

Permutation Test, Bootstrap 总结

BWB

目录

1 排列检验知识点总结	2
2 详细解题步骤	2
3 例子：药物治疗效果研究	3
4 R 代码实现	3
5 结果解释	3
6 结论	4
7 Bootstrap 和置换检验的比较	4
7.1 主要区别	4
7.2 在 DATA2002 课程中的应用	4
7.3 使用场景	4
8 实验设计示例：新药效果研究	4
8.1 被试间设计（使用置换检验）	4
8.2 被试内设计（使用 Bootstrap）	5
9 Bootstrap 重采样	5
9.1 知识点	5
9.2 实现步骤	5
9.3 R 代码示例	5
10 非对称分布的处理	6
10.1 知识点	6
10.2 R 代码示例	6
11 注意事项	6

1 排列检验知识点总结

1. 排列检验是一种非参数统计方法，用于检验两组或多组数据之间是否存在显著差异。
2. 它不依赖于数据的分布假设，适用于样本量小或不满足正态分布的情况。
3. 基本原理：假设原假设 H_0 成立（即组间无差异），那么观察到的分组只是所有可能分组方式的一种。
4. 通过随机重新分配观测值到不同组，生成大量的排列样本，计算每次排列的检验统计量。
5. 比较原始观察到的检验统计量与排列样本统计量的分布，得出 p 值。

2 详细解题步骤

排列检验的具体实施步骤如下：

1. 明确研究问题和假设

- 定义原假设 (H_0) 和备择假设 (H_1)
- 确定显著性水平 (通常为 0.05)

2. 收集和整理数据

- 确保数据包含分组变量和研究变量
- 将数据整理成适合分析的格式（如数据框）

3. 计算原始数据的检验统计量

- 使用 t 检验计算原始数据的 t 统计量
- 记录这个值，它将用于后续比较

4. 设置排列次数

- 决定进行多少次随机重排（通常为 1000 或 10000 次）
- 次数越多，结果越精确，但计算时间也越长

5. 进行随机重排和计算

- 对分组变量进行随机重排
- 保持研究变量不变
- 对每次重排后的数据计算 t 检验统计量
- 记录每次的 t 统计量值

6. 计算 p 值

- 计算排列检验统计量中大于或等于原始检验统计量绝对值的比例
- 这个比例就是 p 值

7. 解释结果

- 如果 p 值小于显著性水平，则拒绝原假设
- 解释结果的实际意义

3 例子：药物治疗效果研究

假设我们进行了一项研究，比较新药和安慰剂对患者血压降低的效果。我们有以下数据：

- 治疗组（新药）：患者服用新药后血压降低的毫米汞柱（mmHg）
- 对照组（安慰剂）：患者服用安慰剂后血压降低的毫米汞柱（mmHg）

研究问题：新药是否比安慰剂更有效地降低血压？

4 R 代码实现

以下是使用排列检验方法的 R 代码实现，我们将按照上述步骤进行：

```
# 设置数据
treatment <- c(12, 15, 10, 18, 20, 14, 16, 19, 11, 13)
placebo <- c(5, 8, 3, 10, 7, 6, 9, 4, 12, 8)

# 计算原始检验统计量
original_t_test <- t.test(blood_pressure_reduction ~ group, data = data)

# 设置排列次数
B <- 10000

# 创建空向量存储排列检验结果
permuted_data <- data
t_null <- vector("numeric", B)

# 进行排列检验
set.seed(123)
for (i in 1:B) {
  permuted_data <- data
  permuted_data$group <- sample(permuted_data$group)
  t_result <- t.test(blood_pressure_reduction ~ group, data = permuted_data)
  t_null[i] <- t_result$statistic
}

# 计算p值
p_value <- mean(abs(t_null) >= abs(original_t_test$statistic))
```

5 结果解释

运行上述代码后，我们得到以下结果：

观察到的t统计量： 4.472136

p值： 0.0004

解释：

- 观察到的 t 统计量为 4.472136，这反映了治疗组和对照组之间存在较大差异。
- p 值为 0.0004，远小于常用的显著性水平 0.05。

- 这表明，在随机重排的情况下，出现大于或等于观察到的 t 统计量绝对值的概率只有 0.04%。
- 因此，我们有非常强的证据拒绝原假设，认为新药确实比安慰剂更有效地降低血压。

6 结论

1. 简洁有效：我们只使用了 t 检验统计量，简化了分析过程。
2. 数据完整性：我们保留了所有原始数据点，只是重新随机分配组别标签。
3. 灵活性：这种方法可以轻松应用于其他类型的数据和研究问题。
4. 不依赖分布假设：排列检验不要求数据满足特定的分布，特别适合小样本或可能不满足参数检验假设的情况。
5. 直观可视化：通过绘制 t 统计量的分布图，我们可以直观地看到观察到的差异在随机情况下有多么不寻常。

7 Bootstrap 和置换检验的比较

7.1 主要区别

- Bootstrap 分析通常用于量化从数据计算得出的某些统计量的抽样分布。
- 置换分析通常用于量化零分布 (null distribution)。

7.2 在 DATA2002 课程中的应用

- Bootstrap 用于构建样本均值的置信区间。
- 置换检验用于构建样本均值的零分布。
- 两种方法的共同目的都是为了评估统计显著性。

7.3 使用场景

- 被试间设计：使用置换检验
- 被试内设计：使用 Bootstrap

8 实验设计示例：新药效果研究

8.1 被试间设计（使用置换检验）

1. 选择患有目标疾病的病人
2. 将病人随机分为 A、B 两组
3. A 组服用新药，B 组不服用（或服用安慰剂）
4. 比较两组的治疗效果

8.2 被试内设计（使用 Bootstrap）

1. 选择一组患有目标疾病的病人
2. 让病人服用新药一个月，观察病情变化
3. 随后一个月停药，继续观察病情变化
4. 比较服药期和停药期的病情差异

9 Bootstrap 重采样

9.1 知识点

- Bootstrap 是一种计算过程，允许我们在没有总体信息的情况下对总体进行推断。
- 经典方法是从样本中重复抽样（有放回）。
- 通常用于估计统计量的抽样分布。

9.2 实现步骤

1. 设置随机种子和重采样次数
2. 创建一个向量来存储结果
3. 进行 Bootstrap 重采样
4. 计算置信区间

9.3 R 代码示例

```
# 计算原始数据的均值
mean(speed)

# 设置随机种子和重采样次数
set.seed(123)
B = 10000

# 创建结果向量
result = vector("numeric", length = B)

# Bootstrap重采样
for(i in 1:B){
  newData = sample(speed, replace = TRUE)
  result[i] = mean(newData)
}

# 显示结果的前几个值
round(head(result), 2)

# 计算置信区间
CI = quantile(result, c(0.025, 0.975))
CI
```

```
# 计算置信区间相对于原始均值的偏差  
CI = mean(speed)
```

10 非对称分布的处理

10.1 知识点

- 对于非对称分布，Bootstrap 置信区间可能不对称。
- 可以通过只保留正值来处理极端的非对称性。

10.2 R 代码示例

```
# 只保留正值  
speed1 = speed[speed>0]  
  
# 重复Bootstrap过程  
B = 10000  
result = vector("numeric", length = B)  
for(i in 1:B){  
  newData = sample(speed1, replace = TRUE)  
  result[i] = mean(newData)  
}  
  
# 计算新的置信区间  
CI = quantile(result, c(0.025, 0.975))  
CI  
  
# 计算相对于新均值的偏差  
CI - mean(speed1)
```

11 注意事项

- Bootstrap 方法对于非对称分布很有用。
- 对于近似正态分布的数据，Bootstrap 和 t 分布方法结果应该相似。
- 在处理极端值时，可以考虑只保留正值，但要注意这可能会影响结果的解释。
- 绘制直方图和比较不同方法的结果有助于理解数据分布和方法的适用性。