P-hacking

TadamasaSawada, Mher Movsisyan

2023-09-01

Q2. P-hacking

The scripts (Q2_Intro1-3) show how p-hacking can inflate the probability of Type-1 error. Three techniques of p-hacking are implemented: 1. Multiple measurements 2. Multiple statistical tests 3. Data peeking (Optimal stopping) If statistical tests are conducted properly, the probability of Type-1 error is 0.05 (See AUA_DAE_00 Review_p.Rmd). But these techniques individually inflate the probability of Type-1 error. The script (Q2_Intro4) shows how (1) the multiple measurements and (2) the multiple tests can be combined. It shows how the combination even further inflates the probability of Type-1 error.

(Q2a) Revise the script (Q2_Intro4) for the combination of (1) the multiple measurements and (3) the data peeking (but not (2) the multiple tests) and report the inflated probability of Type-1 error (Q2b) Revise the script (Q2_Intro4) for the combination of (1) the multiple measurements, (2) the multiple tests, and (3) the data peeking and report the inflated probability of Type-1 error.

Note that Wilcoxon-Mann-Whitney test (WMW) and Kruskal-Wallis test (KW) are statistical tests that are similar to the t-test. They can be used to compare data in two conditions as the t-test does. The WMW and KW tests do not assume normality of populations (more precisely, normality of distributions of averages) while the t-test does. Thier results often correlate to one another but there are always some tolerance and this tolerance can occasionally cross the threshold (5%). https://en.wikipedia.org/wiki/Mann%E2%80%93Whitney_U_test https://en.wikipedia.org/wiki/Kruskal%E2%80%93Wallis_one-way_analysis_of_variance

(Q2_Intro1) Multiple Measurements

```
nSessions = 1000
nMeasurements = 5

nSamples = 100
countSignificant = 0

for(s in 1:nSessions)
{
  checkSignificant = FALSE
  for(m in 1:nMeasurements)
  {
    samplesA = rnorm(nSamples)
    samplesB = rnorm(nSamples)
    dTest = data.frame("Conditions" = c(rep('A',nSamples),rep('B',nSamples)),
                       "Data" = c(samplesA, samplesB) )

    resultT1  = t.test(samplesA, samplesB, paired = FALSE, var.equal = TRUE)

    pT1  = resultT1$p.value
```

```
    if(pT1<0.05)
      checkSignificant = TRUE
  }
  if(checkSignificant)
    countSignificant = countSignificant+1
}

countSignificant/nSessions
```

```
## [1] 0.238
```

(Q2_Intro2) Multiple Tests

```
nSessions = 1000

nSamples = 100
countSignificant = 0
for(s in 1:nSessions)
{
  samplesA = rnorm(nSamples)
  samplesB = rnorm(nSamples)
  dTest = data.frame("Conditions" = c(rep('A',nSamples),rep('B',nSamples)),
                     "Data" = c(samplesA, samplesB) )

  resultT1  = t.test(samplesA, samplesB, paired = FALSE, var.equal = TRUE)
  resultT2  = t.test(samplesA, samplesB, paired = FALSE, var.equal = FALSE)
  resultWMW = wilcox.test(samplesA,samplesB, alternative = "two.sided", paired = FALSE) # Wilcoxon Mann
  resultKW  = kruskal.test(Data ~ Conditions, data = dTest) # Kruskal Wallis test

  pT1  = resultT1$p.value
  pT2  = resultT2$p.value
  pWMW = resultWMW$p.value
  pKW  = resultKW$p.value

  if(pT1<0.05 | pT2<0.05 | pWMW<0.05 | pKW<0.05)
    countSignificant = countSignificant+1
}

countSignificant/nSessions
```

```
## [1] 0.057
```

(Q2_Intro3) Data peeking (Optional stopping)

```
nSessions = 1000

nSamples_min = 10 # The initial sample size
nSamples_max = 20 # Increasing the sample size one by one up to nSamples_max
countSignificant = 0
for(s in 1:nSessions)
{

  for(nSamples in nSamples_min:nSamples_max)
```

```
  {
    samplesA = rnorm(nSamples)
    samplesB = rnorm(nSamples)
    dTest = data.frame("Conditions" = c(rep('A',nSamples),rep('B',nSamples)),
                       "Data" = c(samplesA, samplesB) )

    resultT1  = t.test(samplesA, samplesB, paired = FALSE, var.equal = TRUE)

    pT1  = resultT1$p.value

    if(pT1<0.05)
    {
      countSignificant = countSignificant+1
      break
    }
  }
}

countSignificant/nSessions
```

```
## [1] 0.434
```

(Q2_Intro4a) Multiple Measurements and Data peeking

```
nSessions = 1000
nMeasurements = 5

nSamples = 100
countSignificant = 0


for(s in 1:nSessions)
{
  checkSignificant = FALSE
  for(m in 1:nMeasurements)
  {
    samplesA = rnorm(nSamples)
    samplesB = rnorm(nSamples)
    dTest = data.frame("Conditions" = c(rep('A',nSamples),rep('B',nSamples)),
                       "Data" = c(samplesA, samplesB) )

    resultT1  = t.test(samplesA, samplesB, paired = FALSE, var.equal = TRUE)

    pT1  = resultT1$p.value

    if(pT1<0.05)
    {
      checkSignificant = TRUE
      break
    }
  }

  if(checkSignificant)
```

```
    countSignificant = countSignificant+1
}

countSignificant/nSessions
```

```
## [1] 0.247
```

(Q2_Intro4b) Multiple Measurements and Multiple Tests

```
nSessions = 1000
nMeasurements = 5

nSamples = 100
countSignificant = 0


for(s in 1:nSessions)
{
  checkSignificant = FALSE
  for(m in 1:nMeasurements)
  {
    samplesA = rnorm(nSamples)
    samplesB = rnorm(nSamples)
    dTest = data.frame("Conditions" = c(rep('A',nSamples),rep('B',nSamples)),
                       "Data" = c(samplesA, samplesB) )

    resultT1  = t.test(samplesA, samplesB, paired = FALSE, var.equal = TRUE)
    resultT2  = t.test(samplesA, samplesB, paired = FALSE, var.equal = FALSE)
    resultWMW = wilcox.test(samplesA,samplesB, alternative = "two.sided", paired = FALSE) # Wilcoxon Ma
    resultKW  = kruskal.test(Data ~ Conditions, data = dTest) # Kruskal Wallis test
    pT1  = resultT1$p.value
    pT2  = resultT2$p.value
    pWMW = resultWMW$p.value
    pKW  = resultKW$p.value

    if(pT1<0.05 | pT2<0.05 | pWMW<0.05 | pKW<0.05)
      checkSignificant = TRUE
  }
  if(checkSignificant)
    countSignificant = countSignificant+1
}

countSignificant/nSessions
```

```
## [1] 0.244
```