

talk09 练习与作业

目录

0.1 练习和作业说明	1
0.2 talk09 内容回顾	1
0.3 练习与作业：用户验证	2
0.4 练习与作业 1：基础做图 & ggplot2	3
0.5 练习与作业 2：多图组合，将多个图画在一起	20
0.6 练习与作业 3：作图扩展	27

0.1 练习和作业说明

将相关代码填写入以 “{r}” 标志的代码框中，运行并看到正确的结果；

完成后，用工具栏里的”Knit” 按键生成 PDF 文档；

将 PDF 文档改为：姓名-学号-talk09 作业.pdf，并提交到老师指定的平台/钉群。

0.2 talk09 内容回顾

- basic plot
- ggplot2

0.2.1 layered grammar (图层语法) 的成分

- 图层 (geom_XXX)
- scale (scale_XXX)
- faceting (facet_XXX)
- 坐标系统

0.2.2 图象类型

- 点图
- bars
- boxplots

0.2.3 其它重要内容（部分需要自学）

- colours
- theme
- 其它图像类型
- 图例 (legends) 和坐标轴
- 图形注释和其它定制

0.3 练习与作业：用户验证

请运行以下命令，验证你的用户名。

如你当前用户名不能体现你的真实姓名，请改为拼音后再运行本作业！

```
Sys.info()[["user"]]
```

```
## [1] "mingyuwang"
```

```
Sys.getenv("HOME")
```

```
## [1] "C:/Users/rhong/Documents"
```

0.4 练习与作业 1: 基础做图 & ggplot2

0.4.1 用 `swiss` 数据做图

1. 用直方图 `histogram` 显示 `Catholic` 列的分布情况;
2. 用散点图显示 `Eduction` 与 `Fertility` 的关系; 将表示两者关系的线性公式、相关系数和 `p` 值画在图的空白处。

注: 每种图提供基础做图函数和 `ggplot2` 两个版本!

```
## 代码写这里, 并运行;
```

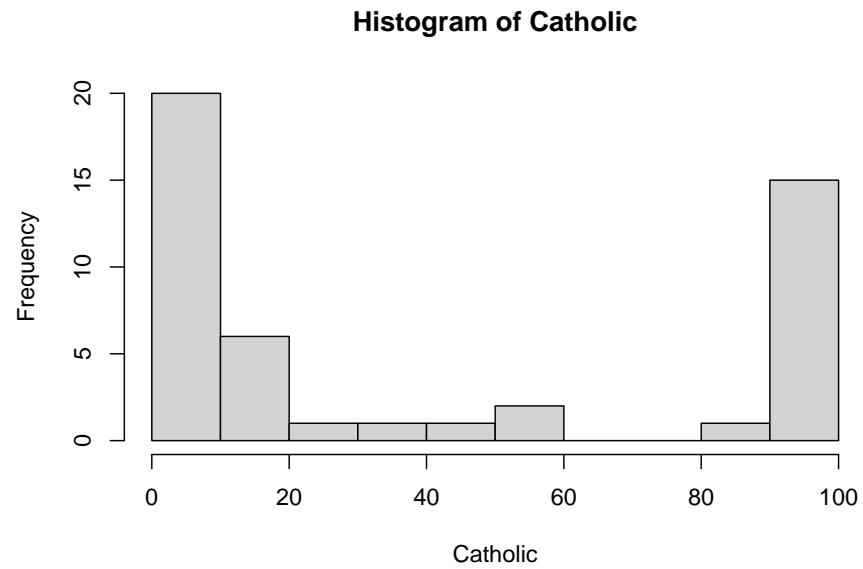
```
library(tidyverse);
```

```
## -- Attaching packages ----- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.8      v dplyr  1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
## 用直方图`histogram`显示`Catholic`列的分布情况
```

```
## 基础做图函数
```

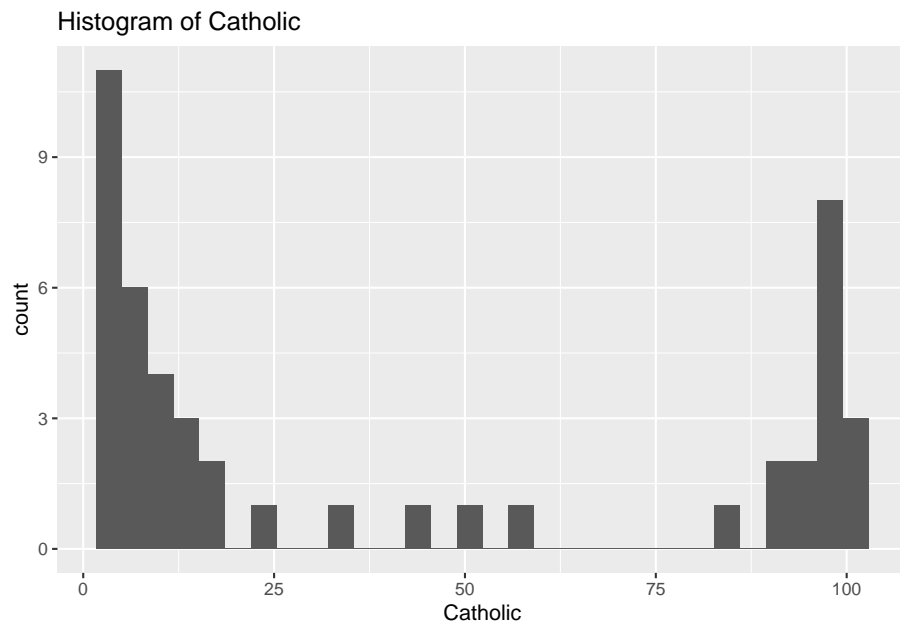
```
with(swiss,hist(Catholic));
```



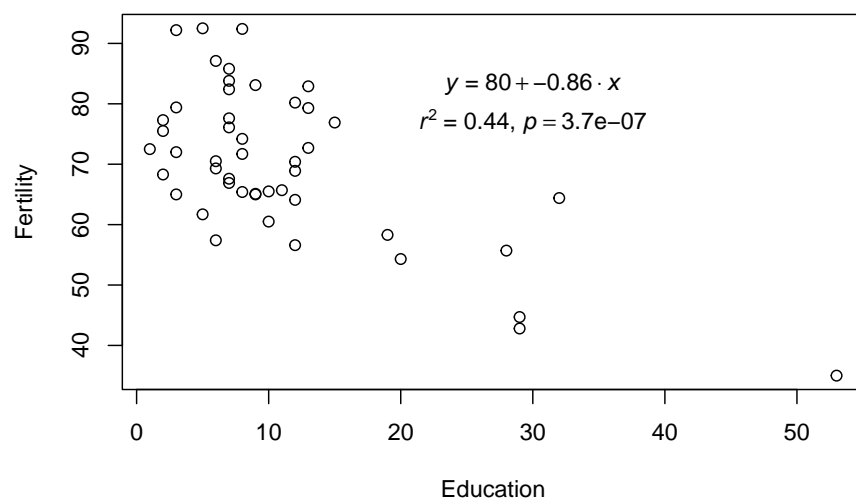
```
##ggplot2
```

```
ggplot(swiss,aes(x = Catholic)) + geom_histogram() +  
  ggtitle("Histogram of Catholic")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



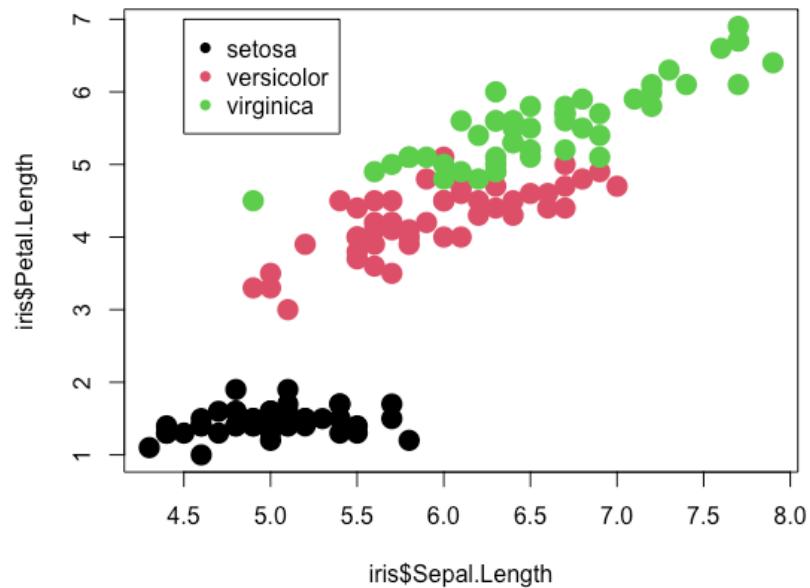
```
## 用散点图显示`Education`与`Fertility`的关系；将表示两者关系的线性公式、相关系数和  $p$  值画
m = lm(Fertility ~ Education, swiss);
c = cor.test( swiss$Fertility, swiss$Education );
eq <- substitute( atop(paste(italic(y), " = ", a + b %.% italic(x), sep = ""),
                        paste(italic(r)^2, " = ", r2, ", ", italic(p)==pvalue, sep="")),
                  list(a = as.vector( format(coef(m)[1], digits = 2) ),
                       b = as.vector( format(coef(m)[2], digits = 2) ),
                       r2 = as.vector( format(summary(m)$r.squared, digits = 2) ),
                       pvalue = as.vector( format( c$p.value , digits = 2) ) )
                  );
## 基础做图函数
with(swiss,plot(Education, Fertility,text(x = 30, y = 80, labels = as.expression(eq))))
```



```
##ggplot2
eq <- as.character(as.expression(eq));
# ggplot(swiss, aes(x = Education, y = Fertility)) + geom_point( shape = 20 ) +
#   geom_text( data = NULL,
#             aes( x = 30, y = 80, label= eq, hjust = 0, vjust = 1),
#             size = 4, parse = TRUE, inherit.aes = F)
```

0.4.2 用 iris 作图

1. 用散点图显示 Sepal.Length 和 Petal.Length 之间的关系; 按 species 为散点确定颜色, 并画出 legend 以显示 species 对应的颜色;

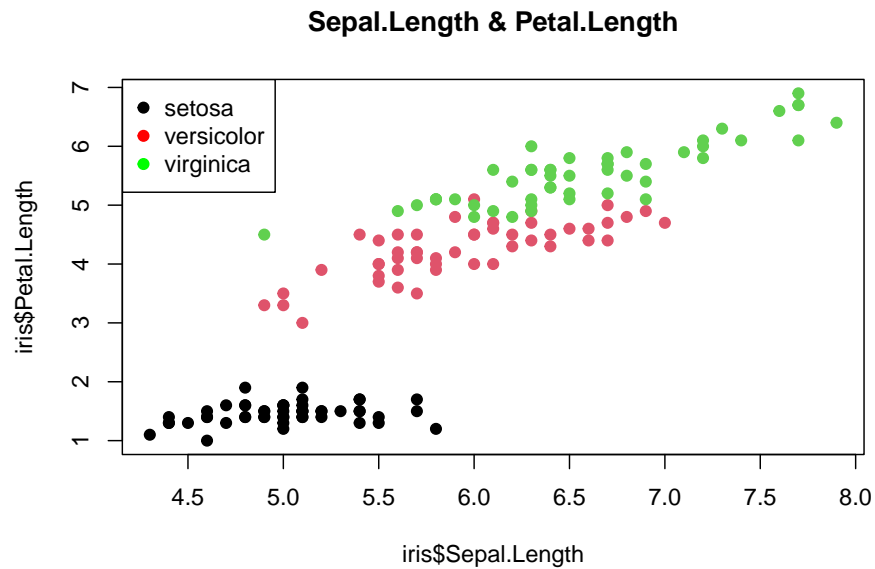


如下图所示：

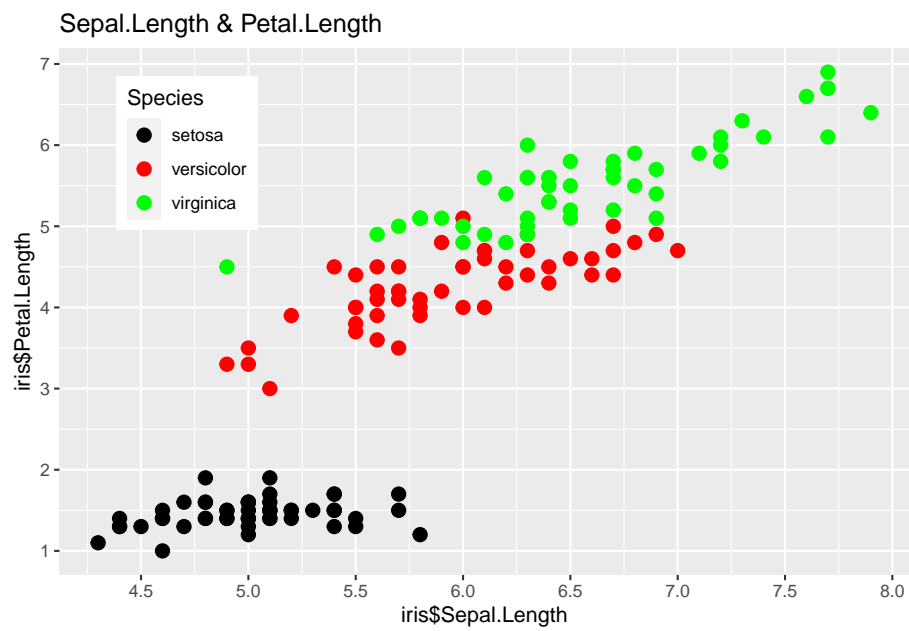
2. 用 boxplot 显示 species 之间 Sepal.Length 的分布情况；

注：每种图提供基础做图函数和 ggplot2 两个版本！

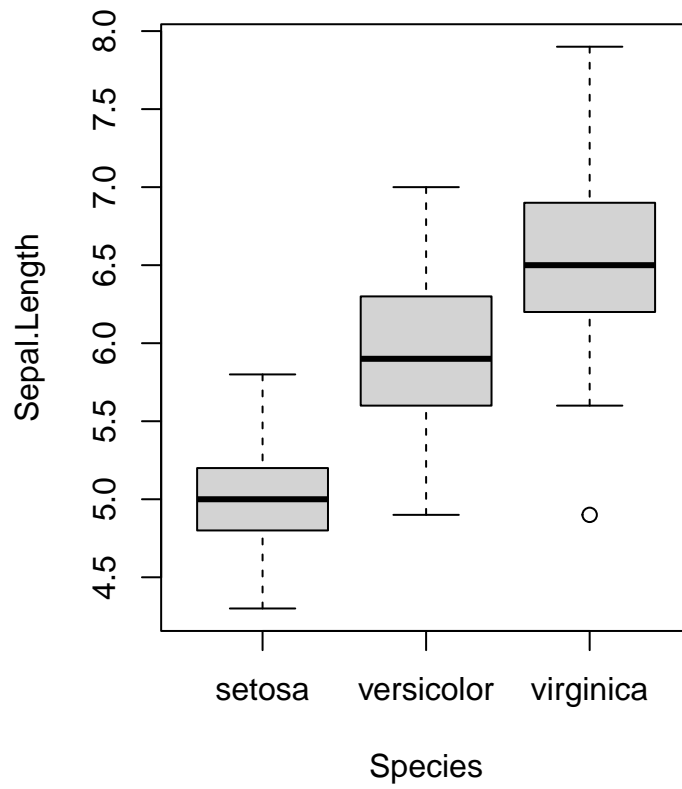
```
## 代码写这里，并运行；  
## 用散点图显示 `Sepal.Length` 和 `Petal.Length` 之间的关系；按 `species` 为散点确定颜色，  
## 基础做图函数  
with(iris, plot( Sepal.Length, Petal.Length, type = "p",  
                main = "Sepal.Length & Petal.Length", col = Species,  
                xlab = "iris$Sepal.Length", ylab = "iris$Petal.Length",  
                pch = 19));  
legend("topleft", legend = c("setosa", "versicolor", "virginica"),  
      col = c("black", "red", "green"), pch = 19);
```



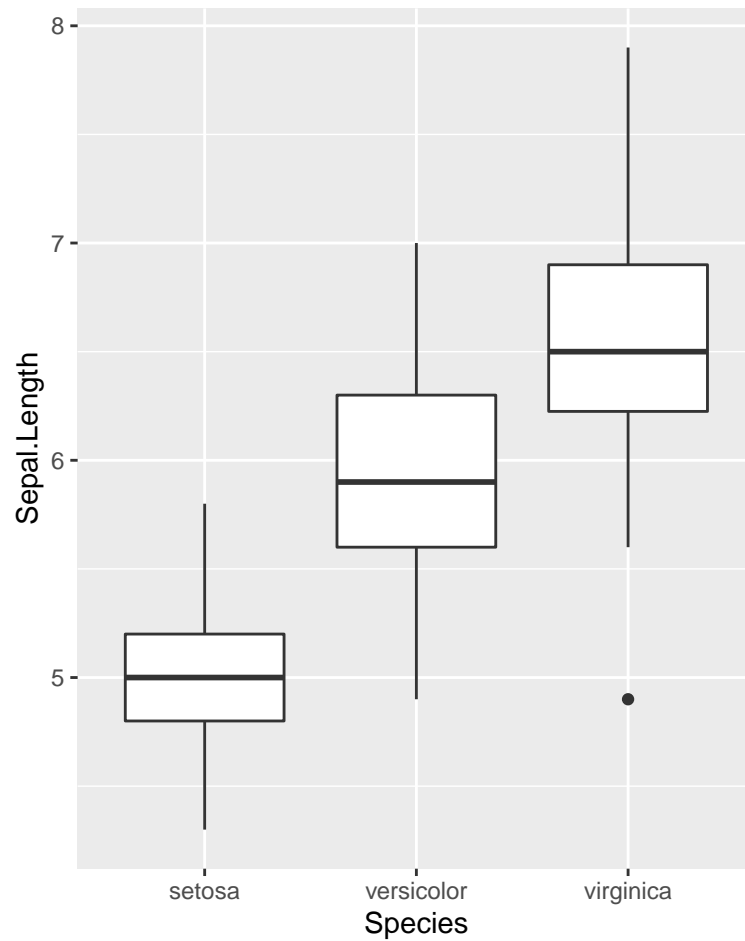
```
##ggplot2
ggplot(iris, aes(x = Sepal.Length, y = Petal.Length, colour = Species)) +
  scale_color_manual( breaks = c("setosa","versicolor","virginica"),
                      values = c("black","red","green") ) +
  xlab( "iris$Sepal.Length" ) + ylab( "iris$Petal.Length" ) +
  ggtitle("Sepal.Length & Petal.Length") +
  geom_point(size = 3) + theme( legend.position = c(0.15,0.8) ) +
  scale_x_continuous(breaks = seq(4.5, 8, 0.5)) +
  scale_y_continuous(breaks = seq(1, 7, 1));
```

```
## 代码写这里，并运行；  
## 用 boxplot 显示`species`之间 `Sepal.Length` 的分布情况  
## 基础做图函数  
with(iris, boxplot(Sepal.Length ~ Species))
```



```
##ggplot2  
ggplot(iris, aes( x = Species, y = Sepal.Length)) + geom_boxplot()
```



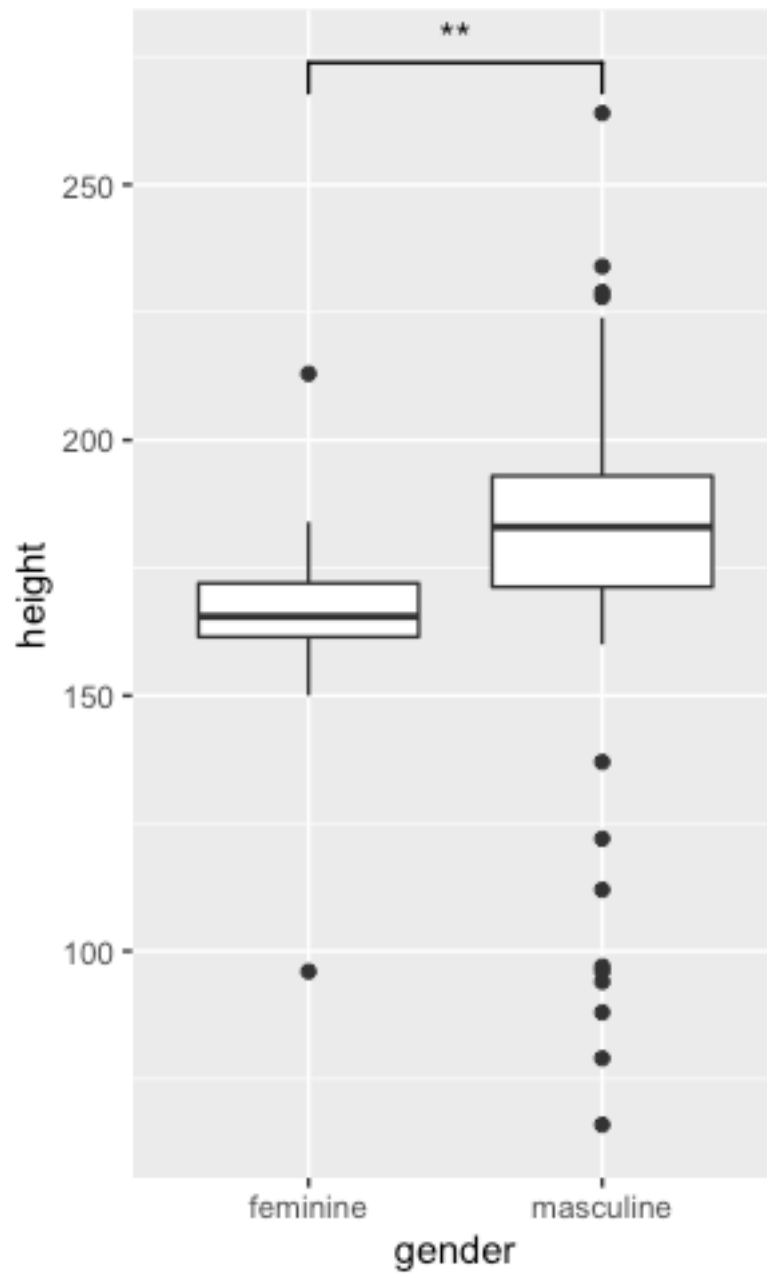
0.4.3 用 ggplot 作图: boxplot

用 `starwars` 的数据作图，画 boxplot 显示身高 `height` 与性别 `gender` 的关系。要求：

1. `height` 为 NA 的，不显示；
2. 用 `ggsignif` 包计算 `feminine` 和 `masculine` 两种性别的身高是否有显著区别，并在图上显示。

3. 将此图的结果保存为变量 **p1**，以备后面使用；

最终结果如图所示：

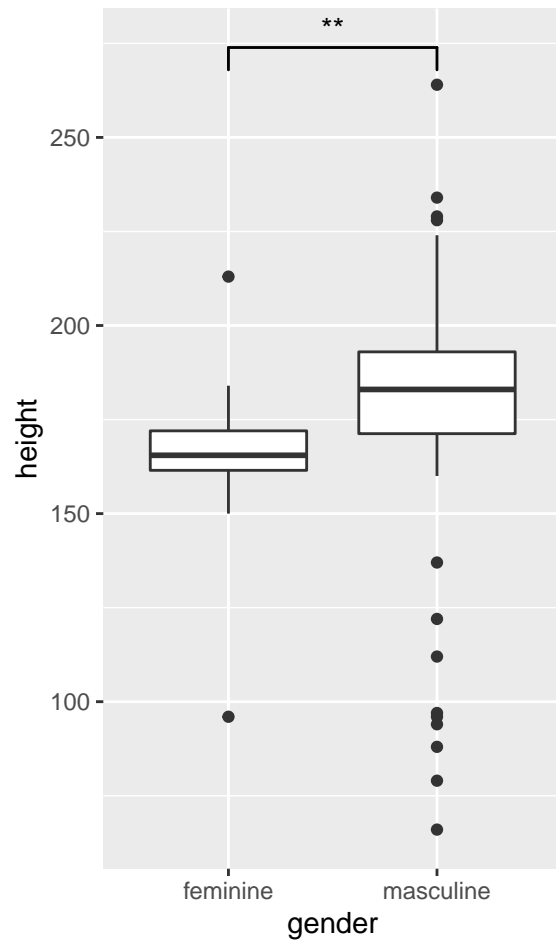


```
## 代码写这里，并运行；  
## 用 `starwars` 的数据作图，画 `boxplot` 显示身高 `height` 与性别 `gender` 的关系  
library(ggsignif);  
p1 <- ggplot(starwars, aes(x = gender, y = height )) +  
  geom_boxplot( ) + scale_x_discrete( na.translate = FALSE ) +  
  geom_signif(comparisons = list(c("feminine", "masculine")), map_signif_level = T );  
p1;
```

```
## Warning: Removed 4 rows containing missing values (stat_boxplot).
```

```
## Warning: Removed 5 rows containing non-finite values (stat_boxplot).
```

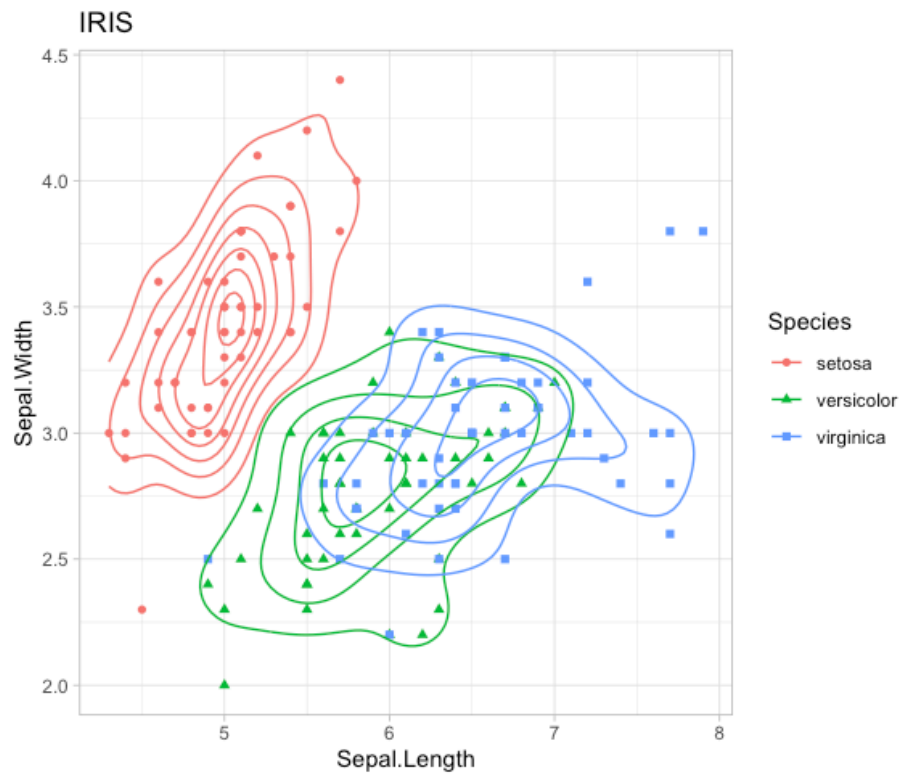
```
## Warning: Removed 9 rows containing non-finite values (stat_signif).
```



0.4.4 用 ggplot 作图：使用 iris 做图

用 `geom_density2d` 显示 `Sepal.Length` 和 `Sepal.Width` 之间的关系，同时以 `Species` 为分组，结果如图所示：

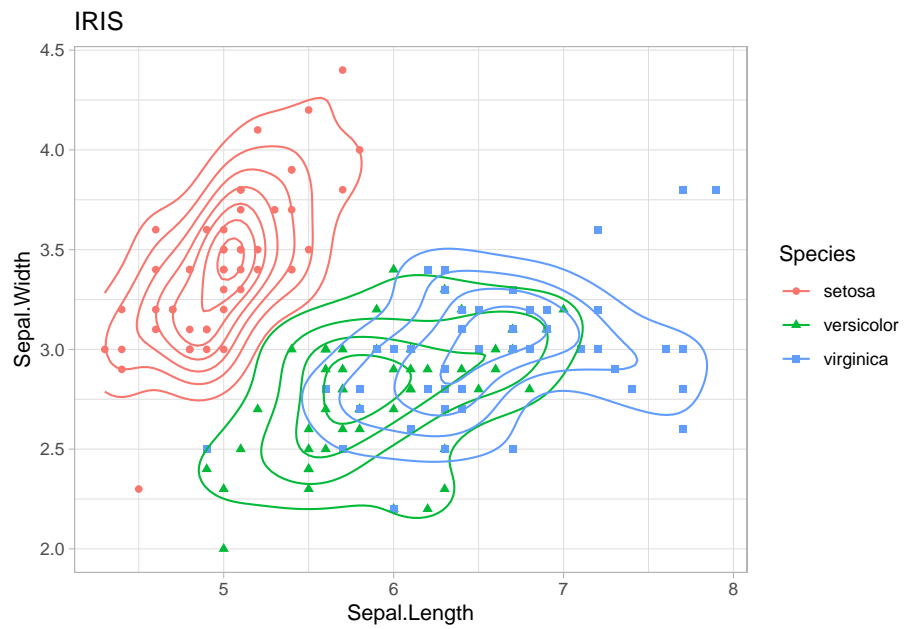
将此图的结果保存为变量 `p2`，以备后面使用：



```
## 代码写这里，并运行；
```

```
## 用`geom_density2d`显示`Sepal.Length`和`Sepal.Width`之间的关系，同时以`Species`为分组
```

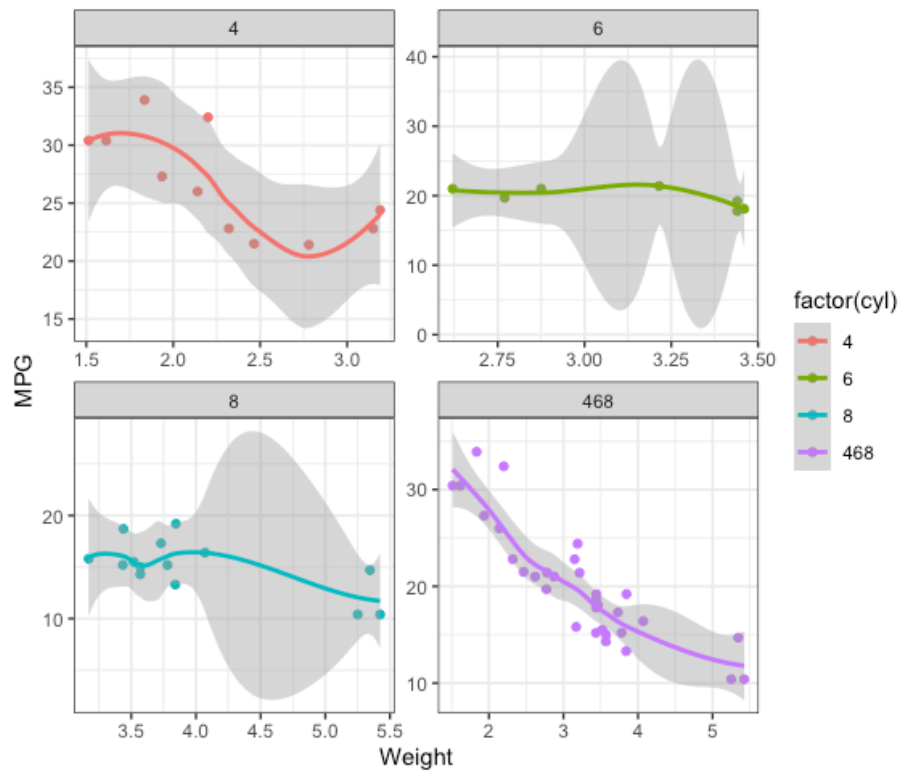
```
p2 <- ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width,  
                      colour = Species, shape = Species)) +  
  geom_point() + geom_density2d() + theme_light() + ggtitle("IRIS");  
p2;
```



0.4.5 用 ggplot 作图: facet

用 `mtcars` 作图，显示 `wt` 和 `mpg` 之间的关系，但用 `cyl` 将数据分组：见下图：

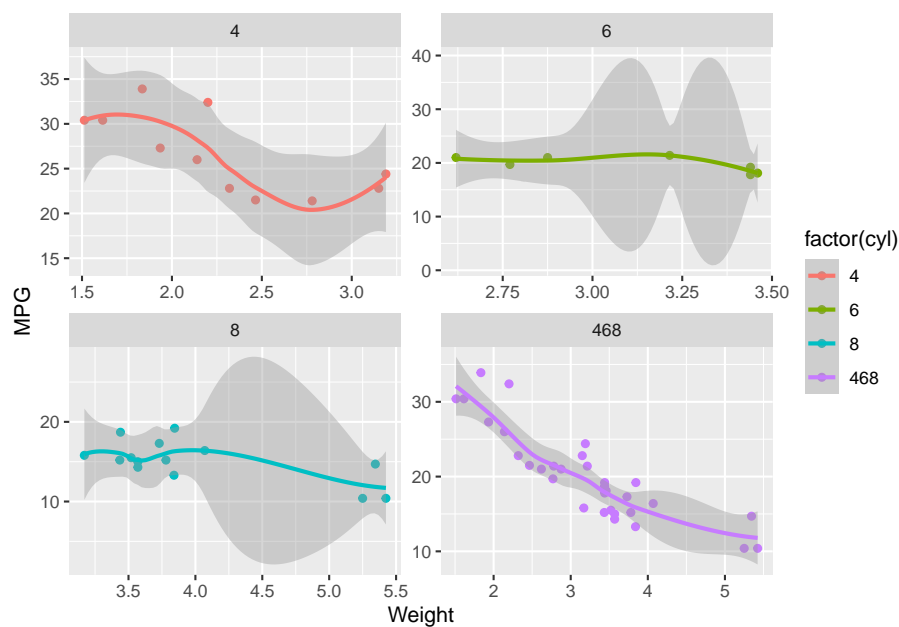
将此图的结果保存为变量 `p3`，以备后面使用；



注此图中的 468 组为所有数据合在一起的结果。

```
## 代码写这里，并运行；
mtcars1 <- mtcars;
mtcars1[["cyl"]] <- 468;
mtcars2 <- bind_rows(mtcars, mtcars1);
p3 <- ggplot(mtcars2, aes(x = wt, y = mpg, colour = factor(cyl))) + geom_point() +
  geom_smooth( se = T ) + xlab( "Weight" ) + ylab( "MPG" ) +
  facet_wrap( .~ cyl, ncol = 2, scales = "free", dir = "h" )
p3;
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

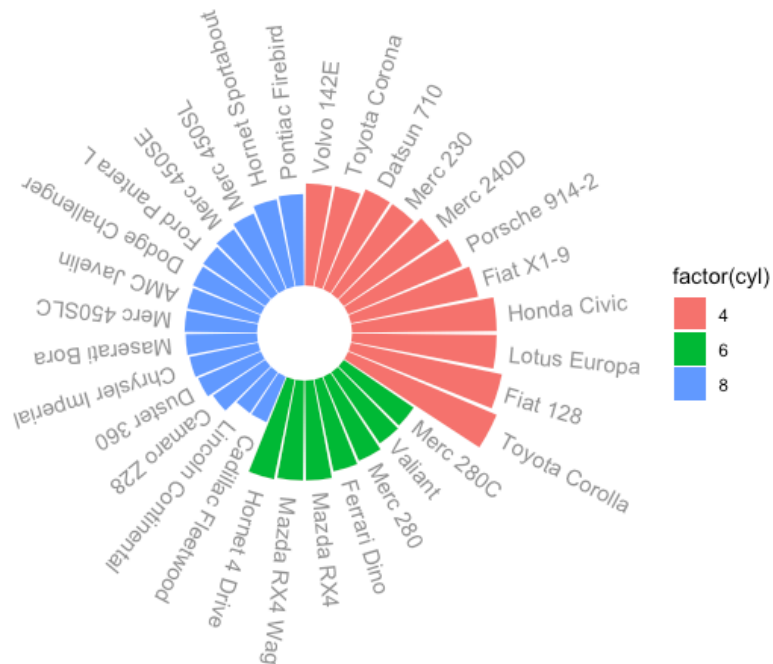


0.4.6 用 ggplot 作图：用 mtcars 做 polar 图

用 mtcars 的 mpg 列做如下图，要求：先按 cyl 排序；每个 cyl 组内按 mpg 排序；将此图的结果保存为变量 p4，以备后面使用；

提示

1. 先增加一列,用于保存 rowname: `mtcars %>% rownames_to_column()`
注：将行名变为列，列名为 rowname
2. 完成排序
3. 更改 rowname 的 factor
4. 计算每个 rowname 的旋转角度: `mutate(id = row_number(),
angle = 90 - 360 * (id - 0.5) / n())`

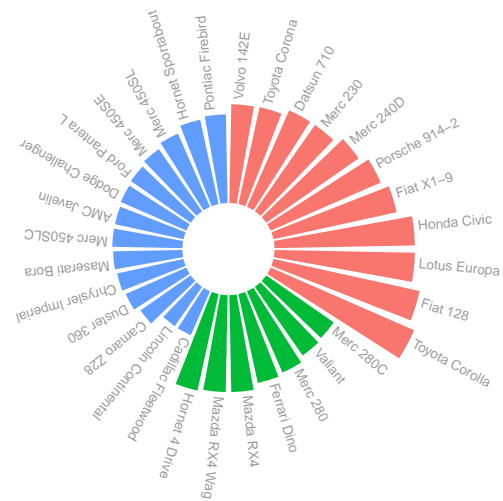


代码写这里，并运行；

```
m <- mtcars %>% rownames_to_column() %>%
  arrange(cyl, mpg) %>%
  mutate( id = row_number(), angle = 90 - 360 * (id - 0.5) / n());
p4 <- ggplot(m, aes(x = factor(m$rowname, levels = rev(m$rowname)),
                    y = mpg, fill = factor(cyl))) +
  theme_void() + ylim(-10,45) + coord_polar(direction = -1) +
  geom_bar(stat = "identity", width = 0.8) +
  geom_text( aes(x = rowname, y = mpg + 1, label = rowname, hjust = 0,vjust = 0),
            size = 2.5, alpha = 0.4, color = "black", angle = m$angle)
p4
```

Warning: Use of `m\$rowname` is discouraged. Use `rowname` instead.

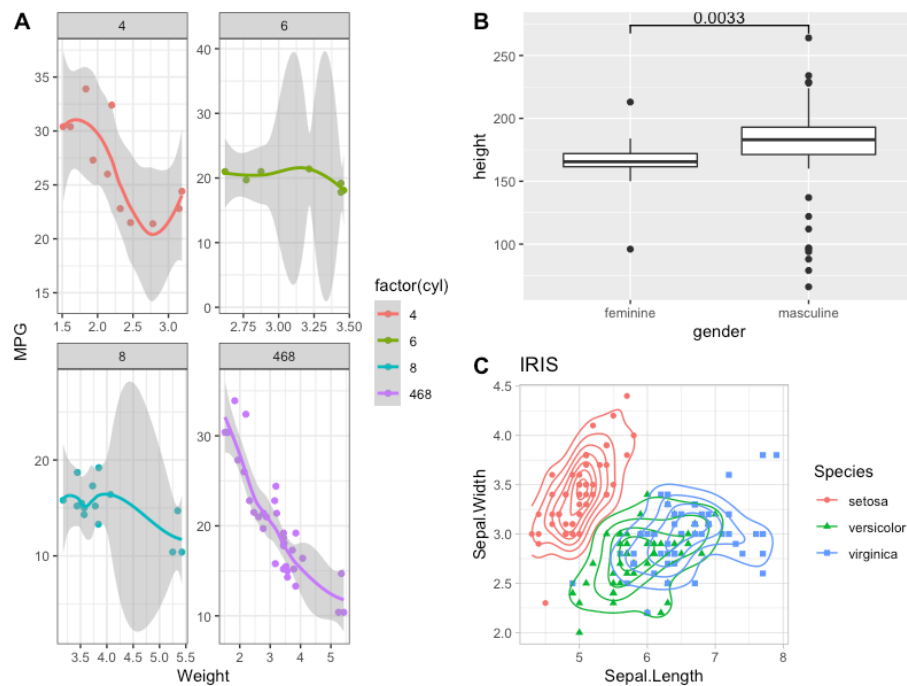
Use of `m\$rowname` is discouraged. Use `rowname` instead.



0.5 练习与作业 2：多图组合，将多个图画在一起

0.5.1 用 `cowplot::ggdraw` 将 p1, p2 和 p3 按下面的方式组合在一起

注：需要先安装 `cowplot` 包



```
## 代码写这里，并运行；
library(cowplot);
gg <- ggdraw() +
  draw_plot(p3, 0, 0, 0.5, 1) + draw_plot(p1, 0.5, 0.5, 0.5, 0.5) +
  draw_plot(p2, 0.5, 0, 0.5, 0.5) +
  draw_plot_label(c("A", "B", "C"), c(0, 0.5, 0.5), c(1, 1, 0.5),
    size = 15, colour = "black")
```

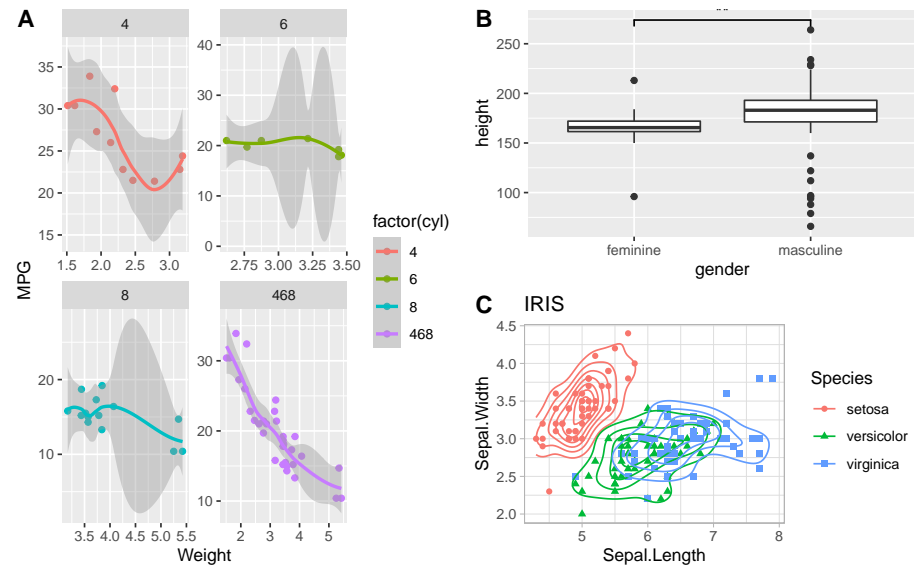
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

```
## Warning: Removed 4 rows containing missing values (stat_boxplot).
```

```
## Warning: Removed 5 rows containing non-finite values (stat_boxplot).
```

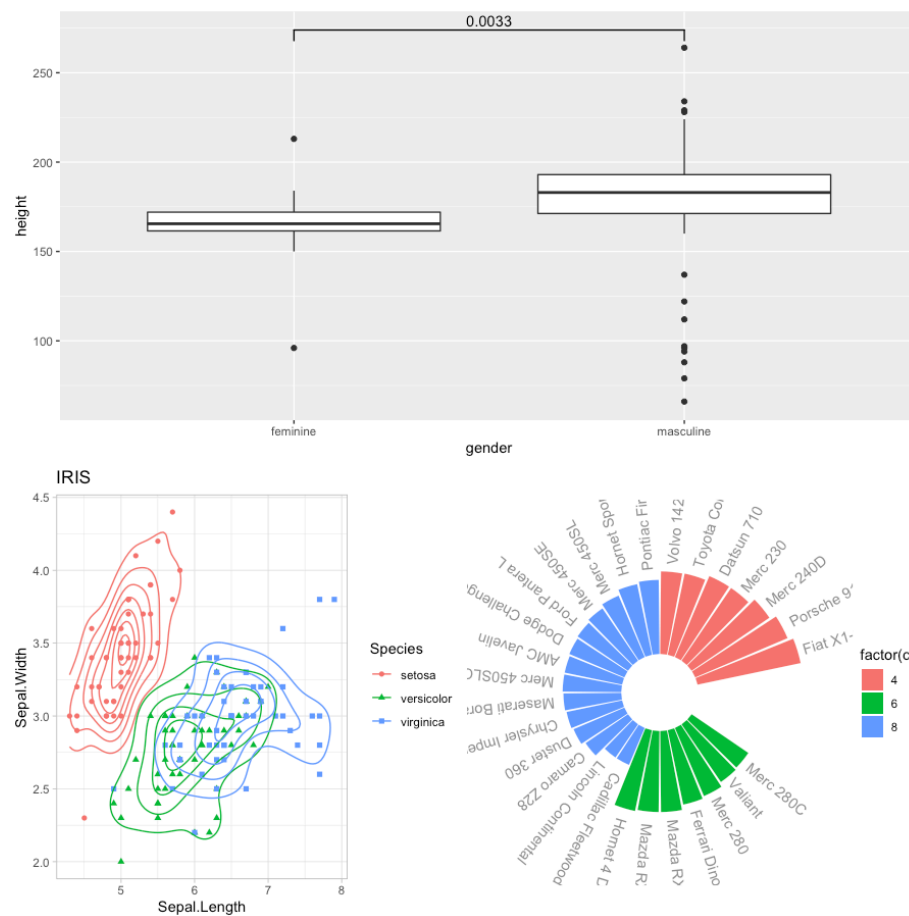
```
## Warning: Removed 9 rows containing non-finite values (stat_signif).
```

```
gg;
```



0.5.2 用 `gridExtra::grid.arrange()` 函数将 p1, p2, p4 按下面的方式组合在一起

注：需要安装 `gridExtra` 包；



```
## 代码写这里，并运行；
```

```
library(gridExtra);
```

```
##
```

```
## 载入程辑包: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
```

```
##
```

```
## combine
```

```
grid.arrange(p1, arrangeGrob(p2, p4, ncol=2))
```

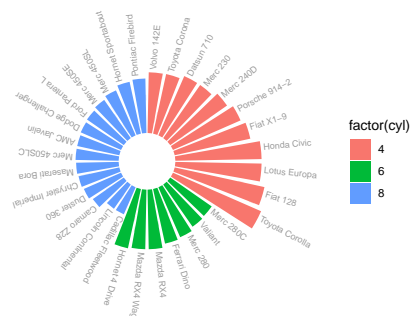
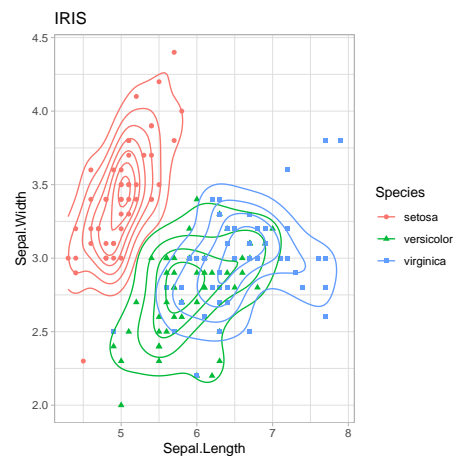
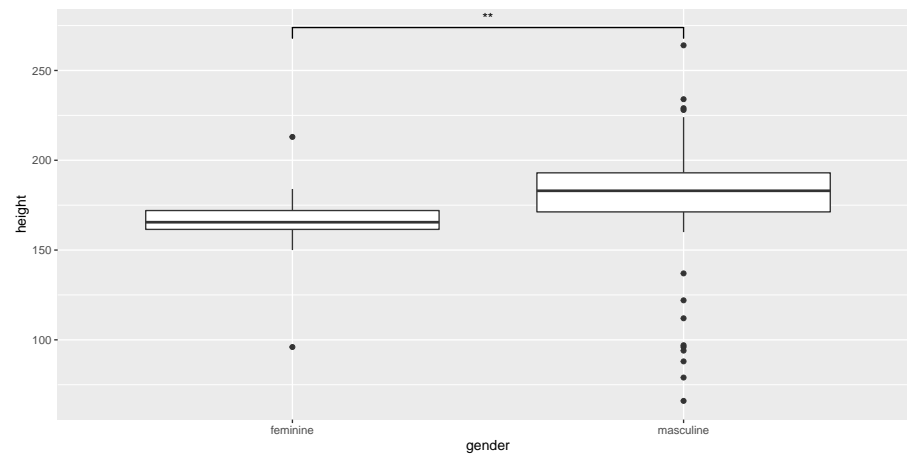
```
## Warning: Use of `m$rowname` is discouraged. Use `rowname` instead.
```

```
## Use of `m$rowname` is discouraged. Use `rowname` instead.
```

```
## Warning: Removed 4 rows containing missing values (stat_boxplot).
```

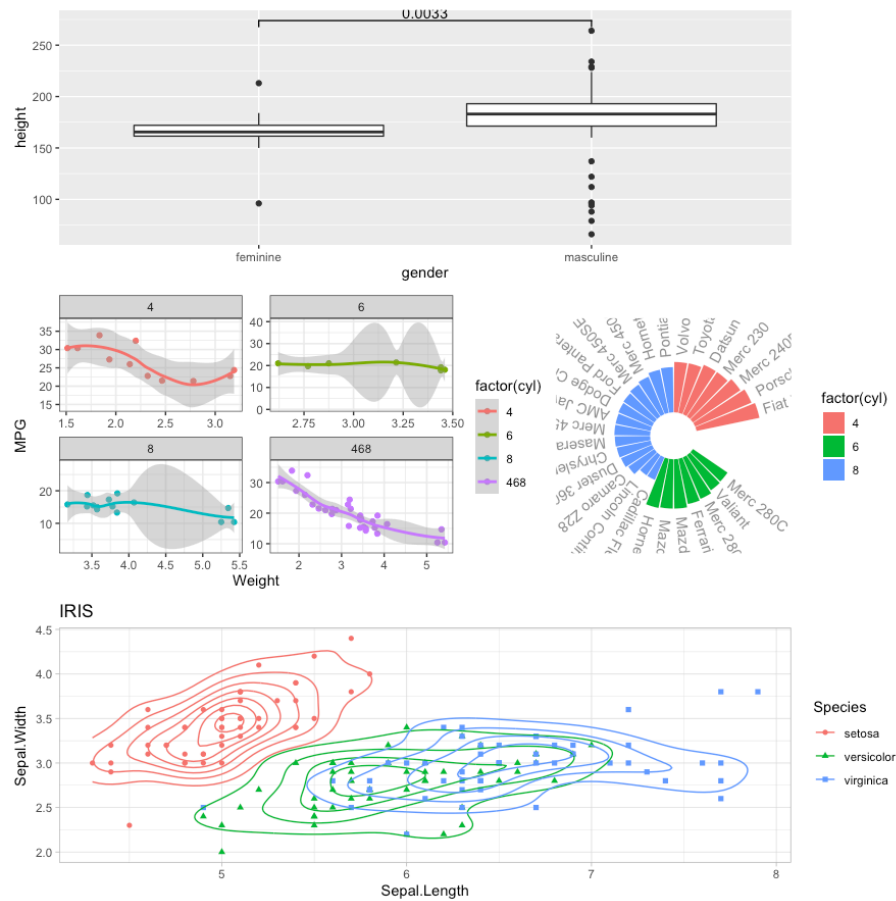
```
## Warning: Removed 5 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 9 rows containing non-finite values (stat_signif).
```



0.5.3 用 patchwork 包中的相关函数将 p1, p2, p3, p4 按下面的方式组合在一起

注：需要安装 patchwork 包；



```
## 代码写这里，并运行；
```

```
library(patchwork);
```

```
##
```

```
## 载入程辑包：'patchwork'
```

```
## The following object is masked from 'package:cowplot':
```

```
##
```

```
##      align_plots
```

```
p1 / (p3 + p4) / p2;
```

```
## Warning: Removed 4 rows containing missing values (stat_boxplot).
```

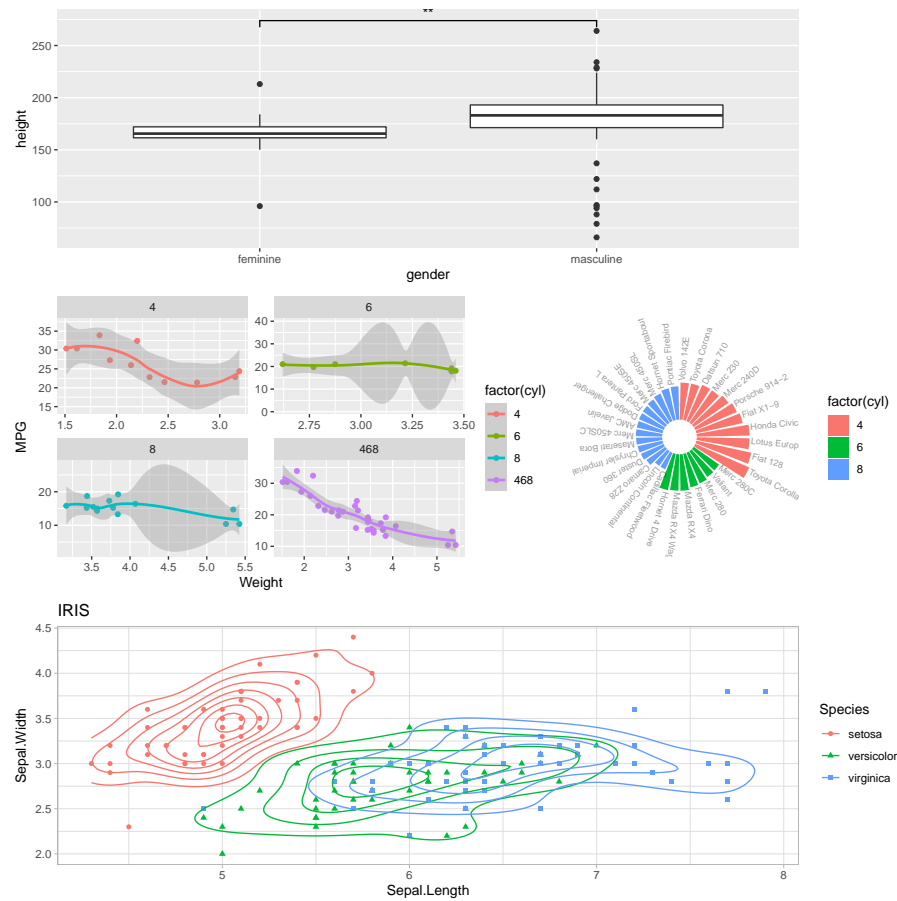
```
## Warning: Removed 5 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 9 rows containing non-finite values (stat_signif).
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

```
## Warning: Use of `m$rowname` is discouraged. Use `rowname` instead.
```

```
## Warning: Use of `m$rowname` is discouraged. Use `rowname` instead.
```



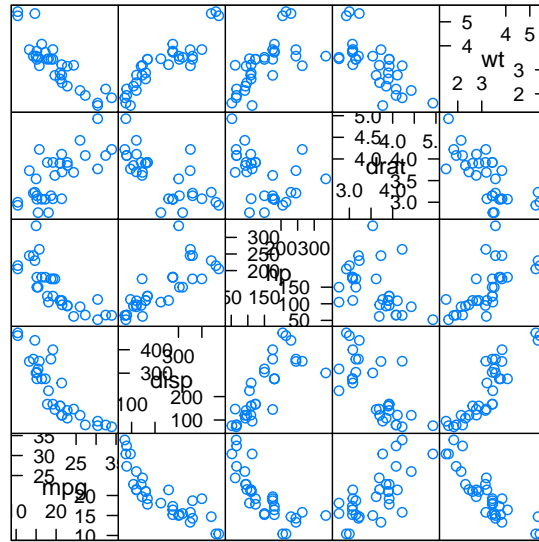
0.6 练习与作业 3：作图扩展

0.6.1 scatterplot

安装 `lattice` 包，并使用其 `splom` 函数作图：

```
lattice::splom( mtcars[c(1,3,4,5,6)] )
```

```
## 代码写这里，并运行；  
library(lattice);  
splom( mtcars[c(1,3,4,5,6)] );
```



Scatter Plot Matrix