

R Notebook

0. remove warnings

```
windowsFonts(Times_New_Roman=windowsFont("Times New Roman"))  
windowsFonts(Arial=windowsFont("Arial"))  
windowsFonts(Comic_Sans_MS=windowsFont("Comic Sans MS"))  
windowsFonts(Impact=windowsFont("Impact"))
```

1. bar plot basic

```
library(ggplot2)

# 1. Load the data
# df_raw = read.csv("/media/windows11/PhD_PQS/ggplot_tutorial/mouse_drug.csv")
# df_stat = read.csv("/media/windows11/PhD_PQS/ggplot_tutorial/mouse_drug_stat.csv")
# save_path = "/media/windows11/PhD_PQS/ggplot_tutorial"

df_raw = read.csv("F:\\PhD_PQS\\ggplot_tutorial\\mouse_drug.csv")
df_stat = read.csv("F:\\PhD_PQS\\ggplot_tutorial\\mouse_drug_stat.csv")
save_path = "F:\\PhD_PQS\\ggplot_tutorial"

# 2. know about your data - can be done using MS Excel
# head(df_raw)
# summary(df_raw)
# View(df_raw)

# 3. bar plot

# 3.1 bar plot on df_stat - day1
df_stat_day1 = df_stat[which(df_stat$day == 1),]

# head(df_stat_day1)

barplot_day1 = ggplot(df_stat_day1, aes(x=group, y=weight_mean)) +
  geom_col(aes(fill=group)) +
  theme(

    panel.background=element_blank(),
    axis.line=element_line(color="black", linewidth=0.5),

    axis.text=element_text(color="black", size=10, face="italic", family="Times_New_Roma
n"),

    axis.title=element_text(color="black", size=15, face="bold"),
    axis.title.x=element_text(family="Arial"),
    axis.title.y=element_text(family="Comic_Sans_MS"),

    axis.ticks.length=unit(0.25, units="cm"),

    legend.position="bottom",
    legend.title=element_text(color="black", size=12, face="bold", family="Impact"),
    legend.text=element_text(color="black", size=12, face="bold.italic", family="Times_Ne
w_Roman")
  ) +
  scale_fill_manual("Test group", values=c("red", "blue")) +
  scale_x_discrete("Test group", labels=c("Control", "Experiment")) +
  scale_y_continuous("Weight (g)", expand=c(0, 0.5), breaks=seq(0, 45, 5),
    limits=c(NA, 40))

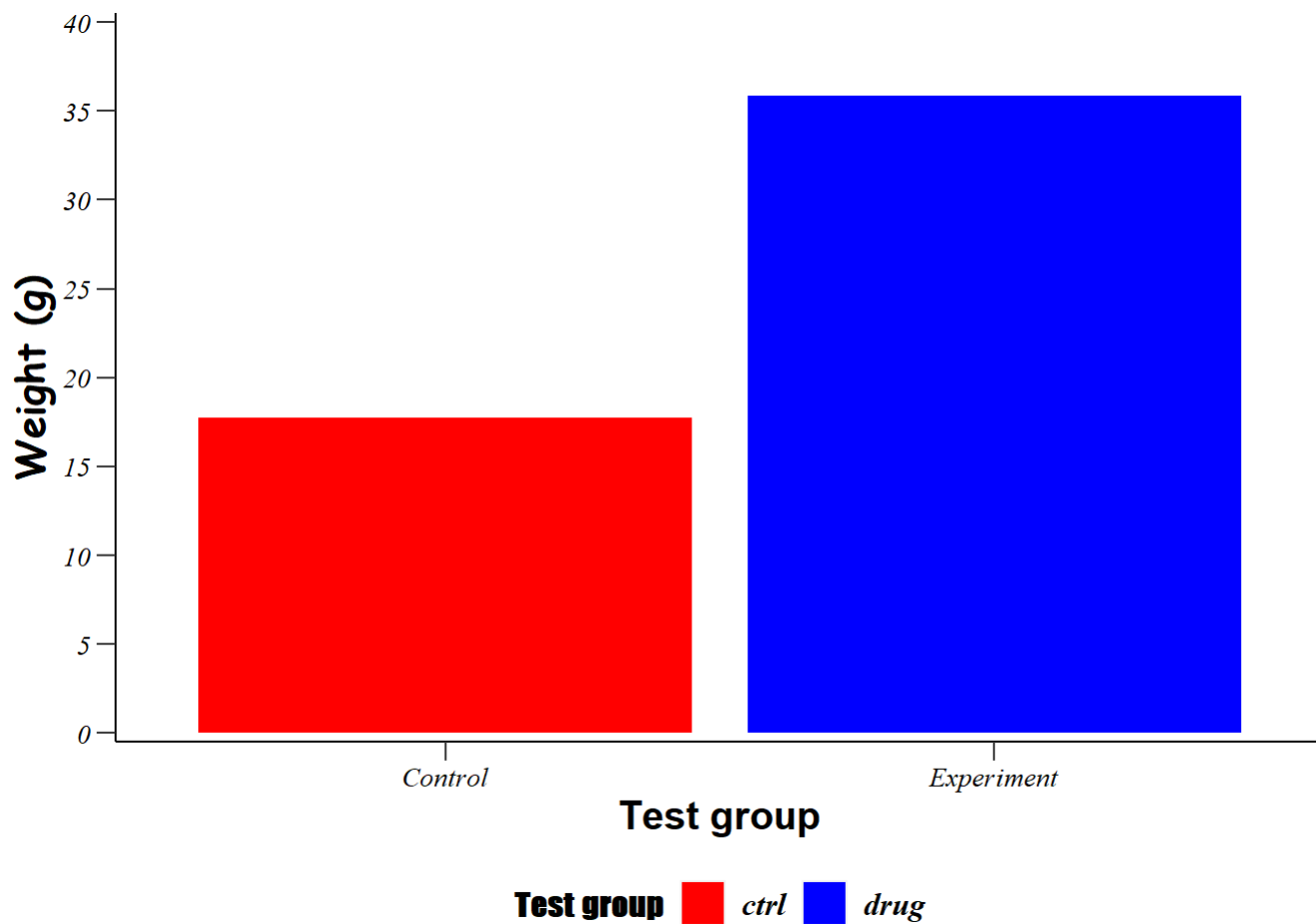
face_list = c("bold", "italic", "bold.italic")
# family_list = c("Times New Roman", "Arial", "mono", "Comic Sans MS", "Calibri")

# theme_bw()
# theme_classic()
# theme_light()
```

```
# theme_dark()
# theme_get()
# theme_minimal()

# barplot_day1

barplot_day1
```



```
# Practise

# 1. plot the day 3

# 2. change the legend position to the top

# 3. make the font family consistent to be "Arial"

# 4. change the color into "#12CF62" and "#CF5F12"
```

2. bar plot - error bar and p-value

```
library(ggplot2)
library(ggprism)

# 1. Load the data
# df_raw = read.csv("/media/windows11/PhD_PQS/ggplot_tutorial/mouse_drug.csv")
# df_stat = read.csv("/media/windows11/PhD_PQS/ggplot_tutorial/mouse_drug_stat.csv")
# save_path = "/media/windows11/PhD_PQS/ggplot_tutorial"

df_raw = read.csv("F:\\PhD_PQS\\ggplot_tutorial\\mouse_drug.csv")
df_stat = read.csv("F:\\PhD_PQS\\ggplot_tutorial\\mouse_drug_stat.csv")
save_path = "F:\\PhD_PQS\\ggplot_tutorial"

# 2. bar plot
df_raw_day1 = df_raw[which(df_raw$day == 1),]
df_stat_day1 = df_stat[which(df_stat$day == 1),]

barplot_day1 = ggplot(df_stat_day1, aes(x=group, y=weight_mean)) +
  geom_col(aes(fill=group), width=0.8) +
  geom_errorbar(aes(ymin=weight_mean-weight_std, ymax=weight_mean+weight_std, color=group),
    width=0.3, linewidth=1) +
  theme(
    panel.background=element_blank(),
    axis.line=element_line(color="black", linewidth=0.5),
    axis.text=element_text(color="black", size=10, family="Arial"),
    axis.title=element_text(color="black", size=15, face="bold", family="Arial"),
    axis.title.x=element_blank(),
    legend.position="bottom",
    legend.title=element_text(color="black", size=12, face="bold", family="Arial"),
    legend.text=element_text(color="black", size=12, family="Arial")
  ) +
  scale_fill_manual("Group", values=c("black", "#CF1212"), labels=c("Control", "Drug")) +
  scale_color_manual("Group", values=c("black", "#CF1212"), labels=c("Control", "Drug")) +
  scale_x_discrete("Group", expand=c(0.65, 0), labels=c("Control", "Drug")) +
  scale_y_continuous("Weight (g)", expand=c(0, 0.5), breaks=seq(0, 50, 5),
    limits=c(NA, 42))

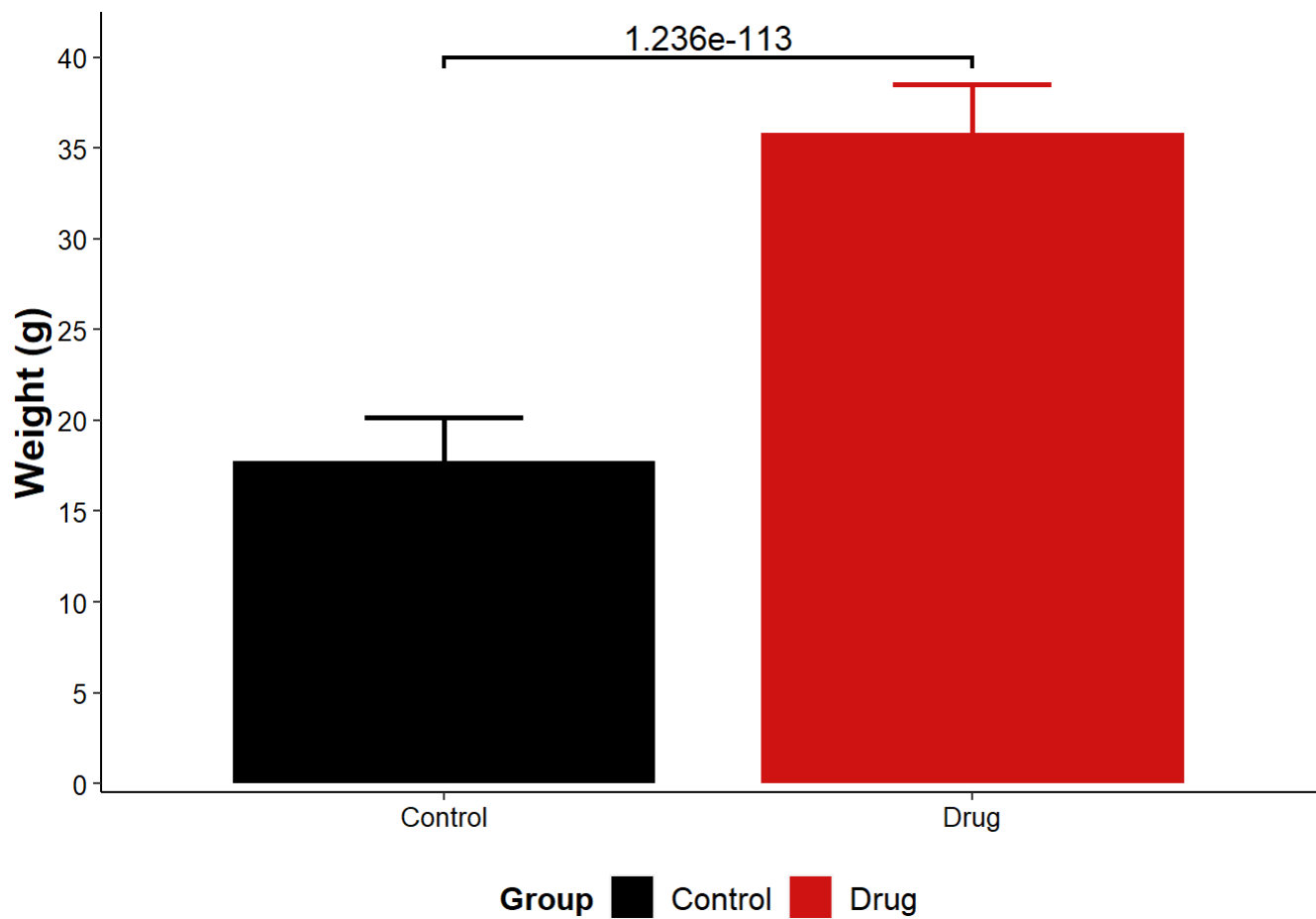
# add the p_value
t_test_results = t.test(weight ~ group, data=df_raw_day1)
p_value = t_test_results$p.value

p_value = signif(p_value, digits=4)

t_test_p_val = data.frame(group1="ctrl", group2="drug", label=p_value)

barplot_day1 = barplot_day1 + add_pvalue(t_test_p_val, y.position=40, bracket.size=0.8,
  tip.length=0.02, label.size=4.5,
  fontfamily="Arial")

barplot_day1
```



```
# Practice
```

```
# plot day 3, and adjust the figure, such as p_value_position
```

3. line plot

```
library(ggplot2)
```

```
# 1. Load the data
```

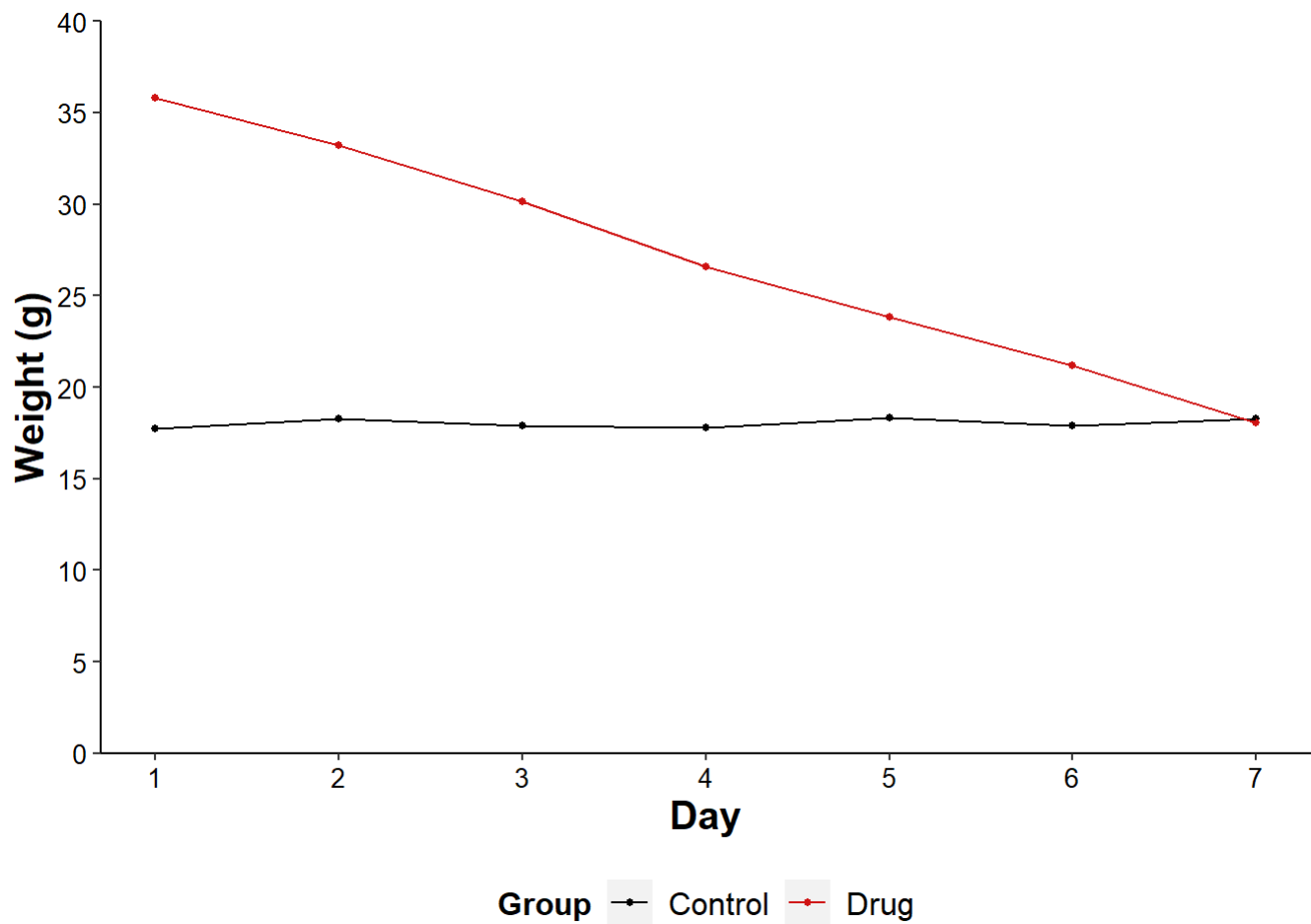
```
# df_raw = read.csv("/media/windows11/PhD_PQS/ggplot_tutorial/mouse_drug.csv")  
# df_stat = read.csv("/media/windows11/PhD_PQS/ggplot_tutorial/mouse_drug_stat.csv")  
# save_path = "/media/windows11/PhD_PQS/ggplot_tutorial"
```

```
df_raw = read.csv("F:\\PhD_PQS\\ggplot_tutorial\\mouse_drug.csv")  
df_stat = read.csv("F:\\PhD_PQS\\ggplot_tutorial\\mouse_drug_stat.csv")  
save_path = "F:\\PhD_PQS\\ggplot_tutorial"
```

```
# 2. Line plot
```

```
lineplot = ggplot(df_stat, aes(x=day, y=weight_mean)) +  
  geom_point(aes(color=group), size=1) +  
  geom_line(aes(color=group), linewidth=0.5) +  
  # geom_errorbar(aes(ymin=weight_mean-weight_std, ymax=weight_mean+weight_std, color=group),  
  #  
  width=0.3, linewidth=0.5) +  
  theme(  
    panel.background=element_blank(),  
    axis.line=element_line(color="black", linewidth=0.5),  
    axis.text=element_text(color="black", size=10, family="Arial"),  
    axis.title=element_text(color="black", size=15, face="bold", family="Arial"),  
    legend.position="bottom",  
    legend.title=element_text(color="black", size=12, face="bold", family="Arial"),  
    legend.text=element_text(color="black", size=12, family="Arial")  
  ) +  
  scale_fill_manual("Group", values=c("black", "#CF1212"), labels=c("Control", "Drug")) +  
  scale_color_manual("Group", values=c("black", "#CF1212"), labels=c("Control", "Drug")) +  
  scale_x_continuous("Day", breaks=seq(0, 7, 1)) +  
  scale_y_continuous("Weight (g)", expand=c(0, 0), breaks=seq(0, 50, 5),  
    limits=c(0, 40))
```

```
lineplot
```



```
# Practice
```

```
# 1. change the step of the y-axis: seq(0, 40, 10)
```

```
# 2. change the legend position to right, and adjust the size of the size of figure
```

```
# 3. add error bar
```

4. histogram

```
library(ggplot2)
library(ggprism)

# 1. Load the data
# df_raw = read.csv("/media/windows11/PhD_PQS/ggplot_tutorial/mouse_drug.csv")
# df_stat = read.csv("/media/windows11/PhD_PQS/ggplot_tutorial/mouse_drug_stat.csv")
# save_path = "/media/windows11/PhD_PQS/ggplot_tutorial"

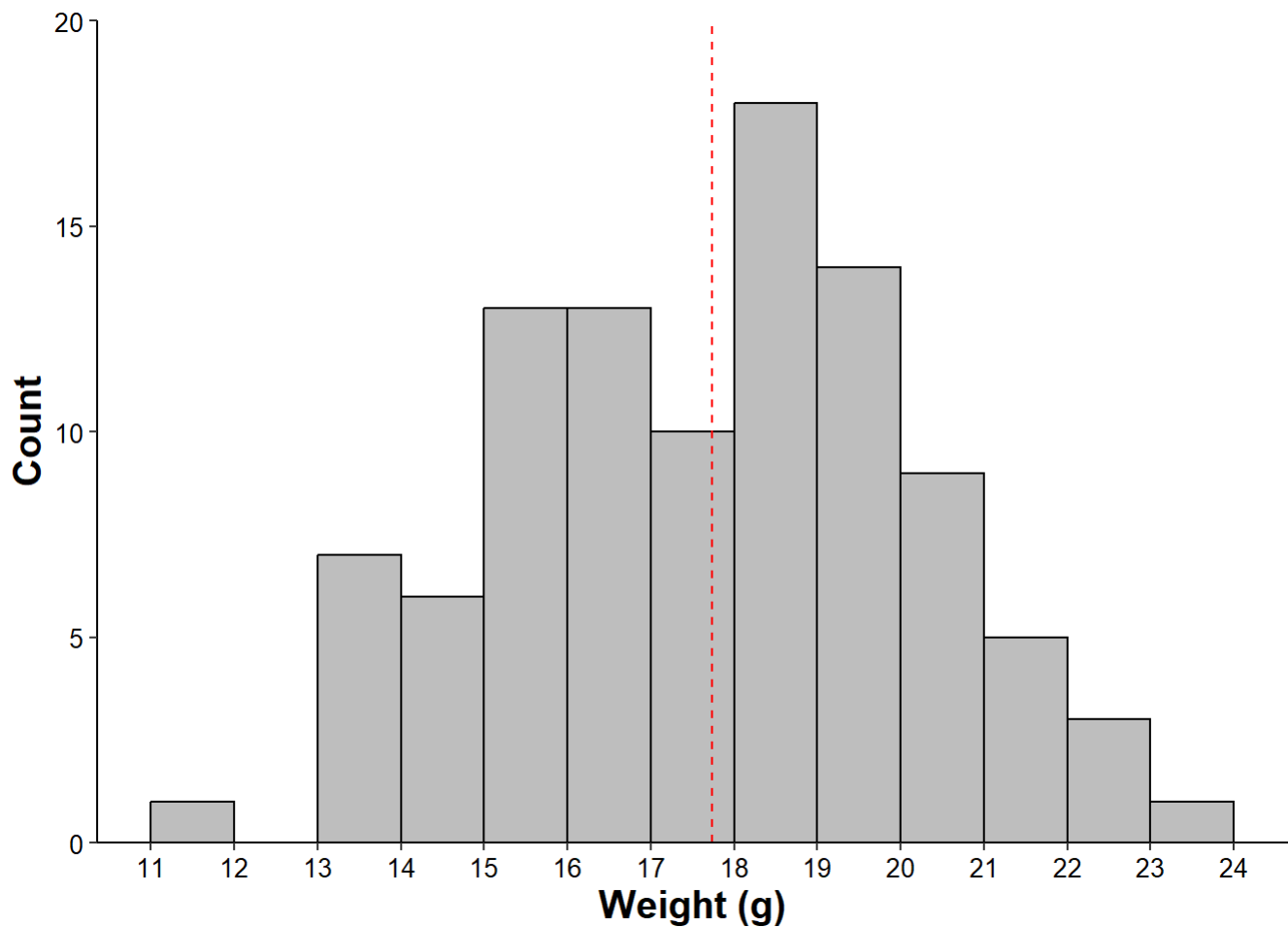
df_raw = read.csv("F:\\PhD_PQS\\ggplot_tutorial\\mouse_drug.csv")
df_stat = read.csv("F:\\PhD_PQS\\ggplot_tutorial\\mouse_drug_stat.csv")
save_path = "F:\\PhD_PQS\\ggplot_tutorial"

# 2. histogram plot
df_raw_ctrl_day1 = df_raw[which((df_raw$day == 1) & (df_raw$group == "ctrl")), ]

mean_weight = mean(df_raw_ctrl_day1$weight)

histplot_ctrl = ggplot(df_raw_ctrl_day1, aes(x=weight)) +
  geom_histogram(binwidth=1, color="black", fill="grey", boundary = 0) +
  geom_vline(xintercept=mean(mean_weight), linetype=2, color="red") +
  theme(
    panel.background=element_blank(),
    axis.line=element_line(color="black", linewidth=0.5),
    axis.text=element_text(color="black", size=10, family="Arial"),
    axis.title=element_text(color="black", size=15, face="bold", family="Arial"),
    legend.position="bottom",
    legend.title=element_text(color="black", size=12, face="bold", family="Arial"),
    legend.text=element_text(color="black", size=12, family="Arial")
  ) +
  scale_x_continuous("Weight (g)", breaks=seq(0, 26, 1)) +
  scale_y_continuous("Count", expand=c(0, 0), breaks=seq(0, 20, 5),
    limits=c(0, 20))

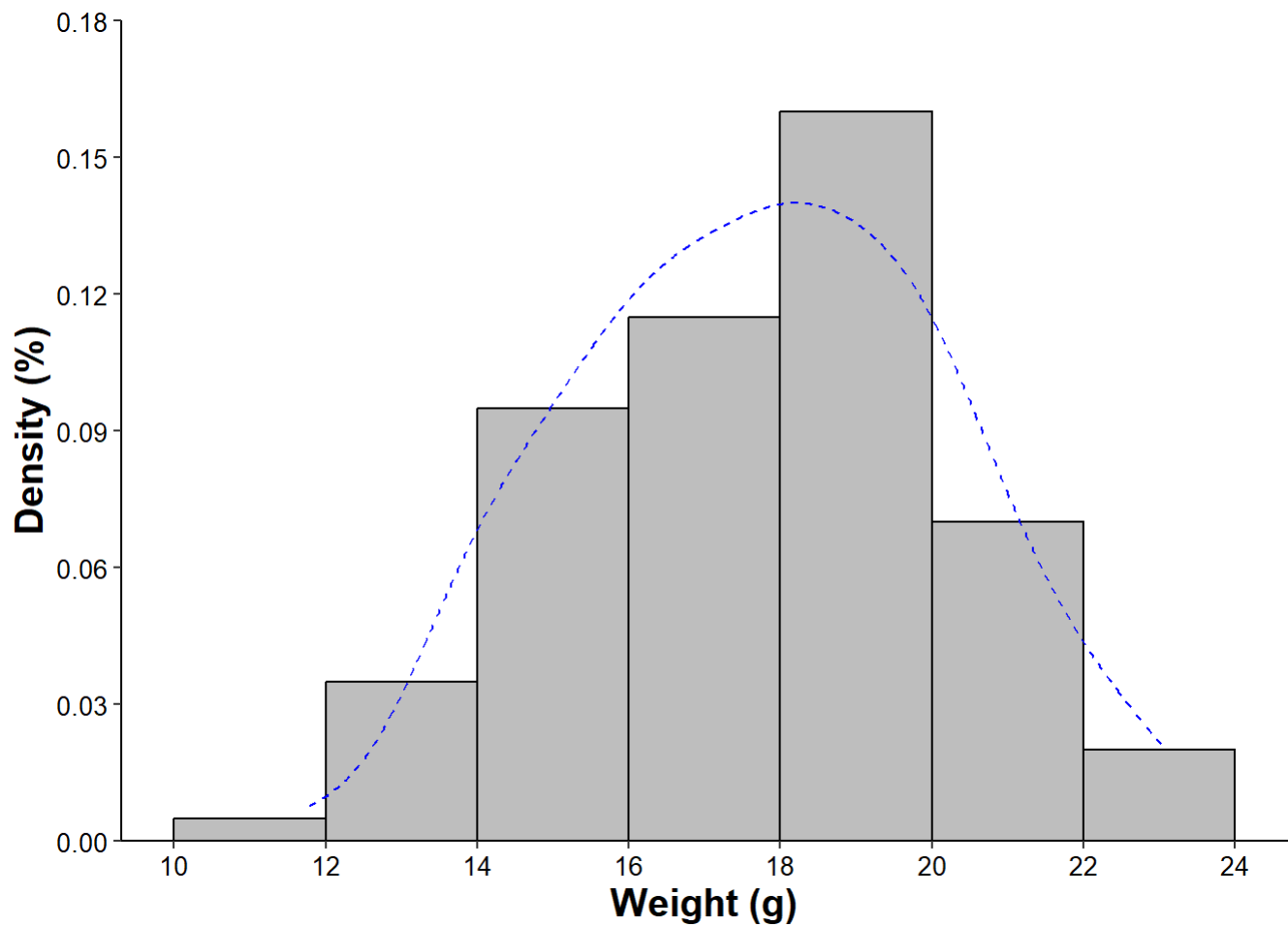
histplot_ctrl
```

```
# 3. density plot
```

```
histplot_ctrl_density = ggplot(df_raw_ctrl_day1, aes(x=weight, y=after_stat(density))) +
  geom_histogram(binwidth=2, color="black", fill="grey", boundary=0) +
  geom_density(linewidth=0.5, color="blue", linetype=2) +
  theme(
    panel.background=element_blank(),
    axis.line=element_line(color="black", linewidth=0.5),
    axis.text=element_text(color="black", size=10, family="Arial"),
    axis.title=element_text(color="black", size=15, face="bold", family="Arial"),
    legend.position="bottom",
    legend.title=element_text(color="black", size=12, face="bold", family="Arial"),
    legend.text=element_text(color="black", size=12, family="Arial")
  ) +
  scale_x_continuous("Weight (g)", breaks=seq(0, 30, 2)) +
  scale_y_continuous("Density (%)", expand=c(0, 0), breaks=seq(0, 2, 0.03),
    limits=c(0, 0.18))
```

```
histplot_ctrl_density
```



```
# Practice  
  
# 1. add a mean line on the density in green  
  
# 2. plot the drug_group on day 2
```

5. violin plot and boxplot

```
library(ggplot2)
library(ggprism)

# 1. Load the data
# df_raw = read.csv("/media/windows11/PhD_PQS/ggplot_tutorial/mouse_drug.csv")
# df_stat = read.csv("/media/windows11/PhD_PQS/ggplot_tutorial/mouse_drug_stat.csv")
# save_path = "/media/windows11/PhD_PQS/ggplot_tutorial"

df_raw = read.csv("F:\\PhD_PQS\\ggplot_tutorial\\mouse_drug.csv")
df_stat = read.csv("F:\\PhD_PQS\\ggplot_tutorial\\mouse_drug_stat.csv")
save_path = "F:\\PhD_PQS\\ggplot_tutorial"

# 2. violin plot
df_raw_day1 = df_raw[which(df_raw$day == 1),]

violinplot_day1 = ggplot(df_raw_day1, aes(x=group, y=weight)) +
  geom_violin(aes(fill=group)) +
  geom_jitter(size=0.5) +
  theme(
    panel.background=element_blank(),
    axis.line=element_line(color="black", linewidth=0.5),
    axis.text=element_text(color="black", size=10, family="Arial"),
    axis.title=element_text(color="black", size=15, face="bold", family="Arial"),
    axis.title.x=element_blank(),
    legend.position="bottom",
    legend.title=element_text(color="black", size=12, face="bold", family="Arial"),
    legend.text=element_text(color="black", size=12, family="Arial")
  ) +
  scale_fill_manual("Group", values=c("lightgray", "#CF1212"), labels=c("Control", "Drug"))
+
  scale_color_manual("Group", values=c("lightgray", "#CF1212"), labels=c("Control", "Drug")) +
  scale_x_discrete("Group", expand=c(0.65, 0), labels=c("Control", "Drug")) +
  scale_y_continuous("Weight (g)", expand=c(0, 0.5), breaks=seq(0, 50, 5),
    limits=c(0, 47))

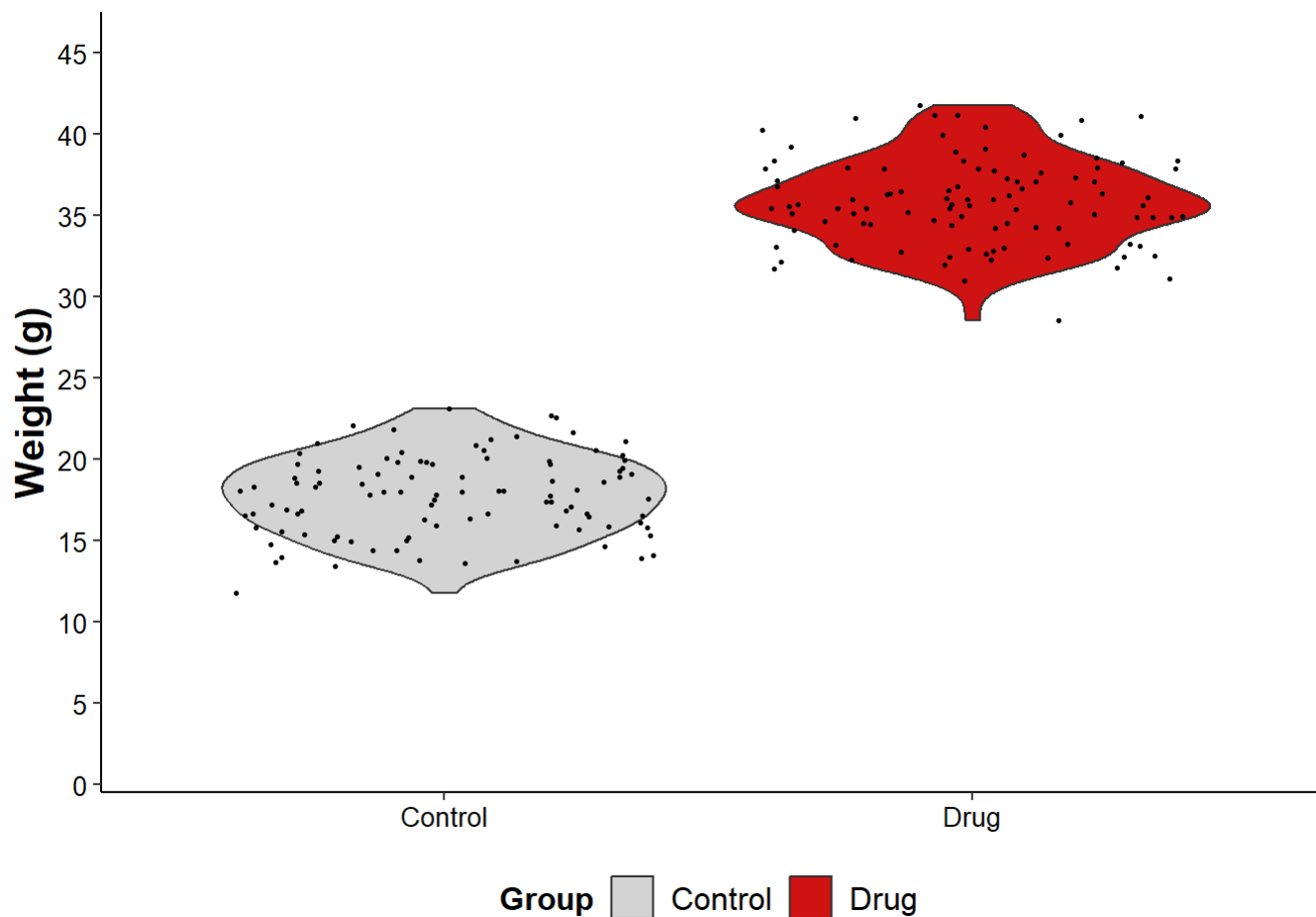
# add the p_value
# t_test_results = t.test(weight ~ group, data=df_raw_day1)
# p_value = t_test_results$p.value

# p_value = signif(p_value, digits=4)

# t_test_p_val = data.frame(group1="ctrl", group2="drug", label=p_value)

# violinplot_day1 = violinplot_day1 + add_pvalue(t_test_p_val, y.position=43.5, bracket.size=
0.5,
#
tip.length=0.02, label.size=4.5,
#
fontfamily="Arial")

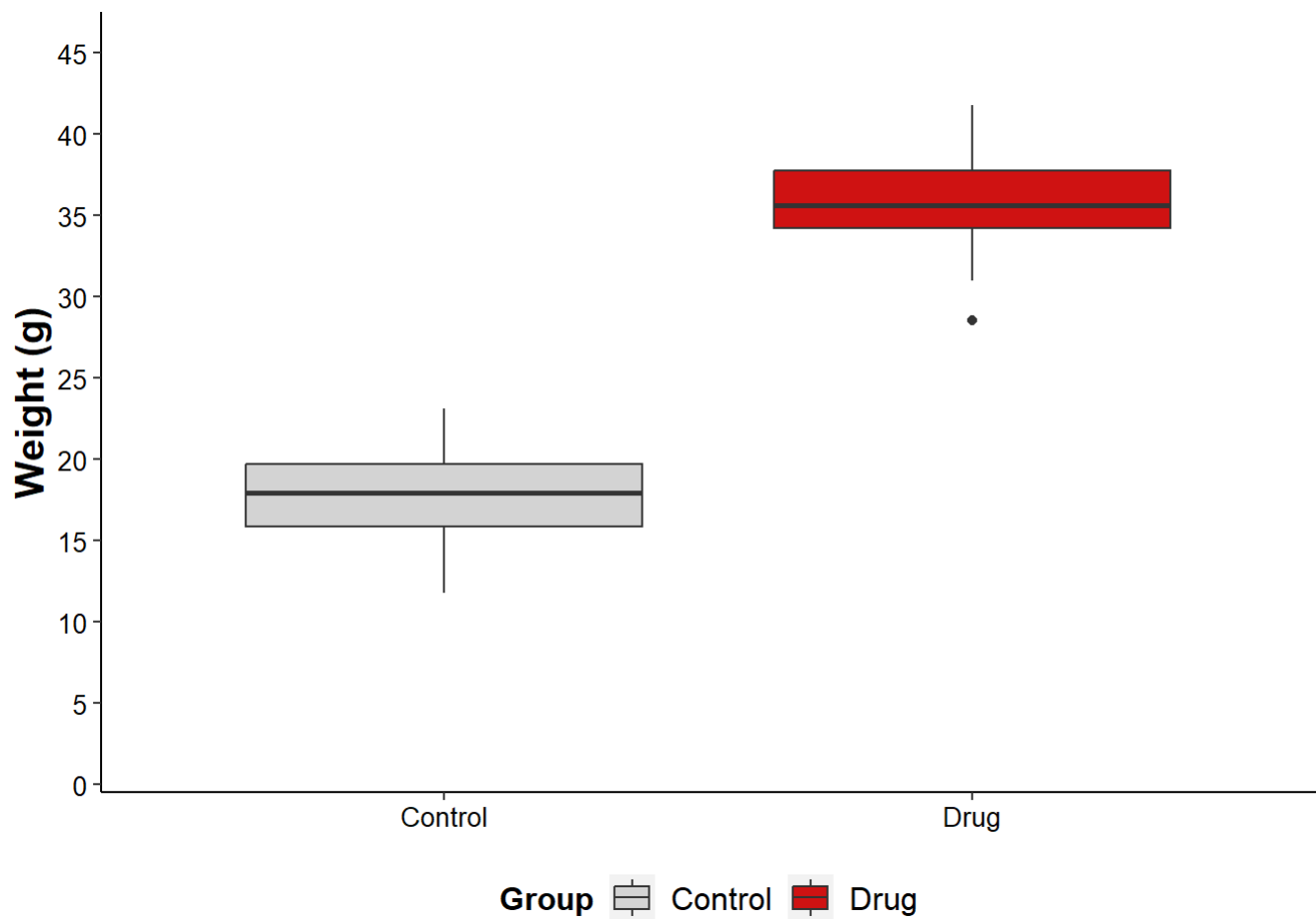
violinplot_day1
```



```
# 3. boxplot
df_raw_day1 = df_raw[which(df_raw$day == 1),]

boxplot_day1 = ggplot(df_raw_day1, aes(x=group, y=weight)) +
  geom_boxplot(aes(fill=group)) +
  theme(
    panel.background=element_blank(),
    axis.line=element_line(color="black", linewidth=0.5),
    axis.text=element_text(color="black", size=10, family="Arial"),
    axis.title=element_text(color="black", size=15, face="bold", family="Arial"),
    axis.title.x=element_blank(),
    legend.position="bottom",
    legend.title=element_text(color="black", size=12, face="bold", family="Arial"),
    legend.text=element_text(color="black", size=12, family="Arial")
  ) +
  scale_fill_manual("Group", values=c("lightgray", "#CF1212"), labels=c("Control", "Drug"))
+
  scale_color_manual("Group", values=c("lightgray", "#CF1212"), labels=c("Control", "Drug")) +
  scale_x_discrete("Group", expand=c(0.65, 0), labels=c("Control", "Drug")) +
  scale_y_continuous("Weight (g)", expand=c(0, 0.5), breaks=seq(0, 50, 5),
    limits=c(0, 47))

boxplot_day1
```



```
# Practice
```

```
# plot day 3, and adjust the figure, such as p_value_position
```

```
# add the p-value
```

6. heatmap

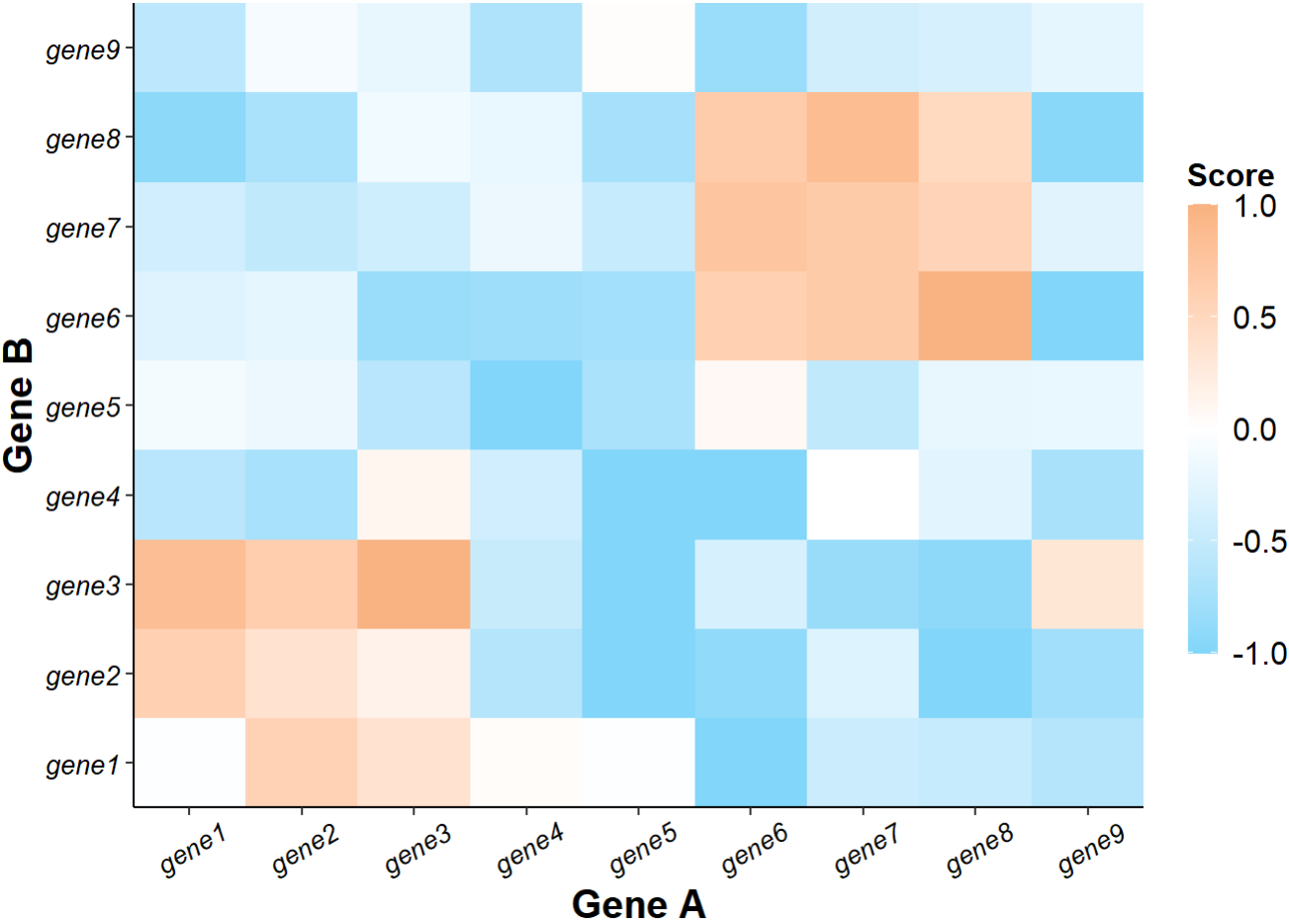
```
library(ggplot2)

# 1. Load the data
# df_heatmap = read.csv("/media/windows11/PhD_PQS/ggplot_tutorial/gene_interaction.csv")
# save_path = "/media/windows11/PhD_PQS/ggplot_tutorial"

df_heatmap = read.csv("F:\\PhD_PQS\\ggplot_tutorial\\gene_interaction.csv")
save_path = "F:\\PhD_PQS\\ggplot_tutorial"

# 2. heatmap
gene_heatmap = ggplot(df_heatmap, aes(x=gene_A, y=gene_B)) +
  geom_tile(aes(fill=score)) +
  theme(
    panel.background=element_blank(),
    axis.line=element_line(color="black", linewidth=0.5),
    axis.text=element_text(color="black", size=10, face="italic", family="Arial"),
    axis.text.x=element_text(angle=30, vjust=0.6),
    axis.title=element_text(color="black", size=15, face="bold", family="Arial"),
    legend.position="right",
    legend.title=element_text(color="black", size=12, face="bold", family="Arial"),
    legend.text=element_text(color="black", size=12, family="Arial"),
    legend.key.width=unit(0.4, units="cm"),
    legend.key.height=unit(1.2, units="cm"),
  ) +
  scale_fill_gradient2("Score", low="#81D6F9", mid="white", high="#F9B281", midpoint=0,
    breaks=seq(-1, 1, 0.5), limits=c(-1, 1)) +
  scale_x_discrete("Gene A", expand=c(0.05, 0)) +
  scale_y_discrete("Gene B", expand=c(0.05, 0))

gene_heatmap
```



```
# Practice
```

```
# change the colour scheme that you like
```

```
# Advanced
```

```
gene_heatmap_ad = ggplot(df_heatmap, aes(x=gene_A, y=gene_B)) +  
  geom_tile(aes(fill=score)) +  
  theme(  
    panel.border=element_rect(color="black", linewidth=0.5, fill=NA),  
    axis.line=element_blank(),  
    axis.text=element_text(color="black", size=10, face="italic", family="Arial"),  
    axis.text.x=element_text(angle=30, vjust=0.6),  
    axis.title=element_blank(),  
    legend.position="bottom",  
    legend.title=element_text(color="black", size=12, face="bold", family="Arial"),  
    legend.text=element_text(color="black", size=12, family="Arial"),  
    legend.key.width=unit(1.2, units="cm"),  
    legend.key.height=unit(0.4, units="cm"),  
  ) +  
  guides(fill=guide_colorbar(title.position="bottom", label.position="bottom",  
                             title.hjust=0.5, ticks=FALSE, frame.colour="black",  
                             frame.linewidth=0.4)) +  
  scale_fill_gradient2("Score", low="#81D6F9", mid="white", high="#F9B281", midpoint=0,  
                       breaks=seq(-1, 1, 0.5), limits=c(-1, 1)) +  
  scale_x_discrete("Gene A", expand=c(0.05, 0)) +  
  scale_y_discrete("Gene B", expand=c(0.05, 0))
```

```
gene_heatmap_ad
```