

Milkfat: boxplots and confidence interval plots

Introduction

In this exercise we will make some different types of graphs to visualise a continuous variable that is observed for different levels of a categorical variable. This might be interesting in applications where you consider using analysis of variance.

Further examples of plots that can be used for this purpose can be found in Chapter 4 of *Modern Statistics with R*.

Before you start, make sure you have installed and loaded the package `tidyverse` package.

```
library(tidyverse)
```

Read the milk fat data set

This dataset contains data on the milkfat contents for two different breeds of cows (SLB and SRB). 12 animals of each breed get one of two different diets (a high fat diet or a low fat diet).

```
milkfat <- read_csv("Milk_fat.csv")
```

Make a summary for the whole dataset

A first overview can be obtained by making a `summary`:

```
summary(milkfat)
```

Make a summary for different combination of breed and treatment

It is more interesting to know the means that result from the two different diets for each of the breeds. Also standard deviations and standard errors¹ could be interesting. The number of observations per combination can be determined by the function `n()` - it should be 12. To get these computations per breed and diet the `group_by()` statement is used. `summarize` is used to compute the summary statistics for each combination.

```
milkfat_mean <- milkfat %>%  
  group_by(Breed, Treatment) %>%  
  summarize(meanfat = mean(`Milk fat`),  
            sd = sd(`Milk fat`),  
            se = sd(`Milk fat`) / sqrt(n()),  
            n = n())
```

Boxplots per treatment and breed

The data can be visualised with one boxplot per diet and breed. We specify one of the factors (diet) as factor on the x-axis and the other one is given using `fill`. `fill` allows us to split the boxplots in the different

¹Try to remember from your statistics class what the difference between standard deviation and standard error is

breeds and give them individual colors.

```
milkgfat %>%  
  ggplot(aes(x = Treatment,  
             y = `Milk fat`,  
             fill = Breed)) +  
  geom_boxplot()
```

Including `labs` let you determine the labels on the x- and y-axis.

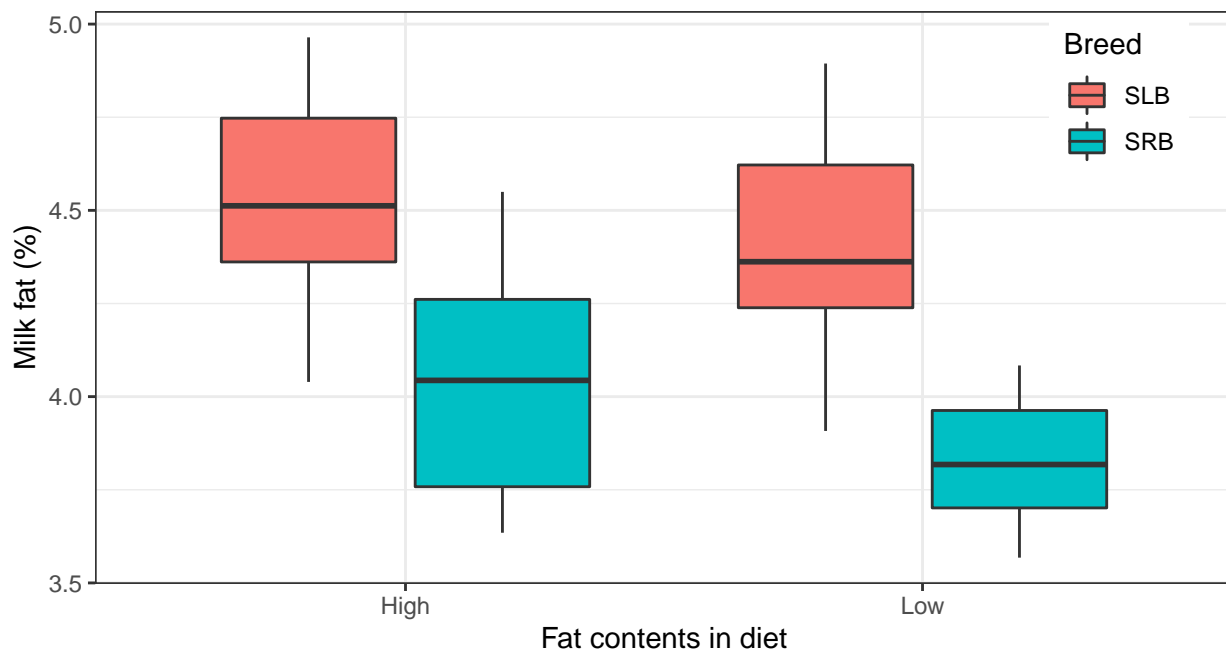
```
milkfat %>%  
  ggplot(aes(x = Treatment,  
            y = `Milk fat`,  
            fill = Breed)) +  
  geom_boxplot() +  
  labs(x = "Fat contents in diet",  
       y = "Milk fat (%)")
```

Legend positions can be chosen to be on either side or on the top or bottom of the plots:

```
milkfat %>%  
  ggplot(aes(x = Treatment,  
            y = `Milk fat`,  
            fill = Breed)) +  
  geom_boxplot() +  
  labs(x = "Fat contents in diet",  
       y = "Milk fat (%)") +  
  theme(legend.position = "top")
```

Sometimes it is more practical to set the legend inside the plot. `legend.position` can be specified for this. The positions can be thought of as percentages of the x- and y-range. In addition, we add `theme_bw`, which makes the background white with grey grid lines.

```
milkfat %>%  
  ggplot(aes(x = Treatment,  
            y = `Milk fat`,  
            fill = Breed)) +  
  geom_boxplot() +  
  labs(x = "Fat contents in diet", y = "Milk fat (%)") +  
  theme_bw() +  
  theme(legend.position = c(0.9, 0.85))
```



To produce plots for publications it could be a good idea to only use grey colors.

```

milkfat %>%
  ggplot(aes(x = Treatment,
             y = `Milk fat`,
             fill = Breed)) +
  geom_boxplot() +
  labs(x = "Fat contents in diet",
       y = "Milk fat (%)") +
  theme_bw() +
  theme(legend.position = c(0.9, 0.85)) +
  scale_fill_brewer(palette = "Greys")

```

Using the command below you can see other types of color palettes:

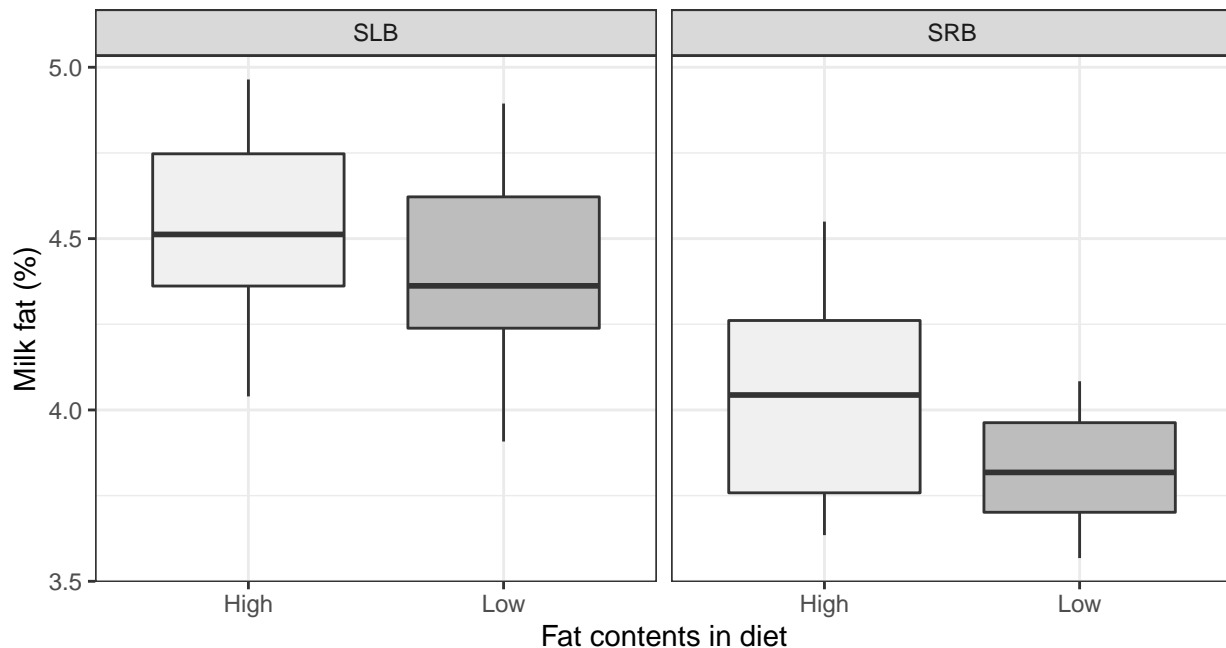
```
RColorBrewer::display.brewer.all()
```

Another possibility is to split the data into different plots, e.g. one for each breed. In this plot the legend is no longer needed and therefore removed.

```

milkfat %>%
  ggplot(aes(x = Treatment,
             y = `Milk fat`,
             fill = Treatment)) +
  geom_boxplot() +
  labs(x = "Fat contents in diet",
       y = "Milk fat (%)") +
  theme_bw() +
  theme(legend.position = "none") +
  scale_fill_brewer(palette = "Greys") +
  facet_wrap("Breed")

```



Plots of mean values with error bars, e.g. confidence interval plots

Earlier we have computed the mean, standard deviations and standard errors for all combinations. The following plot uses the output from that computation `\texttt{milkfat_mean}`.

First we plot only the mean values

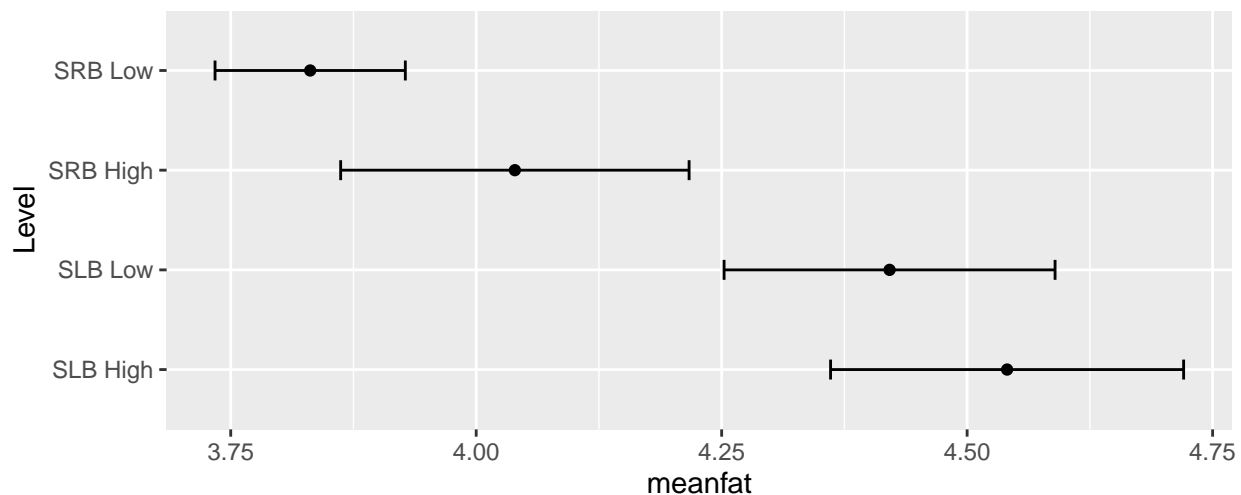
```
milkfat_mean %>%  
  ggplot(aes(x = Treatment,  
            y = meanfat)) +  
  geom_point() +  
  facet_wrap("Breed")
```

Assuming that the data is normally distributed approximate confidence intervals² can be computed by adding and subtracting $1.96 * \text{se}$ the standard error.

```
milkfat_mean %>%  
  ggplot(aes(x = Treatment,  
            y = meanfat,  
            ymin = meanfat - 1.96 * se,  
            ymax = meanfat + 1.96 * se)) +  
  geom_point() +  
  geom_errorbar(width = 0.1) +  
  facet_wrap("Breed")
```

Another possibility is to show all confidence intervals in the same plot. For this a common factor level is created. This factor combines treatment and breed.

```
milkfat_mean1 <- milkfat_mean %>%  
  mutate(Level = paste(Breed, Treatment, sep = " "))  
  
milkfat_mean1 %>%  
  ggplot(aes(y = Level,  
            x = meanfat,  
            xmin = meanfat - 1.96 * se,  
            xmax = meanfat + 1.96 * se)) +  
  geom_point() +  
  geom_errorbarh(height = 0.2)
```



²More correct would be to use the t-distribution