Answer for QQ Test

Huang Tao

2020/3/11

#### A: Consider tossing a fair coin 10n times. The coin is totally fair: probabilities of its head and of its tail in each trial are 50%. Compute probability observing the head for (1) 50% and (2) 70% of the trials when n is 1, 2, and 3.

* This is a Binomial Distribution. The probability of an event happened time in trails can be calculated using this formula:

$P = \tbinom{m}{n}\*p^n$

* When n=3 (in 1000 trails), the P is very small, approach to 0.
* My understanding is that: when n is small, the distrubition is like a historgram, we can calculate the event’s probability; while when n is big, it is like a PDF(probability density fuction), every **single specific event** ’s probability is 0.

calculate\_binomial <- function(a, b){  
 n = 10^b  
 m = a\*n  
 p = (1/2^n) \*factorial(n)/(factorial(m)\*factorial(n-m))  
 print(p)   
  
}  
  
for (i in c(0.5,0.7)){  
 for(j in 1:3) {  
 calculate\_binomial(i,j)  
 }  
}

## [1] 0.2460938  
## [1] 0.07958924  
## [1] NaN  
## [1] 0.1171875  
## [1] 2.317069e-05  
## [1] NaN

#### B:Consider a distribution of annual household income in some country where a few people are super rich. Why is it often recommended to use “median” rather than “average” to represent population in the country?

* This super rich people can be **outliers** of the data. And the normally-called “average”(**mean**) is very sensitive to outliers, while the **median** and **mode** are not.
* We can simulate these two Scenarios.

# marjority salary-earning peopele  
mean1 <- 10000 # US dollars  
sd1 <- 500 # US dollars  
n1 <- 10^5 # thousands people  
  
# top rich people   
mean2 <- 10^9 # billionaires  
sd2 <- 10^7  
n2 <- 1 # refers to the top 1 thousands rich people  
  
  
income1 <- rnorm(n1, mean1, sd1)  
income2 <- rnorm(n2,mean2,sd2)  
  
average\_income\_exclude = mean(income1)  
median\_income\_exclude = median(income1)  
  
average\_income\_include = mean( c(income1, income2))  
median\_income\_include = median( c(income1, income2))  
  
print(average\_income\_exclude)

## [1] 10002.6

print(median\_income\_exclude)

## [1] 10001.87

print(average\_income\_include)

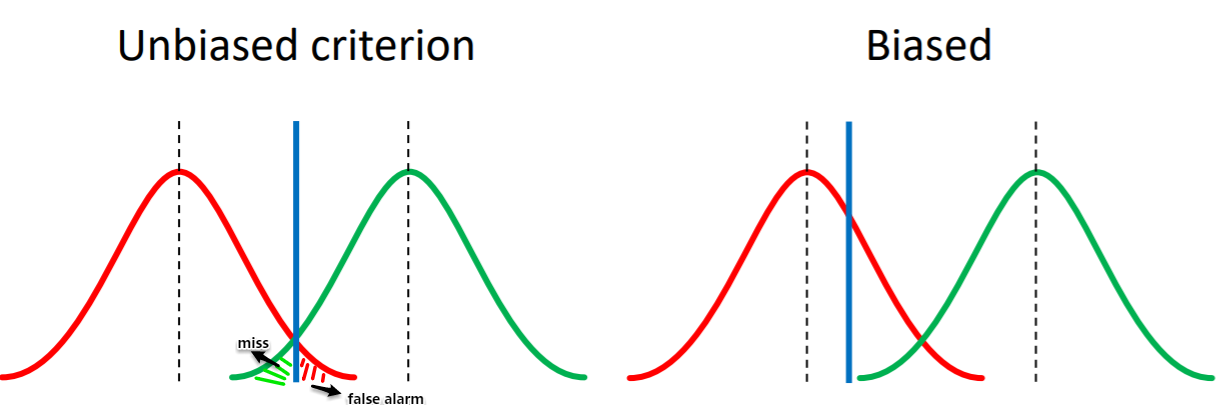
## [1] 19775.71

print(median\_income\_include)

## [1] 10001.88

#### C:Consider a signal detection experiment of a binary task. Why is a participant often encouraged to use a neutral criterion (rather than an extremely-biased criterion) so that hit and correct-rejection rates become roughly the same?

* The fundamental reason is that: there is a trade-off between false alarm and miss.
* In the following picture the red line is noise, the green line stands for signal. If you move the criterion to the left(liberal), you will have higher hit rates with the price of increased false alarm; if you move the criterion to the right(conservative), you will have higher correct-reject rate with the price of increased miss rate.

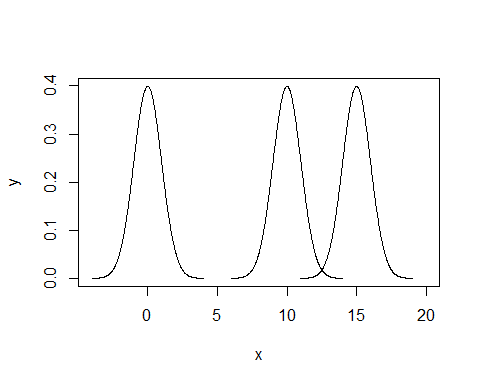


#### D: Consider running a signal detection experiment of a binary task. Numbers of both signal and noise trials were 100 (200 trials in total) in each session of the experiment. You are interested in difference of performance (dʹ) between conditions-A and -B. The conditions are blocked in each session. There is one session for each condition. Based on some theory or meta-analysis, Expected performance for condition-A is dʹ=10 and for condition-B is dʹ=15.

* The key point is: the of condition A and condition B are 10 and 15 respectively. With such **huge** values, it means the signal and noise are very well separated, and extremely easy to be distinguished by participants.
* **In the two 100 trails experiments of condition A and condition B, participants’ performance will be identical: that is 100% hit rate and 0% false alarm.**
* This can be seen form the ROC curve.
* This can also be verified by the last homework—— even when , the hit rate approaches to 100% under the unbiased criterion.

**The following simulation is meaningless.**

sd=1  
  
x <- seq(-4, 4, length=500)  
y <- dnorm(x,mean=0,sd)  
plot(x,y,type="l",xlim=c(-4, 20))  
  
mean1=10  
x1 <- seq(-4,4,length=500)\*sd + mean1  
y1 <- dnorm(x1,mean1,sd)  
lines(x1,y1)  
  
mean2=15  
x2 <- seq(-4,4,length=500)\*sd + mean2  
y2<- dnorm(x2,mean2,sd)  
lines(x2,y2)

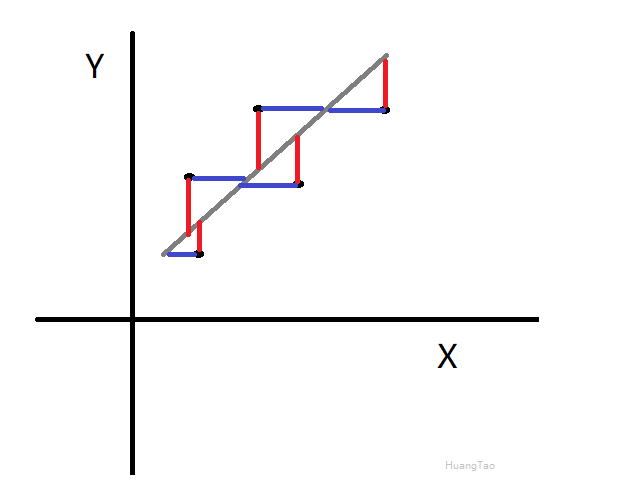


v1 <- rnorm(100, mean1, sd)  
v2 <- rnorm(100, mean2, sd)  
  
  
t.test(v1, v2, alternative = "two.sided", var.equal = TRUE)

##   
## Two Sample t-test  
##   
## data: v1 and v2  
## t = -35.865, df = 198, p-value < 2.2e-16  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -5.215022 -4.671424  
## sample estimates:  
## mean of x mean of y   
## 9.995046 14.938270

#### E:A result of regression analysis between two variables X and Y is unchanged no matter whether Y is formulated as a function of X (Y = aX + b) or X is formulated as a function of Y (X = cY + d). Namely, c =1/a and d = -b/a.

* Geometrically, there two scenarios are different.
* Y is formulated as a function of X (Y = aX + b): is to find the minimum squared of the length of **red** lines (if using least squared method)
* X is formulated as a function of Y (X = cY + d):is to find the minimum squared of the length of **blue** lines (if using least squared method)



### F:A result of regression analysis between two variable X and Y is identical with a result of principal component analysis.

At first glance, I try to just consturct two variable X, Y and do the regression and PCA. Then I think doing this is NOT meaningful.

My understanding:

* PCA and regression analysis are intrinsically different.
* For a dataset, if there are too many columns (too many attributes), we do PCA to reduce the columns. PCA is usually conducted if the dateset is **wide** , for example have 4 or more of variables(/columns/attributes).
* Regression analysis is usually **between two variables**——we try to find the relationship between the 2 varibales. For linear regression, theoretically, We just need two point(two observations/two rows) to draw a line. But in reality, We have a lot of observations(rows), and these observations has noise.
* In 2D situation (have two variable X, Y), these two seems to have some connections if we draw both geometrically, but they are **just completely different things.**
* The regression (least squared method) is to find the minimum squared of the length of **red** lines, while the PCA is to find the minimum squared of the length of **blue** lines (with the final goal of find the maximum value of the projections).
* If only the final line is horizontal, then the values of regression (Least squared method) and PCA maybe the same. But in this situation, there is no need to do PCA (the x axis is the direction of principal component), there is also no need to do regression (the correlation of X, Y is 0)
* Usually, we first use PCA to reduce demention to 2, then we using regression to find the relationship between these 2 variables.
* In data analysis perspective, we have a huge datset with numurous columns and rows, we first use PCA to reduce columns, then we use regression to reduce rows. At last, we get two columns with two rows—–that is the regressed line we got.

