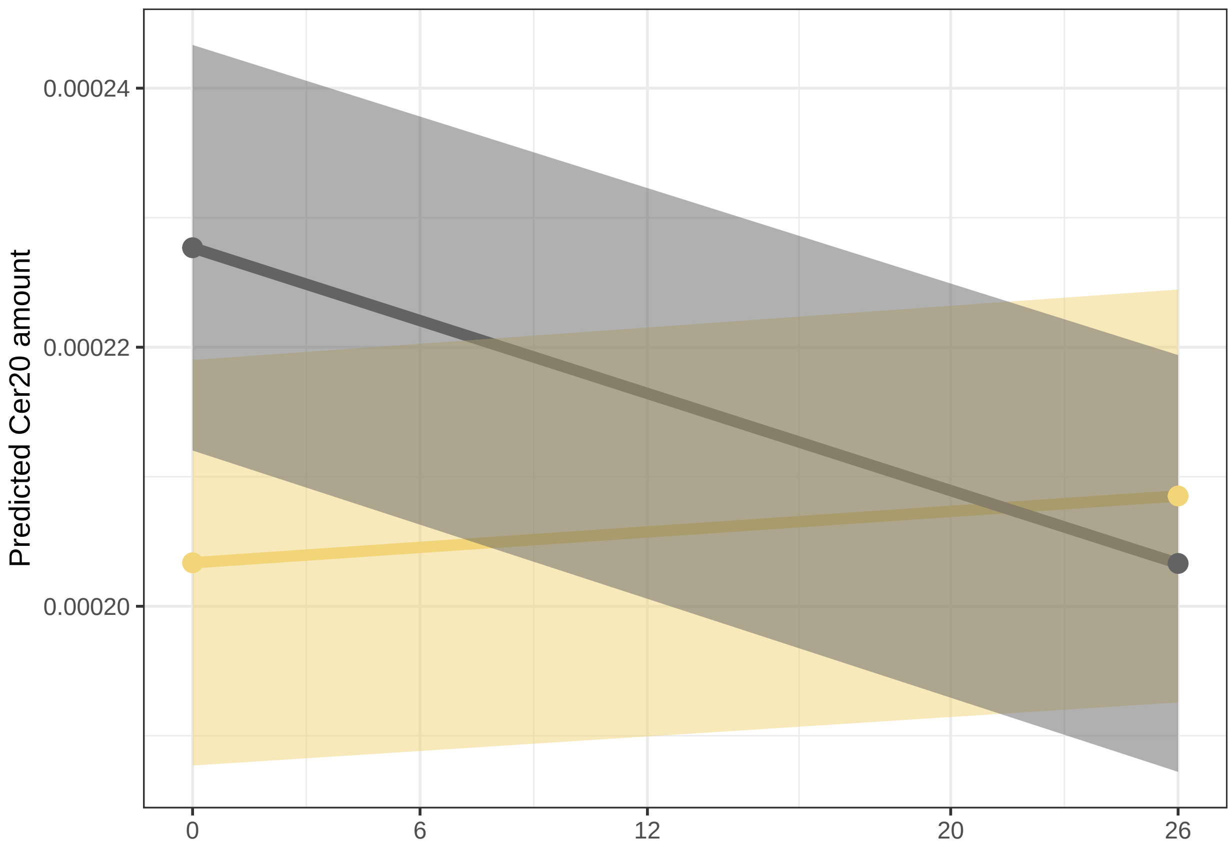

Predicted Cer24_0 (ug/mL)
10
9
8
7
0
6
12
20
26

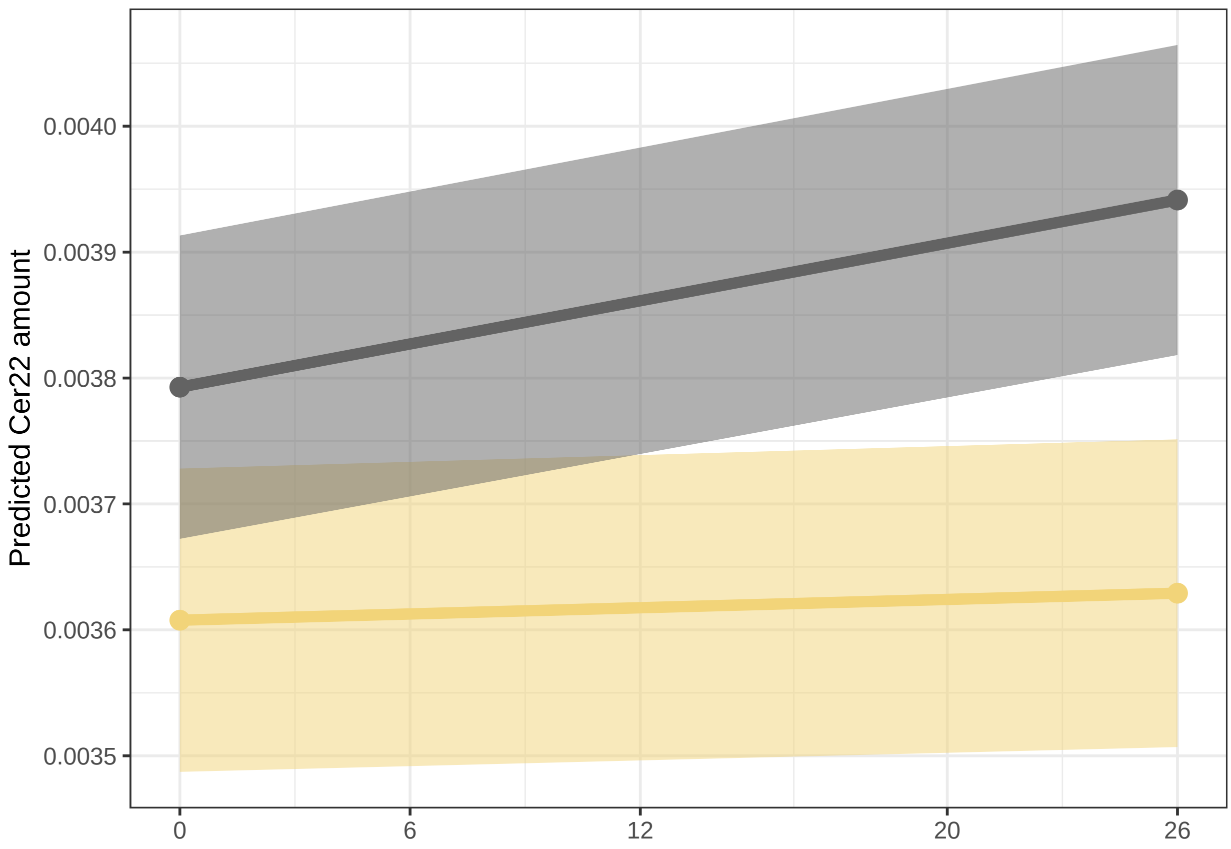


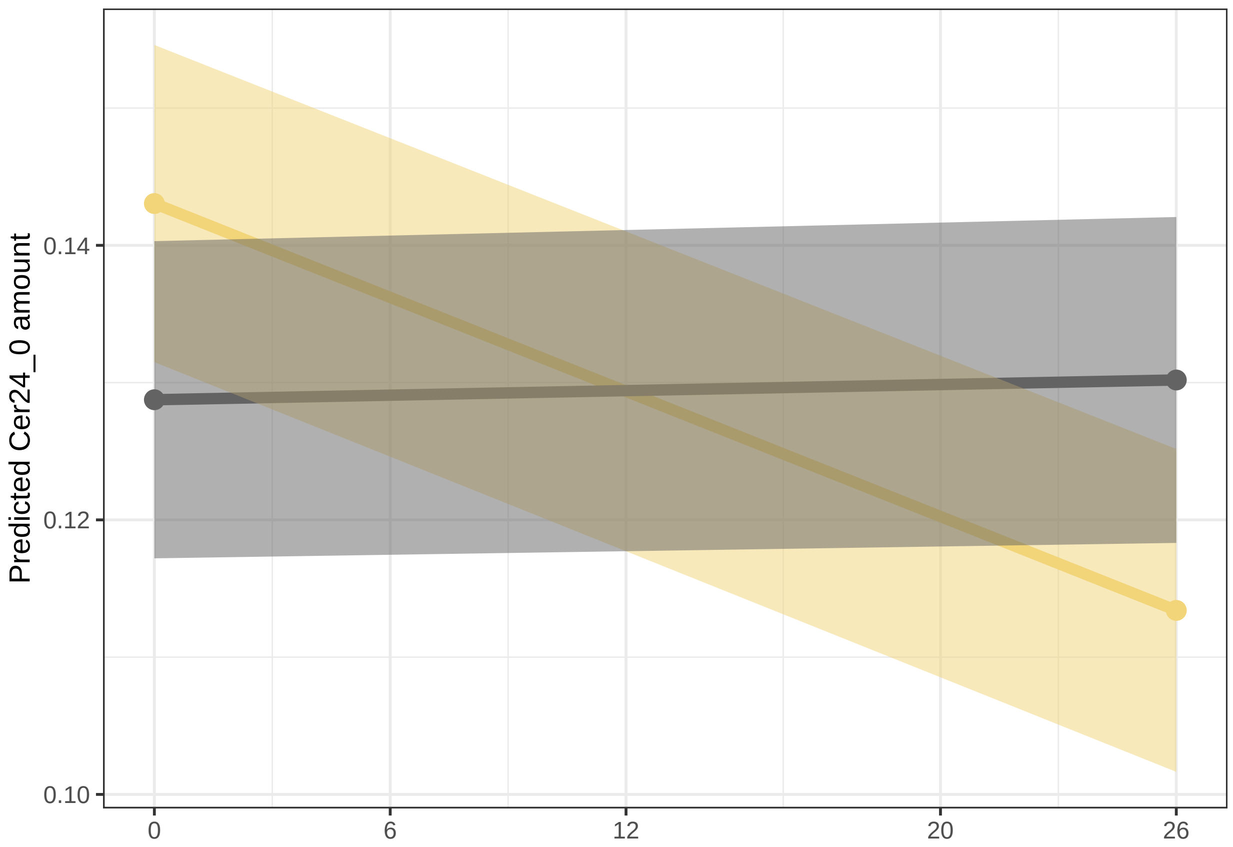
```
#change names back to contain ":"
colnames(data_albu) <- gsub("Cer24_", "Cer24:", colnames(data_albu))

##LiraFlame

#Dirty fix on for error in names with ":"
colnames(data_flame) <- gsub("Cer24:", "Cer24_", colnames(data_flame))

#Empty list to populate in loop
LMM_flame_list <- list()

#Loop through each ceramide and save LMM model, prediction and plot into LMM_flame_list
for(i in 1:length(tmp_cer)){

    #LMM model
    model <- data_flame %>%
        mutate(Treatment = recode(Treatment, `1` = "Liraglutide", `2` = "Placebo")) %>%
        rename(Visit = Time_point) %>%
        #mutate(Visit = recode(Visit, `1` = 0, `2` = 26)) %>%
        lmer(formula =  paste0("`", tmp_cer[i], "`",
                               " ~ Treatment+Visit + Treatment:Visit + (1|Sample_ID)"))

    #LMM prediction
    prediction <- ggpredict(model, terms = c("Visit", "Treatment"))
    prediction$x <- c(0, 0, 26, 26)
    #LMM plot
    plot <- ggplot(prediction)+
        geom_line(aes(x = x, y = predicted, color = group), lwd = 2, alpha = 1) +
        geom_ribbon(aes(x = x, ymin = predicted - std.error,
                               ymax = predicted + std.error, fill = group),
                    alpha = 0.5) +
        geom_point(aes(x = x, y = predicted, color = group), size = 3)+

        scale_colour_manual(values = c("#F2D479", "#636363"))+
        scale_fill_manual(values = c("#F2D479", "#636363"))+
        scale_x_continuous(breaks = c(0, 6, 12, 20, 26), limits = c(0, 26))+
        #ylim(-2.65, -2.55)+
        ylab(paste0("Predicted ", tmp_cer[i], " amount"))+
        xlab("Weeks") +
        #ggtitle("Trial 1") +
        theme_bw()+
        theme(legend.position= "none",#c(0.15,0.1),
              legend.background = element_rect(fill = NA),
              axis.title.x = element_blank(),
              #axis.title.y = element_blank(),
              rect = element_rect(fill = "transparent"),
              panel.background = element_rect(fill = "transparent"))

    #Populate LMM_flame_list with more lists
    LMM_flame_list[tmp_cer[i]] <- list(c(Model = list(model),
                                   Prediction = list(prediction),
                                   Plot = list(print(plot))))

    rm(model, prediction, plot)
}
```

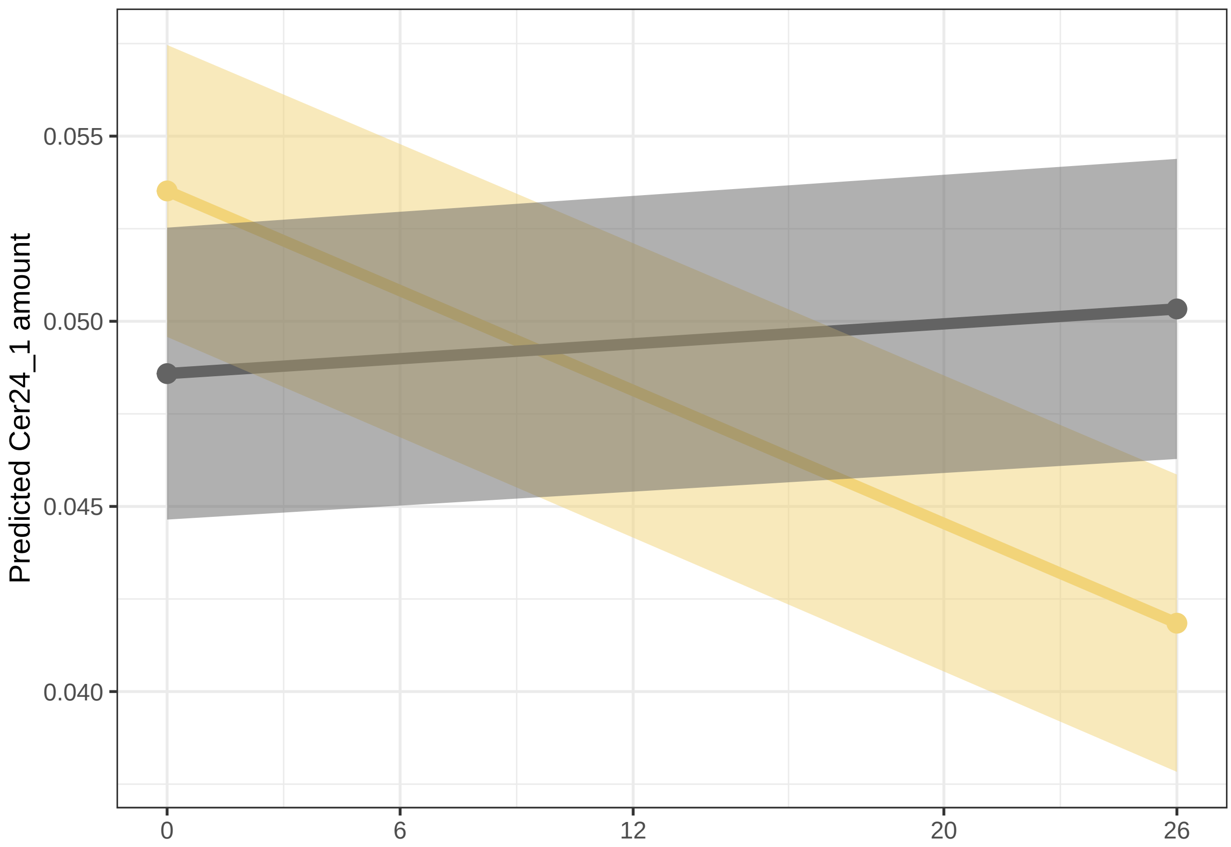

Predicted Cer16 amount
0.0028
0.0027
0.0026
0.0025
0.0024
0
6
12
20
26


Predicted Cer18 amount
0.00375
0.00350
0.00325
0.00300
0
6
12
20
26

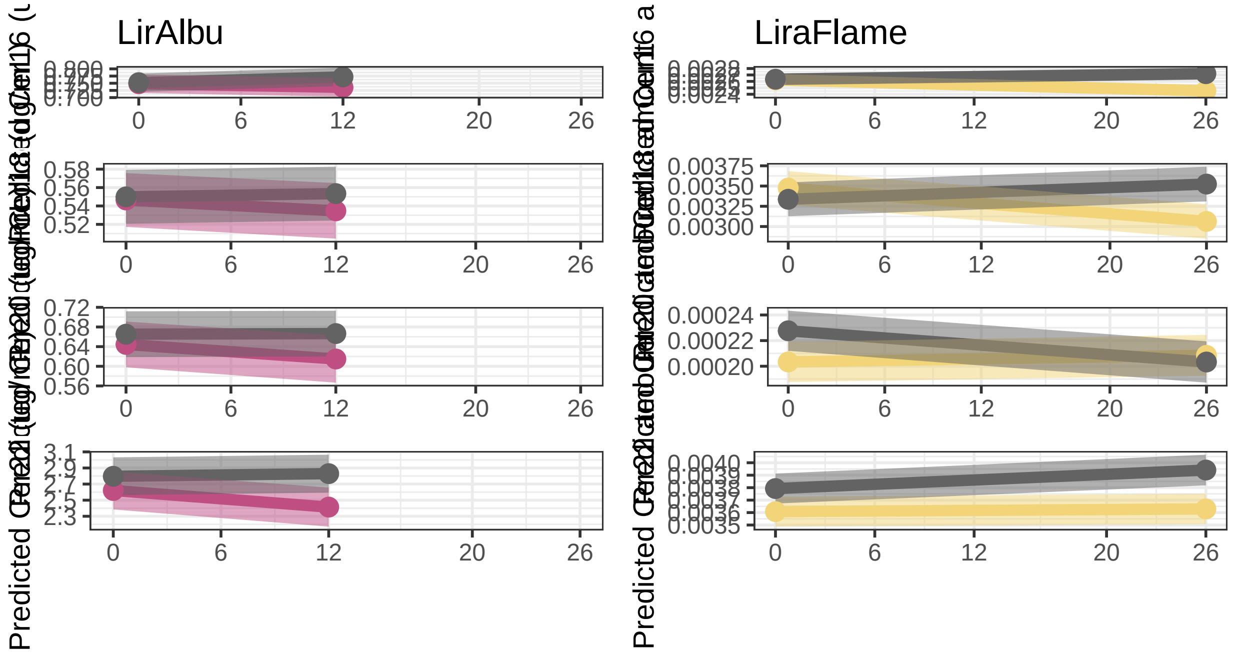

Predicted Cer20 amount
0.00024
0.00022
0.00020
0
6
12
20
26


Predicted Cer22 amount
0.0040
0.0039
0.0038
0.0037
0.0036
0.0035
0
6
12
20
26


Predicted Cer24_0 amount
0.10
0.12
0.14
0
6
12
20
26



```
#change names back to contain ":"
colnames(data_flame) <- gsub("Cer24_", "Cer24:", colnames(data_flame))

## Combine plots
library(ggpubr)

#Combining LMM plots from LiraFlame and LirAlbu
ggarrange(LMM_albu_list$Cer16$Plot + ggtitle("LirAlbu"),
          LMM_flame_list$Cer16$Plot +ggtitle("LiraFlame"),
          LMM_albu_list$Cer18$Plot, LMM_flame_list$Cer18$Plot,
          LMM_albu_list$Cer20$Plot, LMM_flame_list$Cer20$Plot,
          LMM_albu_list$Cer22$Plot, LMM_flame_list$Cer22$Plot,
          LMM_albu_list$`Cer24:0`$Plot, LMM_flame_list$`Cer24:0`$Plot,
          LMM_albu_list$`Cer24:1`$Plot + theme(legend.position= c(0.8,0.2),
                                               legend.title=element_blank(),
                                               axis.title.x = element_text()),
          LMM_flame_list$`Cer24:1`$Plot + theme(legend.position= c(0.15,0.2),
                                                legend.title=element_blank(),
                                                axis.title.x = element_text()),
          ncol = 2, nrow = 6)
```

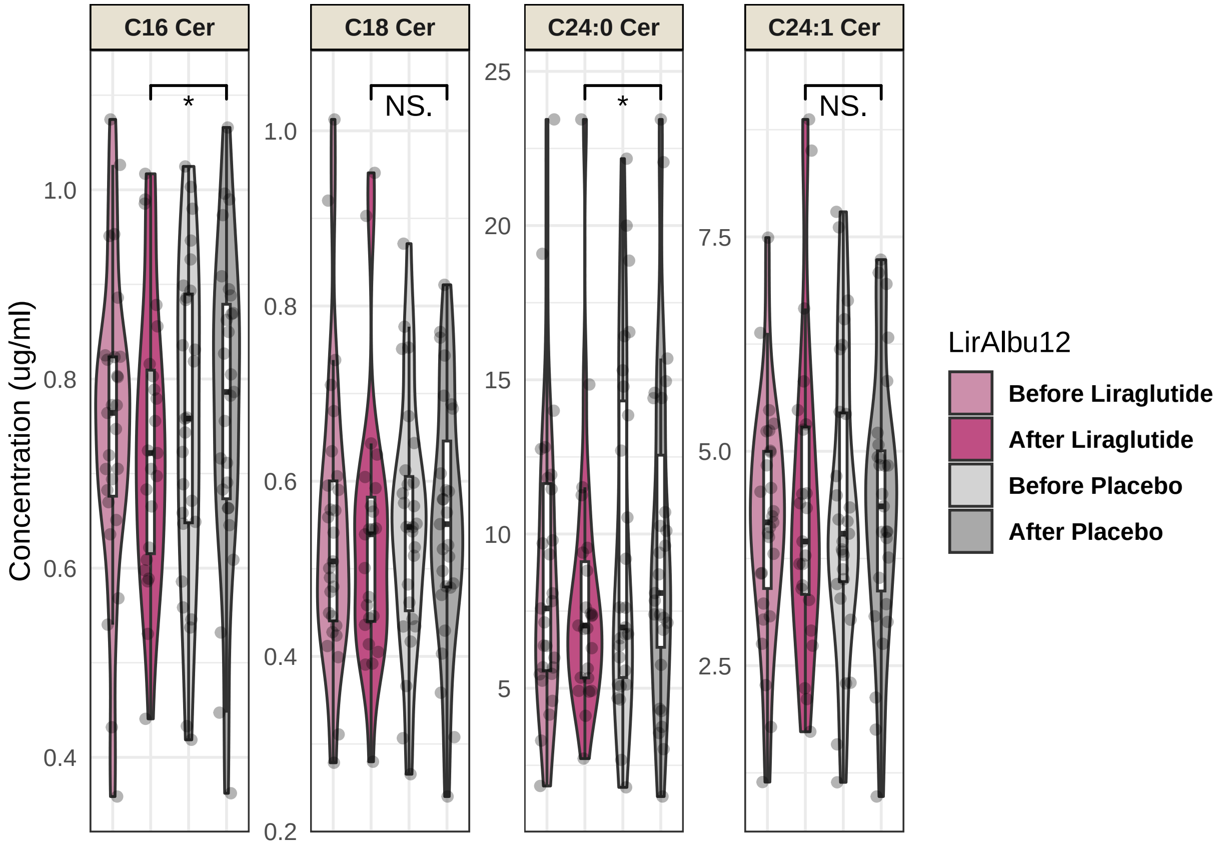


```
#export 12x12

#rm(LMM_albu_list, LMM_flame_list)
rm(tmp_cer, i)
```

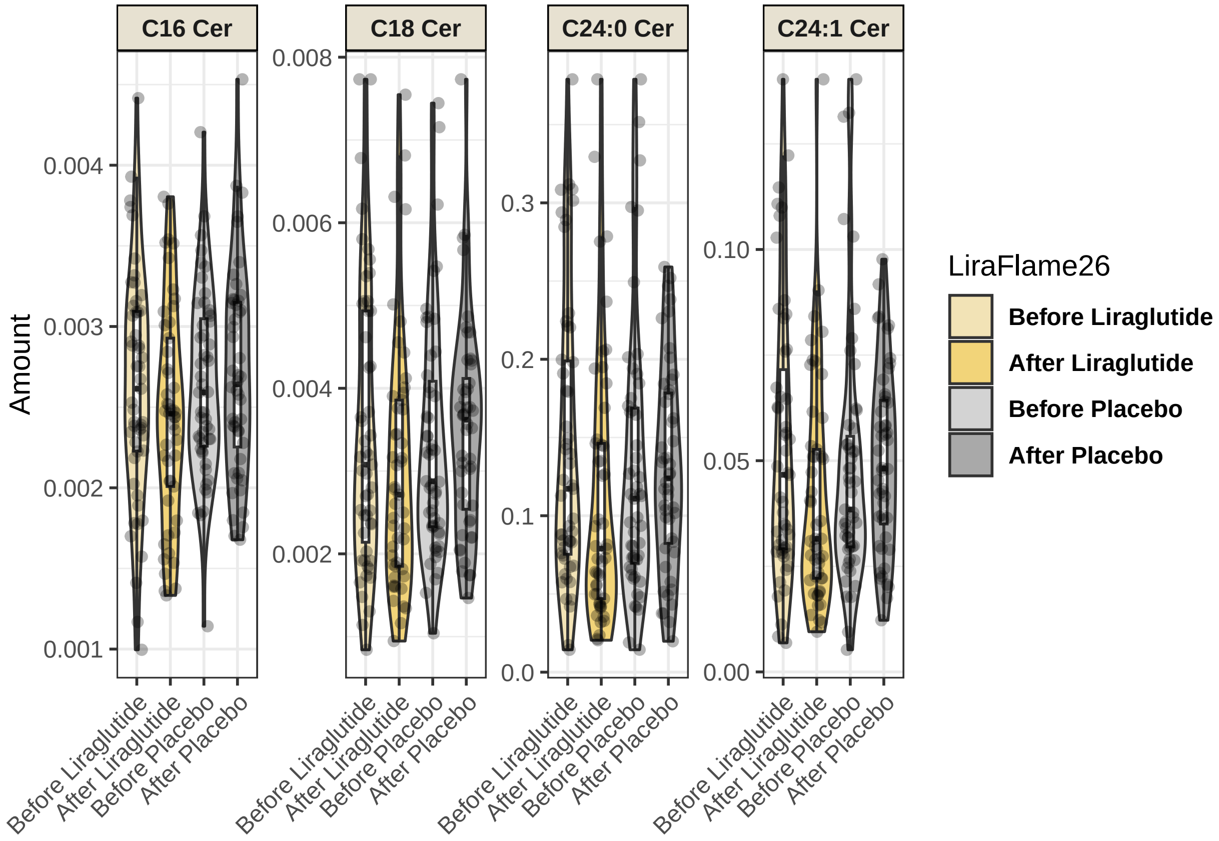
```
#changes names from CerX to CX Cer
CerX_to_CX_Cer <- function(Cer_name){
  if (is.vector(Cer_name)) {
    Cer_name[grepl("Cer", Cer_name)] <- gsub("Cer", "C", Cer_name[grepl("Cer", Cer_name)])
    Cer_name[grepl("Cer", Cer_name)] <- gsub(":0", "", Cer_name[grepl("Cer", Cer_name)])
    Cer_name[grepl("C\\d{2}", Cer_name)] <- paste0(Cer_name[grepl("C\\d{2}", Cer_name)], " Cer")
  } else {
    if (grepl("Cer", Cer_name)) {
      Cer_name <- gsub("Cer", "C", Cer_name)
      Cer_name <- gsub(":0", "", Cer_name)
      Cer_name <- paste0(Cer_name, " Cer")
    } else {
      Cer_name <- Cer_name
    }
  }
  return(Cer_name)
}

# #Example
# CerX_to_CX_Cer("Cer24:0")
# CerX_to_CX_Cer("test")
```

```r
# CerX_to_CX_Cer(c("Cer24:0", "test"))
# CerX_to_CX_Cer(c("Cer24:0", "test", "Computer"))
```

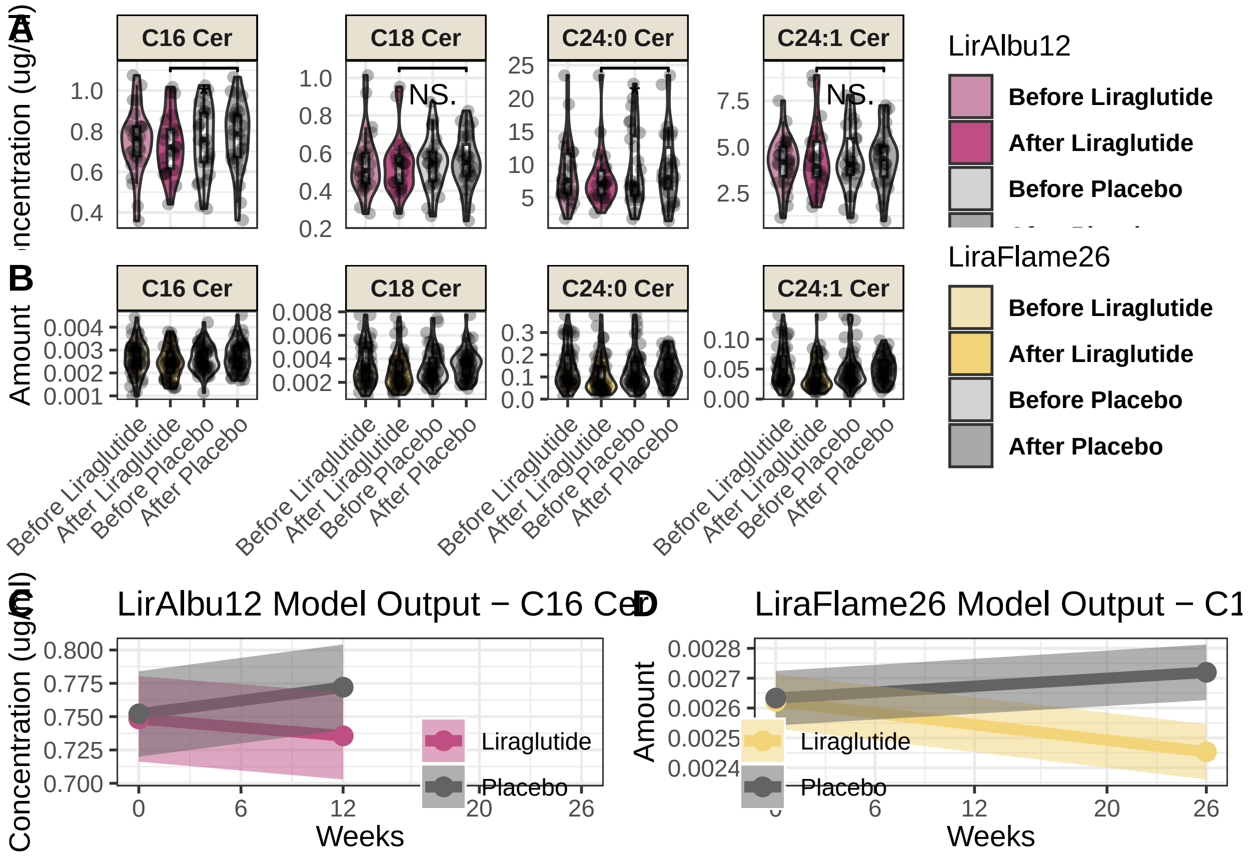
```r
library(ggpubr)

#Boxplot LirAlbu
plot_A <- data_albu %>%
    select(PrePost, Sample_ID, starts_with("Cer")) %>%
    select(-c(Cer20, Cer22)) %>%
    pivot_wider(names_from = c(PrePost, Sample_ID), values_from = starts_with("Cer")) %>%
    pivot_longer(starts_with("Cer")) %>%
    separate(name, sep = "_", c("Ceramide", "PrePost", "Sample_ID")) %>%
    pivot_wider(names_from = c(Ceramide, PrePost), values_from = value) %>%
    pivot_longer(starts_with("Cer")) %>%
    separate(name, sep = "_", c("Ceramide", "PrePost")) %>%
    mutate(PrePost = factor(PrePost,
                             levels = c("LiraPre", "LiraPost", "PlaceboPre", "PlaceboPost"),
                             labels = c("Before Liraglutide",
                                        "After Liraglutide",
                                        "Before Placebo",
                                        "After Placebo"))) %>%
    mutate(Ceramide = CerX_to_CX_Cer(Ceramide)) %>%
    ggplot(aes(x = PrePost, y = value)) +
    geom_violin(aes(fill = PrePost))+
    scale_fill_manual(values = c("#cc8fab", "#BF4E83", "lightgrey", "darkgrey"))+
    guides(fill = guide_legend(title = "LirAlbu12"))+
    geom_boxplot(outlier.shape = NA,
                 width = 0.2)+
    geom_jitter(alpha = 0.3,
                position = position_jitter(width = 0.2))+
    # stat_compare_means(comparisons = list(c("After Liraglutide", "After Placebo")),
    #                    method = "t.test", paired = TRUE,
    #                    label = "p.signif", hide.ns = TRUE)+
    geom_signif(comparisons = list(c("After Liraglutide", "After Placebo")),
                test = "t.test", test.args = list(paired = TRUE),
                tip_length = 0.02, map_signif_level = TRUE, vjust = 1.8
                )+
    ylab("Concentration (ug/ml)")+
    theme_bw()+
    facet_wrap(~ Ceramide, scales = "free_y", nrow = 1)+
    theme(
         #axis.title.y = element_blank(),
          axis.title.x = element_blank(),
          #legend.title = element_blank(),
          legend.position = "right",
          legend.text = element_text(face = "bold"),
          axis.text.x = element_blank(),
          axis.ticks = element_blank(),
          strip.text = element_text(face = "bold"),
          strip.background = element_rect(colour="black",
                                          fill="#E7E1D1"))

plot_A
```



```
plot_B <- data_flame %>%
    mutate(PrePost = factor(paste0(Treatment, "_", Time_point),
                            levels = c("1_1", "1_2", "2_1", "2_2"),
                            label = c("Before Liraglutide",
                                      "After Liraglutide",
                                      "Before Placebo",
                                      "After Placebo"))) %>%
    select(Sample_ID, PrePost, starts_with("Cer")) %>%
    select(-c(Cer20, Cer22)) %>%
    pivot_longer(cols = starts_with("Cer"), names_to = "Ceramide", values_to = "Cer_val") %>%
    mutate(Ceramide = CerX_to_CX_Cer(Ceramide)) %>%
    ggplot(aes(x = PrePost, y = Cer_val)) +
    geom_violin(aes(fill = PrePost))+
    scale_fill_manual(values = c("#f2e3b6", "#F2D479", "lightgrey", "darkgrey"))+
    guides(fill = guide_legend(title = "LiraFlame26"))+
    geom_boxplot(outlier.shape = NA,
                 width = 0.2)+
    geom_jitter(alpha = 0.3,
                position = position_jitter(width = 0.2))+
    # stat_compare_means(comparisons = list(c("After Liraglutide", "After Placebo")),
    #                    method = "t.test", label = "p.signif", hide.ns = TRUE)+
    ylab("Amount")+
    theme_bw()+
    facet_wrap(~ Ceramide, scales = "free_y", nrow = 1)+
    theme(
         #axis.title.y = element_blank(),
          axis.title.x = element_blank(),
          #legend.title = element_blank(),
          legend.position = "right",
          axis.text.x = element_text(angle = 45, vjust = 1, hjust= 1),
          legend.text = element_text(face = "bold"),
          strip.text = element_text(face = "bold"),
          strip.background = element_rect(colour="black",
                                          fill="#E7E1D1"))

plot_B
```



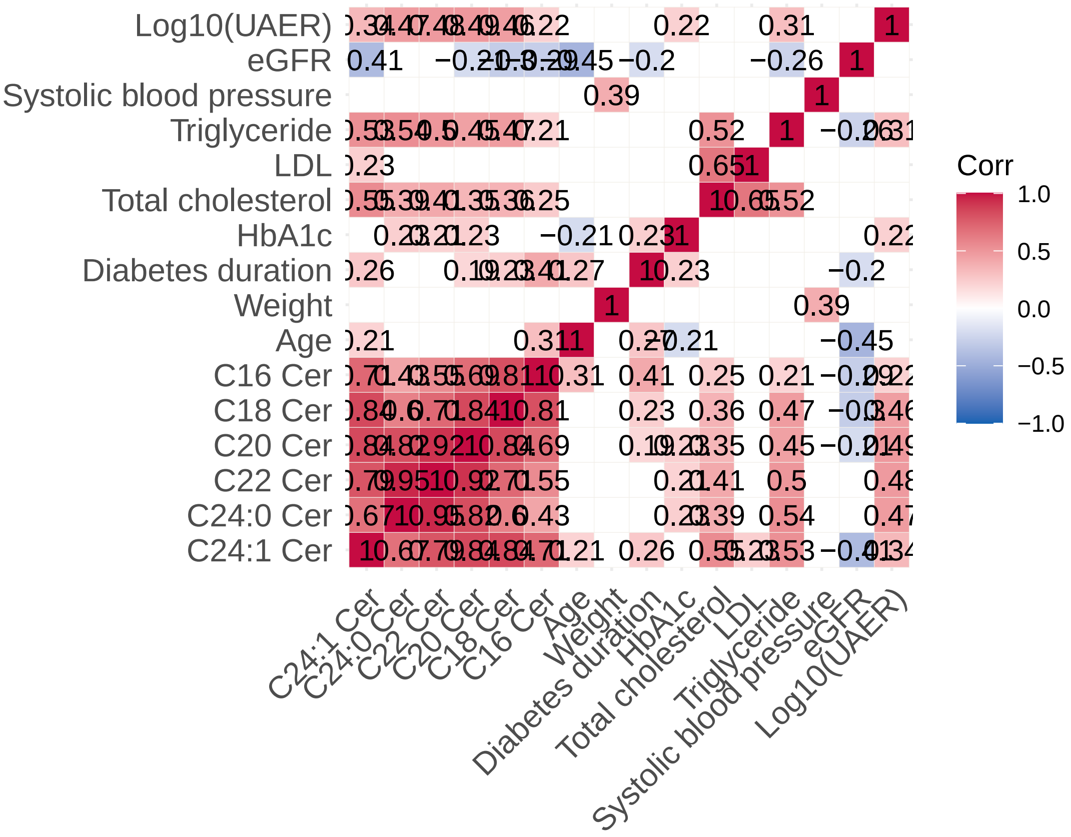
```
#Combined plot A + B
plot_AB <- ggarrange(plot_A, plot_B,
                     labels = c("A", "B"),
                     heights = c(1, 1.3),
              ncol = 1, align = "v")

#Combined plot LMM 1 + 2
plot_LMM12 <- ggarrange(
    LMM_albu_list$Cer16$Plot +
        ggtitle("LirAlbu12 Model Output - C16 Cer") +
        ylab("Concentration (ug/ml)")+
        theme(legend.position = c(0.8, 0.2),
              legend.title = element_blank(),
              axis.title.x = element_text()),
    LMM_flame_list$Cer16$Plot +
        ggtitle("LiraFlame26 Model Output - C16 Cer") +
        ylab("Amount")+
        theme(legend.position = c(0.15, 0.2),
              legend.title = element_blank(),
              axis.title.x = element_text()),
    labels = c("C", "D"), ncol = 2)

#Combine plot_AB and plot_LMM12
ggarrange(plot_AB, plot_LMM12, ncol = 1, heights = c(2,1), align = "v")
```



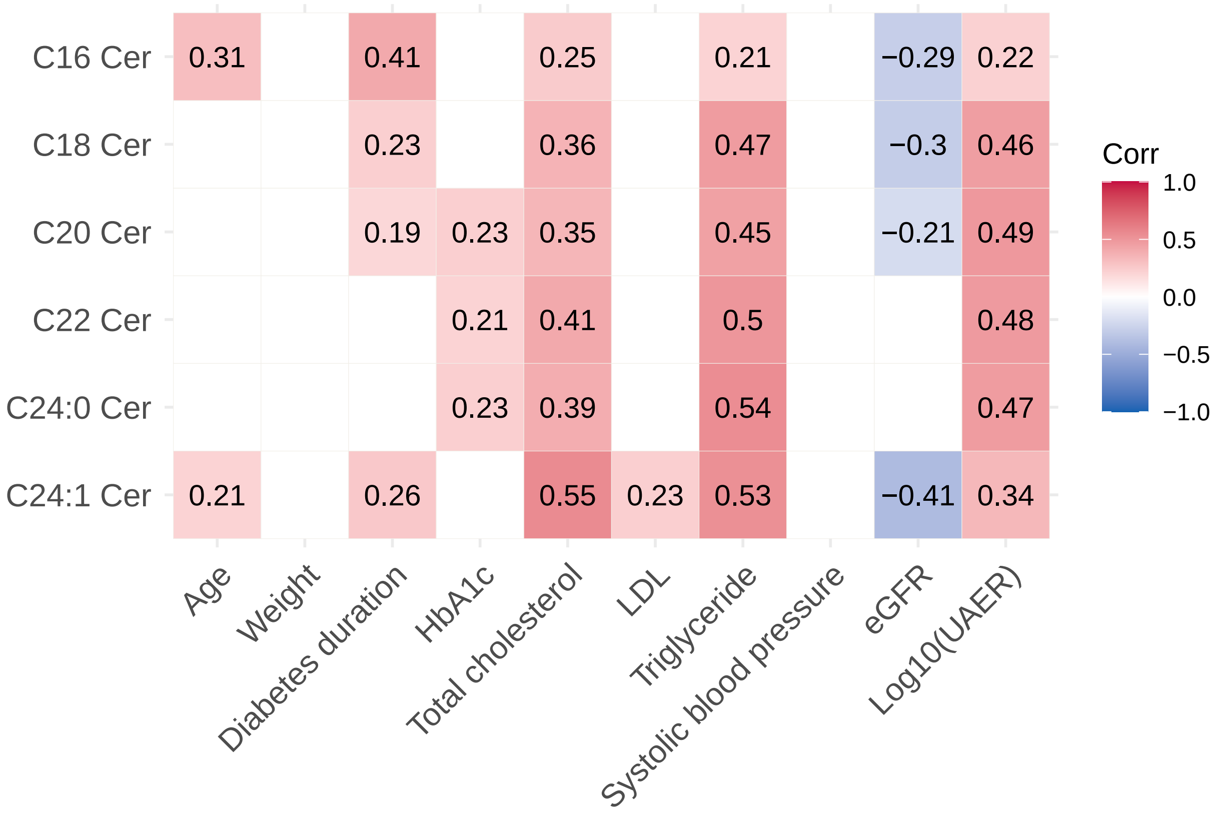
```
#export 10x12

rm(plot_A, plot_B, plot_AB, LMM_albu_list, LMM_flame_list, plot_LMM12)
```

```
library(ggcorrplot)
library(Hmisc)
library(ggpubr)

##LirAlbu
#Variables of interest
tmp_voi <- c("Age", "Sex", "Weight", "Diabetes_Duration", "HbA1c",
             "Total_Cholesterol", "LDL", "Triglyceride", "OfficeSBP",
             "eGFR_CKDEPI", "Log10MeanUAER")
```

```r
#Prepare data for heatmap, select variables of interest, remove non-numeric, impute missing values
tmp_cor <- data_albu %>%
    select(rev(starts_with("Cer")), all_of(tmp_voi)) %>%
    select_if(., is.numeric) %>%
    mutate(across(everything(), ~ if_else(is.na(.x), median(., na.rm = TRUE), .)))

#Fix names
tmp_names <- c("Age", "Weight", "Diabetes duration", "HbA1c", "Total cholesterol",
               "LDL", "Triglyceride", "Systolic blood pressure", "eGFR", "Log10(UAER)")

colnames(tmp_cor)[!grepl("Cer", colnames(tmp_cor))] <- tmp_names
colnames(tmp_cor) <- CerX_to_CX_Cer(colnames(tmp_cor))

#Calculate correlation matrix
tmp_cor <- Hmisc::rcorr(as.matrix(tmp_cor), type = "pearson")

#set NA p-values to 0
tmp_cor$P[is.na(tmp_cor$P)] <- 0

#Full
ggcorrplot::ggcorrplot(tmp_cor$r,
                       hc.order = FALSE,
                       outline.col = "#F2EFE9",
                       p.mat = tmp_cor$P,
                       sig.level = 0.05,
                       insig = "blank",
                       lab = TRUE,
                       ggtheme = ggplot2::theme_minimal,
                       colors = c("#1262B2", "white", "#C50B42"))
```

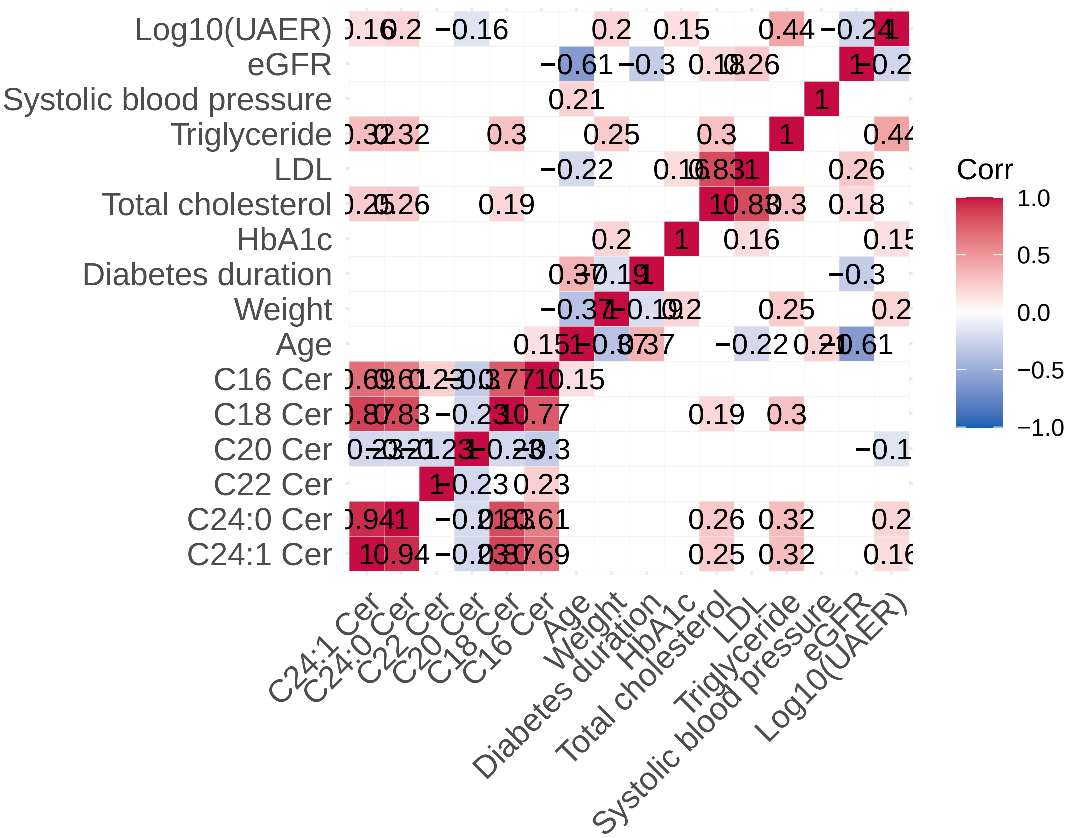


```
#Subset
heatmap_albu <- ggcorrplot::ggcorrplot(tmp_cor$r[!grepl("Cer", colnames(tmp_cor$r)),
                                 grepl("Cer", colnames(tmp_cor$r))],
                       outline.col = "#F2EFE9",
                       p.mat = tmp_cor$P[!grepl("Cer", colnames(tmp_cor$P)),
                                         grepl("Cer", colnames(tmp_cor$P))],
                       sig.level = 0.05,
                       insig = "blank",
                       lab = TRUE,
                       ggtheme = ggplot2::theme_minimal,
                       colors = c("#1262B2", "white", "#C50B42"))

heatmap_albu
```



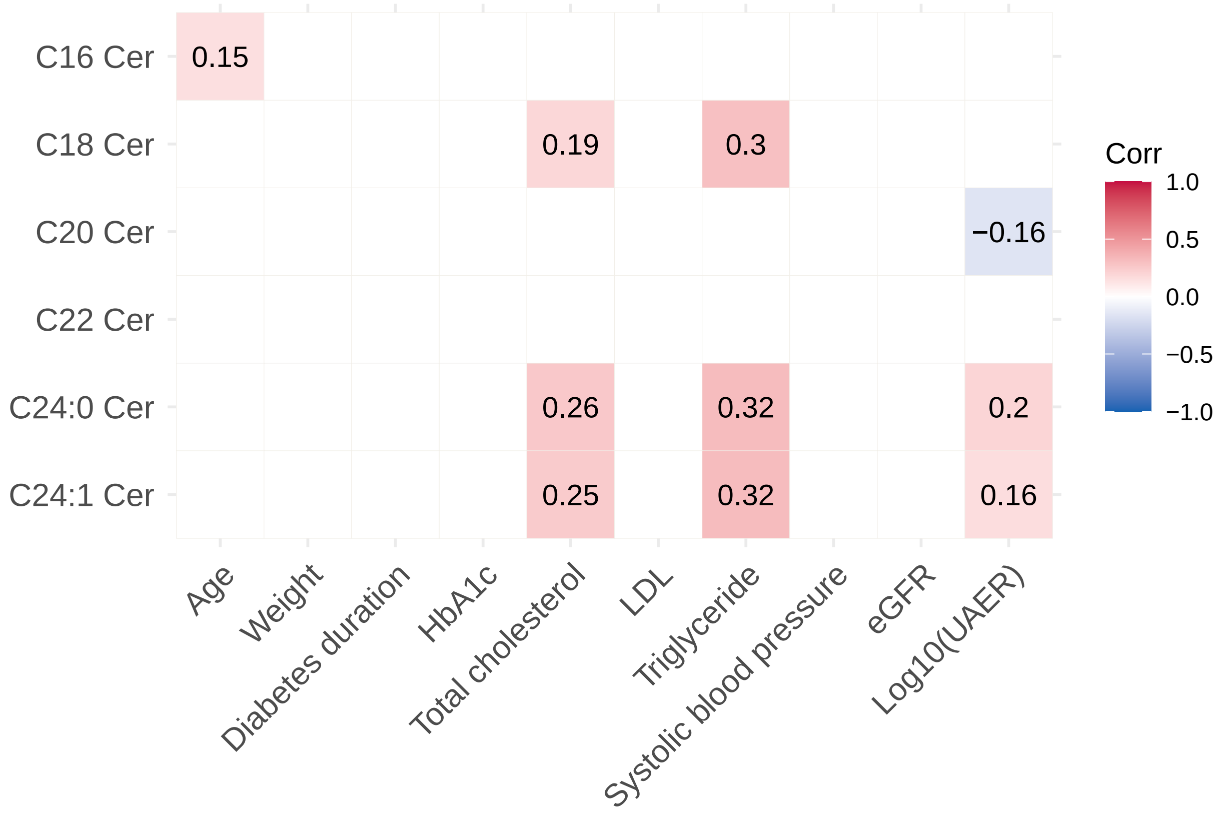
```
#export 5x6

rm(tmp_cor, tmp_voi, tmp_names)


##LiraFlame
#Variables of interest
tmp_voi <- c("age", "Sex", "V2_B_VAEGT", "dm_var", "V2_B_HBA1C",
             "CHOL_v2", "LDL_v2", "TRIG_v2", "Sys_gen_v2_beregnet",
             "GFRepi_V2", "V2_U_LogMeanUAER")

#Prepare data for heatmap, select variables of interest, remove non-numeric, impute missing values
tmp_cor <- data_flame %>%
    select(rev(starts_with("Cer")), all_of(tmp_voi)) %>%
    select_if(., is.numeric) %>%
    mutate(across(everything(), ~ if_else(is.na(.x), median(., na.rm = TRUE), .)))

#Fix names
tmp_names <- c("Age", "Weight", "Diabetes duration", "HbA1c", "Total cholesterol",
               "LDL", "Triglyceride", "Systolic blood pressure", "eGFR", "Log10(UAER)")

colnames(tmp_cor)[!grepl("Cer", colnames(tmp_cor))] <- tmp_names
colnames(tmp_cor) <- CerX_to_CX_Cer(colnames(tmp_cor))

#Calculate correlation matrix
tmp_cor <- Hmisc::rcorr(as.matrix(tmp_cor), type = "pearson")
```

```r
#set NA p-values to 0
tmp_cor$P[is.na(tmp_cor$P)] <- 0

#Full
ggcorrplot::ggcorrplot(tmp_cor$r,
                       hc.order = FALSE,
                       outline.col = "#F2EFE9",
                       p.mat = tmp_cor$P,
                       sig.level = 0.05,
                       insig = "blank",
                       lab = TRUE,
                       ggtheme = ggplot2::theme_minimal,
                       colors = c("#1262B2", "white", "#C50B42"))
```

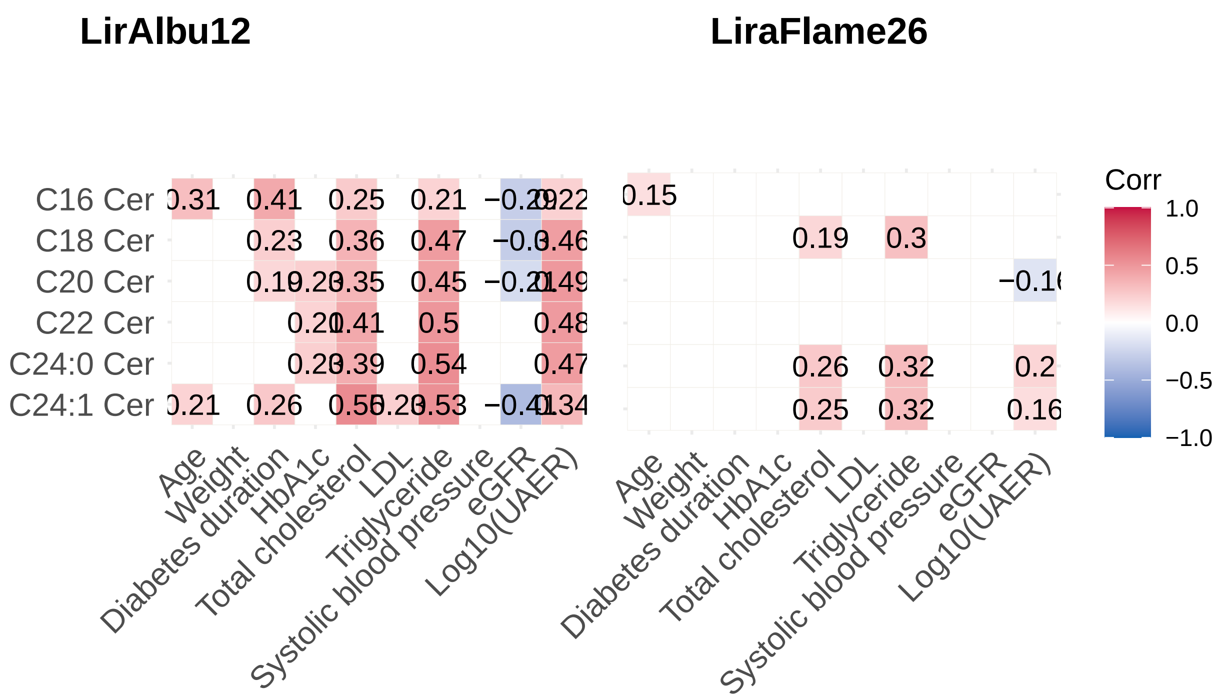


```r
#Subset
heatmap_flame <- ggcorrplot::ggcorrplot(tmp_cor$r[!grepl("Cer", colnames(tmp_cor$r)),
                                    grepl("Cer", colnames(tmp_cor$r))],
                       outline.col = "#F2EFE9",
                       p.mat = tmp_cor$P[!grepl("Cer", colnames(tmp_cor$P)),
                                         grepl("Cer", colnames(tmp_cor$P))],
                       sig.level = 0.05,
                       insig = "blank",
                       lab = TRUE,
                       ggtheme = ggplot2::theme_minimal,
                       colors = c("#1262B2", "white", "#C50B42"))
```

```
heatmap_flame
```



```
#export 5x6

#combine
ggarrange(heatmap_albu+
              theme(legend.position = "none"),
          heatmap_flame+
              theme(axis.text.y = element_blank()),
          labels = c("LirAlbu12", "LiraFlame26"),
          widths = c(1, 1.055), align = "h")
```



```
#export 4.5x10

rm(tmp_cor, tmp_voi, tmp_names, heatmap_albu, heatmap_flame)
```

```
#Calculate variable change
data_albu_change <- data_albu %>%
    select(PrePost, Sample_ID, starts_with("Cer"), `Log10MeanUAER`, Weight, HbA1c) %>%
    pivot_wider(names_from = c(PrePost, Sample_ID),
                values_from = c(starts_with("Cer"), Log10MeanUAER, Weight, HbA1c)) %>%
    pivot_longer(c(starts_with("Cer"),
                   starts_with("Log10MeanUAER"),
                   starts_with("Weight"),
                   starts_with("HbA1c"))) %>%
    separate(name, sep = "_", c("Variable", "PrePost", "Sample_ID")) %>%
    pivot_wider(names_from = c(Variable, PrePost), values_from = value) %>%
    mutate(across(ends_with("Post"),
                  ~ . - get(glue::glue(str_replace(cur_column(), "Post", "Pre")))
                  )) %>%
    select(-ends_with("Pre")) %>%
    rename_with(~str_replace(., "Post", ""), everything()) %>%
    pivot_longer(c(starts_with("Cer"),
                   starts_with("Log10MeanUAER"),
                   starts_with("Weight"),
                   starts_with("HbA1c"))) %>%
    separate(name, sep = "_", c("Variable", "Treatment")) %>%
    pivot_wider(names_from = c(Variable), values_from = value) %>%
    rename_with(~str_replace(., ":", ""), everything())


#Loop through lipids of interest, and potential mediators
#Lipids of interest
Lipid_list <- data_albu_change %>%
    select(starts_with("Cer")) %>%
    colnames()

#Mediators of interest
Mediator_list <- data_albu_change %>%
    select(Weight, HbA1c, Log10MeanUAER) %>%
    colnames()

#Empty table to populate
Mediator_table <- data.frame()

#Nested loop going through all the lipid and mediators
#Each combination is tested in 3 linear regression models:
#Model 1: Lipid ~ Treatment
#Model 2: Mediator ~ Treatment
#Model 3: Lipid ~ Treatment + Mediator
#Model 4: Causal Mediation Analysis of Model 2 and 3 (bootstrap 1000)
#Note: this step quite slow, get a cup of coffee
if(TRUE){ #Set this to TRUE in order to run
for(l in Mediator_list){
    for(i in Lipid_list){

        #Step 1: Total effect - Treatment on lipid: i
        form1 <- paste(i, "~ Treatment")
        model1 <- lm(form1, data_albu_change)
        model1_sum <- summary(model1)

        #Step 2: Effect on mediator - Treatment on Mediator l
        form2 <- paste(l, "~ Treatment")
        model2 <- lm(form2, data_albu_change)
        model2_sum <- summary(model2)

        #Step 3: Effect of mediator on response - Medi_var on Resp_Var
        form3 <- paste(i, "~ Treatment +", l)
        model3 <- lm(form3, data_albu_change)
        model3_sum <- summary(model3)

        #Step 4: Casual mediation analysis
        model4 <- mediation::mediate(model.m = model2, model.y = model3,
                                     treat = "Treatment",
                                     mediator = l,
                                     boot = TRUE)

        #Extract explanatory variable(s)
        explanetory_var <- rownames(model1_sum$coefficients)[nrow(model1_sum$coefficients)]

        #Populate
        #Model 1
        Mediator_table[paste0("Model1_Estimate"), i] <-
            model1_sum$coefficients[explanetory_var, "Estimate"]
        Mediator_table[paste0("Model1_StdError"), i] <-
            model1_sum$coefficients[explanetory_var, "Std. Error"]
        Mediator_table[paste0("Model1_Pval"), i] <-
            model1_sum$coefficients[explanetory_var, "Pr(>|t|)"]
        Mediator_table[paste0("Model1_R2"), i] <- model1_sum$adj.r.squared
        #Model2
        Mediator_table[paste0("Model2_", l, "_Estimate"), i] <-
            model2_sum$coefficients[explanetory_var, "Estimate"]
        Mediator_table[paste0("Model2_", l, "_StdError"), i] <-
            model2_sum$coefficients[explanetory_var, "Std. Error"]
        Mediator_table[paste0("Model2_", l, "_Pval"), i] <-
            model2_sum$coefficients[explanetory_var, "Pr(>|t|)"]
        Mediator_table[paste0("Model2_", l, "_R2"), i] <- model2_sum$adj.r.squared
        #Model3
        Mediator_table[paste0("Model3_", l, "_Estimate"), i] <-
            model3_sum$coefficients[l, "Estimate"]
        Mediator_table[paste0("Model3_", l, "_StdError"), i] <-
            model3_sum$coefficients[l, "Std. Error"]
        Mediator_table[paste0("Model3_", l, "_Pval"), i] <-
            model3_sum$coefficients[l, "Pr(>|t|)"]
        Mediator_table[paste0("Model3_", l, "_R2"), i] <- model3_sum$adj.r.squared
        #Model4
        Mediator_table[paste0("Model4_", l, "_Total_Estimate"), i] <- model4$tau.coef
        Mediator_table[paste0("Model4_", l, "_Total_CIlower"), i] <- model4$tau.ci[["2.5%"]]
        Mediator_table[paste0("Model4_", l, "_Total_CIupper"), i] <- model4$tau.ci[["97.5%"]]
        Mediator_table[paste0("Model4_", l, "_Total_Pval"), i] <- model4$tau.p

        Mediator_table[paste0("Model4_", l, "_ADE_Estimate"), i] <- model4$z.avg
        Mediator_table[paste0("Model4_", l, "_ADE_CIlower"), i] <- model4$z.avg.ci[["2.5%"]]
        Mediator_table[paste0("Model4_", l, "_ADE_CIupper"), i] <- model4$z.avg.ci[["97.5%"]]
        Mediator_table[paste0("Model4_", l, "_ADE_Pval"), i] <- model4$z.avg.p

        Mediator_table[paste0("Model4_", l, "_ACME_Estimate"), i] <- model4$d.avg
        Mediator_table[paste0("Model4_", l, "_ACME_CIlower"), i] <- model4$d.avg.ci[["2.5%"]]
        Mediator_table[paste0("Model4_", l, "_ACME_CIupper"), i] <- model4$d.avg.ci[["97.5%"]]
        Mediator_table[paste0("Model4_", l, "_ACME_Pval"), i] <- model4$d.avg.p
    }
}
}
# # #Save Mediator_table as CSV
# write.csv(Mediator_table, here("data/Mediator_table.csv"))

rm(model1, model1_sum, model2, model2_sum, model3, model3_sum, model4,
   data_albu_change, explanetory_var, form1, form2, form3, i, l,
   Lipid_list, Mediator_list)

# #Step 1: Total effect - Treatment on Cer16
# model1 <- lm(Cer16 ~ Treatment, data = data_albu_change)
# summary(model1)
#
```

```
# #Step 2: Effect on mediator - Treatment on weight
# model2 <- lm(Weight ~ Treatment, data = data_albu_change)
# summary(model2)
#
# #Step 3: Effect of mediator on dependent - Weight on Cer
# model3 <- lm(Cer16 ~ Treatment+Weight, data = data_albu_change)
# summary(model3)
#
# #Step 4: Casual mediation analysis
# model4 <- mediate(model.m = model2, model.y = model3,
#                    treat = "Treatment", mediator = "Weight", boot = TRUE)
# summary(model4)


#Formatting mediator table
Mediator_table_compact <- Mediator_table %>%
    mutate(Model = rownames(.)) %>%
    as_tibble(.) %>%
    filter(grepl("Model4", Model)) %>%
    separate(Model, sep = "_", c("Model", "Mediator", "Measure", "Output")) %>%
    select(-Model) %>%
    filter(!grepl("ADE", Measure)) %>%
    pivot_longer(contains("Cer"), names_to = "Ceramide") %>%
    pivot_wider(names_from = c(Mediator, Measure, Output), values_from = value) %>%
    mutate(across(ends_with("_Estimate"),
                  ~ paste0(round(., 3), " (",
                    round(get(glue::glue(str_replace(cur_column(), "_Estimate", "_CIlower"))), 3), ";",
                    round(get(glue::glue(str_replace(cur_column(), "_Estimate", "_CIupper"))), 3), ")"
                           ))) %>%
    select(-ends_with("_CIlower"), -ends_with("_CIupper"))

#NOTE: There might be small variations in the values due the randomness of the bootstrapping
# #Save Mediator_table_compact as CSV
# write.csv(Mediator_table_compact, here("data/Mediator_table_compact.csv"))


rm(Mediator_table, Mediator_table_compact)
```

```
#Calculate variable change
data_flame_change <- data_flame %>%
    select(Time_point, Treatment, Sample_ID, starts_with("Cer"),
           V2_U_LogMeanUAER, V5_U_LogMeanUAER, V2_B_VAEGT, V5_B_VAEGT, V2_B_HBA1C, V5_B_HBA1C) %>%
    mutate(Time_point = factor(Time_point,
                               labels = c("Pre", "Post"))) %>%
    mutate(Treatment = factor(Treatment,
                               labels = c("Lira", "Placebo"))) %>%
    rename_with(~ gsub("_._", "", .)) %>%
    pivot_wider(names_from = c(Treatment, Time_point, Sample_ID),
                values_from = c(starts_with("Cer"), V2LogMeanUAER, V5LogMeanUAER,
                                V2VAEGT, V5VAEGT, V2HBA1C, V5HBA1C)) %>%
    pivot_longer(everything()) %>%
    separate(name, sep = "_", c("Variable","Treatment", "Time_point", "Sample_ID")) %>%
    pivot_wider(names_from = c(Variable, Time_point), values_from = value) %>%
    select(-c(V2LogMeanUAER_Post, V5LogMeanUAER_Pre,
              V2VAEGT_Post, V5VAEGT_Pre,
              V2HBA1C_Post, V5HBA1C_Pre)) %>%
    rename_with(~ gsub("V2", "", .)) %>%
    rename_with(~ gsub("V5", "", .)) %>%
    mutate(across(ends_with("_Post"),
                  ~ . - get(glue::glue(str_replace(cur_column(), "_Post", "_Pre")))
                  )) %>%
    select(-ends_with("_Pre")) %>%
    rename_with(~str_replace(., "_Post", ""), everything()) %>%
    rename_with(~str_replace(., ":", ""), everything())


#Loop through lipids of interest, and potential mediators
#Lipids of interest
Lipid_list_flame <- data_flame_change %>%
    select(starts_with("Cer")) %>%
    colnames()

#Mediators of interest
Mediator_list_flame <- data_flame_change %>%
    select(VAEGT, HBA1C, LogMeanUAER) %>%
    colnames()

#Empty table to populate
Mediator_table_flame <- data.frame()

#Nested loop going through all the lipid and mediators
#Each combination is tested in 3 linear regression models:
#Model 1: Lipid ~ Treatment
#Model 2: Mediator ~ Treatment
#Model 3: Lipid ~ Treatment + Mediator
#Model 4: Causal Mediation Analysis of Model 2 and 3 (bootstrap 1000)
#Note: this step quite slow, get a cup of coffee
if(TRUE){ #Set this to TRUE in order to run
for(l in Mediator_list_flame){
    for(i in Lipid_list_flame){

        #Step 1: Total effect - Treatment on lipid: i
        form1 <- paste(i, "~ Treatment")
        model1 <- lm(form1, data_flame_change)
        model1_sum <- summary(model1)

        #Step 2: Effect on mediator - Treatment on Mediator l
        form2 <- paste(l, "~ Treatment")
        model2 <- lm(form2, data_flame_change)
        model2_sum <- summary(model2)

        #Step 3: Effect of mediator on response - Medi_var on Resp_Var
        form3 <- paste(i, "~ Treatment +", l)
        model3 <- lm(form3, data_flame_change)
        model3_sum <- summary(model3)
        #Step 4: Casual mediation analysis
        model4 <- mediation::mediate(model.m = model2, model.y = model3,
                                     treat = "Treatment",
                                     mediator = l,
                                     boot = TRUE)

        #Extract explanatory variable(s)
        explanetory_var <- rownames(model1_sum$coefficients)[nrow(model1_sum$coefficients)]

        #Populate
        #Model 1
        Mediator_table_flame[paste0("Model1_Estimate"), i] <-
            model1_sum$coefficients[explanetory_var, "Estimate"]
        Mediator_table_flame[paste0("Model1_StdError"), i] <-
            model1_sum$coefficients[explanetory_var, "Std. Error"]
        Mediator_table_flame[paste0("Model1_Pval"), i] <-
            model1_sum$coefficients[explanetory_var, "Pr(>|t|)"]
        Mediator_table_flame[paste0("Model1_R2"), i] <- model1_sum$adj.r.squared
        #Model2
        Mediator_table_flame[paste0("Model2_", l, "_Estimate"), i] <-
            model2_sum$coefficients[explanetory_var, "Estimate"]
        Mediator_table_flame[paste0("Model2_", l, "_StdError"), i] <-
            model2_sum$coefficients[explanetory_var, "Std. Error"]
        Mediator_table_flame[paste0("Model2_", l, "_Pval"), i] <-
            model2_sum$coefficients[explanetory_var, "Pr(>|t|)"]
        Mediator_table_flame[paste0("Model2_", l, "_R2"), i] <- model2_sum$adj.r.squared
        #Model3
        Mediator_table_flame[paste0("Model3_", l, "_Estimate"), i] <-
            model3_sum$coefficients[l, "Estimate"]
        Mediator_table_flame[paste0("Model3_", l, "_StdError"), i] <-
            model3_sum$coefficients[l, "Std. Error"]
        Mediator_table_flame[paste0("Model3_", l, "_Pval"), i] <-
            model3_sum$coefficients[l, "Pr(>|t|)"]
        Mediator_table_flame[paste0("Model3_", l, "_R2"), i] <- model3_sum$adj.r.squared
        #Model4
        Mediator_table_flame[paste0("Model4_", l, "_Total_Estimate"), i] <- model4$tau.coef
        Mediator_table_flame[paste0("Model4_", l, "_Total_CIlower"), i] <- model4$tau.ci[["2.5%"]]
        Mediator_table_flame[paste0("Model4_", l, "_Total_CIupper"), i] <- model4$tau.ci[["97.5%"]]
        Mediator_table_flame[paste0("Model4_", l, "_Total_Pval"), i] <- model4$tau.p

        Mediator_table_flame[paste0("Model4_", l, "_ADE_Estimate"), i] <- model4$z.avg
        Mediator_table_flame[paste0("Model4_", l, "_ADE_CIlower"), i] <- model4$z.avg.ci[["2.5%"]]
        Mediator_table_flame[paste0("Model4_", l, "_ADE_CIupper"), i] <- model4$z.avg.ci[["97.5%"]]
        Mediator_table_flame[paste0("Model4_", l, "_ADE_Pval"), i] <- model4$z.avg.p

        Mediator_table_flame[paste0("Model4_", l, "_ACME_Estimate"), i] <- model4$d.avg
        Mediator_table_flame[paste0("Model4_", l, "_ACME_CIlower"), i] <- model4$d.avg.ci[["2.5%"]]
        Mediator_table_flame[paste0("Model4_", l, "_ACME_CIupper"), i] <- model4$d.avg.ci[["97.5%"]]
        Mediator_table_flame[paste0("Model4_", l, "_ACME_Pval"), i] <- model4$d.avg.p
    }
}
}
# #Save Mediator_table as CSV
# write.csv(Mediator_table_flame, here("data/Mediator_table_flame.csv"))

rm(model1, model1_sum, model2, model2_sum, model3, model3_sum, model4,
   data_flame_change, explanetory_var, form1, form2, form3, i, l,
   Lipid_list_flame, Mediator_list_flame)

#Formatting mediator table
Mediator_table_flame_compact <- Mediator_table_flame %>%
    mutate(Model = rownames(.)) %>%
    as_tibble(.) %>%
    filter(grepl("Model4", Model)) %>%
    separate(Model, sep = "_", c("Model", "Mediator", "Measure", "Output")) %>%
    select(-Model) %>%
    filter(!grepl("ADE", Measure)) %>%
    pivot_longer(contains("Cer"), names_to = "Ceramide") %>%
    pivot_wider(names_from = c(Mediator, Measure, Output), values_from = value) %>%
        mutate(across(ends_with("_Estimate"),
                  ~ paste0(format(., digits = 4), " (",
                    format(get(glue::glue(str_replace(cur_column(),
                                                      "_Estimate", "_CIlower"))), digits = 4), ";",
                    format(get(glue::glue(str_replace(cur_column(),
                                                      "_Estimate", "_CIupper"))), digits = 4), ")"
                           ))) %>%
    select(-ends_with("_CIlower"), -ends_with("_CIupper"))

#NOTE: There might be small variations in the values due the randomness of the bootstrapping
# #Save Mediator_table_flame_compact as CSV
# write.csv(Mediator_table_flame_compact, here("data/Mediator_table_flame_compact.csv"))

rm(Mediator_table_flame, Mediator_table_flame_compact)
```