talk10 练习与作业

目录

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-talk10 作业.pdf`，并提交到老师指定的平台/钉群。

0.2 Talk10 内容回顾

- data summarisation functions (vector data)
  - median, mean, sd, quantile, summary
- 图形化的 data summarisation (two-D data/ tibble/ table)
  - dot plot

  - smooth
  - linear regression
  - correlation & variance explained
  - groupping & bar/ box/ plots

- statistics

  - parametric tests

    * t-test
    * one way ANNOVA
    * two way ANNOVA
    * linear regression
    * model / prediction / coefficients

  - non-parametric comparison

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] "lucas"
```

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

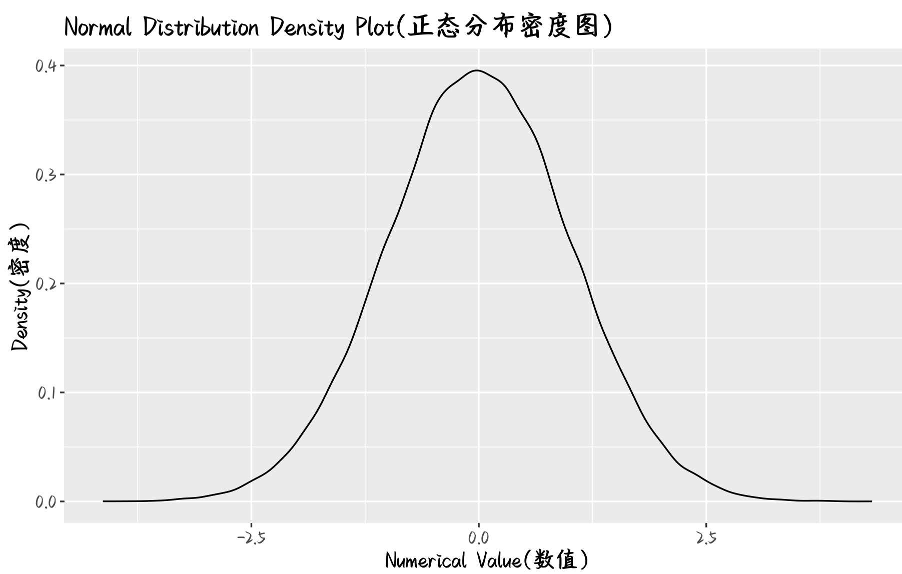
```
## [1] "/Users/lucas"
```

0.4 练习与作业 1：数据查看

- **正态分布**

1. 随机生成一个数字（numberic）组成的 vector，长度为 10 万，其值符合正态分布；
2. 用 ggplot2 的 density plot 画出其分布情况；
3. 检查 mean +- 1 * sd，mean +- 2 * sd 和 mean +- 3 * sd 范围内的取值占总值数量的百分比。

```
## 代码写这里，并运行；
library(ggplot2)

# Setting random seeds to ensure reproducibility
set.seed(123)
data = rnorm(100000)
df = data.frame(value = data)

# Creating density maps with ggplot2
ggplot(df, aes(x = data)) +
  geom_density() +
  # Prevent the GBK character to show as block
  theme(
    text=element_text(
      family="RLQDMSWR",
      size=14)) +
  labs(
    title = "Normal Distribution Density Plot(正态分布密度图)",
    x = "Numerical Value(数值)",
    y = "Density(密度)"
  )
```




```
# Calculate the percentage of the total number
# of values taken in different ranges:
mean_value = mean(data)
sd_value = sd(data)

# Calculate the percentage in the range mean +- 1 * sd
within_1_sd =
  sum(data >= mean_value - sd_value &
        data <= mean_value + sd_value) / length(data)

# Calculate the percentage in the range mean +- 2 * sd
within_2_sd =
  sum(data >= mean_value - 2 * sd_value &
        data <= mean_value + 2 * sd_value) / length(data)

# Calculate the percentage in the range mean +- 3 * sd
within_3_sd =
  sum(data >= mean_value - 3 * sd_value &
        data <= mean_value + 3 * sd_value) / length(data)
```

```r
# Convert results to percentages
within_1_sd * 100
```

```
## [1] 68.149
```

```r
within_2_sd * 100
```

```
## [1] 95.463
```

```r
within_3_sd * 100
```

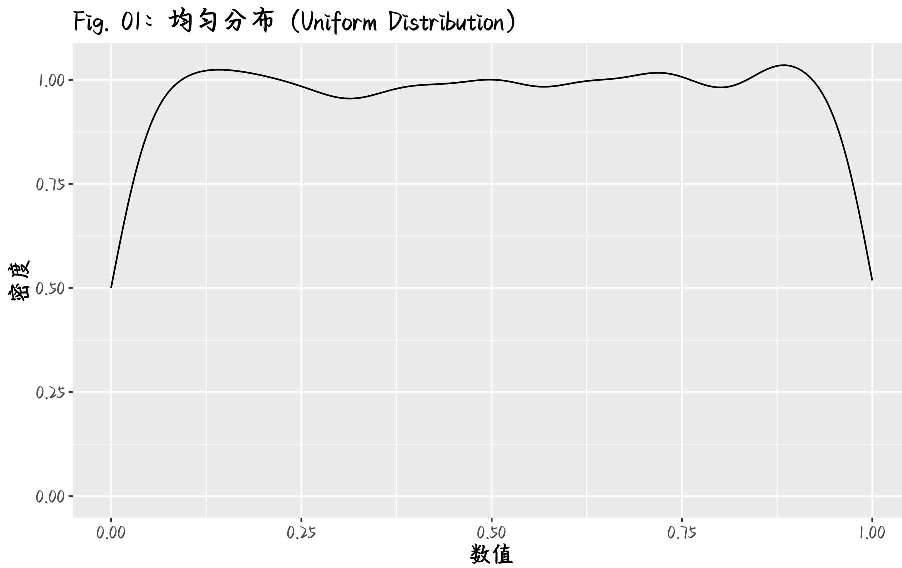
```
## [1] 99.735
```

- **用函数生成符合以下分布的数值，并做图：**

另外，在英文名后给出对应的中文名：

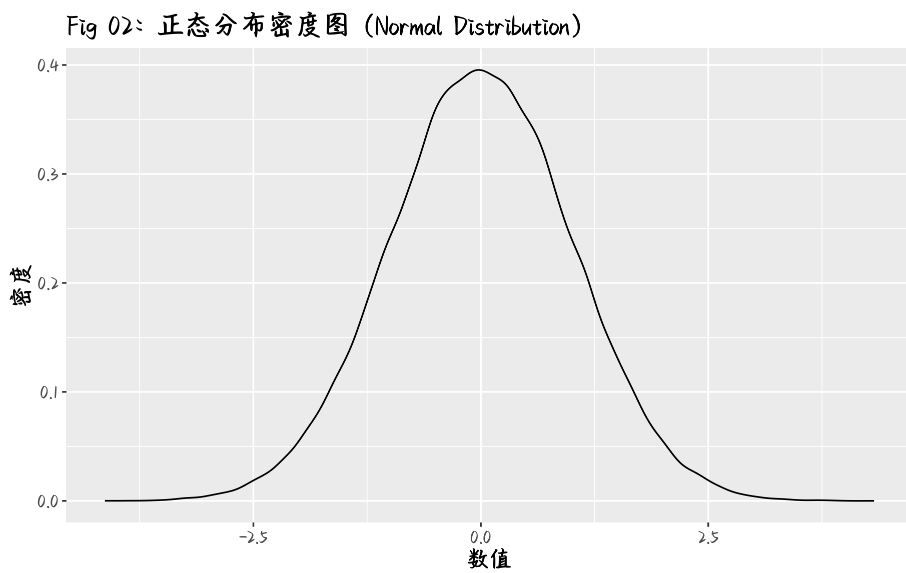
```
-   Uniform Distribution

-   Normal Distribution

-   Binomial Distribution

-   Poisson Distribution

-   Exponential Distribution

-   Gamma Distribution
```

```r
## 代码写这里，并运行;
# Task 01: Uniform Distribution
# Generate uniformly distributed random values
uniform_data =
  runif(10000,
        min = 0,
        max = 1)

# Creating Density Maps
ggplot(
  data.frame(
    x = uniform_data),
  aes(x)) +
  geom_density() +
  # Prevent the GBK character to show as block
  theme(
    text=element_text(
      family="RLQDMSWR",
      size=14)) +
  labs(
    title = "Fig. 01: 均匀分布 (Uniform Distribution)",
    x = " 数值",
    y = " 密度")
```



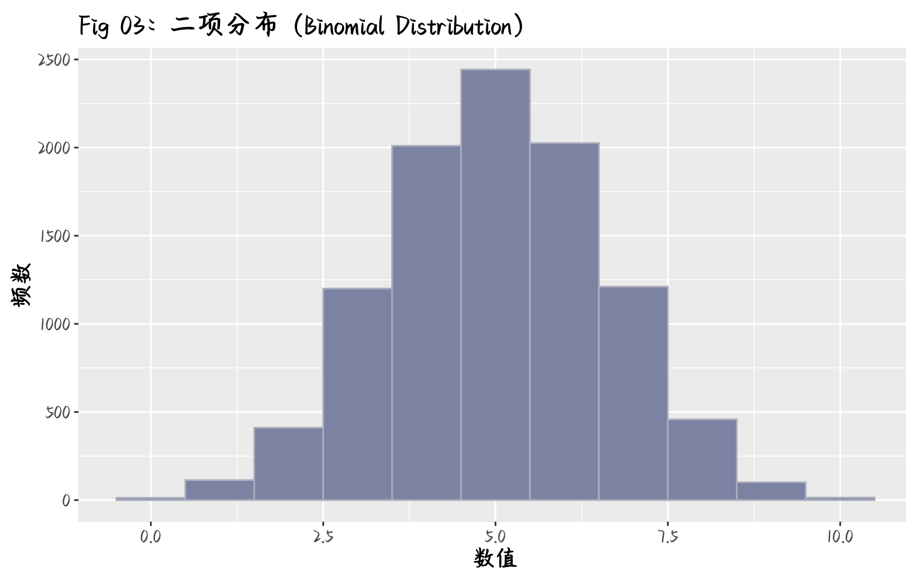

```r
# Task 02: Normal Distribution
# Setting random seeds to ensure reproducibility
set.seed(123)

# Generate normally distributed random values
normal_data =
  rnorm(100000)

# Creating data.frame
normal_df =
  data.frame(value = normal_data)

# Creating Density Plots with ggplot2
library(ggplot2)

ggplot(normal_df,
       aes(x = value)) +
  geom_density() +
  labs(
```

```
    title = "Fig 02: 正态分布密度图 (Normal Distribution)",
    x = " 数值",
    y = " 密度") +
  # Prevent the GBK character to show as block
  theme(
    text = element_text(
      family = "RLQDMSWR",
      size = 14))
```



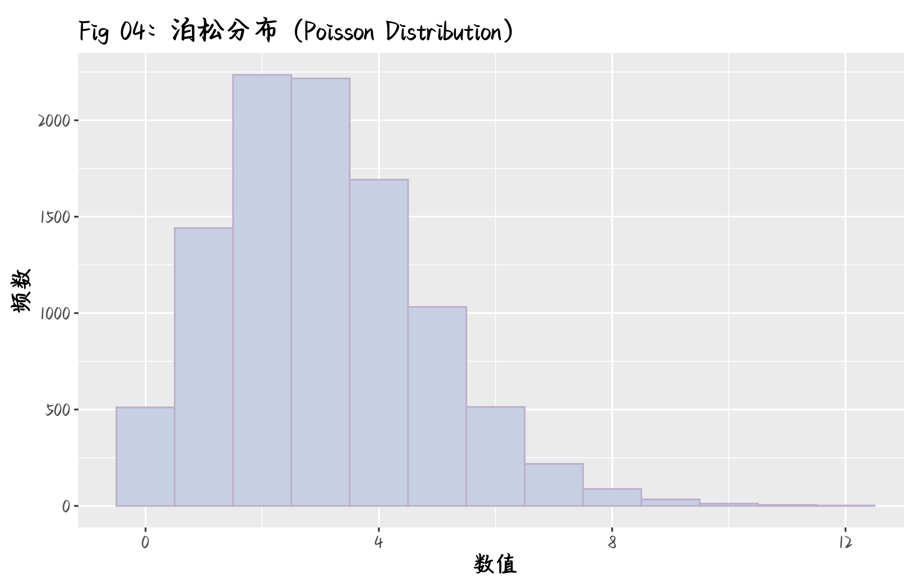

```
# Task 03: Binomial Distribution
# Generate random values for binomial distribution
binomial_data =
  rbinom(10000, size = 10, prob = 0.5)

# Creating Histograms
ggplot(
  data.frame(
    x = binomial_data),
  aes(x)) +
```

```r
  geom_histogram(
    binwidth = 1,
    fill = "#7C82A2",
    color = "#ABADBC") +
  labs(
    title = "Fig 03: 二项分布 (Binomial Distribution)",
    x = " 数值",
    y = " 频数") +
  # Prevent the GBK character to show as block
  theme(
    text = element_text(
      family = "RLQDMSWR",
      size = 14))
```

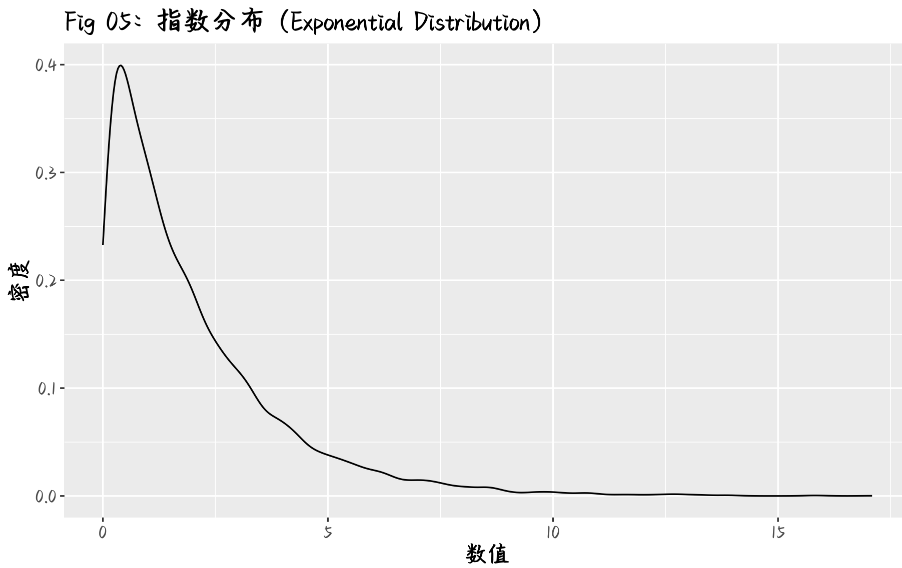



```r
# Task 04: Poisson Distribution
# Generate random values for poisson distribution
poisson_data =
  rpois(10000, lambda = 3)
```

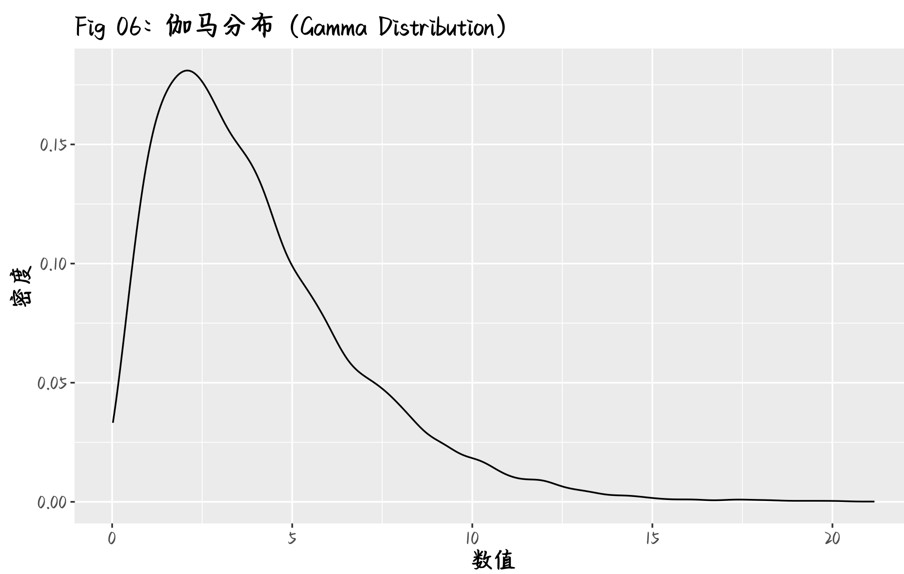
```r
# Creating Histograms
ggplot(
  data.frame(
    x = poisson_data),
  aes(x)) +
  geom_histogram(
    binwidth = 1,
    fill = "#C7CFE4",
    color = "#BDAFCA") +
  labs(
    title = "Fig 04: 泊松分布 (Poisson Distribution)",
    x = " 数值",
    y = " 频数") +
  # Prevent the GBK character to show as block
  theme(
    text = element_text(
      family = "RLQDMSWR",
      size = 14))
```



```
# Task 05: Exponential Distribution
# Generate random values for exponential distribution
exponential_data =
  rexp(10000, rate = 0.5)

# Generate density plot
ggplot(
  data.frame(
    x = exponential_data),
  aes(x)) +
  geom_density() +
  labs(
    title = "Fig 05: 指数分布 (Exponential Distribution)",
    x = " 数值",
    y = " 密度") +
  # Prevent the GBK character to show as block
  theme(
    text = element_text(
      family = "RLQDMSWR",
      size = 14))
```




```
# Task 06: Gamma Distribution
# Generate random values for gamma distribution
gamma_data =
  rgamma(10000, shape = 2, rate = 0.5)

# Generate density plot
ggplot(
  data.frame(
    x = gamma_data),
  aes(x)) +
  geom_density() +
  labs(
    title = "Fig 06: 伽马分布 (Gamma Distribution)",
    x = " 数值",
    y = " 密度") +
  # Prevent the GBK character to show as block
  theme(text = element_text(
    family = "RLQDMSWR",
    size = 14))
```



- **分组的问题**
  - 什么是 `equal-sized bin` 和 `equal-distance bin`? 以 `mtcars` 为例，将 `wt` 列按两种方法分组，并显示结果。

Answer:

- Equal-sized bin (等宽分组)：在这种分组方法中，数据被分成具有相同宽度或大小的区间（bin）或组。这意味着每个分组具有相同数量的数据点，但这可能导致一些分组的范围内包含更多或更少的数据点，具体取决于数据的分布。
- Equal-distance bin (等距分组)：这种分组方法将数据按照等距离间隔的方式分成分组。区间的宽度可能不同，但每个区间都覆盖了相等的数值范围。这意味着每个分组的数据点的范围是相等的，但每个分组可能包含不同数量的数据点，具体取决于数据的分布。

```r
## 代码写这里，并运行;
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
# Use equal-width grouping to
# divide wt columns into 5 groups
mtcars_equal_width =
  mtcars %>%
  mutate(
    wt_group_equal_width =
      cut(wt,
          breaks = 5,
          labels = FALSE))

head(mtcars_equal_width)
```

```
##                    mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

```
##                    wt_group_equal_width
## Mazda RX4                             2
## Mazda RX4 Wag                         2
## Datsun 710                            2
## Hornet 4 Drive                        3
## Hornet Sportabout                     3
## Valiant                               3
```

```
# Use isometric grouping to
# divide wt columns into 5 groups
mtcars_equal_distance =
  mtcars %>%
  mutate(
    wt_group_equal_distance =
      cut_interval(wt,
                   n = 5))

head(mtcars_equal_distance)
```

```
##                    mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
##                   wt_group_equal_distance
## Mazda RX4                      (2.3,3.08]
## Mazda RX4 Wag                  (2.3,3.08]
## Datsun 710                     (2.3,3.08]
## Hornet 4 Drive                (3.08,3.86]
## Hornet Sportabout             (3.08,3.86]
## Valiant                       (3.08,3.86]
```

- **boxplot 中 outlier 值的鉴定**
  - 以 swiss$Infant.Mortality 为例，找到它的 outlier 并打印出来；

```
## 代码写这里，并运行；
# Load the data
data(swiss)

# Select the columns to look for outliers
infant_mortality =
  swiss$Infant.Mortality

# Calculate quartiles
Q1 =
  quantile(infant_mortality, 0.25)
Q3 =
  quantile(infant_mortality, 0.75)

# Calculate IQR
IQR = Q3 - Q1

# Calculate upper and lower limit
upper_limit =
  Q3 + 1.5 * IQR
lower_limit =
  Q1 - 1.5 * IQR

# Find the outlier
outliers =
  infant_mortality[
    infant_mortality > upper_limit |
```

```
      infant_mortality < lower_limit]

# 打印异常值
cat(
  "Infant Mortality Outlier(s):",
  outliers,
  "\n")
```

```
## Infant Mortality Outlier(s): 10.8
```

- **以男女生步数数据为例，进行以下计算：**

首先用以下代码装入 Data:

```
source("../data/talk10/input_data1.R"); ## 装入 Data data.frame ...
head(Data);
```

```
##   Student    Sex Teacher Steps Rating
## 1       a female  Catbus  8000      7
## 2       b female  Catbus  9000     10
## 3       c female  Catbus 10000      9
## 4       d female  Catbus  7000      5
## 5       e female  Catbus  6000      4
## 6       f female  Catbus  8000      8
```

- 分别用`t.test`和`wilcox.test`比较男女生步数是否有显著差异；打印出`p.value`

```
## 代码写这里，并运行；
data_df = as.data.frame(Data)
rm(Data)
```

```r
# Task 01:
# Use t.test to compare
# whether there is a significant difference
# in the number of steps
# taken by male and female students
t_test_result = t.test(Steps ~ Sex, data = data_df)
cat("t-test p.value:", t_test_result$p.value, "\n")
```

```
## t-test p.value: 0.01461209
```

```r
# Task 02:
# Using wilcox.test to compare
# whether there is a significant difference
# in the number of steps
# taken by male and female students
wilcox_test_result = wilcox.test(Steps ~ Sex, data = data_df)
cat("Wilcoxon rank sum test p.value:", wilcox_test_result$p.value, "\n")
```

```
## Wilcoxon rank sum test p.value: 0.01773304
```

- 两种检测方法的`p.value`哪个更显著？为什么？

答：

在统计学中，`t.test` 和 `wilcox.test` 是两种不同的检验方法，用于比较两个样本之间的差异。它们的显著性水平（`p-value`）的解释方式有所不同，而哪个更显著取决于数据的性质和问题的背景。

1. `t.test`:
    - `t.test` 是一种用于比较两组数值型数据之间平均值差异的方法。
    - 当两组数据来自正态分布的总体时，通常可以使用 `t.test`。
    - `p-value` 衡量了两组数据的平均值之间是否存在显著差异。较小的 `p-value` 表示更显著的差异。

2. `wilcox.test`:

    - `wilcox.test` 是一种非参数检验方法，用于比较两组数据的中位数差异。
    - 这个方法适用于数据不满足正态分布假设的情况，或者数据的中位数差异比平均值差异更值得关注时。
    - `p-value` 衡量了两组数据的中位数之间是否存在显著差异。较小的 `p-value` 表示更显著的差异。

哪个方法的 `p-value` 更显著取决于所研究的数据和问题。

- 如果数据满足 `t.test` 的假设，且平均值差异更重要，那么 `t.test` 可能会产生更显著的 `p-value`。
- 如果数据不满足 `t.test` 的假设，或者中位数差异更重要，那么 `wilcox.test` 可能会产生更显著的 `p-value`。

  (Based on ChatGPT-3.5)

- **以下是学生参加辅导班前后的成绩情况，请计算同学们的成绩是否有普遍提高?**

注：先用以下代码装入数据:

```
source("../data/talk10/input_data2.R");
head(scores);
```

```
##     Time Student Score
## 1 Before       a    65
## 2 Before       b    75
## 3 Before       c    86
## 4 Before       d    69
## 5 Before       e    60
## 6 Before       f    81
```

注：计算时请使用 paired = T 参数；

```
## 代码写这里，并运行；
score_df = as.data.frame(scores)
rm(scores)

# Execute paired t-test
t_test_result =
  t.test(Score ~ Time,
         data = score_df,
         paired = TRUE)

# Print the result
cat(
  "Paired t-test p.value:",
  t_test_result$p.value,
  "\n")
```

```
## Paired t-test p.value: 0.004163495
```

```
# Check if the p-value is
# less than the level of significance,
# e.g. 0.05
if (t_test_result$p.value < 0.05) {
  cat("Student achievement has generally improved.\n")
} else {
  cat("There is insufficient evidence of a general improvement in student achievement.\
}
```

```
## Student achievement has generally improved.
```

0.5 练习与作业 2：作图

- **利用 talk10 中的 data.fig3a 作图**
  - 首先用以下命令装入数据:

```
library(tidyverse);
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v stringr   1.5.0
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.0
## v readr     2.1.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts
```

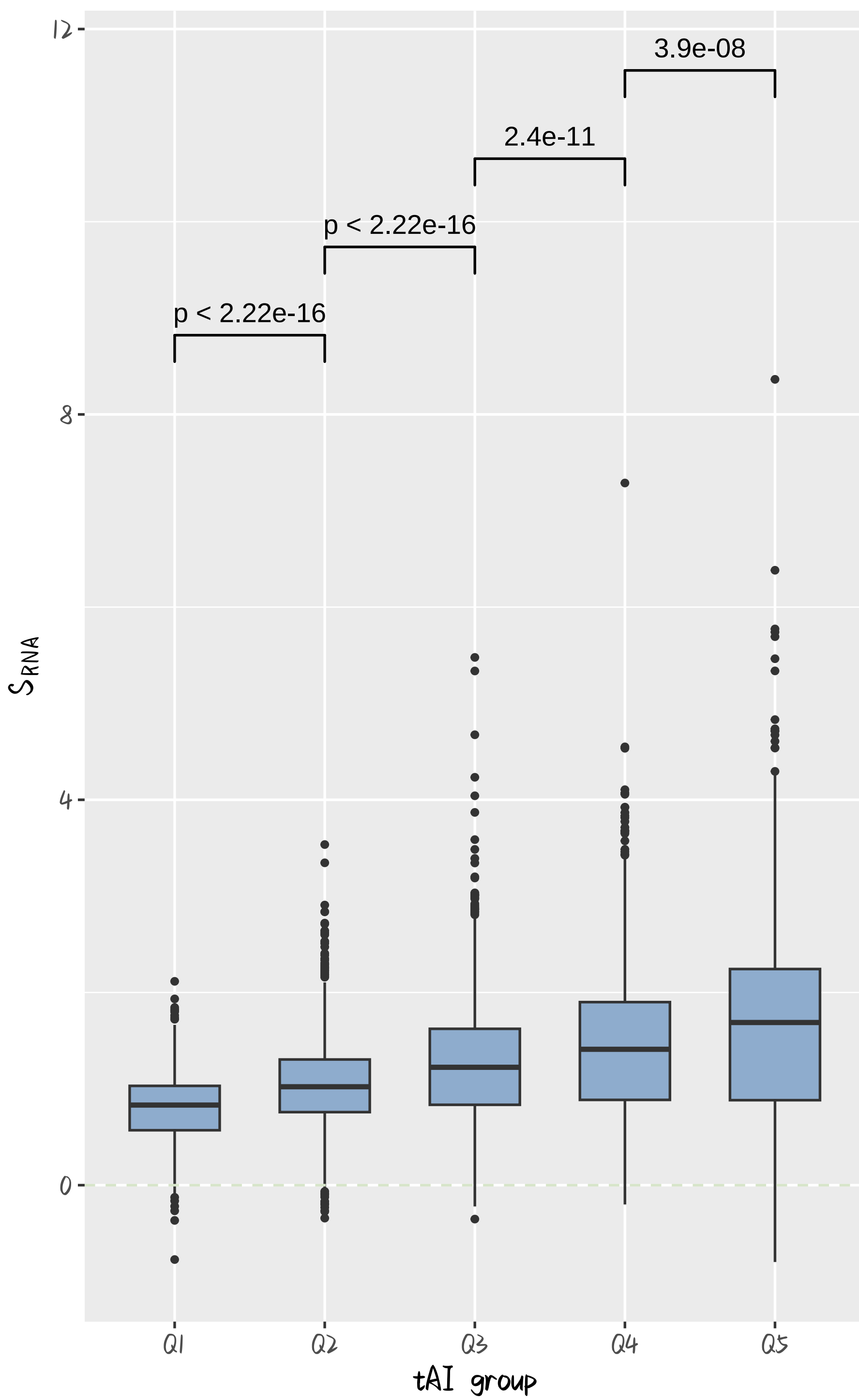
```
data.fig3a <- read_csv( file = "../data/talk10/nc2015_data_for_fig3a.csv" ,show_col_typ
```

-   利用两列数据: `tai` `zAA1.at` 做`talk10`中的`boxplot`（详见: `fig3a`的制作）;

-   用`ggsignif`为相邻的两组做统计分析（如用 `wilcox.test` 函数），并画出`p.value`;

```
## 代码写这里, 并运行;
library(tidyverse)
library(ggsignif)

fig3a_df = as.data.frame(data.fig3a)
rm(data.fig3a)

# Draw the basic boxplot
fig3a_boxplot =
  ggplot( fig3a_df,
          aes( factor(tai), zAA1.at ) ) +
  geom_boxplot(
```

```
    fill = "#8EACCD",
    linetype = 1 ,
    outlier.size = 1,
    width = 0.6) +
  xlab( "tAI group" ) +
  ylab( expression( paste( italic(S[RNA]) ) ) ) +
  scale_x_discrete(
    breaks= 1:5 ,
    labels= paste("Q", 1:5, sep = "") ) +
  geom_hline(
    yintercept = 0,
    colour = "#D7E5CA",
    linetype = 2) +
  theme(
    text=element_text(
      family="RLQDMSWR",
      size=14))


# Add p.value
fig3a_boxplot_signif =
  fig3a_boxplot +
  geom_signif(
    comparisons = list(1:2, 2:3, 3:4, 4:5),
    test = wilcox.test,
    step_increase = 0.1 )

print(fig3a_boxplot_signif)
```



3.9e-08
2.4e-11
p < 2.22e-16
p < 2.22e-16
12
8
4
0
S_RNA
Q1
Q2
Q3
Q4
Q5
tAI group

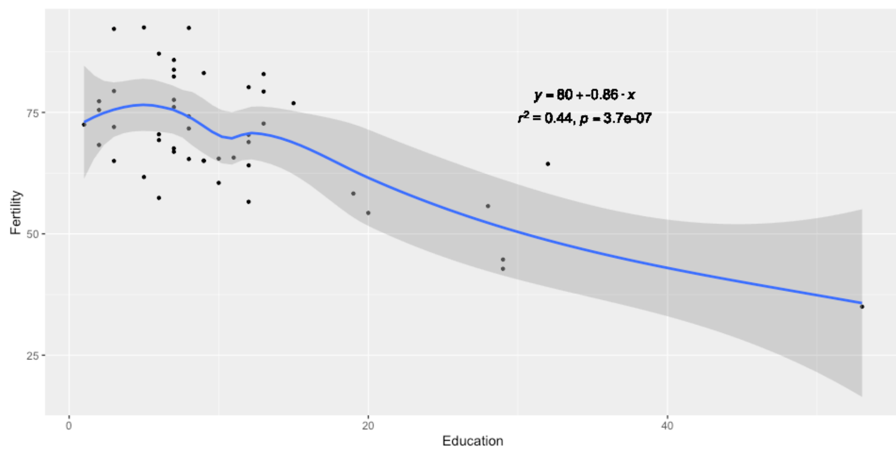
问：这组数据可以用 `t.test` 吗？为什么？

答：

不可以，不满足 t-test 的先决条件，如果运行下列 t.test 的代码将会报错:

```
# Take the first 200 rows of data
# to avoid exceeding the test range
fig3a_df_test = fig3a_df[1:200,]



# Carry out the examination
shapiro_test =
  shapiro.test(fig3a_df_test$zAA1.at)
levene_test =
  car::leveneTest(
    fig3a_df_test$zAA1.at ~ fig3a_df_test$tai,
    data = fig3a_df_test)
```

- **用系统自带变量 `mtcars` 做图**
  - 用散点图表示 `wt`（x-轴）与 `mpg`（y-轴）的关系
  - 添加线性回归直线图层
  - 计算 `wt` 与 `mpg` 的相关性，并将结果以公式添加到图上。其最终效果如下图所示（注：相关代码可在 `talk09` 中找到）:

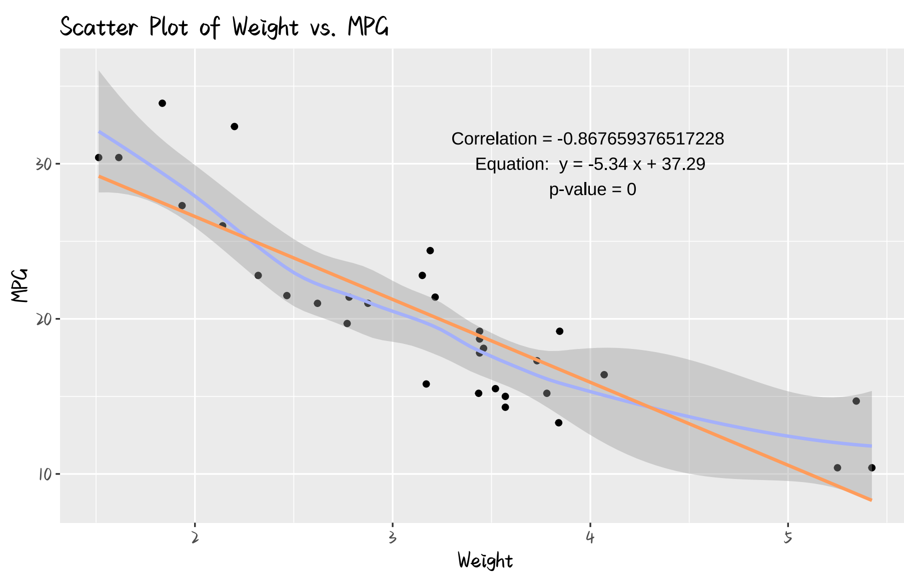



```
## 代码写这里，并运行；
library(ggplot2)

data("mtcars")
# Creating Scatterplots
scatter_plot_mtcars =
  ggplot(mtcars,
         aes(x = wt, y = mpg)) +
  geom_point() +
  geom_smooth(color = "#A4B0FA") +
  labs(
    title = "Scatter Plot of Weight vs. MPG",
    x = "Weight",
    y = "MPG") +
  theme(
    text=element_text(
      family="RLQDMSWR",
      size=14))

# Calculate lm_model
mtcars_lm_model =
  lm(mpg ~ wt, data = mtcars)
```

```
# Calculate Correlation
mtcars_correlation =
  cor(mtcars$wt, mtcars$mpg)

# Generate the equation
mtcars_equation =
  paste("y =",
        round(coef(mtcars_lm_model)[2], 2),
        "x +",
        round(coef(mtcars_lm_model)[1], 2))

# Calculate the p-value
mtcars_p_value =
  round(
    summary(mtcars_lm_model)$coefficients[2, 4], 4)

# Draw the plot
scatter_plot_with_regression =
  scatter_plot_mtcars +
  # Add a linear regression line
  geom_smooth(method = "lm", se = FALSE, color = "#FF9C5B") +

  # Add correlation formula
  annotate("text",
           # Locate the annotation
           x = 4, y = 30,
           # Contents of the annotation
           label = paste("Correlation =",
                         mtcars_correlation,
                         "\n",
                         "Equation: ",
                         mtcars_equation,
```

```
                                    "\n",
                                    "p-value =",
                                    mtcars_p_value),)







# Print scatter plots and
# linear regression lines
print(scatter_plot_with_regression)
```

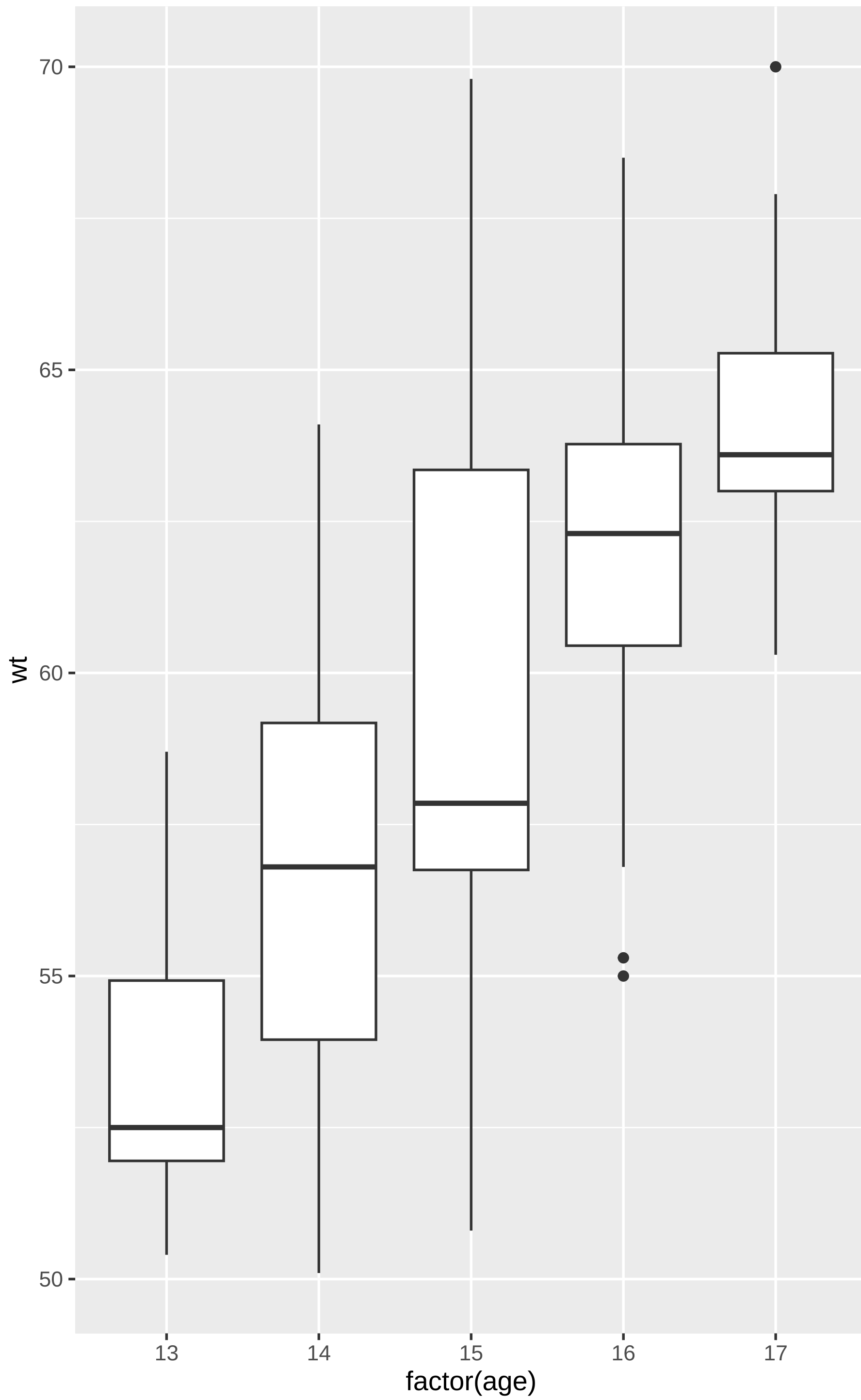
```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```



0.6 练习与作业 3：线性模型与预测

- 使用以下代码产生数据进行分析

```
wts2 <- bind_rows(
   tibble( class = 1, age = sample( 13:15, 20, replace = T ), wt = sample( seq(50, 60,
   tibble( class = 2, age = sample( 14:16, 20, replace = T ), wt = sample( seq(55, 65,
   tibble( class = 3, age = sample( 15:17, 20, replace = T ), wt = sample( seq(60, 70,
);

ggplot(wts2, aes( factor( age ), wt ) ) + geom_boxplot() ;
```


70
65
60
55
50
wt
13
14
15
16
17
factor(age)

- 用线性回归检查`age`，`class` 与 `wt` 的关系，构建线性回归模型；

- 以`age`，`class`为输入，用得到的模型预测`wt`；

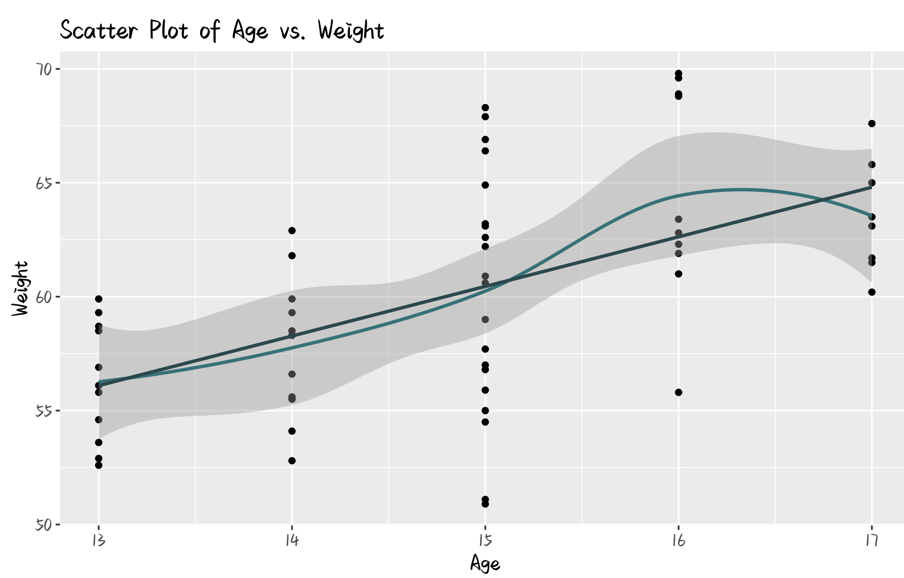
- 计算预测的`wt`和实际`wt`的相关性；

- 用线性公式显示如何用`age`，`class`计算`wt`的值。

```
## 代码写这里，并运行；
library(ggplot2)
library(dplyr)

# Create data.frame
wts2 = bind_rows(
   tibble( class = 1, age = sample( 13:15, 20, replace = T ), wt = sample( seq(50, 60,
   tibble( class = 2, age = sample( 14:16, 20, replace = T ), wt = sample( seq(55, 65,
   tibble( class = 3, age = sample( 15:17, 20, replace = T ), wt = sample( seq(60, 70,
)

# Draw Scatter Plot
wts2_scatter_plot =
  ggplot(wts2, aes(x = age, y = wt)) +
  geom_point() +
  geom_smooth(color = "#377177") +
  theme(
    text=element_text(
      family="RLQDMSWR",
      size=14)) +
  labs(
    title = "Scatter Plot of Age vs. Weight",
    x = "Age",
    y = "Weight")
```

```r
# Add a linear regression line
wts2_scatter_plot_with_regression =
  wts2_scatter_plot +
  geom_smooth(
    method = "lm",
    se = FALSE,
    color = "#2B484E")+
  theme(
    text=element_text(
      family="RLQDMSWR",
      size=14))

print(wts2_scatter_plot_with_regression)
```

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



```r
# Constructing a linear regression model
linear_model =
  lm(wt ~ age + class, data = wts2)

# Forecast wt
predicted_wt =
  predict(
    linear_model,
    newdata =
      data.frame(
        age = wts2$age,
        class = wts2$class))

# Calculate the correlation
# between actual wt and predicted wt
correlation =
  cor(wts2$wt, predicted_wt)

# Print relevance
cat("Correlation between Actual Weight and Predicted Weight:", correlation, "\n")
```

```
## Correlation between Actual Weight and Predicted Weight: 0.8240552
```

```r
# Print the formula
cat("Linear Regression Formula: wt = ", round(coef(linear_model)[1], 2), " + ", round(c
```

```
## Linear Regression Formula: wt =  57.11  +  -0.54  * age +  5.56  * class
```

```r
# AWAITING PERFECTION:
# Insert the formula into the figure
```