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] "sicheng.wu"
```

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

```
## [1] "/home/vkorpela"
```

0.4 练习与作业 1：数据查看

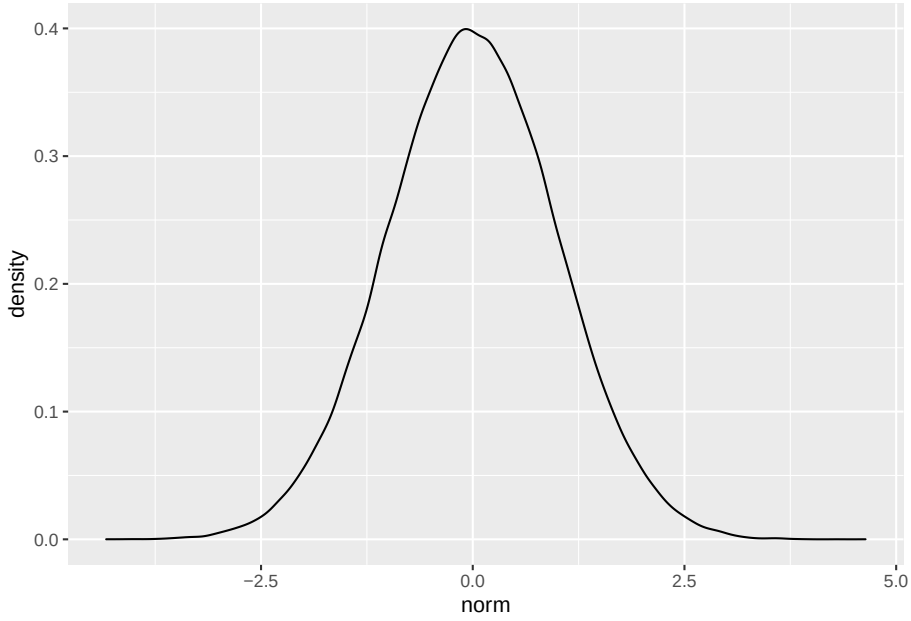
- **正态分布**

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(tidyverse)
```

```
## Warning in system("timedatectl", intern = TRUE): running command 'timedatectl'
## had status 1

## -- 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   1.2.0      v stringr 1.4.1
## v readr   2.1.2      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
# 生成 vector
x <- rnorm(100000, mean = 0, sd = 1)

# 画出分布状况
ggplot(data.frame(norm = x), aes(x = norm)) +
  geom_density()
```




```
# 计算各范围取值百分比
c("mean+-1" = paste(sum(x <= 1 & x >= -1) / 1000, "%", sep = ""),
  "mean+-2" = paste(sum(x <= 2 & x >= -2) / 1000, "%", sep = ""),
  "mean+-3" = paste(sum(x <= 3 & x >= -3) / 1000, "%", sep = ""))
```

```
##   mean+-1   mean+-2   mean+-3
## "68.343%" "95.425%" "99.751%"
```

- **用函数生成符合以下分布的数值，并做图：**

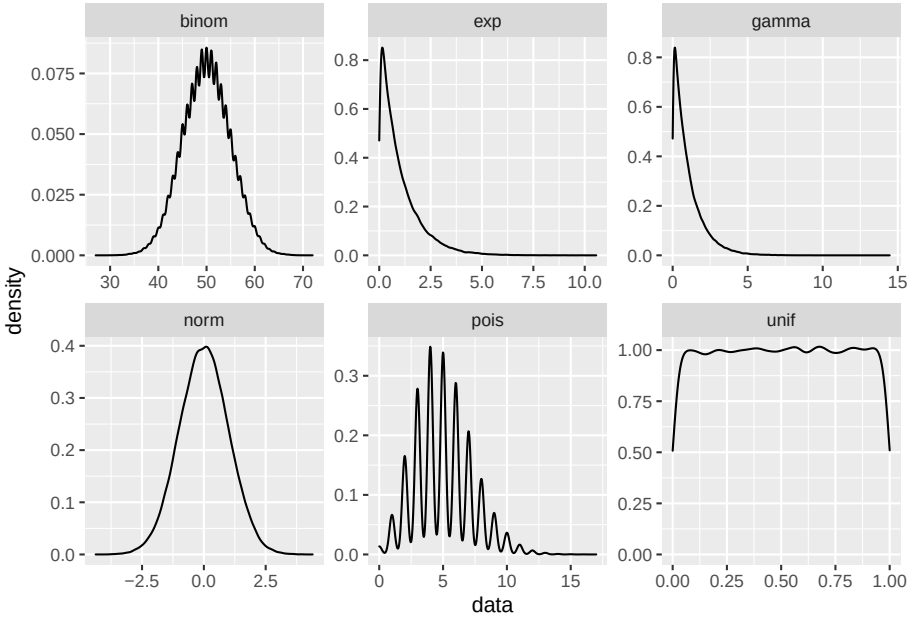
另外，在英文名后给出对应的中文名：

- Uniform Distribution：均匀分布

- Normal Distribution：正态分布

- Binomial Distribution：二项分布
- Poisson Distribution：泊松分布
- Exponential Distribution：指数分布
- Gamma Distribution：伽马分布

```
## 代码写这里，并运行；
# 均匀分布
dist = bind_rows(
  tibble(type = "unif", data = runif(100000)),
  tibble(type = "norm", data = rnorm(100000)),
  tibble(type = "binom", data = rbinom(100000, 100, 0.5)),
  tibble(type = "pois", data = rpois(100000, 5)),
  tibble(type = "exp", data = rexp(100000)),
  tibble(type = "gamma", data = rgamma(100000, 1))
)

ggplot(dist, aes(x = data)) +
  geom_density() +
  facet_wrap(~type, ncol = 3, scales = "free")
```



- **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。

```
## 代码写这里，并运行;
mtcars.bin <- mtcars %>%
  mutate(
    equal.sized = ntile(wt, n = 4),
    equal.distance = cut(
      wt,
      seq(min(wt), max(wt), (max(wt) - min(wt)) / 4),
      include.lowest = TRUE
    )
  )
```

```
# equal-sized bin 在分组时确保每组元素个数相同
table(mtcars.bin$equal.sized)
```

```
##
## 1 2 3 4
## 8 8 8 8
```

```
# equal-distance bin 在分组时确保每组的间隔一样大
table(mtcars.bin$equal.distance)
```

```
##
## [1.51,2.49] (2.49,3.47] (3.47,4.45] (4.45,5.42]
##           8          13           8           3
```

- **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；
s.iqr <- IQR(swiss$Infant.Mortality)
s.sum <- summary(swiss$Infant.Mortality)
swiss %>%
  filter(Infant.Mortality < s.sum["1st Qu."] - 1.5 * s.iqr |
         Infant.Mortality > s.sum["3rd Qu."] + 1.5 * s.iqr) %>%
  .$Infant.Mortality
```

```
## [1] 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;
# t-Test
with(Data, t.test(Steps ~ Sex)["p.value"])
```

```
## $p.value
## [1] 0.01461209
```

```
# Wilcoxon test
with(Data, wilcox.test(Steps ~ Sex)["p.value"])
```

```
## Warning in wilcox.test.default(x = DATA[[1L]], y = DATA[[2L]], ...): cannot
## compute exact p-value with ties
```

```
## $p.value
## [1] 0.01773304
```

- 两种检测方法的`p.value`哪个更显著? 为什么?

答：两种检测方法中 t-Test 的 p.value 更显著。因为 t-Test 评估的是均值的差异，而 Wilcoxon test 评估的是中值的差异，在此例中，两个性别步数的均值差异略大于中值差异。

- **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；
scores.wide <- scores %>%
  spread(key = Time, value = Score)

with(scores.wide, t.test(After, Before, paired = TRUE))
```

```
##
##  Paired t-test
##
## data:  After and Before
```

```
## t = 3.8084, df = 9, p-value = 0.004163
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
##   4.141247 16.258753
## sample estimates:
## mean difference
##            10.2
```

答：计算得 p-value 为 $0.004163 < 0.01$，存在显著差异，说明同学们的成绩有普遍提高。

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：

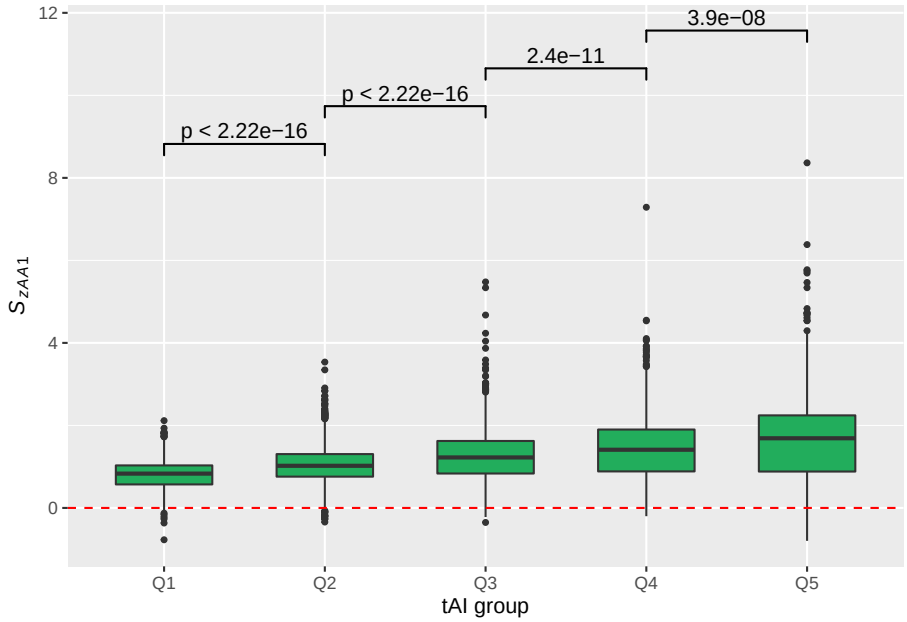
```
library(tidyverse);
data.fig3a <- read_csv( file = "../data/talk10/nc2015_data_for_fig3a.csv" );
```

```
## Rows: 7109 Columns: 8
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): acc
## dbl (7): tai, trans.at, trans.gc, zAA2.at, zAA2.gc, zAA1.at, zAA1.gc
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

- 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`；

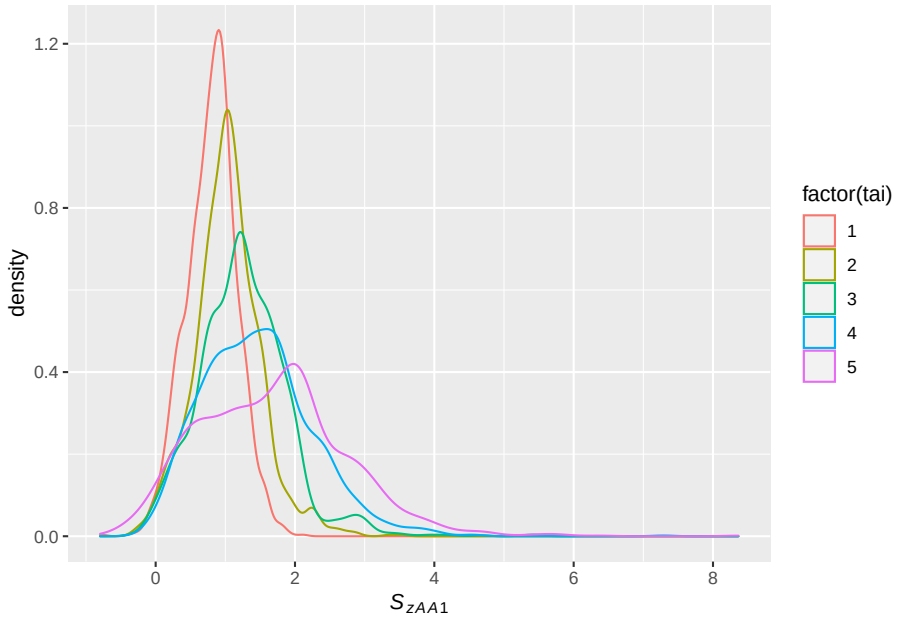
```
## 代码写这里，并运行；
library(ggsignif)

ggplot(data.fig3a, aes(x = factor(tai), y = zAA1.at)) +
  geom_boxplot(fill = "#22AD5C", linetype = 1,
               outlier.size = 1, width = 0.6) +
  xlab("tAI group") +
  ylab(expression(italic(S[zAA1]))) +
  scale_x_discrete(breaks = 1:5, labels = paste("Q", 1:5, sep = "")) +
  geom_hline(yintercept = 0, colour = "red", linetype = 2) +
  geom_signif(comparisons = list(1:2, 2:3, 3:4, 4:5),
              test = "wilcox.test", step_increase = 0.1)
```



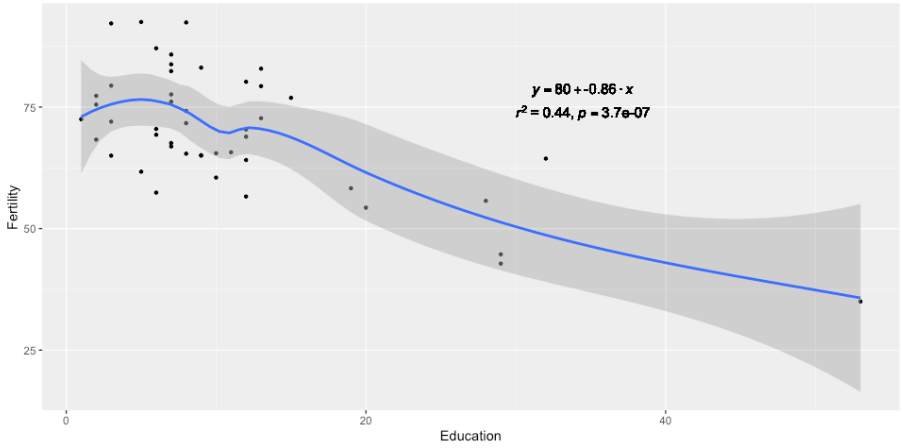
问： 这组数据可以用 `t.test` 吗？为什么？

答：不能，因为部分数据中存在大量离群值，呈偏态分布，不适合使用 t-Test。

```
## 代码写这里，并运行；
ggplot(data.fig3a, aes(x = zAA1.at, color = factor(tai))) +
  geom_density() +
  xlab(expression(italic(S[zAA1])))
```



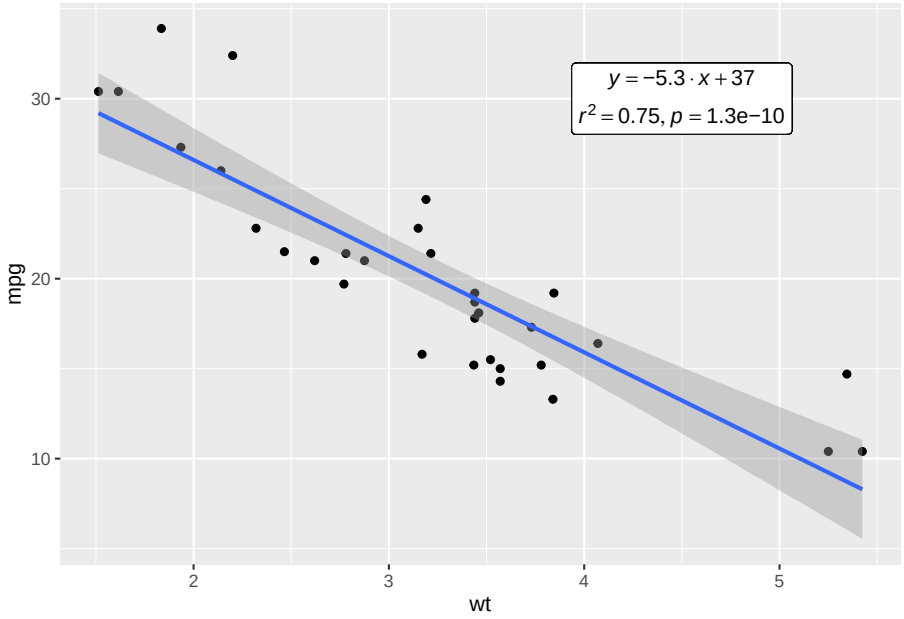
- **用系统自带变量 `mtcars` 做图**
  - 用散点图表示 `wt`（x-轴）与 `mpg`（y-轴）的关系
  - 添加线性回归直线图层
  - 计算 `wt`）与 `mpg` 的相关性，并将结果以公式添加到图上。其最终效果如下图所示（注：相关代码可在 `talk09` 中找到）：




```r
## 代码写这里，并运行；
mtcars.linear <- lm(mtcars$mpg ~ mtcars$wt)
mtcars.cor = cor.test(mtcars$wt, mtcars$mpg)
eq_text <- substitute(
  atop(
    italic(y) == a %.% italic(x) + b,
    list(italic(r)^2 == r2, italic(p) == pvalue)
  ),
  list(
    a = format(coef(mtcars.linear)[[2]], digits = 2),
    b = format(coef(mtcars.linear)[[1]], digits = 2),
    r2 = format(summary(mtcars.linear)$r.squared, digits = 2),
    pvalue = format(mtcars.cor$p.value, digits = 2)
  )
)

eq_text <- as.expression(eq_text)
eq_text <- as.character(eq_text)

ggplot(mtcars, aes(x = wt, y = mpg)) +
  geom_point() +
  geom_smooth(method = "lm") +
```

```
  geom_label(
    data = NULL,
    aes(x = 4.5, y = 30, label = eq_text),
    parse = TRUE,
    inherit.aes = FALSE
  )
```

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

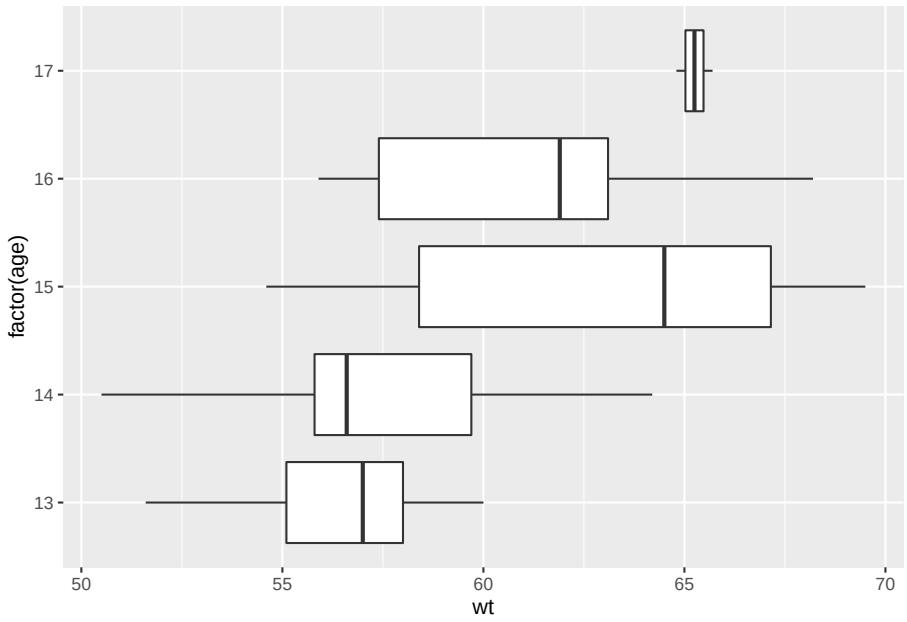


0.6 练习与作业 3：线性模型与预测

- **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() + coord_flip();
```



- 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(FSA)
```

```
## ## FSA v0.9.3. See citation('FSA') if used in publication.
```

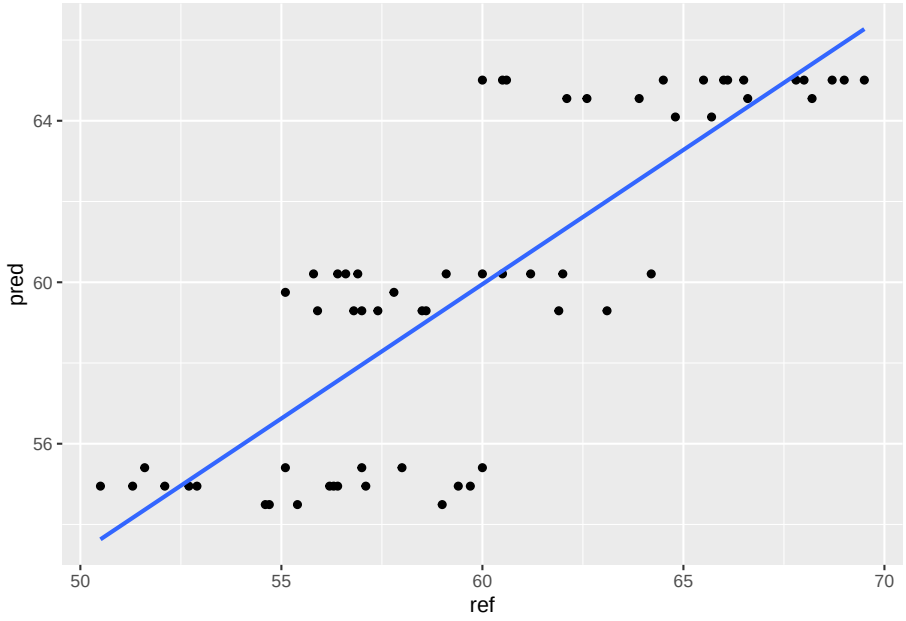
```
## ## Run fishR() for related website and fishR('IFAR') for related book.
```

```
# 构建线性回归模型
wts2.model <- lm(wt ~ age + class, data = wts2)
anova(wts2.model)
```

```
## Analysis of Variance Table
##
## Response: wt
##           Df Sum Sq Mean Sq F value    Pr(>F)
## age        1 291.71  291.71  33.937 2.773e-07 ***
## class      1 682.15  682.15  79.359 2.180e-12 ***
## Residuals 57 489.95    8.60
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# 进行预测
wts2.pred <- predict(
  wts2.model,
  wts2 %>% select(age, class)
)
wts2.cmp <- data.frame(
  ref = wts2$wt,
  pred = wts2.pred
)

ggplot(wts2.cmp, aes(x = ref, y = pred)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE)
```

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




```
# 计算相关性
wts2.cor <- with(wts2.cmp, cor.test(pred, ref))
wts2.cor$estimate
```

```
##       cor
## 0.8156523
```

```
# 线性公式
wts2.coef <- coef(wts2.model)

paste0(
  "wt = ",
  format(wts2.coef["age"], digits = 3),
  " * age + ",
  format(wts2.coef["class"], digits = 3),
  " * class + ",
  format(wts2.coef[1], digits = 3)
)
```

```
## [1] "wt = -0.457 * age + 5.26 * class + 56.1"
```