Assignment 8 solutions

EMATM0061: Statistical Computing and Empirical Methods, TB1, 2024

Introduction

This is the 8th assignment for Statistical Computing and Empirical Methods. This assignment is mainly based on Lectures 21, 22, 23 and 24 (see the Blackboards).

Load the tidyverse package:

```r
library(tidyverse)
```

1. A chi-squared test of population variance

Suppose we have an i.i.d. sample $X_1, \cdots, X_n \sim \mathcal{N}(\mu, \sigma^2)$ and a conjectured value for the population variance σ_0^2. We wish to test the null hypothesis that σ_0^2.

(Q1)

Implement a function called “chi_square_test_one_sample_var” which takes as input a sample called ““sample”” and a null value for the variance called “sigma_square_null”.

Answer

```r
chi_square_test_one_sample_var <- function(sample, sigma_square_null){
  sample <- sample[!is.nan(sample)] # remove any missing values
  n <- length(sample) # sample length
  #. compute test statistic
  chi_squared_statistic <- (n-1)*var(sample)/sigma_square_null
  # compute p-valu e
  p_value <- 2*min(pchisq(chi_squared_statistic, df=n-1),
                   1-pchisq(chi_squared_statistic, df=n-1))
  return (p_value)
}
```

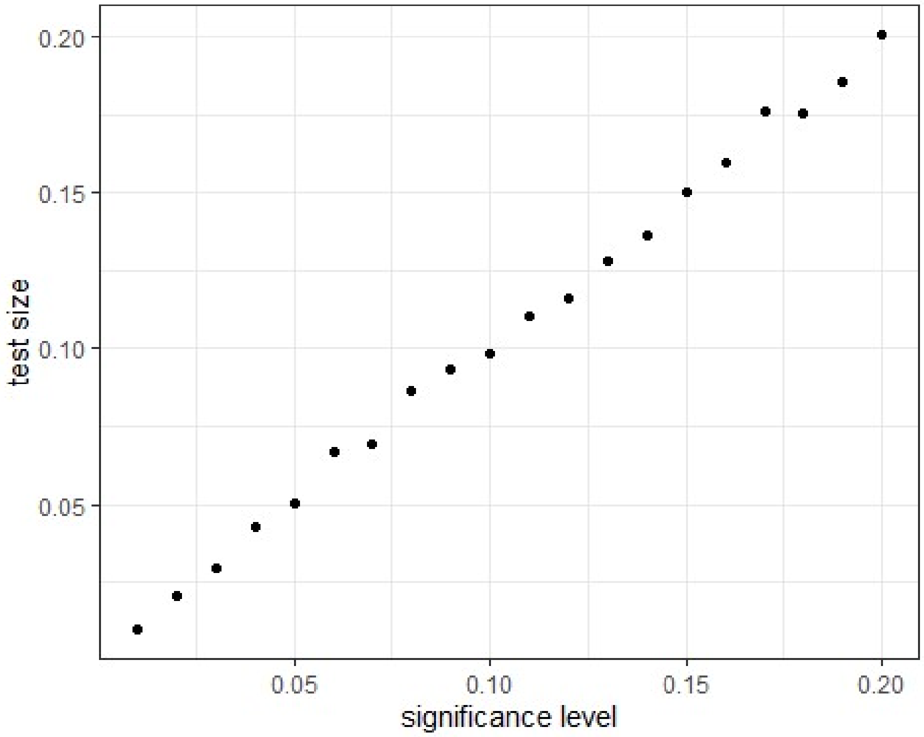
(Q2)

Conduct a simulation study to see how the size of the test varies as a function of the significance level. You can consider a sample size of 100, $\mu = 1, \sigma^2 = 4$.

Answer

```r
trials <- seq(10000)
sample_size <- 100
```

```
significance_levels <- seq(0.01, 0.2, 0.01)

mu_0 <- 1
sigma_0 <- 2

set.seed(0)

df_simulated_var_test <- crossing(trials=trials,

significance_levels=significance_levels) %>%
  mutate(sample=map(trials,~rnorm(sample_size, mean=mu_0, sd=sigma_0))
) %>%
  mutate(p_value = map_dbl(sample, ~chi_square_test_one_sample_var(.x,
sigma_0^2) ) ) %>%
  mutate(reject=(p_value<significance_levels))


df_test_size <- df_simulated_var_test %>%
  group_by(significance_levels) %>%
  summarise(test_size=mean(reject))

df_test_size %>% ggplot( aes(x=significance_levels, y=test_size) ) +
geom_point() +
  theme_bw() + xlab('significance level') + ylab('test size')
```

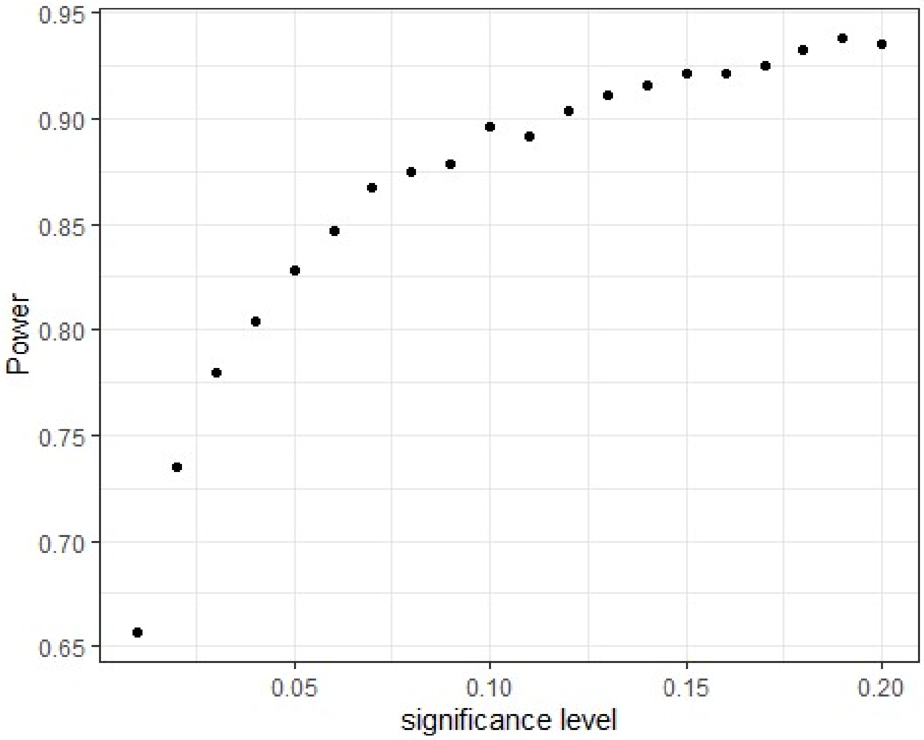


(Q4)

Conduct a simulation study to see how the statistical power of the test varies as a function of the significance level. You can consider a sample size of 100, $\mu = 1, \sigma^2 = 6$ and $\sigma_0^2 = 4$.

Answer

```
trials <- seq(10000)
sample_size <- 100
significance_levels <- seq(0.01, 0.2, 0.01)

mu_0 <- 1
sigma_0 <- 2
sigma <- sqrt(6)

set.seed(0)

df_simulated_var_test <- crossing(trials=trials,

significance_levels=significance_levels) %>%
  mutate(sample=map(trials,~rnorm(sample_size, mean=mu_0, sd=sigma)) )
%>%
  mutate(p_value = map_dbl(sample, ~chi_square_test_one_sample_var(.x,
sigma_0^2) ) ) %>%
  mutate(reject=(p_value<significance_levels))


df_power <- df_simulated_var_test %>%
  group_by(significance_levels) %>%
  summarise(power=mean(reject))

df_power %>% ggplot( aes(x=significance_levels, y=power) ) +
geom_point() +
  theme_bw() + xlab('significance level') + ylab('Power')
```



(Q5)

Load the “Palmer penguins” library and extract a vector called ““bill_adelie”” consisting of the bill lengths of the Adelie penguins belonging to the Adelie species.

Suppose we model the sequence of bill lengths as a sample of independent and identically distributed Gaussian random variables $X_1, \cdots, X_n \sim \mathcal{N}(\mu, \sigma^2)$ with a population mean μ and population standard deviation σ.

Now apply your function ““chi_square_test_one_sample_var”” to test the null hypothesis that the population standard deviation is 3 mm at a significance level of $\alpha = 0.1$.

Answer

```
library(palmerpenguins)
bill_adelie_df <- penguins %>% filter(species=='Adelie') %>%
  select(bill_length_mm) %>% drop_na()

bill_adelie <- pull(bill_adelie_df)
# you can also use: bill_adelie <- bill_adelie_df$bill_length_mm

chi_square_test_one_sample_var(bill_adelie, 3^2)

## [1] 0.05196711
```

Therefore, given significance level 0.1, we should reject the null hypothesis.

2. Obstacles to valid scientific inference

(Q1)

The following concepts have been introduced in Lecture 21. For each of the following concepts, give

(A) an explanation of the concept and
(B) an example of a situation (real or hypothetical) where they create a barrier to drawing scientific conclusions based on data.

You are encouraged to discuss these concepts with your colleagues.

1. Measurement distortions
2. Selection bias
3. Confounding variables

Answer

1. Measurement distortions

Measurement distortions occur whenever there is a mismatch between the quantities recorded within the data and the true variable of interest.

Example: Suppose we are investigating the effect of a new type of feed on chickens. For the purpose of the experiment a large sample of chickens is split into two groups. One group is given feed type A, and the other feed type B. After some time has elapsed, the chicken's weights are all measured. Suppose the measuring equipment used to weight the chickens given feed type A had a downward bias. On the other hand, the measuring device used to weight the chickens given feed type B has no such bias. We could then observe a data set where the recorded weights for chickens receiving feed source B typically exceed those receiving feed type A. However, this difference is purely an artefact of the faulty measurement equipment.

2. Selection bias

Selection bias occurs whenever the sample included in the analysis misrepresents the underlying population of interest.

a) Sample bias: When some members of the population are more likely to be selected than others.
b) Self selection bias: Occurs whenever people decide whether or not they should be assigned to a particular group.
c) Attrition bias: Occurs whenever the sample is distorted by people leaving the study.

d) Post-hoc selection: Occurs when a subset of the data is chosen based on the sample itself.

Example: A classic historical example of sample bias is the Literary Digest poll of 1936. In this people were asked about their voting intentions. The prediction was for a 57% victory for Republican Landon over Roosevelt. In fact Roosevelt won the election. The data was collected based upon surveys. These suveys were carried out based upon lists from telephone books and club membership. At the time both telephone ownership and club membership were indicators of wealth, so the sample was biased towards people with higher levels of wealth. At the time, higher levels of wealth were slightly more likely to vote Republican than Democrat. Hence, the selection bias resulted in a misleading result. Subsequent Gallop polls created more accurate results with smaller sample sizes.

3. Confounding variables

Suppose that we are interested in understanding the causal effect of an independent variable X on a dependent variable Y. A confounding variable is a third causal factor Z which effects both X and Y. This makes it difficult to disentangle causal effects from purely correlative behavior.

Example: As an example consider a scientific study into the causal effect of regular cardio-vascular exercise on longevity. Here a confounding variable could be someones overall interest in a healthy lifestyle. This is likely to effect someones via the causal effect of increased cardio-vascular exercise. However, it is likely that an interest in a healthy lifestyle will also effect the amount of fresh fruit and vegetables someone eats, for example. This may also have a causal effect on longevity. Hence, in light of the confounding variables it is difficult to distinguish a causal effect from a purely correlatative relationship.

3. Multivariate distributions and parameter estimation

Suppose that we have a sample of red-tailed hawks and we want to investigate the distribution of several features (Wing, Weight and Tail) of red-tailed hawks. We model the Wing, Weight and Tail with a multivariate Gaussian distribution. First, load the “Hawks” data frame from the “Stat2Data” library.

```r
library(Stat2Data)
data(Hawks)
```

(Q1)

Now extract a subset of the data frame called ““hawks_rt”” that contains only the rows corresponding to hawks from the “Red-tailed” (RT) species and three columns - ““Wing”“, “”Weight”“, “”Tail”“. Remove any rows of “”hawks_rt”” with missing values from one of the relevant columns.

Answer

```
hawks_rt <- Hawks %>% filter(Species=='RT') %>% select(Wing, Weight,
Tail)
```

(Q2)

Now, lets model the three features ““Wing”“, “”Weight”” and ““Tail”” with a multivariate Gaussian distribution. Suppose that your data frame “hawks_rt” consists of a i.i.d. sample $X_1, \cdots, X_n \sim \mathcal{N}(\mu, \Sigma)$. Here we model the three features ““Wing”“, “”Weight”” and ““Tail”” with a multivariate Gaussian distribution with population mean μ and population covariance matrix Σ. Compute the minimum variance unbiased estimates (MVUE) of the μ and Σ.

Answer

```
mu_mvue <- map_dbl(hawks_rt, ~mean(.x, na.rm=TRUE))
mu_mvue

##      Wing    Weight      Tail
##  383.3036 1094.4301  222.1490

Sigma_mvue <- cov(hawks_rt, use="complete.obs")
Sigma_mvue

##             Wing     Weight     Tail
## Wing    970.2672  1983.0489 148.8996
## Weight 1983.0489 35800.5187 705.7409
## Tail    148.8996   705.7409 211.4762
```

4. Basic concepts in classification

In lecture 24, we introduced some concepts in classification. Try to refresh your memory of these concepts by explaining them.

(Q1) Write down your explanation of each of the following concepts. Give an example where appropriate.

1. A classification rule
2. A learning algorithm
3. Training data
4. Feature vector
5. Label
6. Expected test error
7. Train error
8. The train test split

Answer

1. A classification rule

A classification rule, also known as a classifier, is a mapping $\phi: \mathcal{X} \rightarrow \mathcal{Y}$ where X is a space of feature vectors and Y is a set of discrete class labels, typically finite in number. As an example, consider a classifier which maps images of cats and dogs, represented as a vector, into a single label corresponding to whether the image is of a dog or a cat.

2. A learning algorithm

In the context of classification, a learning algorithm is an algorithm which takes as input a set of training data and outputs a classification rule. This classification rule may then be applied to previously unseen data. Examples include linear discriminant analysis and logistic regression.

3. Training data

Training data is a sequence of ordered pairs of the form $\left((X_1, Y_1), \cdots, (X_n, Y_n)\right)$ where X_i is a feature vector and Y_i is an associated class label. For example, when trying to learn a cat/dog image classifier, X_i corresponds to an image of a dog or cat and Y_i is the label - “cat” or “dog”.

4. Feature vector

A feature vector X is a vector containing one or more variables (also known as features) which we shall use to predict the class label, in the context of a classification rule. For example, in the context of the cat/dog image classifier, the feature vector is a vector corresponding to a grey-scale image where each element corresponds to the numerical value of a particular pixel. As another example, in the case of penguin classification, the feature vector is a vector containing several morphological properties such as the bill length, the flipper length, the weight etc.

5. Label

The label Y is a unique identifier that signifies that the corresponding feature vector X is associated with an item belonging to a particular class. For example Y could be 1 when associated with an image X of a dog and 0 when associated with an image X of a cat. The set of labels is in one-to-one correspondence with the number of classes eg. if there are three classes then there must be three labels.

6. Expected test error

The test error is the average number of mistakes a classification rule makes on unseen data. More precisely, suppose we have a classification rule $\phi: \mathcal{X} \rightarrow \mathcal{Y}$ then given a distribution P which describes the distribution over random pairs (X, Y) where X is a feature vector and Y is a label, the test error is

$$\mathcal{R}(\phi) = \mathbb{P}_{(X,Y)}[\phi(X) \neq Y].$$

This is typically estimated by the average error on a particular test data set.

7. Train error

The train error is the average number of mistakes made by a classifier on the training data. More precisely, suppose we have set of training data $\big((X_1, Y_1), \cdots, (X_n, Y_n)\big)$ where X_i and a classifier $\phi: \mathcal{X} \to \mathcal{Y}$, the train error is given by

$$\hat{\mathcal{R}}(\phi) = \frac{1}{n}\sum_{i=1}^{n}[\phi(X) \neq Y].$$

8. The train test split

The train test split refers to a split of your data set into two pieces - containing two groups of examples: The training data and the testing data. The training data is used as input into the learning algorithm. Based on the training data (and not the test data), the learning algorithm generates a classification rule. The test data plays no role in the learning of the classification rule. Instead, the test data is used to assess the classification rule’s performance on previously unseen data, not used within the training process. Hence, we compute the number of mistakes on the test data, which gives rise to an estimate for the expected test error.

5. The train test split

Suppose you want to build a classifier to predict whether a hawk belongs to either the “Sharp-shinned” or the “Cooper’s” species of hawks. The feature vector will be a four-dimensional row vector containing the weight, and the lengths of the wing, the tail and the hallux. The labels will be binary: 1 if the hawk is “Sharp-shinned” and 0 if the hawk belongs to “Cooper’s” species.

(Q1)

Begin by loading the “Hawks” data frame from the “Stat2Data” library. Now extract a subset of the data frame called ““hawks_total”” with five columns - ““Weight”“, “”Wing”“, “”Hallux”“, “”Tail”” and ““Species””. The data frame should only include rows corresponding to hawks from either the “Sharp-shinned” (SS) or the “Cooper’s” (CH) species, and not the “Red-tailed” (RT) species. Convert the Species column to a binary variable with a 1 if the hawk belongs to the sharp-shinned species and 0 if the hawk belongs to Cooper’s species. Finally, remove any rows with missing values from one of the relevant columns.

Answer

```
library(Stat2Data)
data(Hawks)
```

```
hawks_total <- Hawks %>% select( Weight, Wing, Hallux, Tail, Species)
%>%
  filter(Species=='SS' | Species =='CH') %>% drop_na() %>%
  mutate(Species=as.numeric(Species=='SS'))
```

(Q2)

Now implement a train test split for your ““hawks_total”” data frame. You should use 60% of your data within your training data and 40% in your test data. You should create a data frame consisting of training data called ““hawks_train”” and a data frame consisting of test data called ““hawks_test””. Display the number of rows in each data frame.

Answer

```
num_total <- hawks_total %>% nrow() # number of penguin data
num_train <- floor(num_total*0.6) # number of train examples
num_test <- num_total-num_train # number of test samples
set.seed(123) # set random seed for reproducibility

test_inds <- sample(seq(num_total),num_test) # random sample of test
indicies
train_inds <- setdiff(seq(num_total),test_inds) # training data
indicies

hawks_train <- hawks_total %>% filter(row_number() %in% train_inds) #
train data
hawks_test <- hawks_total %>% filter(row_number() %in% test_inds) #
test data
hawks_train %>% nrow()

## [1] 194

hawks_test%>%nrow()

## [1] 130
```

(Q3)

Next extract a data frame called ““hawks_train_x”” from your training data (from ““hawks_train””) containing the feature vectors and no labels. In addition, extract a vector called ““hawks_train_y”” consisting of labels from your training data. Similarly, create data frames called ““hawks_test_x”” and ““hawks_test_y”” corresponding to the feature vectors and labels within the test set, respectively.

Answer

```
hawks_train_x <- hawks_train %>% select(-Species) # train feature
vectors
```

```
hawks_train_y <- hawks_train %>% pull(Species) # train labels
hawks_test_x <- hawks_test %>% select(-Species) # test feature vectors
hawks_test_y <- hawks_test %>% pull(Species) # test labels
```

(Q4)

Now let's consider a very simple (and not very effective) classifier which entirely ignores the feature vectors. The classifier is defined as follows.

Let $\hat{y} \in \{0,1\}$ be a fixed value. For any input x, the output of classifier is always the fixed value $\hat{y}$. In other words, your classifier is of the form $\phi_{\hat{y}}(x) \equiv \hat{y}$ for all $x \in \mathbb{R}^4$ (note that $\hat{y}$ is fixed).

Based on the training data from the previous questions, your task is to choose a value $\hat{y}$ from $\{0,1\}$ such that the classifier has a smaller training error.

Answer

```
# only two possible phi in this case: one for y_hat=0, the other for
y_hat=1
train_error_phi_0 <- mean(abs(hawks_train_y-0))
train_error_phi_1 <- mean(abs(hawks_train_y-1))

# finding y-hat with minimum training error
if(train_error_phi_0<train_error_phi_1){
  y_hat<-0
  }else{
    y_hat<-1
  }

# alternatively, you can also compute y_hat by (why?):
# y_hat<-as.numeric(mean(hawks_train_y)>=0.5)

y_hat

## [1] 1
```

(Q5)

Next compute the train and test error of $\phi_{\hat{y}}$.

In general, $\phi_{\hat{y}}$ performs poorly, as it does not use any information of the feature vector. However, in the example, the train error and test error seems relatively low (much less than 50%). Try to explain why the errors are relatively low. In which cases we might have a error of $\phi_{\hat{y}}$ close to 50%?

Answer

```
train_error_simple <- mean(abs(y_hat-hawks_train_y)) # train error
test_error_simple <- mean(abs(y_hat-hawks_test_y)) # test error
train_error_simple

## [1] 0.2371134

test_error_simple

## [1] 0.1769231
```

The very low error rates demonstrate that the classes are very imbalanced. Here the large majority of the hawks in the data set belong to the sharp-shinned species. If the two classes were more balanced we wouldn’t be able to do much better than a 50% error rate.