Benedict Kongyir Qualifying Exams

Rules: you may only use any textbooks and any articles you have cited already in your dissertation to solve problems. You may not use AI, search engines, or any other software on any questions.

1. Please revise the draft to be more scientific. I do enjoy your writing style, but unfortunately a statistics dissertation has to be more scientific, i.e. more formal. Provide annotations of your changes (ex. cross out deleted words, identify additions). I've provided some tips on the Introduction to get you started.

   A. In the introduction, the following words are not well defined or not scientific.

      a. Silent threats

      b. Faulty findings

      c. Seriously

      d. Real

      e. True

   B. In the introduction the following statements are speculative without references, and hence non-scientific.

      a. affecting everything from medical decisions and business strategies to public policy.

      b. Violations of normality can lead to inflated Type I error rates or reductions in statistical power, thereby undermining the validity of conclusions in applied research (You can say things like, "as we shall see" if it's something that you'll prove later on)

2. Suppose $X_1, X_2, \ldots, X_n$ are iid F. The goal is to test whether the data have mean 1. Consider the following procedure

   Procedure: Use 10% of the data as training data to test for normality with type I error rate 0.05.

   (i) If normality is rejected, apply a nonparametric test to determine if the mean is 1 using the remaining test data.

   (ii) If normality is not rejected, test if the mean is 1 using the remaining test data and a hypothesis test that assumes normality.

   (a) Provide a well documented R function that implements the procedure above. Recall you may not use AI.

   (b) Provide a well documented R function that extends the procedure above to 10 fold cross validation.

   (c) Prove that the procedure (in a) is valid if data are normally distributed and procedures in i. and ii. are valid.

(d) Provide a simulation study for $n = 30, 50, 100$ when data are generated under $\text{Exp}(1)$ and also $N(1, 1)$. You must document and write your own R code. I need to be able to copy and paste your code in R and recover your exact results. You must also concisely describe your simulation study.

3. Provide another simulation study that compare the procedure above to a procedure which just applies a nonparametric test to all the data. Consider the same settings as in d.

Solutions

1. A) I have rephrased paragraphs 1, 2, and 3 to make them more formal. I deleted informal words such as Silent threats and Faulty findings, and replaced Seriously with significantly. I have also replaced words like "troublesome" with a more appropriate, clear word, challenging. "Real" is replaced with practical in paragraph 11. The statement containing the word True in the phrase 'true violations' has been deleted by rephrasing the statement to read, This creates a particularly challenging scenario; precisely when researchers most need to identify non-normality to avoid inappropriate use of parametric methods, the tests are least capable of doing so.

   B) The following phrases have been rephrased.

   affecting everything from medical decisions and business strategies to public policy is rephrased and backed with a proper reference from literature to "they may impact the evaluation of medical treatments, the efficacy of business interventions, and public policy (Ioannidis, 2005)."

   Violations of normality can lead to inflated Type I error rates or reductions in statistical power, thereby undermining the validity of conclusions in applied research is also rephrased and backed with a proper reference from literature to "As we shall demonstrate in our simulation study in chapter 2, violations of normality contributes to inflation of type I error rates and or loss in statistical power for many common downstream procedure, thereby undermining validity of conclusions in statistical research (Rochon et al., 2012)."

2. (a) The following R function implements the procedure in 2 i, and ii.

```
# ------------------------------------------------------
# a) R function for adaptive one-sample location test   #
# ------------------------------------------------------
adaptive_test <- function(x, alpha = 0.05, train_frac = 0.10){
  # randomly choose training set
  train_index <- sample(1:length(x),
  floor(train_frac * length(x)), replace = FALSE)
  # Training data for normality check
  x_train <- x[train_index]
  # Test data for location test
  x_test <- x[-train_index]
  # SW-test on training data
  if(shapiro.test(x_train)$p.value < alpha){
```

```
    # Non-normal: Use Wilcoxon sign rank test
    pval <- wilcox.test(x_test, mu = 1)$p.value
  }else{# Normal: use t-test
    pval <- t.test(x_test, mu = 1)$p.value
  }
  return(pval)
}
```

(b) The following R function extends the procedure in 2 i, and ii to 10-fold cross-validation. Performing the 10-folds cross-validation results to 10 p-values for the downstream test for each iteration, and since we only need just a single p-value for the decision, we need to find a way to appropriately obtain a single p-value from the 10 p-values. I tried taking the average, minimum, etc. but none of them worked. Finally, I decided that if all null hypotheses were true, what's the probability that the smallest p-value we observed would be as extreme or more extreme than what we actually saw? So I decided to take the minimum among all the 10 p-values, but then apply Bonferroni type correction to account for the multiple testing finally obtaining a single combined p-value as:

$$p_{\text{combined}} = 1 - (1 - p_{\min})^{10}$$

where $p_{\min} = \min(p_1, \ldots, p_{10})$

```
# ---------------------------------------------
# b)    10-fold cross-validated adaptive test  #
# ---------------------------------------------
adaptive_test_cv <- function(x, alpha = 0.05, K = 10){
  # Create K folds
  folds <- sample(rep(1:K, length.out = length(x)))
  # Store p-values from each fold
  pvals_fold <- numeric(K)
  for(k in 1:K){
    # Current test fold
    test_index <- which(folds ==k)
    # Remaining as training
    train_index <- setdiff(1:length(x), test_index)
    # Define Training and test sets
    x_train <- x[train_index]
    x_test <- x[test_index]
    if(shapiro.test(x_train)$p.value < alpha){ # Non-normal
      pvals_fold[k] <- wilcox.test(x_test, mu = 1)$p.value
    }else{# Normal
      pvals_fold[k] <- t.test(x_test, mu = 1)$p.value
    }
  }
  ## Combine the K p-values with Minimum P-value method
  p_min <- min(pvals_fold)
```

```
  p_combined <- 1 - (1 - p_min)^K
  return(p_combined)
}
```

(c) *Proof.* Suppose $\mathcal{D} = \{X_1, \ldots, X_n\}$ is an i.i.d. sample from a normal distribution. Consider the partition of the data where $\mathcal{D}_T$ represent the 10% used for normality pre-test and $\mathcal{D}_S$ the 90% used for the downstream test. Since the X_i are i.i.d., $\mathcal{D}_T$ and $\mathcal{D}_S$ are independent under H_0.

Now, consider testing $H_0 : \mu = 1$. Define $A = \{\text{normality rejected on } \mathcal{D}_T\}$. Let ***t-test*** and ***w-test*** denote procedures in i. and ii. respectively.

Suppose that the data are normally distributed and procedures in i. and ii. are valid. Let α denotes the nominal level of the downstream test. By the independence of $\mathcal{D}_T$ and $\mathcal{D}_S$, the conditional Type I error rates satisfy:

$$P(\text{reject } H_0 \mid A^c, H_0) = P(\textit{t-test} \text{ rejects } H_0 \mid H_0) \leq \alpha$$
$$P(\text{reject } H_0 \mid A, H_0) = P(\textit{w-test} \text{ rejects } H_0 \mid H_0) \leq \alpha$$

By the law of total probability the overall type I error rate can be written as:

$$\begin{aligned} P(\text{reject } H_0 \mid H_0) &= P(\text{reject } H_0 \cap A^c \mid H_0) + P(\text{reject } H_0 \cap A \mid H_0) \\ &= P(\text{reject } H_0 \mid A^c, H_0) \cdot P(A^c) + P(\text{reject } H_0 \mid A, H_0) \cdot P(A) \\ &\leq \alpha \cdot P(A^c) + \alpha \cdot P(A) \\ &= \alpha[P(A^c) + P(A)] \\ &= \alpha \end{aligned}$$

Therefore, the procedure (in a) is valid if data are normally distributed and procedures in i. and ii. are valid. □

(d) **Objective:** To evaluate the Type I error rate and power of adaptive testing procedures that use a pre-test for normality to decide between a parametric (one-sample t-test) and a nonparametric (Wilcoxon signed-rank test) test for the hypothesis $H_0 : \mu = 1$. The performance of these adaptive methods is compared against the standard nonparametric (Wilcoxon signed-rank test).

Data Generation and Replication: For each sample size $n \in \{30, 50, 100\}$ and distribution $F \in \{N(1, 1), \text{Exp}(1)\}$, the following steps are repeated for $N_{\text{sim}} = 1000$ replications:

i. Generate a sample $X_1, \ldots, X_n \overset{\text{iid}}{\sim} F$.

ii. Apply Shapiro–Wilk test for normality at level $\alpha = 0.05$ on 10% of the data.

iii. Perform a one–sample t-test to test $H_0 : \mu = 1$ using the remaining data if normality is satisfied; otherwise apply a Wilcoxon Sign Rank test.

iv. For part (b), randomly split the data into 10 folds.

v. Apply Shapiro–Wilk test for normality at level $\alpha = 0.05$ using 9 of the folds.

vi. If normality is satisfied for all the 9 folds, perform the one sample t-test on the remaining one-fold, otherwise, apply the Wilcoxon Sign Rank test.

vii. Obtain a single p-value from the 10-folds p-values from the downstream procedure as:

$$p_{\text{combined}} = 1 - (1 - p_{\min})^{10}$$

where $p_{\min} = \min(p_1, \ldots, p_{10})$

viii. for problem 3, Perform the ordinary Wilcoxon signed rank test to the full sample.

ix. For each procedure, sample size, and distribution, calculate the probability of type I error rate as $\hat{\alpha} = \frac{1}{N_{sim}} \sum_{i=1}^{N_{sim}} I(\text{ p-value}_i < \alpha)$

Table 1 indicates that both adaptive procedures are approximately valid for normal distributions. Type I error rates are however inflated for adaptive procedure where we used 10% of the data for pre-testing and the remaining 90% for adaptive downstream test. For the adaptive cross-validation method, Type I error rates are controlled for smaller sample sizes such as 30, and 50, but inflated for larger sample size, 100.

	Normal		Exponential	
n	Adaptive	CV	Adaptive	CV
30	0.0478	0.0477	0.0808	0.0120
50	0.0513	0.0472	0.0905	0.0006
100	0.0489	0.0493	0.1913	0.1710

Table 1: Estimated Type I error rates for the adaptive and CV procedures under normal and exponential distributions.

Run R codes

```
> # ------------------------------------------------------------
> # d) Implementation:Simulation study for Exp(1) and N(1,1)
> # ------------------------------------------------------------
> # Define run controls
> set.seed(123)
> Nsim         <- 1e4 # Number of simulations
> alpha        <- 0.05 # Significance level
> sample_size  <- c(30, 50, 100) # Sample sizes to test
> distributions <- c("normal", "exponential") # Distributions
> # ------------------------------------------------------
> #   data generation function
> # ------------------------------------------------------
> generate_data <- function(n, dist) {
```

```
+   dist <- tolower(dist)
+   if (dist == "normal") {
+     x <- rnorm(n, mean = 1, sd = 1)
+   } else if (dist == "exponential") {
+     x <- rexp(n, rate = 1)
+   }
+   return(x)
+ }
> # Initialize result matrices
> type1_adaptive <- matrix(NA,
+                 nrow = length(sample_size),
+                 ncol = length(distributions),
+                 dimnames = list(sample_size, distributions))
> type1_adaptive_cv <- type1_adaptive
>
> # loop through each sample size and each distribution
> for (i in seq_along(sample_size)) {
+   n <- sample_size[i]
+   for (j in seq_along(distributions)) {
+     dist <- distributions[j]
+     # store all p-vales
+     pvals1 <- numeric(Nsim)
+     pvals2 <- numeric(Nsim)
+     # calculate p-values for each procedure
+     for (s in 1:Nsim) {
+       x <- generate_data(n, dist = dist)
+       pvals1[s] <- adaptive_test(x, alpha = alpha,
            train_frac = 0.10)
+       pvals2[s] <- adaptive_test_cv(x, alpha = alpha, K = 10)
+     }
+     # calculate type I error rates
+     type1_adaptive[i, j]  <- mean(pvals1 < alpha)
+     type1_adaptive_cv[i, j] <- mean(pvals2 < alpha)
+   }
+ }
> # print results
> type1_adaptive
    normal exponential
30  0.0478      0.0808
50  0.0513      0.0905
100 0.0489      0.1913
> type1_adaptive_cv
    normal exponential
30  0.0477      0.0120
50  0.0472      0.0006
```

```
    100 0.0493      0.1710
```

3. The following simulation studies compare the procedures in 2(d) to the nonparametric Wilcoxon Sign Rank test that does not apply any pre-test adaptive rule. From Table 2, we see that both adaptive procedures have lower probability of Type I error rates compared to the ordinary Wilcoxon Sign Rank test when samples come from exponential distribution. This is because the Wilcoxon Sign Rank test assumption of symmetry is violated for exponential distribution. Thus, the Wilcoxon Sign rank test targets the wrong center. For samples from normal distribution, all tests are valid.

	Normal			Exponential		
n	Adaptive	CV	Wilcoxon	Adaptive	CV	Wilcoxon
30	0.0478	0.0477	0.0456	0.0808	0.0120	0.1527
50	0.0513	0.0472	0.0487	0.0905	0.0006	0.2196
100	0.0489	0.0493	0.0510	0.1913	0.1710	0.3855

Table 2: Estimated Type I error rates for the adaptive, cross-validated adaptive (CV), and Wilcoxon procedures under normal and exponential distributions.

Run R codes for Comparing procedures above

```
> # ------------------------------------------------
> # 3 compare wilcoxon test to adaptive procedures
> # ------------------------------------------------
> set.seed(123)
> type1_wilcox <- matrix(NA,
+                        nrow = length(sample_size),
+                        ncol = length(distributions),
+                        dimnames = list(sample_size, distributions))
>
> for (i in seq_along(sample_size)) {
+   n <- sample_size[i]
+   for (j in seq_along(distributions)) {
+     dist <- distributions[j]
+     pvals3 <- numeric(Nsim)
+     for (s in 1:Nsim) {
+       x <- generate_data(n, dist)
+       pvals3[s] <- wilcox.test(x, mu = 1)$p.value
+     }
+     type1_wilcox[i, j] <- mean(pvals3 < alpha)
+   }
+ }
>
> # organize results in tables
> compare_methods <- list(
+   normal = cbind(
```

```
+     Adaptive    = type1_adaptive[ , "normal"],
+     Adaptive_cv = type1_adaptive_cv[ , "normal"],
+     Wilcoxon    = type1_wilcox[ , "normal"]
+   ),
+   exponential = cbind(
+     Adaptive    = type1_adaptive[ , "exponential"],
+     Adaptive_cv = type1_adaptive_cv[ , "exponential"],
+     Wilcoxon    = type1_wilcox[ , "exponential"]
+   )
+ )
> # print results
> compare_methods$normal
    Adaptive Adaptive_cv Wilcoxon
30    0.0478      0.0477   0.0456
50    0.0513      0.0472   0.0487
100   0.0489      0.0493   0.0510
> compare_methods$exponential
    Adaptive Adaptive_cv Wilcoxon
30    0.0808      0.0120   0.1527
50    0.0905      0.0006   0.2196
100   0.1913      0.1710   0.3855
```

References

John P. A. Ioannidis. Why most published research findings are false. *PLoS Medicine*, 2(8):e124, 2005.

Justine Rochon, Matthias Gondan, and Meinhard Kieser. To test or not to test: Preliminary assessment of normality when comparing two independent samples. *BMC medical research methodology*, 12:1–11, 2012.