advanced_visualization_with_r_part_2_exercises

Exercise 1

Question 1

Read the fast_food_data.csv into a dataset named "fast_food".

Set both the `header` and `stringsAsFactors` arguments equal to TRUE.

Subset the data set to be named "fast_food_subset" and include columns 3, 5, 6, 10, 11.

Then rename those columns "type", "calories","totfat","carbs", & "sugars".

Answer:

```
fast_food = read.csv(file = "data/fast_food_data.csv", header = TRUE, stringsAsFactors = TRUE)
fast_food_subset = fast_food[,c(3,5,6,10,11)]
colnames(fast_food_subset) = c("type", "calories","totfat","carbs", "sugars")
head(fast_food_subset)
```

```
##     type calories totfat carbs sugars
## 1 Burger      240      8    32      6
## 2 Burger      290     11    33      7
## 3 Burger      530     27    47      9
## 4 Burger      520     26    41     10
## 5 Burger      720     40    51     14
## 6 Burger      750     43    42     10
```

Question 2

Create a base plot for a scatterplot of totfat on the x axis and calories on the y axis, and save it to `ffplot1`.

Add a scatterplot layer with color as `tomato2`.

Enhance the scatterplot `ffplot1` with `my_ggtheme`.

Add a regression line to the scatterplot, and save the final figure as `ffplot1`.

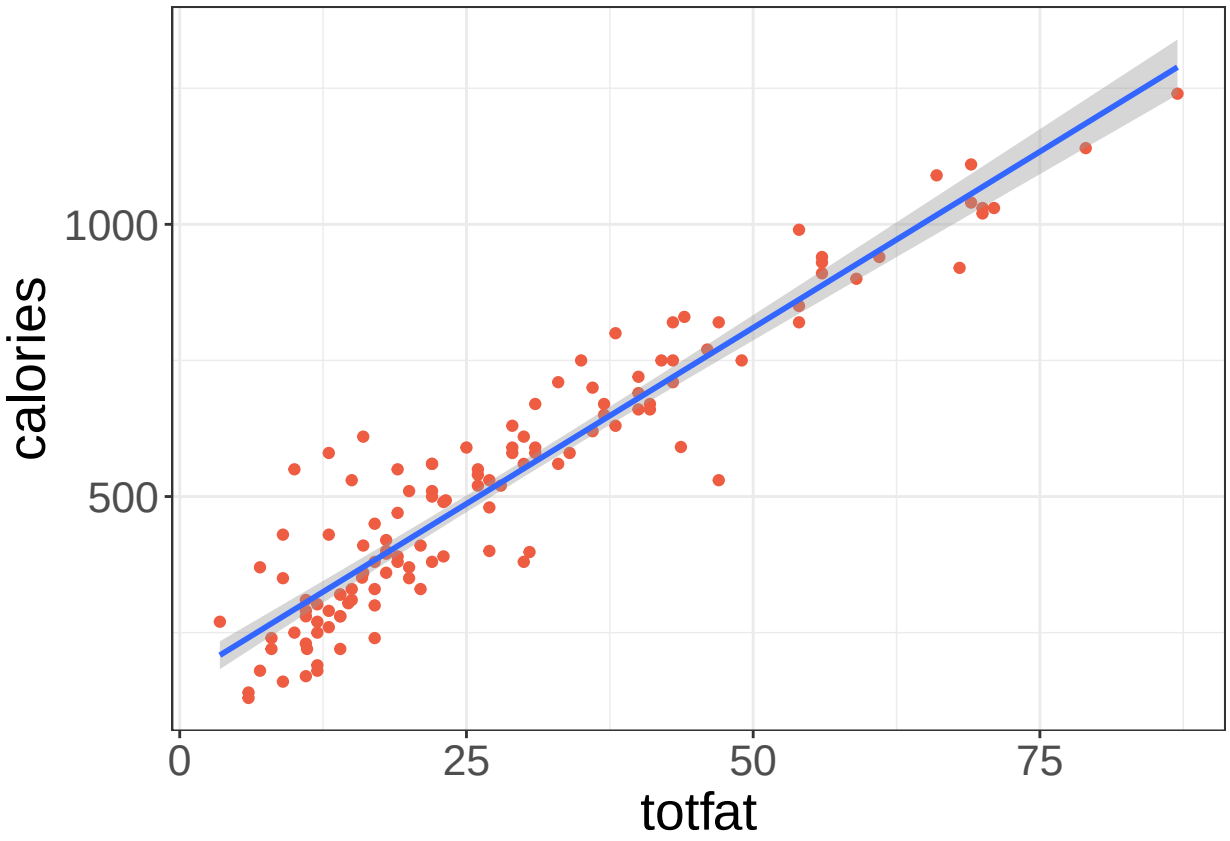
Answer:

```
library(ggplot2)
my_ggtheme = theme_bw() +
  theme(axis.title = element_text(size = 20),
        axis.text = element_text(size = 16),
        plot.title = element_text(size = 25),
        plot.subtitle = element_text(size = 18))

ffplot1 = ggplot(data = fast_food_subset, aes(x = totfat, y = calories)) +
  geom_point(color = "tomato2") +
  my_ggtheme +
  geom_smooth(method = lm)

ffplot1
```

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



Question 3

Load the `tidyverse` package

Create a new subset from `fast_food`, named `fast_food_sub`,

Select only `Calories` and variables that end in "g.",

EXCLUDE variables that start with "Serving" from `fast_food`.

Drop the rows with missing values.

Transform `fast_food_sub` to a long dataset `fast_food_long`.

Gather them using `key` and `value`.

Make sure to check the data afterwards.

Answer:

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
## v tibble  3.1.3     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.0     v forcats 0.5.1
## v purrr   0.3.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
fast_food_sub = fast_food %>%
  select(Calories, ends_with("g."), -starts_with("Serving")) %>%
  drop_na()

fast_food_long = gather(data = fast_food_sub, key = "variable", value = "value")

head(fast_food_long)
```

```
##   variable value
## 1 Calories   240
## 2 Calories   290
## 3 Calories   530
## 4 Calories   520
## 5 Calories   720
## 6 Calories   750
```

Question 4

Set up data:

Use separate `mutate` statements to achieve the following goals:

- Convert all strings in variable to lower case

- Use `substr` and `nchar` to remove the last "." from `variable`

- Remove remaining "."" from variable names. The ideal variable reads "trans_fat___g"

Hint: use `str_replace_all`.

Confirm the changes in `fast_food_long`.

Answer:

```
fast_food_long = fast_food_long %>%
  mutate(variable = str_to_lower(variable)) %>%
  mutate(variable = substr(variable, 1, nchar(variable)-1)) %>%
  mutate(variable = str_replace_all(variable,  "[.]",  "_"))
```

Question 5

Create a base box plot of `fast_food_long` and save it as `ffboxplot`.

Update the aesthetics of the box plot by filling the boxplot with color,

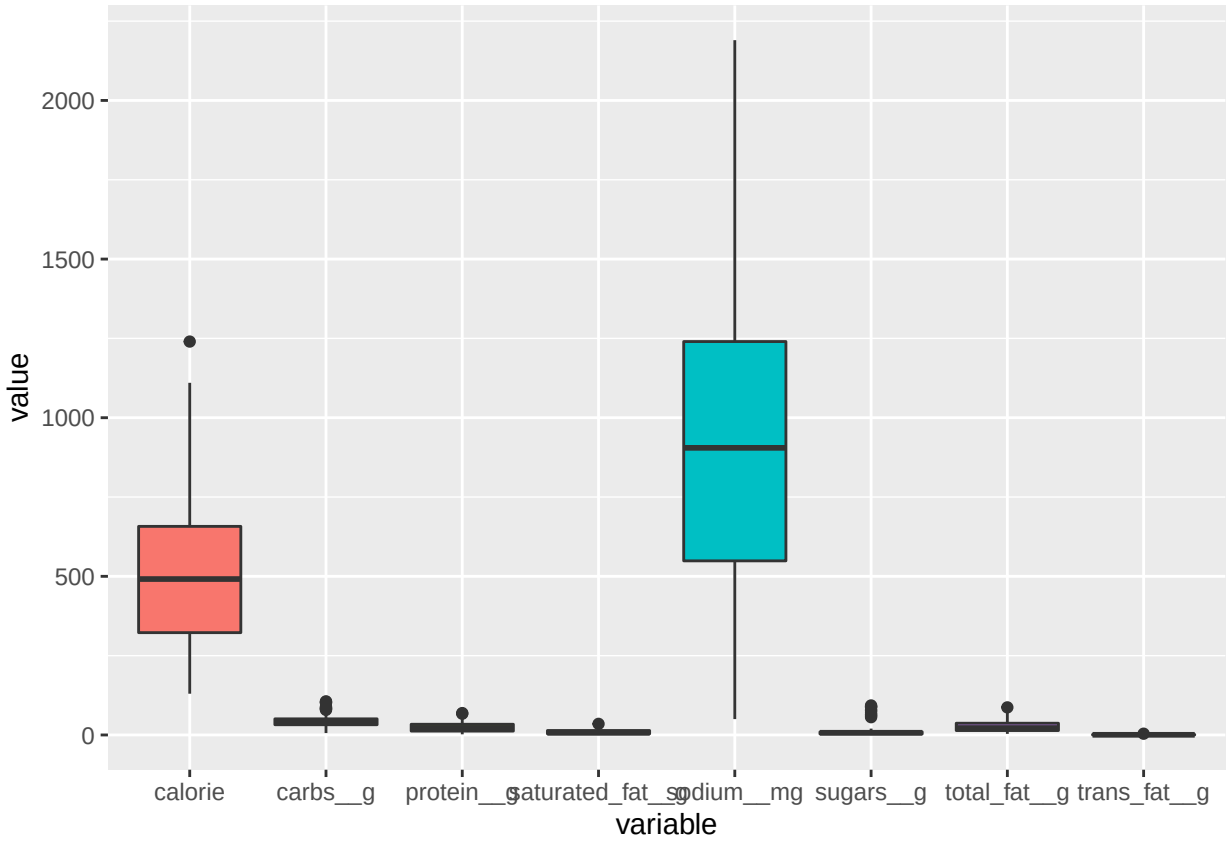
but make sure a legend for color is not included in the plot.

Answer:

```
library(ggplot2)
ffboxplot <- ggplot(data = fast_food_long, aes(x = variable, y = value, fill = variable)) +
  geom_boxplot() +
  guides(fill = FALSE)
```

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```

```
ffboxplot
```



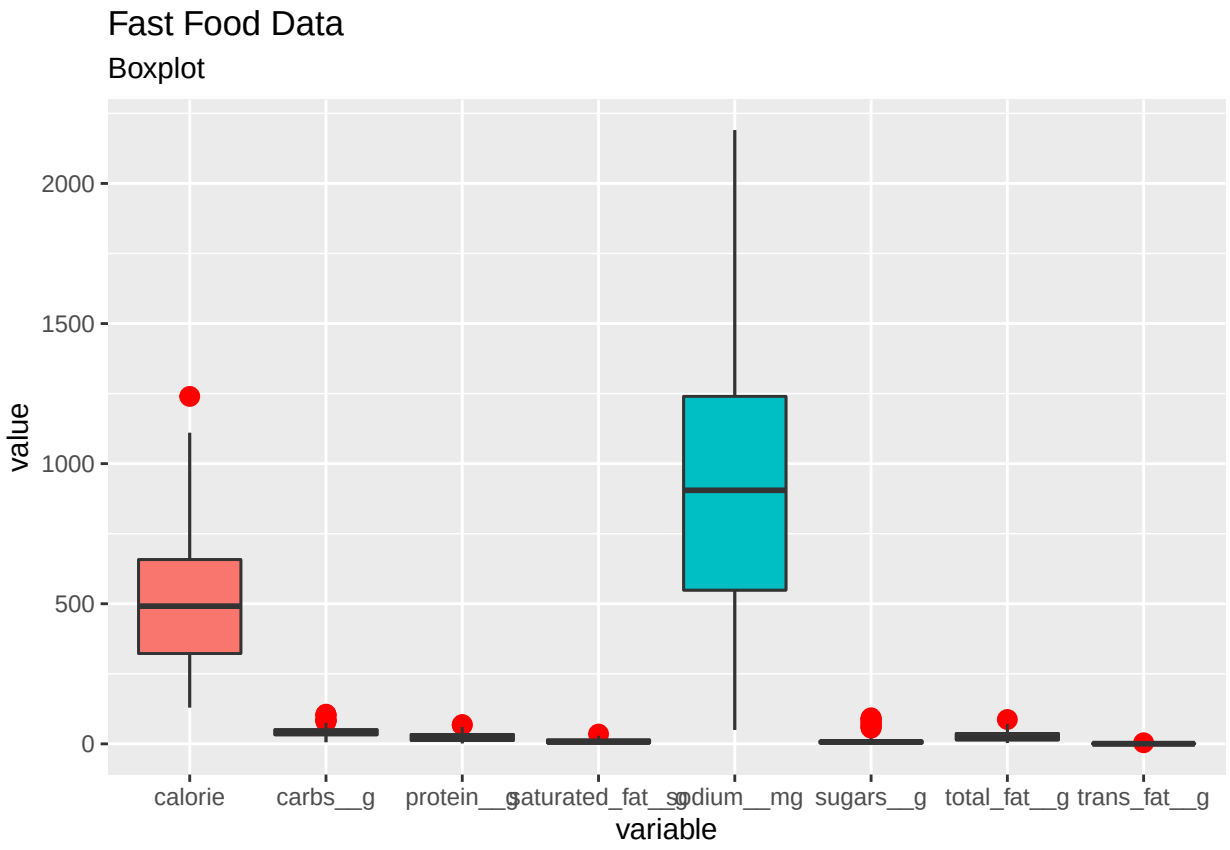
Question 6

Update `ffboxplot` by highlighting the outliers. Make them red and size 3.

Add a title "Fast Food Data" and subtitle "Boxplot" to the plot.

Answer:

```
ffboxplot = ffboxplot +
  geom_boxplot(outlier.colour = "Red",
               outlier.size = 3)+
  labs(title = "Fast Food Data",
       subtitle = "Boxplot")
ffboxplot
```



Question 7

Normalize the values for all variables in `fast_food_long`.

Remove NA's while normalizing with the maximum value.

Create a boxplot with normalized data.

Highlight outliers and add title and subtitle as in Question 6.

Answer:

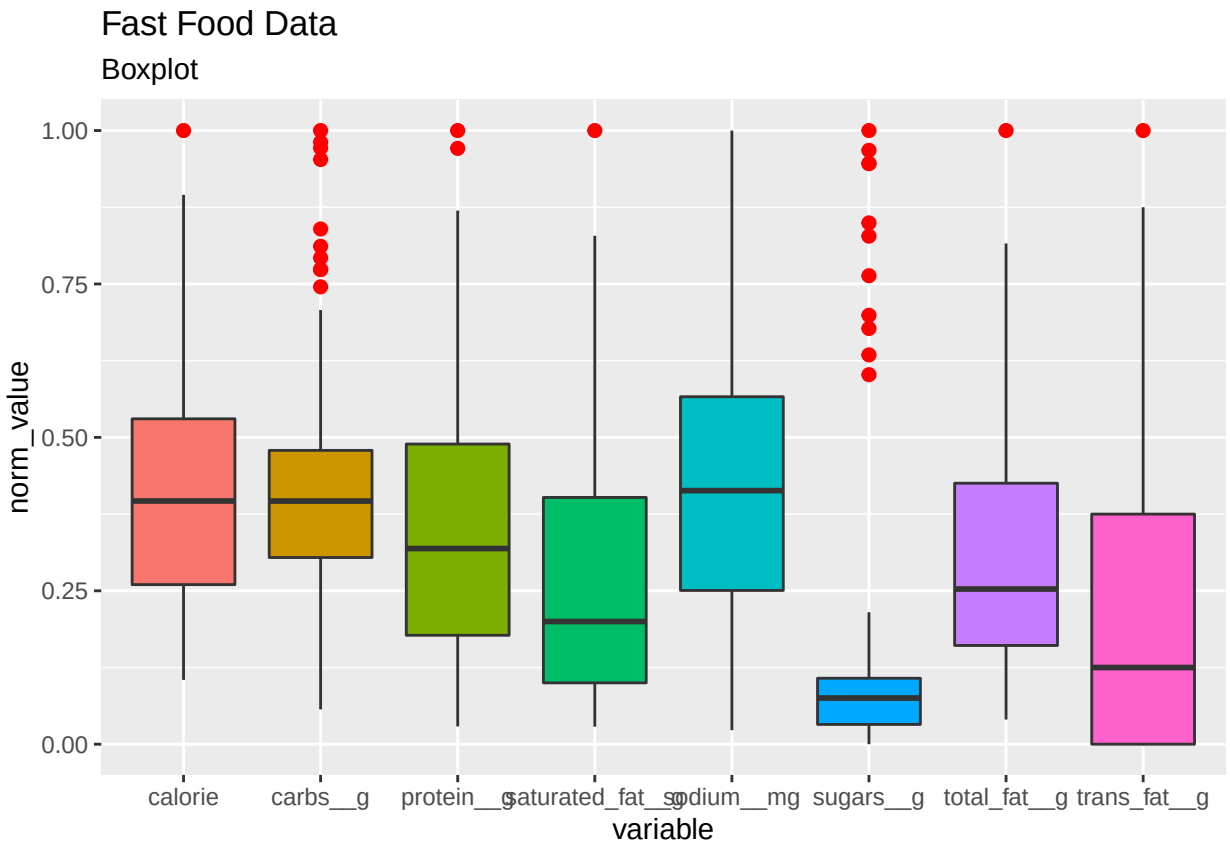
```
fast_food_long= fast_food_long %>% group_by(variable) %>%
  mutate(norm_value = value / max(value, na.rm = TRUE))
fast_food_long
```

```
## # A tibble: 912 x 3
## # Groups:   variable [8]
##    variable value norm_value
##    <chr>    <dbl>      <dbl>
##  1 calorie    240      0.194
##  2 calorie    290      0.234
##  3 calorie    530      0.427
##  4 calorie    520      0.419
##  5 calorie    720      0.581
##  6 calorie    750      0.605
##  7 calorie    530      0.427
##  8 calorie    510      0.411
##  9 calorie    350      0.282
## 10 calorie    190      0.153
## # ... with 902 more rows
```

```
ffboxplot = ggplot(data = fast_food_long,
                   aes(x = variable, y = norm_value, fill = variable)) +
  geom_boxplot(outlier.colour = "Red",
               outlier.size = 2) +
  guides(fill = FALSE) +
  labs(title = "Fast Food Data",
       subtitle = "Boxplot")
```

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```

```
ffboxplot
```



Exercise 2

==========

Question 1

Convert `fast_food_sub` into a long dataset `FF_subset_long2`

Normalize the nutrition components.

Explicitly exclude `Calories` from the `gather` statement.

Use the same code used previously to:

- Change the case of all letters to lower case

- Remove the last character from the end of the nutrition variables

- Replace all remaining . with _ in the variable names

Then use the group_by & mutate statements to normalize the nutritional values, using the MEAN to normalize.

Check the head of FF_subset_long2 to confirm you have four columns. What are those column names?

Answer:

```
FF_subset_long2 = fast_food_sub %>%
  gather(-Calories, key = variable, value = value) %>%
  mutate(variable = str_to_lower(variable)) %>%
  mutate(variable = substr(variable, 1, nchar(variable)-1)) %>%
  mutate(variable = str_replace_all(variable,  "[.]",  "_")) %>%
  group_by(variable) %>%
  mutate(norm_value = value / mean(value, na.rm = TRUE))

FF_subset_long2
```

```
## # A tibble: 798 x 4
## # Groups:   variable [7]
##    Calories variable     value norm_value
##       <int> <chr>        <dbl>      <dbl>
##  1      240 total_fat__g     8      0.292
##  2      290 total_fat__g    11      0.402
##  3      530 total_fat__g    27      0.986
##  4      520 total_fat__g    26      0.950
##  5      720 total_fat__g    40      1.46
##  6      750 total_fat__g    43      1.57
##  7      530 total_fat__g    15      0.548
##  8      510 total_fat__g    22      0.804
##  9      350 total_fat__g     9      0.329
## 10      190 total_fat__g    12      0.438
## # ... with 788 more rows
```

Question 2

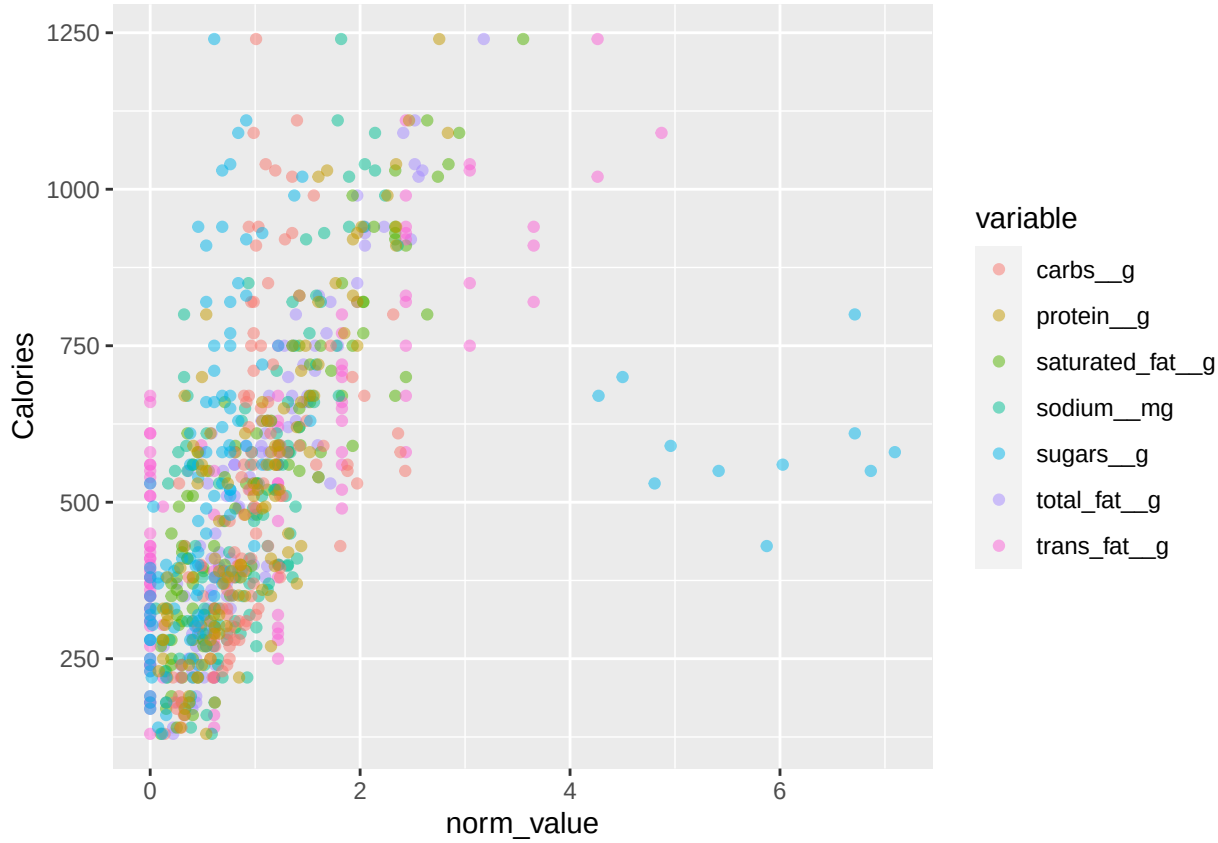
Create a base plot with `FF_subset_long2` and call it `base_norm_plot`.

Make sure to use the normalized values on the x-axis and calories on the y-axis.

Add a scatterplot layer to `base_norm_plot` with point size = 1.5 and 50% opacity. Save it as `scatter_norm`.

Answer:

```
base_norm_plot = ggplot(data = FF_subset_long2,
                        aes(x = norm_value, y = Calories, color = variable))
scatter_norm = base_norm_plot +
                geom_point(size=1.5,
                           alpha=0.5)
scatter_norm
```



Question 3

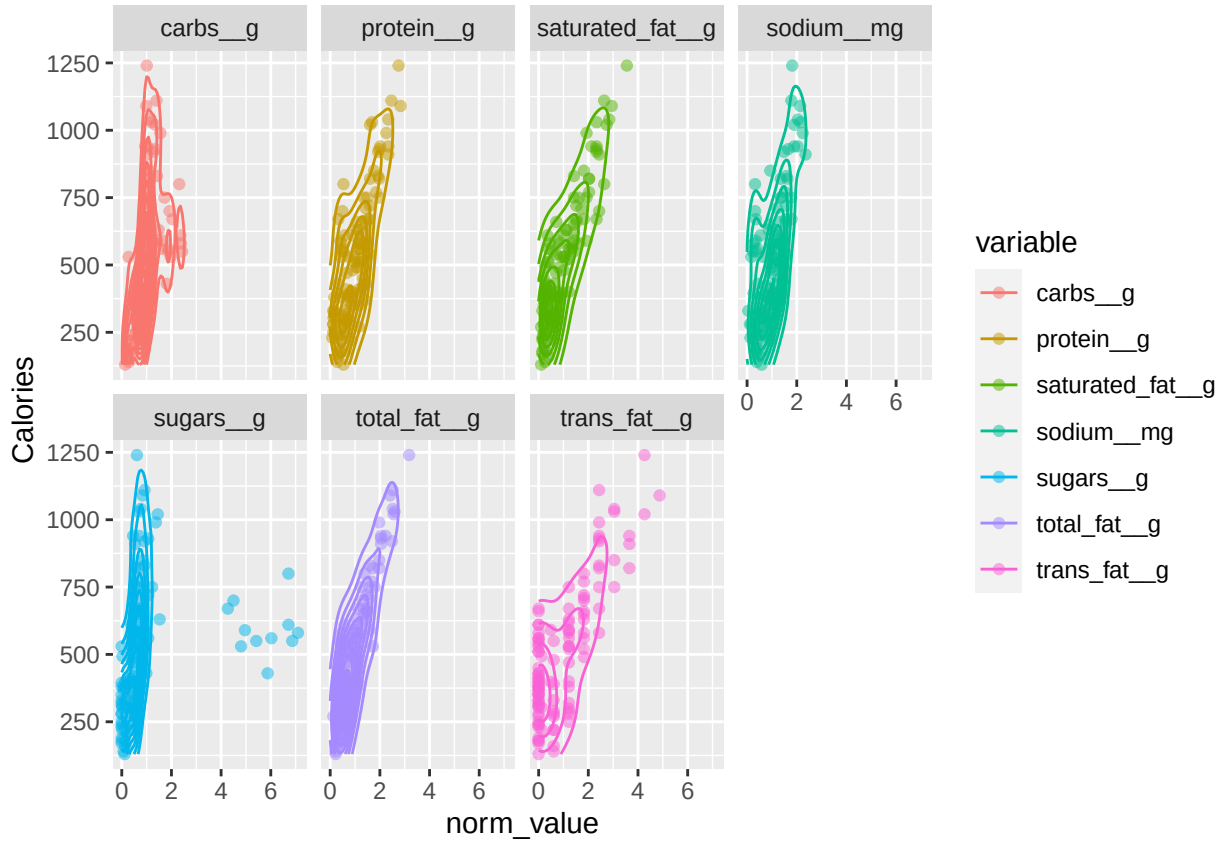
Add a 2d geom_density layer to `scatter_norm` and save it as `scatter_norm`.

Split the scatterplots into different facets for each variable using `facet_wrap`, displayed in 2 rows.

Answer:

```
scatter_norm = scatter_norm +
               geom_density2d()  +
               facet_wrap(~variable, nrow = 2)

scatter_norm
```



Question 4

Add a built-in theme `theme_light()` to the scatterplots.

Remove the redundant legend.

Finally, add title "Fast Food: Calories vs. Other variables" and, subtitle "2D distribution of scaled data" to the plots.

View the updated plot.

Answer:

```
scatter_norm = scatter_norm + theme_light() + guides(color = FALSE) +
               labs(title = "Fast Food: Calories vs. Other variables",
                    subtitle = "2D distribution of scaled data")
```

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```

```
scatter_norm
```