DAVINCI VS PREMIERE

Comparació de temps de renderització entre diferents programes d'edició de vídeo

Desembre 2020

Grup 32
Jia Long Ji
Jiabo Wang
You Wu

1. RESUM

1.1 Objectiu

El nostre propòsit és comprovar si el temps de processament d'un vídeo varia segons el tipus de programa d'edició de vídeo. En el nostre cas, segons si es processa amb *Adobe Premiere Pro 2019*, un programa de pagament, o *Davinci Resolve 16*, un programa gratuït.

1.2 Mètodes

Vam escollir a l'atzar 30 vídeos dels 60 que teníem gravats, i cadascun es va renderitzar dues vegades, una utilitzant l'*Adobe Premiere Pro 2019* i l'altra amb el *Davinci Resolve*, mentre es cronometrava el temps a cada vegada.

1.3 Resultats

Després d'analitzar les dades obtingudes, fent els càlculs adients, hem conclòs que no podem rebutjar que el rendiment dels dos editors de vídeo, a l'hora de renderitzar vídeos, sigui diferent.

2. INTRODUCCIÓ

Avui en dia, el mercat ofereix múltiples opcions a l'usuari d'escollir entre els diferents softwares disponibles per al seu abast. Però, a vegades, de tantes alternatives que tenim no sabem per quin ens hem de decantar. El dubte més sovint que pot sorgir és si escollir software de pagament o, per contra, triar programes gratuïts.

El nostre estudi s'ha basat en la comparació del temps de processament d'un vídeo segons un programa d'edició de vídeo de pagament com *Adobe Premiere Pro 2019*, i una alternativa gratuïta com *Davinci Resolve* i veure'n si hi ha diferència o no.

3. MÈTODES

3.1 Recollida de dades

Per la recollida de dades, vam decidir gravar 60 vídeos qualsevols, ja sigui amb càmera, amb el mòbil, amb l'ordinador... i amb una durada d'entre 5 i 20 minuts. Per tal d'obtenir mostres aleatòries, vam escollir 30 vídeos dels 60 gravats de manera completament aleatòria, assignant un número diferent de l'1 al 60 a cadascun i, mitjançant un generador de números aleatoris, s'escollia el vídeo a partir del número generat.

Cadascun dels 30 vídeos seleccionats van ser processats dues vegades, una amb l'editor Premiere i l'altra amb l'editor Davinci. Evidentment, per tal de poder fer una anàlisi just, tots els renderitzats es van realitzar amb el mateix format de sortida (més detalladament, amb un format de codificació de vídeo H.264 i una configuració de *High Quality 1080p HD*, amb dimensions 1920x1080 i una velocitat de fotograma de 60FPS), treballant sempre amb el mateix ordinador, mantenint en tot moment un mateix consum de recursos, per tal d'evitar un afavoriment entre els editors de vídeo (per tant, durant cada procés de renderització, només teníem obert l'editor de vídeo sense cap altre software que ralentís el procés).

Per tal d'obtenir les mostres, és a dir, els temps de processament, basta amb cronometrar cada procés de renderització. Aquestes mostres es tracten de dades aparellades, ja que a partir de cada vídeo s'extreu un parell de mesures, és a dir, tenim dues mostres (temps de renderització a *Davinci Resolve* i *Adobe Premiere Pro*) aparellades per cada vídeo.

En quant a especificacions tècniques del nostre equip, en el nostre cas hem utilitzat un ordinador portàtil *Lenovo ideapad 330*, que té les següents característiques:

- Processador intel Core i7-8550U (4C, 1.8/4.0 GHz, 8 Mb)
- Memoria RAM 4GB Soldered + 4 GB DIMM DDR4-2133
- Almacenamiento 256 GB SSD
- Gràfica integrada Intel(R) UHD Graphics 620

3.2 Variables

Hem definit les variables:

- Y ~ temps de processament d'un vídeo
- X_{AP} ~ processament amb l'editor *Adobe Premiere Pro 2019*
- X_{DR} ~ processament amb l'editor *Davinci Resolve*

3.3 Pla estadístic

Comparació de mitjanes de mostres aleatòries aparellades ($\mu_{DR} = \mu_{AP}$)

1. **Variable**

   D = diferència de temps de renderització entre els editors de vídeo *Davinci Resolve* i *Adobe Premiere Pro*.

2. **Premisses**
   - Mostres aleatòries aparellades
   - Efecte additiu constant
   - Normalitat de D

3. **Hipòtesi**

   $\mu_D = \mu_{DR} - \mu_{AP}$

   $H_0: \mu_D = 0$

   $H_1: \mu_D \neq 0$

   μ_D és la mitjana poblacional de la diferència de temps de renderització entre els editors de vídeo *Davinci Resolve* i *Adobe Premiere Pro*.

   μ_{DR} és la mitjana poblacional de temps de renderització de l'editor de vídeo *Davinci Resolve*.

   μ_{AP} és mitjana poblacional de temps de renderització de l'editor de vídeo *Adobe Premiere Pro.*

   Es tracta d'una prova és bilateral, ja que volem saber si triguen o no el mateix temps de renderització de vídeo.

4. **Estadístic**

L’estadístic segueix una distribució t-Student t_{29} de 29 graus de llibertat.

$\hat{t} \sim t_{n-1}$ $\qquad \hat{t} = \frac{\bar{D}-\mu_D}{S_D/\sqrt{n}}$

5. **Estudi del P-valor i del punt crític (α = 0.05)**

Punt crític = $t_{29,0.975}$

Si el p-valor < α o el punt crític < estadístic podem rebutjar la hipòtesis nul·la.

6. **Interval de confiança**

$IC(\mu_D, 0.95) = \bar{D} \pm t_{n-1,1-\frac{\alpha}{2}} \sqrt{{S_D}^2/n}$

4. RESULTATS

A partir de les dades obtingudes, hem calculat els següents indicadors juntament amb el boxplot del conjunt:

```
summary(dades$Temps_Davinci_Resolver)
 Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 6.75    8.88   11.02   13.55   15.90   59.90

summary(dades$Temps_Adobe_Premiere)
 Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
2.130   6.532  12.620  13.876  14.330  42.050

> sd(dades$Temps_Davinci_Resolve)
[1] 9.817662

> sd(dades$Temps_Adobe_Premiere)
[1] 8.690621
```

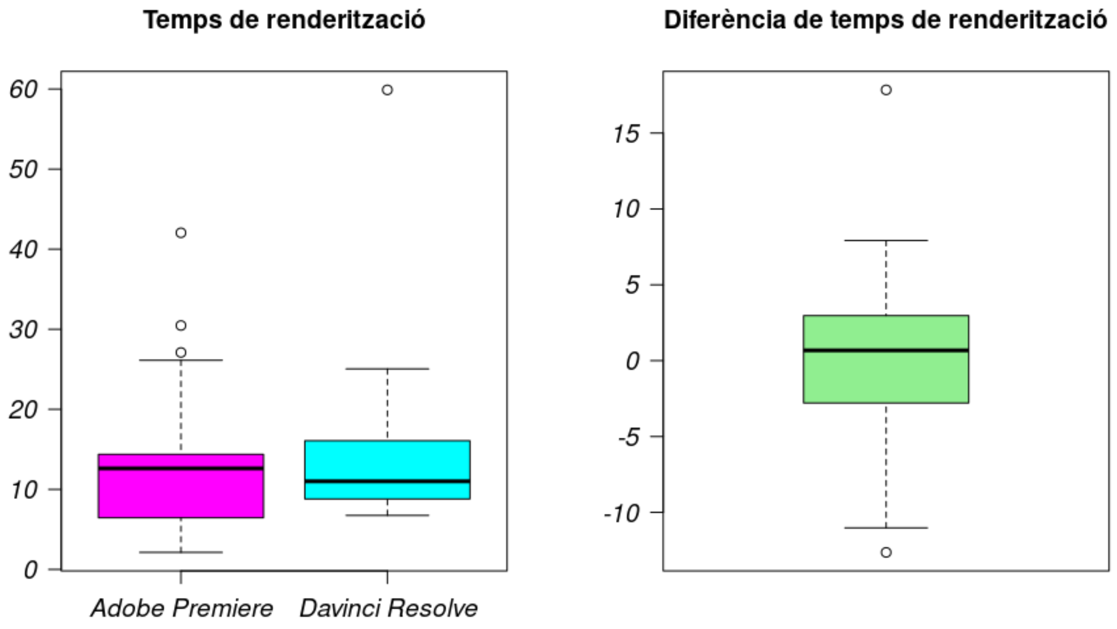

Fig. 1: Boxplot del temps de renderització amb Adobe Premiere i Davinci Resolve (esquerra) i boxplot de la variable diferència de temps de renderització (dreta).

4.1 Validesa de les premisses

Abans de la prova de comparació de mitjanes, és important assegurar que les mostres obtingudes compleixen les premisses. En el nostre cas és necessari comprovar l'efecte additiu constant i la Normalitat de la variable diferència de temps de renderització (Temps de renderització Davinci Resolve - Temps de renderització Adobe Premiere).

Variable Diferència

- **Normalitat**

  Per veure la normalitat de les dades hem fet un gràfic qqnorm amb R (figura 2) i veiem que tots els resultats observats (els punts) s'ajusten prou als valors teòrics de la Normal (la recta), tot i haver-hi alguns valors extrems, i per tant podem acceptar-ne la seva normalitat.

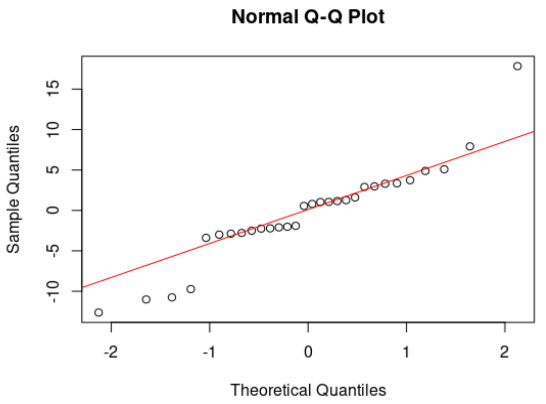

Fig. 2: Gràfic Q-Q Normal de la variable diferència D.

- **Efecte additiu**

  Després de realitzar el gràfic Bland-Altman (figura 3), hem vist que no compleix la premissa d'efecte additiu, ja s'observa un efecte multiplicatiu. I per tant, hem decidit fer servir una transformació logarítmica de les dades.

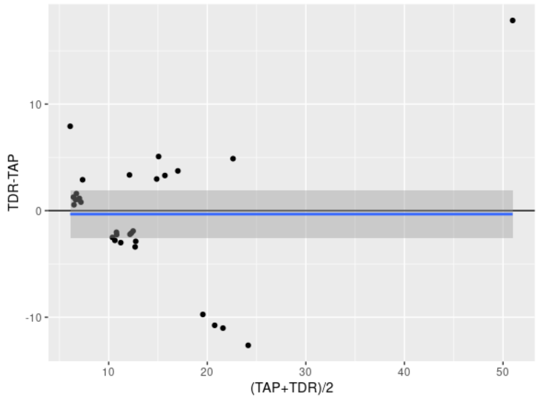

Fig. 3: Gràfic Bland-Altman de la variable diferència D.

Variable diferència de logaritmes

- **Normalitat**

  Tal com hem vist a l'observació anterior respecte a la normalitat de la variable diferència, una vegada transformada amb logaritmes, podem seguir acceptant la seva normalitat, ja que els resultats obtinguts segueixen ajustant-se prou bé als valors teòrics.

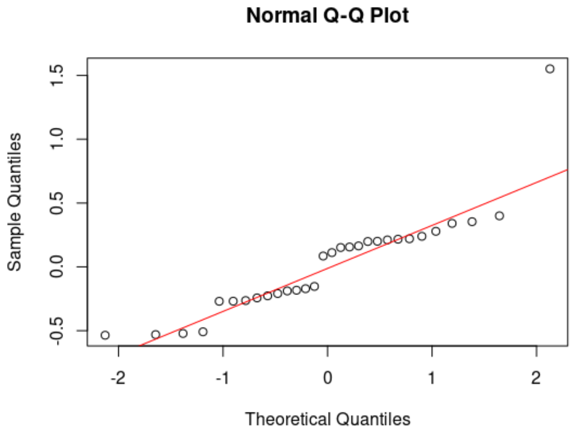

Fig. 4: Gràfic Q-Q Normal de la variable diferència D, amb transformació logarítmica.

- **Efecte additiu**

  Un cop canviat passat la variable diferència a variable diferència de logaritmes, sí que podem observar l'efecte additiu, tal i com es mostra la figura 5 la variabilitat de la diferència de logaritmes més o menys es manté de manera constant a mesura que augmenta el valor de l'eix horitzontal.

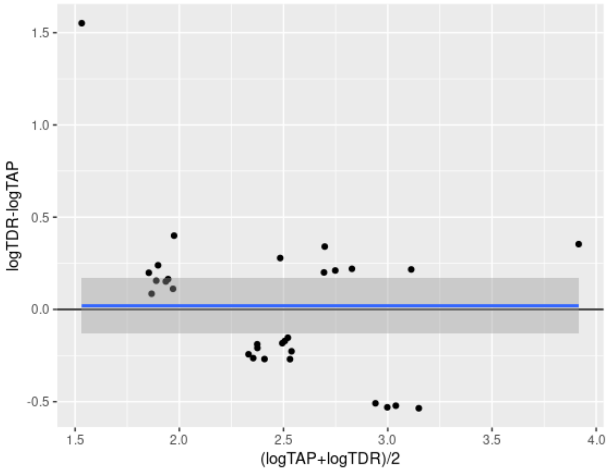

Fig. 5: Gràfic Bland-Altman de la variable diferència D, amb transformació logarítmica.

4.2 Comparació de mitjanes ($\mu_{DR} = \mu_{AP}$)

Una vegada verificades les premisses, ja podem fer el contrast d'hipòtesi sobre la comparació de mitjanes.

a) Hipòtesis

$\mu_D = \mu_{DR} - \mu_{AP}$

$H_0: \mu_D = 0$ $\quad\quad$ $H_1: \mu_D \neq 0$

En el nostre cas, hem escollit fer una prova bilateral, ja que volem saber si triguen o no el mateix temps de renderització de vídeo. La hipòtesi nul·la (H_0) és que la diferència de temps de renderització sigui 0, i la hipòtesi complementària és que la diferència de temps sigui diferent de 0 (H_1).

b) Estadístic, distribució i premisses

L'estadístic segueix una distribució t-Student t_{29} amb 29 graus de llibertat:

$\hat{t} \sim t_{n-1}$

$$\hat{t} = \frac{\overline{D} - \mu_D}{S_D/\sqrt{n}}$$

Les premisses són:

- Mostra aleatòria aparellada
- Premissa de Normalitat de D
- Premissa de l'efecte additiu

c) Càlcul de l'estadístic, punt crític i p-valor

Tal com hem vist a l'apartat anterior, hem hagut d'utilitzar la transformació logarítmica sobre la nostra variable per tal de complir les premisses, per tant, els càlculs també els farem sobre aquesta transformació.

$$\overline{D} = \log(T_{DR}) - \log(T_{AP}) = \log\left(\frac{T_{DR}}{T_{AP}}\right) = 0.020068$$

$$t = \frac{\overline{D} - \mu_D}{S_D/\sqrt{n}} = \frac{0.020068 - 0}{0.4083/\sqrt{30}} = 0.2692$$

$$Punt\ crític\ (\alpha\ = 0.05)\ \ =\ t_{29,0.975} = 2.045$$

$$P\text{-}valor\ =\ P(|t_{29}| > 0.2692)\ =\ 0.7897$$

d) Conclusió sobre la hipòtesi nul·la

No podem rebutjar la hipòtesi nul·la, ja que el p-valor = 0,7897 > 0.05 (o el punt crític = 2,045 > t = 0,2692).

e) Interval de confiança

$$IC(\log(\mu_D),0.95) = \overline{D} \pm t_{29,0.975}\sqrt{{S_D}^2/n} = 0.020068 \pm 2.045 * 0.0745 =$$

$$= [-0.1324, 0.1725]$$

Traduint l'interval a exponencial queda:

$$IC\left(\frac{\mu_{DR}}{\mu_{AP}}, 0.95\right) = [e^a, e^b] = [0.876, 1.188]$$

f) Prova d'hipòtesi sencera amb t-test

A la figura següent mostra el t-test de R de la variable diferència de logaritmes i la seva desviació de tipus mostral:

```
        One Sample t-test

data:  logD
t = 0.26919, df = 29, p-value = 0.7897
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -0.1324067  0.1725430
sample estimates:
 mean of x
0.02006814

> sd(logD)
[1] 0.4083351
```

Fig. 6: Prova d'hipòtesi sencera utilitzant el t-test.

5. DISCUSSIÓ

5.1 Conclusions

Després d'haver fet el contrast d'hipòtesi sobre la comparació de mitjanes, hem arribat a la conclusió de què no hi ha evidència en contra de la hipòtesi nul·la, ja que el p-valor = 0.7897 és molt més major que el llindar d'acceptació $\alpha = 0.05$ (o bé, tenint en compte que el punt crític = 2.045 supera el valor de l'estadístic $t = 0.2692$).

En quant a l'interval de confiança obtingut, [0.876, 1.188], després d'haver fet la traducció a exponencial, s'interpreta com que la mitjana de la duració del procés de renderització amb *Davinci Resolve* és entre 0.876 i 1.188 vegades superior a la mitjana amb *Adobe Premiere Pro*, amb una confiança del 95%. Per tant, tal i com es pot observar, aquesta diferència és mínima i reforça la nostra conclusió.

En definitiva, tot i que en un principi havíem pensat que el programa d'edició de vídeo *Adobe Premiere Pro*, pel simple fet de ser pagament, diferiria bastant respecte el *Davinci Resolve*, una alternativa gratuïta, amb els resultats obtinguts, hem pogut observar que no podem descartar que el rendiment a l'hora de renderitzar vídeos sigui el mateix.

5.2 Observacions

Després de l'estudi realitzat, com a proposta de millora, creiem que hauria sigut més adient agafar un rang més ampli de duració dels vídeos gravats, per tal que les diferències observades fossin més notables.

6. ANNEXOS

a) Taula de resultats (temps en minuts)

N	Davinci	Premiere
1	7.05	5.78
2	7.52	5.92
3	7.47	6.42
4	6.75	6.2
5	7.60	6.45
6	7.15	6.12
7	7.58	6.78
8	17.83	30.47
9	15.37	26.13
10	14.68	24.42
11	16.08	27.10
12	11.30	14.18
13	11.25	13.35
14	9.12	11.63
15	11.05	13.27

N	Davinci	Premiere
16	13.78	10.43
17	17.60	12.52
18	9.68	11.93
19	8.80	5.9
20	11.52	13.43
21	18.88	15.15
22	9.77	11.8
23	9.23	12.02
24	10.98	14.38
25	9.72	12.72
26	16.35	13.38
27	10.05	2.13
28	17.35	14.05
29	25.05	20.17
30	59.90	42.05

b) Script en R

```
# Lectura de les dades
dades <- read.table("clipboard", header=TRUE, sep='\t')

# Hipòtesi
H0: Diferència de temps de renderització és 0, mu diferència = 0
H1:  mu diferència != 0, ja que és una prova bilateral

# Premisses
# Abans de la prova de comparació de mitjanes, és molt important assegurar
# les mostres obtingudes compleixen les premisses, a una anàlisi de dades
# aparellades és necessària comprovar l'efecte additiu constant i Normalitat de la D

TDR <-dades$Temps_Davinci_Resolve
# 30 mostres de temps de renderització amb l'editor Davinci Resolve
TAP <- dades$Temps_Adobe_Premiere
# 30 mostres de temps de renderització amb l'editor Adobe Premiere
mTDR <- mean(TDR)
mTAP <- mean(TAP)
sTDR <- sd(TDR)
sTAP <-sd(TAP)
summary(TDR)
summary(TAP)

# Variable Diferència
D <- TDR - TAP # Variable diferència de temps de renderització Davinci-Premiere
boxplot(D, col="lightgreen", main = "Diferència de temps de renderització")
mD <-mean(D)

# Normalitat
qqnorm(D)
qqline(D, col=2)
# La distribució de les diferències segueix una distribució normal.

# Efecte additiu
library(PairedData)
p <- paired(TAP, TDR)
plot(p, type='BA')
boxplot(p, col = 6:5 , names=c("Adobe Premiere", "Davinci Resolve"), main = "Temps
de renderització" )

# L'efecte no és constant, per valors grans, l'efecte és més gran
# Com que l'efecte additiu no és constant, provarem de solucionar-lo fent servir la
# transformació logarítmica sobre les variables


# Variable Diferència de logaritmes
logTDR <-log(TDR)
logTAP <- log(TAP)
logD <- logTDR -logTAP
# Normalitat
```

```
qqnorm(logD)
qqline(logD,col=2)
# La distribució de les diferències es pot assumir com a normal, ja que la variable
# diferència s'ajusten força bé als quantils teòrics de la Normal

# Efecte additiu
library(PairedData)
p2 <- paired(logTAP, logTDR)
plot(p2, type='BA')
# L'efecte additiu constant

# Comparació de mitjanes
mlogD <- mean(logD) # Mitjana mostral de logD
slogD <- sd(logD) # Desviació típica mostral de logD
nlogD <- length(D) # Nombre de mostres de dades
elogD <- slogD/sqrt(nlogD) # Estimació de l'error tipus

t <- (mlogD)/elogD # Càlcul estadístic
# Com que és una prova bilateral (ja que volem saber si en dos programes d'editors
diferents
# el temps de renderització són iguals o no), és necessàri calcular en el p-valor la
# probabilitat: P(|t29| > |t|) on en aquest cas el valor t és positiu, per tant utilitzem la
funció següent:
p_valor <- 2*(1-pt(t, nlogD-1))

#Punt crític
punt_critic <- qt(0.975, nlogD-1)

#Interval de confiança
IC_logD <- c(mlogD - punt_critic*elogD, mlogD + punt_critic*elogD) #IC(μlogD, 0.95)

# No rebutgem la hipòtesi nul·la, ja que el p-valor > 0.05(0.79) o que el punt crític és
major que la t (0.269 < 2.045)
t.test(logD)
```