Important Point

Created	@March 12, 2024 4:21 PM
Class	ADS
Reviewed	☐

统计方面

1. T.test

在做t.test之前要做什么？

首先确定假设，随后进行数据的清洗和处理。最重要的是要先通过hist这个数据看一下数据分布，是否符合正态分布，然后进行正态分布检验（hapiro-Wilk检验）。

当确认使用t.test之后，要确定使用哪一种t.test

1. **单样本 t 检验（One-sample t-test）**：这种类型的 t 检验用于比较单个样本的平均值和已知或理论的平均值。例如，你可能想要测试一个制造过程是否产生的产品尺寸的平均值与规定的尺寸相符。
2. **独立样本 t 检验（Independent two-sample t-test）**：这种类型的 t 检验用于比较两个独立样本的平均值。例如，你可能想要比较两种不同教学方法对学生成绩的影响。
3. **配对样本 t 检验（Paired t-test）**：这种类型的 t 检验用于比较同一组观察对象在不同条件下的平均值。例如，你可能想要比较同一组学生在接受不同教学方法后的学习成绩。

接下来要判断使用单尾检验还是双尾检验，确定检验的目的。

什么情况下不能用t.test? 应该转而用什么test？

1. **非正态数据**：如果你的数据不符合正态分布，可以考虑使用非参数检验，如威尔科克森秩和检验（Wilcoxon rank-sum test）或曼-惠特尼 U 检验（Mann-Whitney U test）。这些检验不需要数据符合正态分布。
2. **方差不等**：如果两个样本的方差不相等，可以使用 Welch 的 t 检验，它不需要假设两个样本的方差相等。在 R 中，你可以通过将 `t.test` 函数的 `var.equal` 参数设为 `FALSE` 来进行 Welch 的 t 检验。

3. **配对数据**：如果你有配对数据，但差异不符合正态分布，可以使用威尔科克森符号秩检验（Wilcoxon signed-rank test）。
4. **分类数据**：如果你的数据是分类的或二元的（例如，成功/失败或是/否），那么 t 检验不适用。在这种情况下，你可以使用卡方检验（Chi-squared test）或费希尔精确概率检验（Fisher's exact test）。

T test标准流程？

先看样本分布，用shapiro分析，hist一下

然后看类别，决定用哪一个，如果是实验前后对照用pair

```
#双样本
data1 = data("ToothGrowth")
str(ToothGrowth)
t.test(len ~ supp, data = ToothGrowth)

t_test_dose_0.5_1.0 <- t.test(len ~ dose, data = subset(Too
thGrowth, dose %in% c(0.5, 1.0)))
print(t_test_dose_0.5_1.0)

#paired
blood_pressure_data <- read.table("blood_pressure.txt", hea
der = TRUE)
treatment_effect <- t.test(blood_pressure_data$bp_before, b
lood_pressure_data$bp_after, paired = TRUE)
print(treatment_effect)

#这里是两种不同的取样方法，针对的分别是两个list，或者是直接用datafra
me的两个参数
```

2. 什么是一类错误和二类错误？

Type I 错误和 Type II 错误是统计假设检验中的两种错误类型，它们与显著性水平（α）之间有着密切的关系。

Type I 错误（假阳性）：发生在当零假设为真时，错误地拒绝了零假设的情况。换句话说，它意味着在实际上不存在效应或差异的情况下，我们错误地认为发现了效应或差异。通常情况下，显著性水平 α 表示了 Type I 错误的概率，即当零假设为真时错误

地拒绝零假设的概率。比如，α 设定为 0.05，则意味着以 5% 的概率错误地拒绝了零假设，即将 5% 的几率认为发现了差异，但实际上却没有。

Type II 错误（假阴性）：发生在当备择假设为真时，未能拒绝零假设的情况。换句话说，它意味着在实际上存在效应或差异的情况下，我们未能检测到这种效应或差异，而错误地接受了零假设。Type II 错误与统计检验的功效（power）相关，即正确地拒绝零假设的概率（1 - Type II 错误的概率）。通常情况下，减小 Type I 错误的概率（通过减小显著性水平 α），可能会增加 Type II 错误的概率。

显著性水平 α 在统计推断中是一种平衡，选择较低的 α 可以减小 Type I 错误的概率，但这可能导致增加 Type II 错误的概率。这种权衡关系表示了在假设检验中需要在两种错误类型之间做出选择的必要性。科学研究中，通常会根据研究的具体情况和重要性来选择合适的显著性水平，以平衡 Type I 和 Type II 错误的概率。

3. POWER 和 SAMPLE SIZE 关系

功效（Power） 和 **样本大小（Sample Size）** 是统计假设检验中的两个关键概念。

1. **功效（Power）**：在统计假设检验中，功效是正确拒绝零假设（即，检测到效应存在）的概率。换句话说，如果实际上存在一个效应（即，零假设不成立），那么功效就是你的实验发现这个效应的概率。功效通常用 1 减第二类错误（β）的概率来表示，第二类错误是指当实际存在效应时，我们错误地接受了零假设。在设计实验时，我们通常希望功效至少为 0.8，这意味着我们有 80% 的机会检测到实际存在的效应。
2. **样本大小（Sample Size）**：样本大小是指我们在进行实验或研究时收集的数据点的数量。样本大小对统计分析的结果有重大影响。一般来说，样本越大，我们对总体参数的估计就越准确，拒绝错误的零假设的能力也就越强。然而，增加样本大小也会增加收集数据的成本和时间。

功效和样本大小之间存在密切的关系。在保持其他条件（如效应大小和显著性水平）不变的情况下，增加样本大小可以增加实验的功效，即提高检测到实际存在效应的概率。

使用下载的包pwr实现

使用方法可以看

https://zhuanlan.zhihu.com/p/129375307

https://zhuanlan.zhihu.com/p/137779235

```
install.packages("pwr")
library(pwr)
```

t test中d值的计算

就t检验而言，它的效应量可以用如下方法来估计：

$$d = \frac{|\mu_1 - \mu_2|}{\sigma}$$

where μ_1 = mean of group 1
μ_2 = mean of group 2
σ^2 = common error variance


```
pwr.t.test(n= , d = , sig.level = , power = , type = c("tw
o.sample","one.sample", "paired"))


delta <- 130 * 0.1
sd <- 30
power <- 0.8
sig.level <- 0.05
result <- power.t.test(delta = delta, sd = sd, sig.level =
sig.level, power = power, type = "two.sample", alternative
= "one.sided")
result$n
```

可以通过设置参数alternative="two.sided"、"less"或者"greater"来指定双侧检验或者单侧检验，默认值是双侧检验。

卡方检验

```
pwr.chisq.test(w=, N = , df = , sig.level =, power = )
```

$$w = \sqrt{\sum_{i=1}^{m} \frac{(p0_i - p1_i)^2}{p0_i}}$$

where $p0_i$ = cell probability in ith cell under H_0
$p1_i$ = cell probability in ith cell under H_1

Cohen建议将w值0.1作为小效应量，0.3作为中等效应量以及0.5作为大效应量。

4. Catagerical data (Chi-square)

对于卡方检验，我们可以通过其知道数据与我们期待的分布是否相符，或者两因素是否独立（独立性检验，其实本质是看和独立分布是否相同）即有两个分类变量，我们想要确定它们之间是否存在显著差异。

卡方适配度检验（Chi-Square Goodness-of-Fit Test）是一种统计方法，用于检验一个分类变量的观察频数分布是否与期望频数分布相符。它通常用于确定样本数据是否符合预期的分布，比如均匀分布或其他已知的理论分布。

卡方独立性检验（Chi-Square Test of Independence）用于检验两个分类变量之间是否存在显著的关联。常用于研究两个变量是否独立，例如性别和吸烟习惯之间是否有关系。

卡方的假设条件：

Assumptions include: – 2 points.

- The variables must be categorical.
  - *Fits.*
- Observations must be independent.
  - Can assume from the task. *Fits.*
- Cells in the contingency table are mutually exclusive.
  - *Fits.*
- The expected value of cells should be 5 or greater in at least 80% of cells.
  - See Table 2. *Fits.*

- **零假设（H0）：** 变量ab无关/数据分布与预期分布没有差异
- **备择假设（H1）：** 变量a受变量b影响/数据分布与预期分布有显著差异

卡方检验是卡方分布为基础的一种检验方法，主要用于**分类变量**，根据样本数据推断总体的分布与期望分布是否有显著差异，或推断两个分类变量是否相关或相互独立。其原假设为：观察频数与期望频数没有差别。

首先导入，清洁数据

然后可视化数据（表格方式）

```
#对于长数据可以用table的方式来查看数据
category_counts = table(data2$genotype)
#还有多的因素的话也可以画出来
```

```
category_counts = table(data2$genotype,data2$sex)
#然后可视化 多因素只需要改变位置就可以了
barplot(category_counts, main="Category Counts", xlab="Cate
gory", ylab="Frequency", col=c("skyblue", "orange", "gree
n"))
#如果使用ggplot要转换成长数据/library(tidyr)
```

然后计算 如果数据数量太少需要用fisher检验而不是卡方

```
#计算预期数据 用nrow来作为总数 注意对应关系
expected_counts <- c("mut/mut" = 0.5 * nrow(data2),
                     "WT/mut" = 0.25 * nrow(data2),
                     "WT/WT" = 0.25 * nrow(data2))
expected_counts
expected_prob = c(0.5,0.25,0.25)
```

对于单因素数据

```
#可以用两个list比较,这里用的是给定概率
chi_test_result <- chisq.test(x = category_counts, p = expe
cted_prob)
chi_test_result
#or
chi_test_result <- chisq.test(x = category_counts, p = expe
cted_counts，, rescale.p = TRUE)
chi_test_result


#也可以暴力一点直接生成表格，这里是作为两组数据计算皮尔森概率（并不好）
tableR<- matrix(c(56,7,17,40,20,20)，nrow=2，ncol=3)
chi_test_result <- chisq.test(tableR)chi_test_result


#或者用三个list生成dataframe
opening_times <- c("Early", "Late")
satisfied <- c(864, 980)
unsatisfied <- c(714, 473)
data <- data.frame(Opening_Times = opening_times, Satisfied
```

```
= satisfied, Unsatisfied = unsatisfied)


# 进行卡方检验
chisq_test_result <- chisq.test(data[,2:3])
```

对于双因素数据,需要一个数据matrix和一个概率matrix，并且数据matrix需要as.numeric

```
category_counts = table(data2$sex,data2$genotype)
expected_prob <- matrix(c(0.25,0.25,0.125,0.125,0.125, 0.125), nrow=2, ncol=3)
chi_test_result <- chisq.test(as.numeric(category_counts), p = expected_prob)
```

```
#对于table的后续可视化
barplot(tableR,beside = TRUE,
                col = c("skyblue", "orange"),
                names.arg = c("het", "mut","WT"))
```

在得到结论的时候需要注意要说明significance，还要同时说明effect是什么，就是你接受了哪个假设，带来了什么影响。

5. correlation

线性拟合用来判断数据的两个变量之间是否有线性关系

需要的数据只有：自变量的list和因变量的list

- **零假设（H0）：** 两个变量之间不能判断有线性关系
- **备择假设（H1）：** 两个变量之间有线性关系

首先是假设的验证，需要满足数据残差normalization和方差齐性

```
model_male <- lm(data_males$val ~ data_males$year)
hist(residuals(model_male), breaks = 5, col = "gray",
     main = "Histogram of the residuals", xlab = "Residuals", cex = 0.6)plot(model_male, which = c(1, 2))
```

```
plot(model_male, 1)
plot(model_male, 2)
```

然后进行分析

```
correlation_males <- cor.test(data_males$val, data_males$ye
ar, use = "complete.obs")
print(correlation_males)
```

6. ANOVA

拿到两个条件，如果是前后对照的就做个减法，如果还是很多条件就做多因素anova（*）

- **零假设（H0）：** 多组数据的means之间没有显著性差异
- **备择假设（H1）：** 多组数据的means之间有显著性差异

H0: means of different supp groups are the same
H1: means of different supp groups are NOT the same

需要先进行样本的残差的正态分布检验和样本的方差齐性检验

分别用 (plot(anova_model, 2)和(plot(anova_model, 1)

或者是

```
shapiro.test(data1$Glucose)
bartlett.test(data1$Glucose, data1$Treatment)
```

符合做anova，不符合做 Kruskal-Wallis test

单因素anova分析

```
anova_result <- aov(Glucose~Treatment, data = data1)
summary(anova_result)
```

多因素分析(如果因素中有数字要转化为as.factor)

H0: means of different supp groups are the same
H1: means of different supp groups are NOT the same

```
anova_result <- aov(len ~ supp * as.factor(dose), data = da
ta1)
```

anova完成后用turkey进行下一步检验

```
tukey_result <- TukeyHSD(anova_result)
tukey_result
```

最后给出建议可以怎么办？为anova计算power

如果power有点低，可以计算power=0.8或者0.9至少需要多少样本

如何计算里面的f？ df E 残差 df A 组数-1

估计各效应的偏η^2：

- $\eta^2_{(A)} = \frac{F_A \times df_A}{F_A \times df_A + df_E} = \frac{8.611 \times 1}{8.611 \times 1 + 24} = 0.264$
- $\eta^2_{(B)} = \frac{F_B \times df_B}{F_B \times df_B + df_E} = \frac{5.975 \times 2}{5.975 \times 2 + 24} = 0.332$
- $\eta^2_{(A\times B)} = \frac{F_{A\times B} \times df_{A\times B}}{F_{A\times B} \times df_{A\times B} + df_E} = \frac{19.350 \times 2}{19.350 \times 2 + 24} = 0.617$

- 效应量$f = \sqrt{\frac{\eta^2_{(A)}}{1 - \eta^2_{(A)}}} = 0.599$

```
pwr.anova2.test(
k = 3,
n = 7,
f = 0.537,
sig.level = 0.05,
power = NULL
)
```

7. (cannot match any assumption): Boostraps

当样本的独立性很差的时候需要用到bootstrap

These data are categorical but seriously lacking in independence. Students may have been customers more than once and, similarly, there are likely to be students who were customers in both trials. The two categories are therefore not **exclusive either**.

不满足chi square的要求

bootstrap本质就是你关心一个样本的某个特征，比如a元素的占比。那么就可以通过bootstrap重新抽样允许重复得到新的样本，计算每个样本的这个特征，随后直接通过这些重复出来的特征的分布计算出置信区间

详细原理步骤

1. **原始样本**：
   - 你有一个样本数据集，这里是各电影类型的喜欢人数。
   - 例子：`comedy: 73, action: 42, romance: 38, ...`，总共有267个学生。
2. **抽取Bootstrap样本**：
   - 从原始样本中随机抽取有放回的样本，每个样本的大小与原始样本相同。
   - 这种抽样方式允许一个数据点被多次抽中或一次也不被抽中。
3. **计算统计量**：
   - 对每个Bootstrap样本计算感兴趣的统计量，如均值、比例等。
   - 在我们的例子中，我们计算每个Bootstrap样本中喜欢某个类型电影的比例。
4. **重复抽样**：
   - 重复上述步骤许多次（如1000次），得到大量的Bootstrap样本。
   - 对每个Bootstrap样本计算统计量，得到统计量的分布。
5. **构建置信区间**：
   - 从统计量的分布中提取特定百分位数，构建置信区间（例如95%的置信区间通常是从2.5%到97.5%的百分位数）。

```
#这是一个比较标准的解法，可以查看，在已知喜欢a的人占总体分布的情况下就可以重抽样

first_satisfied <- 864
first_unsatisfied <- 714
```

```
second_satisfied <- 980
second_unsatisfied <- 473

first_bootstraps <- c()
second_bootstraps <- c()

first_results <- c(rep(1, first_satisfied), rep(0, first_un
satisfied))
second_results <- c(rep(1, second_satisfied), rep(0, second
_unsatisfied))

for (a in 1:100) {
first_sample <- mean(sample(first_results, length(first_res
ults), replace = T))
second_sample <- mean(sample(second_results, length(second_
results), replace = T))
first_bootstraps <- c(first_bootstraps, first_sample)
second_bootstraps <- c(second_bootstraps, second_sample)
}

first_upper <- quantile(first_bootstraps, probs = c(0.975))
second_lower <- quantile(second_bootstraps, probs = c(0.02
5))

boxplot(
first_bootstraps,
second_bootstraps,
notch = T,
names = c('early', 'late'),
ylab = 'Prop. of satisfied button presses'
)
```

另一种重新抽样的方法

```
# 设置参数
total_students <- sum(data1$students)
comedy_students <- data1$students[data1$genre == "comedy"]
comedy_proportion <- comedy_students / total_students
```

```
# 生成Bootstrap样本
set.seed(123)
bootstrap_samples <- replicate(1000, {
  sample_data <- sample(c(1, 0), total_students, replace =
TRUE, prob = c(comedy_proportion, 1 - comedy_proportion))
  mean(sample_data)
})

# 计算置信区间
comedy_ci <- quantile(bootstrap_samples, c(0.025, 0.975))
comedy_ci

boxplot(bootstrap_samples)
```

```
#这是一个复杂版本，可以看看，画的图也更复杂
library(ggplot2)
library(dplyr)

data = read.table("Reporter_assay_4-1-15.txt", header = TRU
E, sep = "\t")

ggplot(data, aes(x = Epigenetic_status, y = ave, color = Tr
anscription_status)) +
  geom_boxplot() +
  labs(x = "Epigenetic Status", y = "Enhancer Activity", co
lor = "Transcription Status")

active_median <- median(data$ave[data$Epigenetic_status ==
"Active"])
repressed_median <- median(data$ave[data$Epigenetic_status
== "Repressed"])
median_diff <- active_median - repressed_median

# 生成一个中位数差异的自助样本
```

```
bootstrap_sample <- function(data) {
  boot_data <- data[sample(nrow(data), replace = TRUE), ]
  with(boot_data, median(ave[Epigenetic_status == "Activ
e"]) - median(ave[Epigenetic_status == "Repressed"]))
}

bootstrap_median_diff <- bootstrap_sample(data)

# 生成1000个自助样本的中位数差异
bootstrap_median_diffs <- replicate(1000, bootstrap_sample
(data))

# 计算这些差异的95%置信区间
conf_interval <- quantile(bootstrap_median_diffs, c(0.025,
0.975))

df <- data.frame(bootstrap_median_diffs)

ggplot(df, aes(x = bootstrap_median_diffs)) +
  geom_histogram(binwidth = 0.1, fill = "lightblue", color
= "black") +
  geom_vline(aes(xintercept = conf_interval[1]), color = "r
ed", linetype = "dashed", size = 1) +
  geom_vline(aes(xintercept = conf_interval[2]), color = "r
ed", linetype = "dashed", size = 1) +
  xlab("Median Difference") +
  ylab("Frequency") +
  ggtitle("Bootstrap Median Differences")

diff2 = with(data, median(ave[Transcription_status == "Acti
ve"]) - median(ave[Transcription_status == "None"]))

bootstrap_sample2 <- function(data) {
  boot_data1 <- data[sample(nrow(data), replace = TRUE), ]
  with(boot_data1, median(ave[Transcription_status == "Acti
ve"]) - median(ave[Transcription_status == "None"]))
}
```

```
bootstrap_median_diffs_1 <- replicate(1000, bootstrap_sampl
e2(data))

conf_interval_1 <- quantile(bootstrap_median_diffs_1, c(0.0
25, 0.975))

df1 <- data.frame(bootstrap_median_diffs_1)

ggplot(df1, aes(x = bootstrap_median_diffs_1)) +
  geom_histogram(binwidth = 0.1, fill = "lightblue", color
= "black") +
  geom_vline(aes(xintercept = conf_interval_1[1]), color =
"red", linetype = "dashed", size = 1) +
  geom_vline(aes(xintercept = conf_interval_1[2]), color =
"red", linetype = "dashed", size = 1) +
  xlab("Median Difference") +
  ylab("Frequency") +
  ggtitle("Bootstrap Median Differences")
```

8. Bayes

例题条件概率(更多见打印文件)

Lie detector problem
In a big store, around 10% of employees are stealing. Everybody has to take a lie detector test that is correct in 80% of cases (and mistakes are equally likely in either direction). Everybody says that they are not a thief.
If the lie detector says that 50 people are lying, how many of them are probably thieves?

```
# Number of employees
n <- 1000
# Generate a vector of employees, where 1 represents a thie
f and 0 represents an honest employee
employees <- c(rep(1, n*0.1), rep(0, n*0.9))
# Shuffle the employees
employees <- sample(employees)
# Generate a vector of lie detector results, where 1 repres
ents a lie and 0 represents truth
```

```
# The lie detector is 80% accurate, so for thieves (employe
es == 1), it correctly identifies them as liars 80% of the
time,
# and for honest employees (employees == 0), it incorrectly
identifies them as liars 20% of the time
results <- ifelse(employees == 1, rbinom(n, 1, 0.8), rbinom
(n, 1, 0.2))
# Find the number of people who the lie detector identified
as liars
liars <- which(results == 1)
# Find the number of these who are actually thieves
thieves <- sum(employees[liars])
print(thieves)
prob = thieves/length(liars)
print(prob*50)
```

```
# Probability that an employee is a thief
P_A <- 0.1

# Probability that the lie detector says an employee is lyi
ng, given that they are a thief
P_B_given_A <- 0.8

# Total probability that the lie detector says an employee
is lying
P_B <- 0.08 + 0.18

# Use Bayes' theorem to find the probability that an employ
ee is a thief, given that the lie detector says they are ly
ing
P_A_given_B <- (P_B_given_A * P_A) / P_B

# Multiply by the number of people the lie detector identif
ied as liars to find the expected number of these who are t
hieves
thieves <- P_A_given_B * 50
```

```
print(thieves)
```

数据方面

1. 数据查看和清洁

```
summary(data1)
str(data1)

anyNA(data2)
anyDuplicated(data2)
```

2. 数据重整（长数据转换）

3. 数据处理

把数据中满足条件的数据提取出来形成一个新的数据，使用subset

```
data_males <- subset(data_clean, sex == "Male")
```

生成dataframe用于表格

```
data <- data.frame(
  Treatmrnt = c("het", "mut","WT"),
  female = c(1, 2, 3),
  male = c(5, 2, 6),
)
```

生成matrix用于计算

```
expected_prob <- matrix(c(0.25,0.25,0.125,0.125,0.125, 0.12
5), nrow=2, ncol=3)
observed <- matrix(c(26, 5, 7, 30, 2, 10), nrow = 2, byrow
= TRUE)
```

数据选取

```
active_median <- median(data$ave[data$Epigenetic_status ==
"Active"])
repressed_median <- median(data$ave[data$Epigenetic_status
== "Repressed"])
median_diff <- active_median - repressed_median
```

4. 画图

```
#绘制箱线图
ggplot(data, aes(x = Epigenetic_status, y = ave, color = Tr
anscription_status)) +  geom_boxplot() +  labs(x = "Epigene
tic Status", y = "Enhancer Activity", color = "Transcriptio
n Status")

# 绘制条形图
ggplot(movie_data, aes(x = genre, y = students)) +
  geom_bar(stat = "identity") +
  theme_minimal() +
  xlab("Movie Genre") +
  ylab("Number of Students") +
  ggtitle("Popularity of Movie Genres among Chinese Univers
ity Students") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
 #or
barplot(data1$students, names.arg = data1$genre)
```