Estadística Inferencial en R

Día 1 del curso:
Análisis de RNA-seq en R

Andree Valle Campos
@avallecam

CDC - Perú
Centro Nacional de Epidemiología,
Prevención y Control de Enfermedades

2019-11-04

Temario

1. **(breve) Introduction a R**
    - Conceptos clave
2. **Variables:**
    - Concepto y clasificación
    - Explorar distribuciones
    - **Prueba de Hipótesis:** limitantes y alternativas
3. **Modelos lineales:**
    - **Regresión lineal y logística**
    - Relación con PH
    - **Comparación múltiple:** corrección del valor p y FDR
    - Aplicación en *microarrays*

Metodología

- Teórico y **práctico**
  - Definiciones y procedimientos
  - Responder a: *¿y cómo lo hago en R?*
- Material
  - 2 archivos de teoría en PDF
  - 5 practicas + solucionario
- Contenido disponible en **https://github.com/avallecam/biostat2019**

¡Pregunta 1!

Del 1 al 5, ¿Qué tan familiarizado estás con cada tema?

(breve) introducción a R

Ir a

`01-biostat2019-slides.pdf`

Práctica 1

```r
#aritmética
2+2
x <- 2
x+2

#funciones
seq(1,10)
rep(1,5)

#vectores
heights <- c(147.2, 153.5, 152.5, 162.0, 153.9)
mean(heights)

#subconjuntos
y <- LETTERS[1:10]
y[3:7]
```

Si necesitas ayuda:

1. Pregúntale a R con la función `help()` o `?`, p.e.: `help(mean)` o `?mean`
2. Nos pasas la voz :)

Variables

Variables: Ejemplo

VEF = Volumen Expiratorio Forzado (mL/s)

¿Existe una asociación entre VEF y edad, talla, etc?

¿Cómo se ve afectado el VEF en las personas que fuman?

- base: `espirometria.dta`
- exposición: `edad`, `talla`
- desenlace: `vef`

caso	codigo	edad	vef	talla	sexo	fumar
1	301	9	1.708	57.0	female	No
2	451	8	1.724	67.5	female	No
3	501	7	1.720	54.5	female	No
4	642	9	1.558	53.0	male	No
5	901	9	1.895	57.0	male	No
6	1701	8	2.336	61.0	female	No

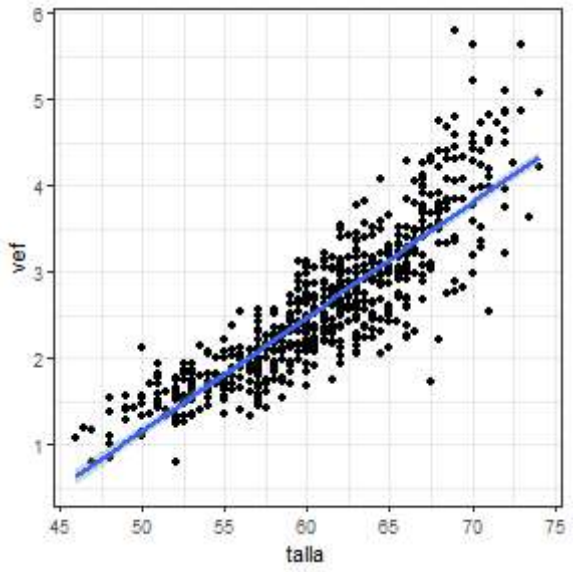
Variables numéricas

```
espir %>%
  select(edad,vef,talla) %>%
  skim()
```

type	variable	missing	complete	n	mean	sd	p0	p25	p50	p75	p100	hist
numeric	edad	0	654	654	9.93	2.95	3	8	10	12	19	
numeric	talla	0	654	654	61.14	5.7	46	57	61.5	65.5	74	
numeric	vef	0	654	654	2.64	0.87	0.79	1.98	2.55	3.12	5.79	

Variables numéricas: PH

```
espir %>%
  ggplot(aes(x=talla,y=vef)) +
  geom_point() +
  geom_smooth(method = "lm")
```




```
espir %>%
  select(edad,vef,talla) %>%
  correlate() %>%
  rearrange() %>%
  shave()
```

```
## # A tibble: 3 x 4
##   rowname   edad  talla   vef
##   <chr>    <dbl>  <dbl> <dbl>
## 1 edad    NA      NA         NA
## 2 talla    0.792 NA         NA
## 3 vef      0.756  0.868    NA
```

Variables categóricas

```
espir %>%
  tabyl(sexo) %>%
  adorn_totals("row") %>%
  adorn_pct_formatting()
```

```
##    sexo   n percent
##    male 336   51.4%
##  female 318   48.6%
##   Total 654  100.0%
```

```
espir %>%
  tabyl(sexo,fumar) %>%
  adorn_totals(c("col")) %>%
  adorn_percentages() %>%
  adorn_pct_formatting() %>%
  adorn_ns(position = "front") %>%
  adorn_title()
```

```
##                fumar
##    sexo           No          Si         Total
##    male 310 (92.3%) 26  (7.7%) 336 (100.0%)
##  female 279 (87.7%) 39 (12.3%) 318 (100.0%)
```

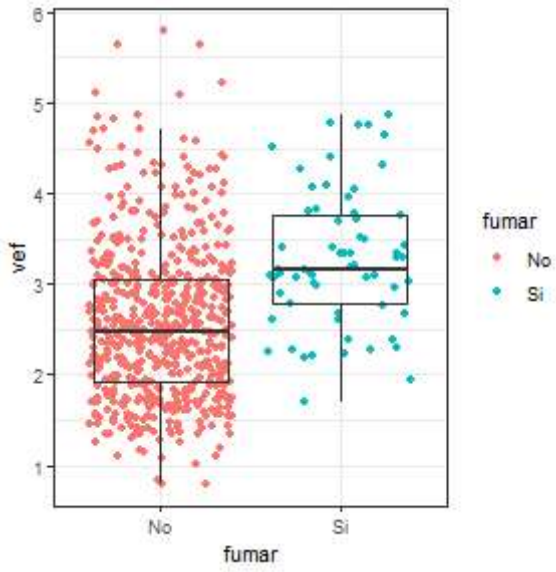
Variables categóricas: PH

```
espir %>%
  tabyl(sexo,fumar) %>%
  chisq.test() %>%
  tidy()
```

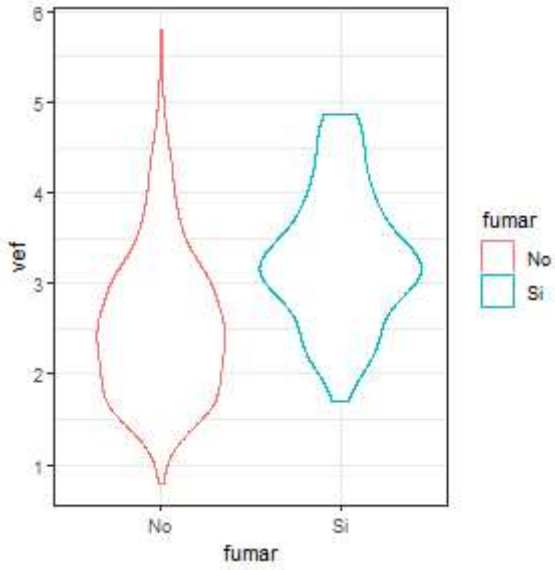
```
## # A tibble: 1 x 4
##   statistic p.value parameter method
##       <dbl>   <dbl>     <int> <chr>
## 1      3.25  0.0714         1 Pearson's Chi-squared test with Yates' conti~
```

Variables numéricas y categóricas

```
espir %>%
  ggplot(aes(x=fumar,y=vef)) +
  geom_point(
    aes(color=fumar),
    position = "jitter") +
  geom_boxplot(alpha=0)
```




```
espir %>%
  ggplot(aes(x=fumar,y=vef)) +
  geom_violin(aes(color=fumar))
```



Variables numéricas y categóricas: PH

```
espir %>%
  select(fumar,vef) %>%
  group_by(fumar) %>%
  skim()
```

type	fumar	variable	missing	complete	n	mean	sd	p0	p25	p50	p75	p100	hist
numeric	No	vef	0	589	589	2.57	0.85	0.79	1.92	2.46	3.05	5.79	
numeric	Si	vef	0	65	65	3.28	0.75	1.69	2.8	3.17	3.75	4.87	

```
t.test(vef ~ fumar, data = espir, var.equal=FALSE) %>%
  tidy()
```

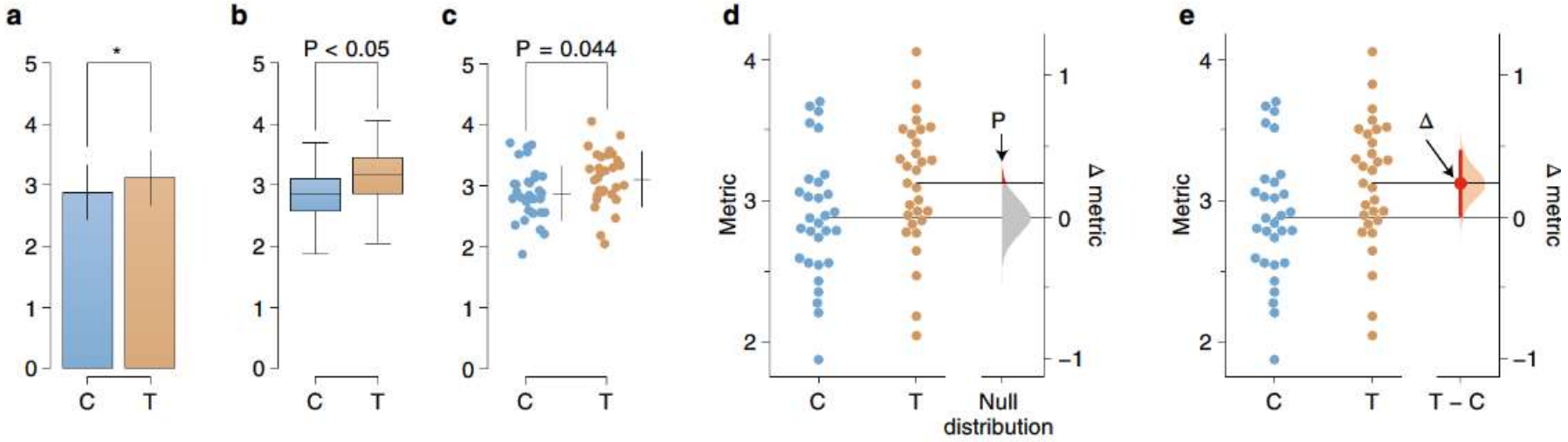
estimate	estimate1	estimate2	statistic	p.value	parameter	conf.low	conf.high	method	alternative
-0.71	2.57	3.28	-7.15	0	83.27	-0.91	-0.51	Welch Two Sample t-test	two.sided

Práctica 2

```r
#importar base en formato DTA
espir <- read_dta("data-raw/espirometria.dta") %>% as_factor()

#generar resumen descriptivo
espir %>% skim()

#grafica distribución
espir %>%
  ggplot(aes(vef)) +
  geom_histogram()

#realizar pruebas de hipótesis
#experimenta y describe qué cambios genera cada uno de los argumentos?
compareGroups(formula = fumar ~ edad + vef + talla + sexo,
              data = espir,
              byrow=T,
              method=c(vef=2)
              ) %>%
  createTable(show.all = T) %>%
  export2xls("table/tab1.xls")
```

Comentario: Pruebas de hipótesis

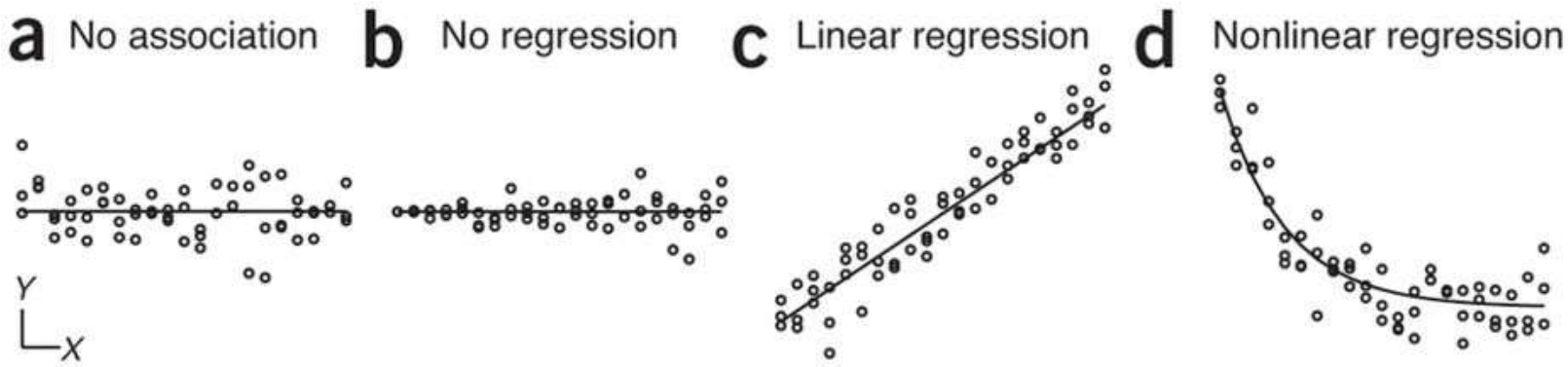
- **Limitante:**
  - dicotomización de resultados en significativos y no significativos
  - valor p da indicativo de *tamaño de efecto*, pero no lo cuantifica
- **Alternativa[1]:**



[1] Moving beyond P values: data analysis with estimation graphics

Modelos lineales

¿Por qué usar una regresión?



- Porque:
  - Fija una **variable independiente** o **exposición** y observa una respuesta en la **variable dependiente** o **desenlace**.
  - Permite **explicar** el cambio promedio de un evento Y en base a cambios en X, usando coeficientes o medidas de asociación.
  - Permite **predecir** la probabilidad asociada a un evento.
  - Permite **cuantificar** el tamaño del efecto de la comparación.

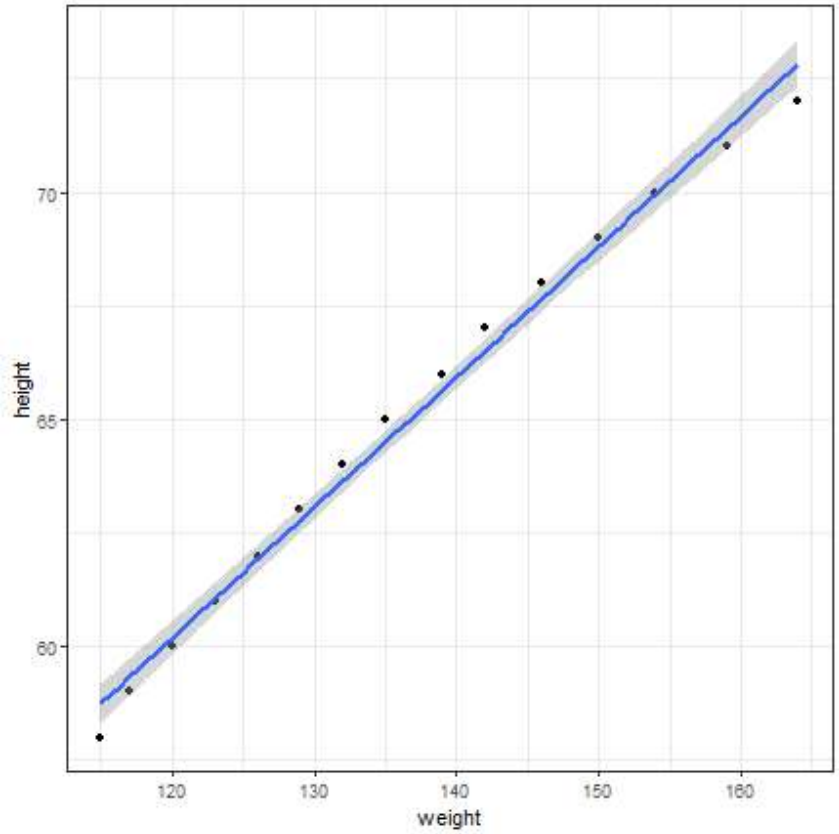
Regresión Lineal Simple

características

- **una** variable independiente (simple)
- **una** variable dependiente (univariada)
- ambas variables deben ser **numéricas**

objetivos

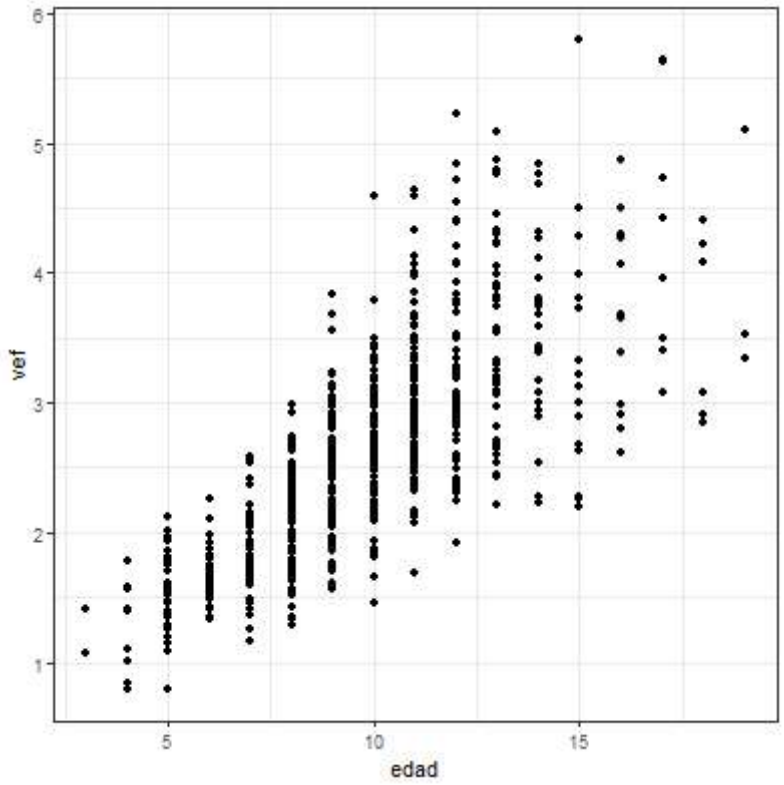
- ajustar datos a una recta
- interpretar medida de bondad de ajuste (R^2) y coeficientes
- evaluar supuestos
- visualizar el modelo



$$Y = \beta_0 + \beta_1 X_1 + \epsilon$$

RLinS: Ejemplo

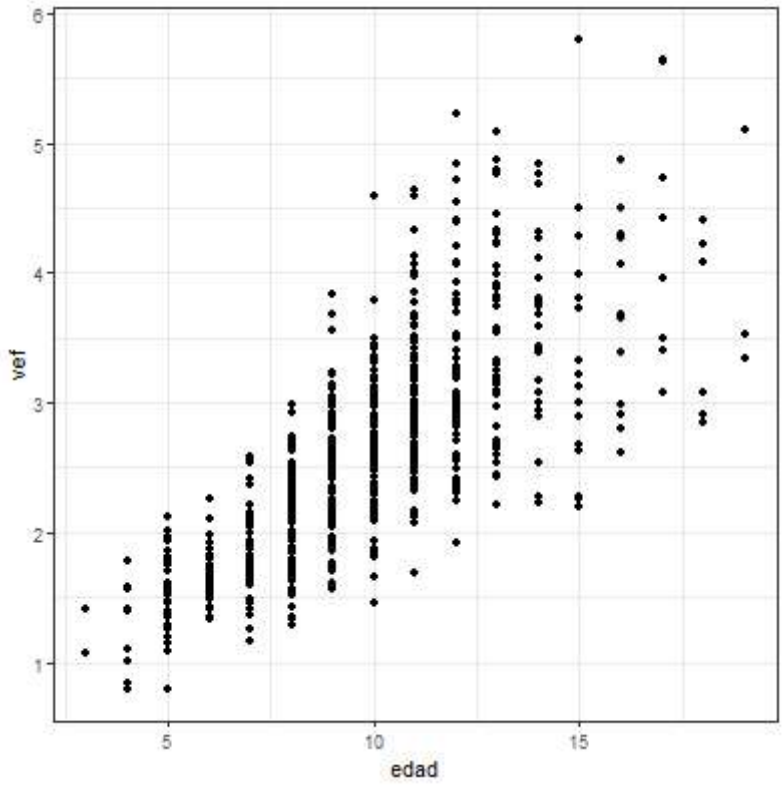
```
espir %>%
  ggplot(aes(x=edad,y=vef)) +
  geom_point()
```



- **Pregunta**
  - **¿Existe una relación lineal entre edad y VEF?**
  - **¿En cuánto incrementa el VEF, por cada incremento en un año de edad?**

RLinS: Ejemplo

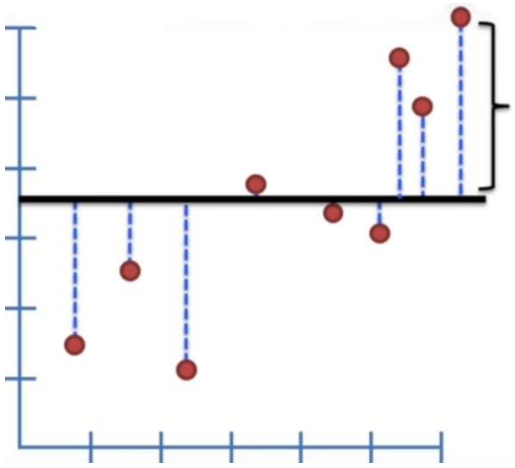
```
espir %>%
  ggplot(aes(x=edad,y=vef)) +
  geom_point()
```

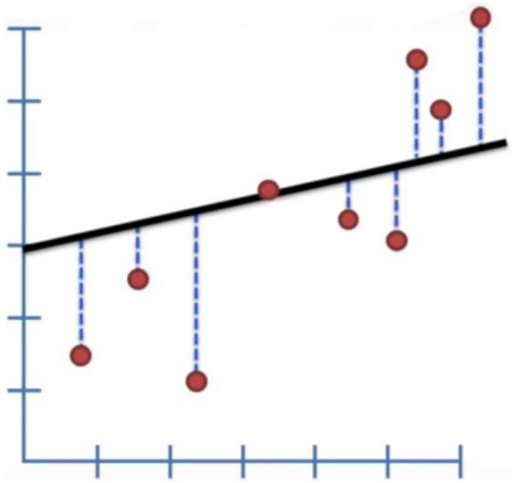


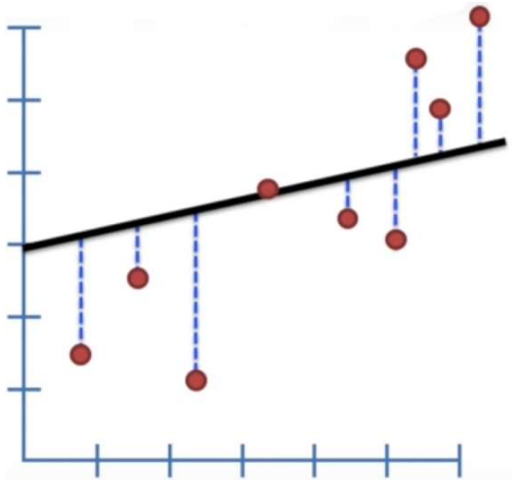
- **Pregunta**
  - **¿Existe una relación lineal entre edad y VEF?**
  - **¿En cuánto incrementa el VEF, por cada incremento en un año de edad?**
- **Evaluación de supuestos:**
  - **#1** independencia de observaciones
  - **#2** linealidad

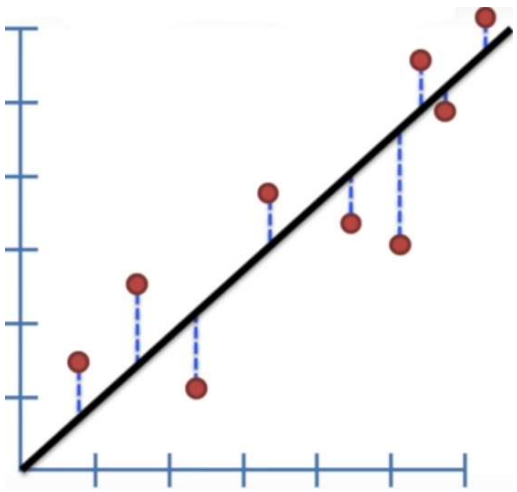
RLinS: residuales

- Definición: Diferencia entre el *valor observado* y el *valor predicho* en el eje vertical.

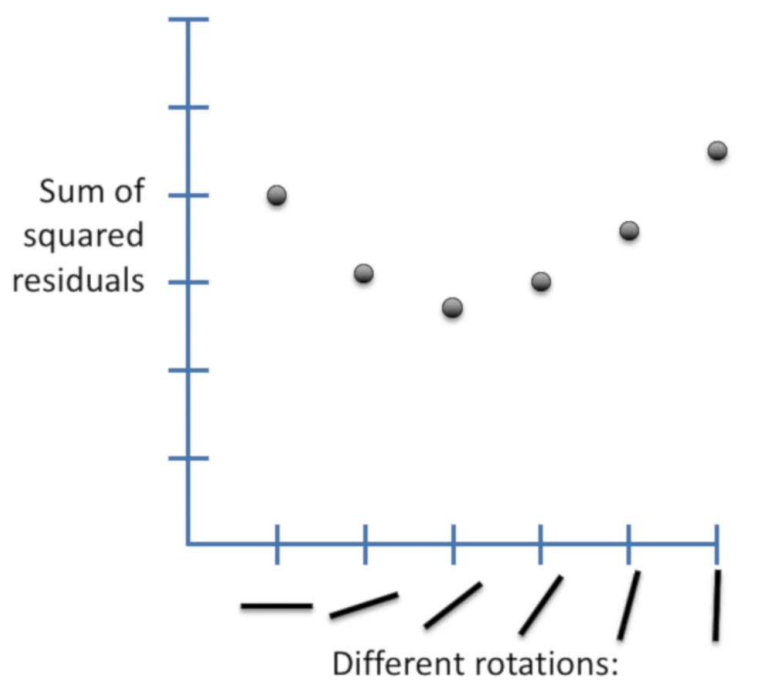








RLinS: suma de mínimos cuadrados

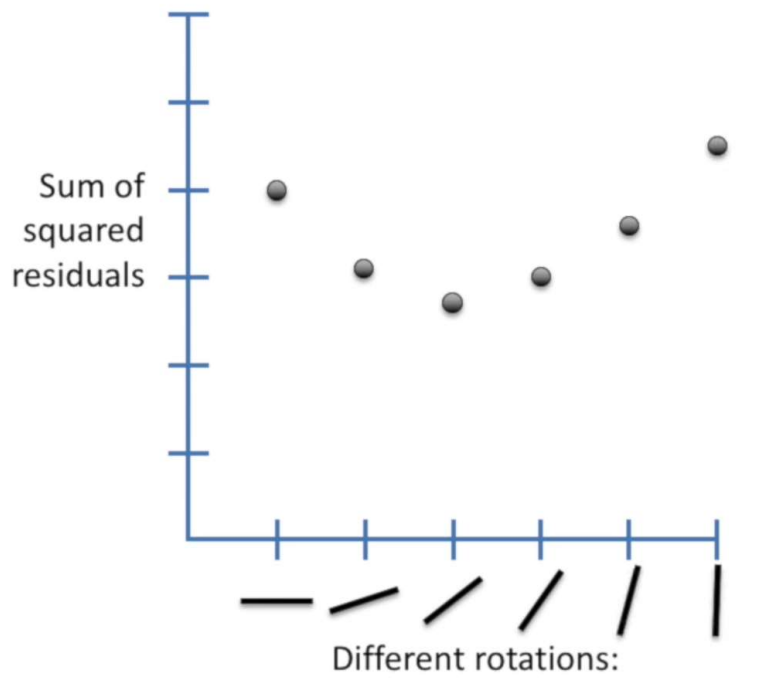


Cálculo de la **sumatoria del cuadrado de los residuales** hacia la media y la recta:

$$SSE(mean) = \sum (data - mean)^2$$

$$SSE(fit) = \sum (data - fit)^2$$

RLinS: suma de mínimos cuadrados



Cálculo de la **sumatoria del cuadrado de los residuales** hacia la media y la recta:

$$SSE(mean) = \sum (data - mean)^2$$

$$SSE(fit) = \sum (data - fit)^2$$

$$Var(x) = \frac{SSE(x)^2}{n}$$

Medida de **bondad de ajuste**:

$$R^2 = \frac{Var(mean) - Var(fit)}{Var(mean)}$$

RLinS: bondad de ajuste

```
wm1 <- lm(vef ~ edad, data = espir)

wm1 %>%
  glance()
```

```
## # A tibble: 1 x 6
##   r.squared adj.r.squared sigma statistic   p.value    df
##       <dbl>         <dbl> <dbl>     <dbl>     <dbl> <int>
## 1     0.572         0.572 0.568      872. 2.45e-122     2
```

INTERPRETACIÓN

- **edad** *explica* el **57%** de la variabilidad de **VEF**
- existe un **57%** *de reducción en* la variabilidad de **VEF** al tomar en cuenta la **edad**

RLinS: coeficientes

```
wm1 %>% tidy()
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic   p.value
##   <chr>          <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept)    0.432   0.0779       5.54 4.36e-  8
## 2 edad           0.222   0.00752     29.5  2.45e-122
```

INTERPRETACIÓN

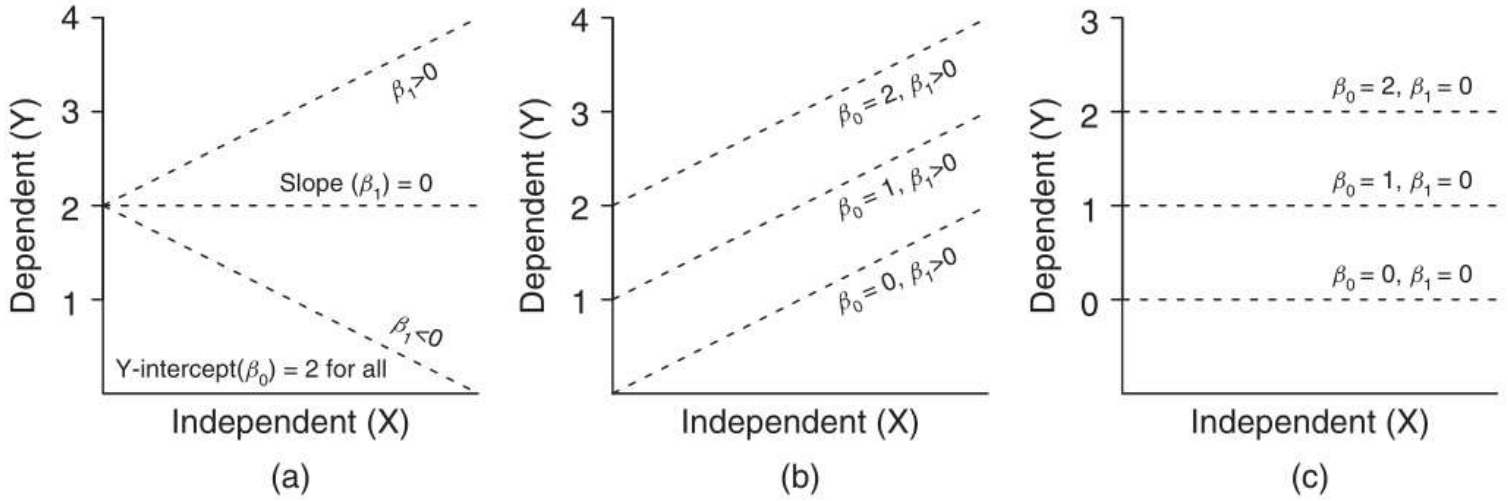


Fig 8.2 Fictitious data contrasting differences in interpretation between slope (β_1) and y-intercept (β_0) parameters.

RLinS: coeficientes

```
wm1 %>% tidy()
```

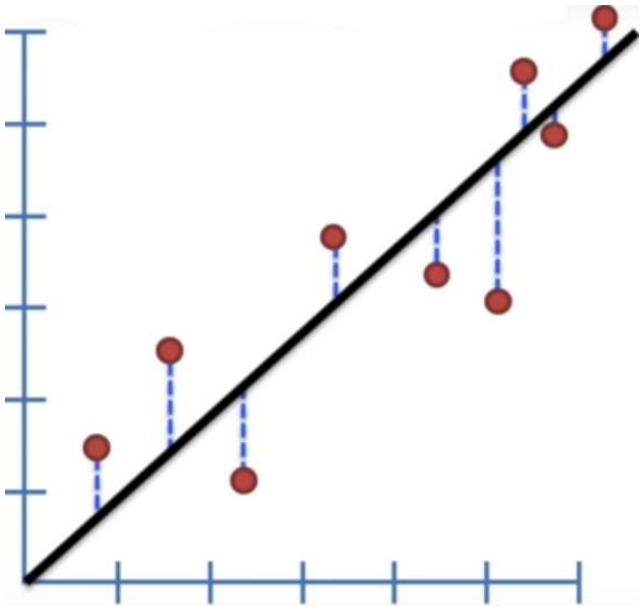
```
## # A tibble: 2 x 5
##   term        estimate std.error statistic   p.value
##   <chr>          <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept)    0.432   0.0779       5.54 4.36e-  8
## 2 edad           0.222   0.00752     29.5  2.45e-122
```

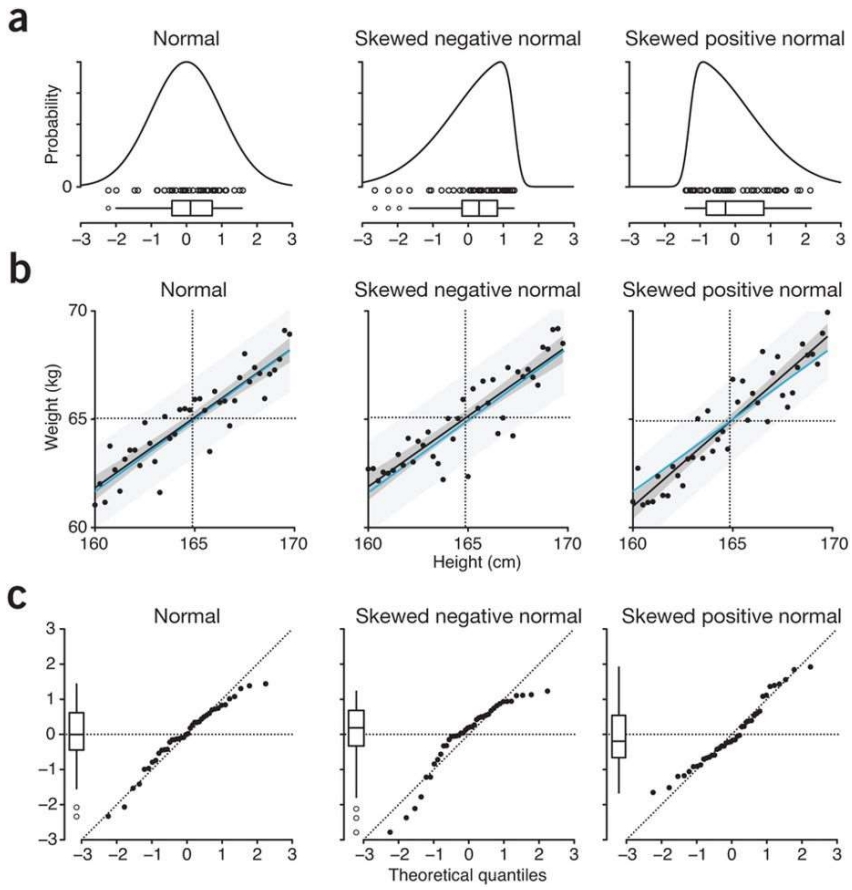
```
wm1 %>% confint_tidy()
```

```
## # A tibble: 2 x 2
##   conf.low conf.high
##      <dbl>     <dbl>
## 1    0.279     0.585
## 2    0.207     0.237
```

- β_{edad}:
- En la población, por cada incremento de **edad** en *una unidad*, el **VEF** en promedio *incrementa* en 0.22 mL/s,
- con un intervalo de confianza al 95% de 0.21 a 0.24 mL/s.
- Este resultado es estadísticamente significativo con un valor **p < 0.001**

RLinS: supuesto **#3** normalidad residuales





RLinS: supuesto #3 normalidad residuales

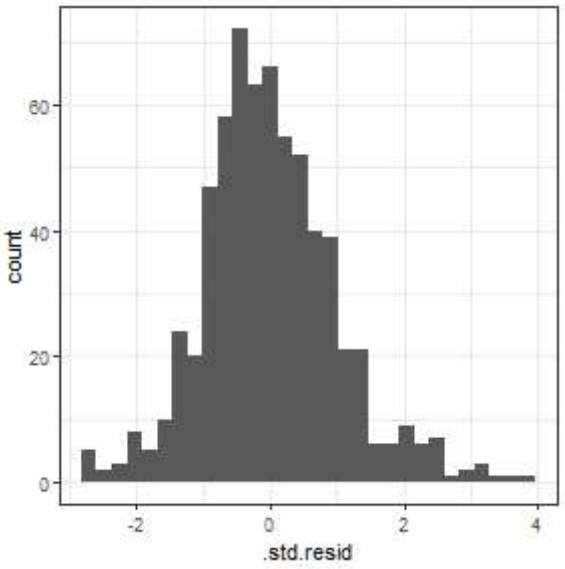
- generación de *dataframe* con:
  - `.fitted` : valor predicho de Y para valor de X ($\hat{Y}$)
  - `.resid` : valor crudo del residual ($Y - \hat{Y}$)
  - `.std.resid` : valor *estudiantizado* del residual

```
wm1 %>%
  augment()
```

```
## # A tibble: 654 x 9
##       vef  edad .fitted .se.fit .resid    .hat .sigma   .cooksd .std.resid
##     <dbl> <dbl>   <dbl>   <dbl>  <dbl>   <dbl>  <dbl>     <dbl>      <dbl>
##  1  1.71      9    2.43  0.0233 -0.722 0.00168  0.567 0.00137       -1.27
##  2  1.72      8    2.21  0.0265 -0.484 0.00218  0.568 0.000797      -0.854
##  3  1.72      7    1.99  0.0313 -0.266 0.00304  0.568 0.000335      -0.469
##  4  1.56      9    2.43  0.0233 -0.872 0.00168  0.567 0.00199       -1.54
##  5  1.89      9    2.43  0.0233 -0.535 0.00168  0.568 0.000750      -0.944
##  6  2.34      8    2.21  0.0265  0.128 0.00218  0.568 0.0000558      0.226
##  7  1.92      6    1.76  0.0370  0.155 0.00424  0.568 0.000160       0.274
##  8  1.41      6    1.76  0.0370 -0.349 0.00424  0.568 0.000808      -0.616
##  9  1.99      8    2.21  0.0265 -0.221 0.00218  0.568 0.000166      -0.390
## 10  1.94      9    2.43  0.0233 -0.488 0.00168  0.568 0.000624      -0.861
## # ... with 644 more rows
```

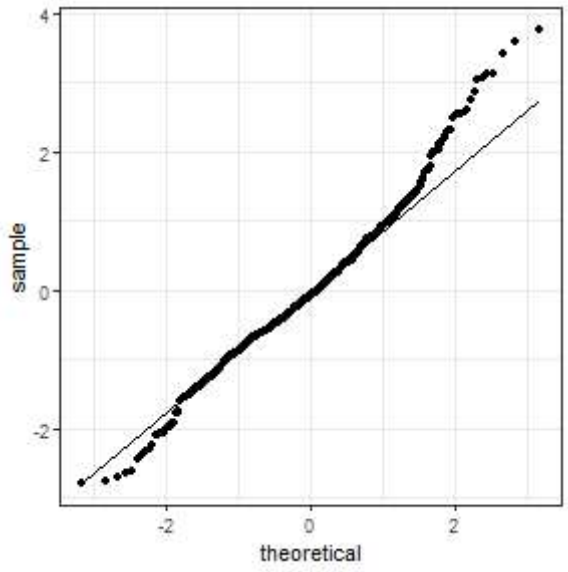
RLinS: supuesto #3 normalidad residuales

```
wm1 %>%
  augment() %>%
  ggplot(aes(.std.resid)) +
  geom_histogram()
```



- **META:** puntos sobre la línea

```
wm1 %>%
  augment() %>%
  ggplot(
    aes(sample=.std.resid)
    ) +
  geom_qq() +
  geom_qq_line()
```



RLinS: supuesto **#4** homoscedasticidad

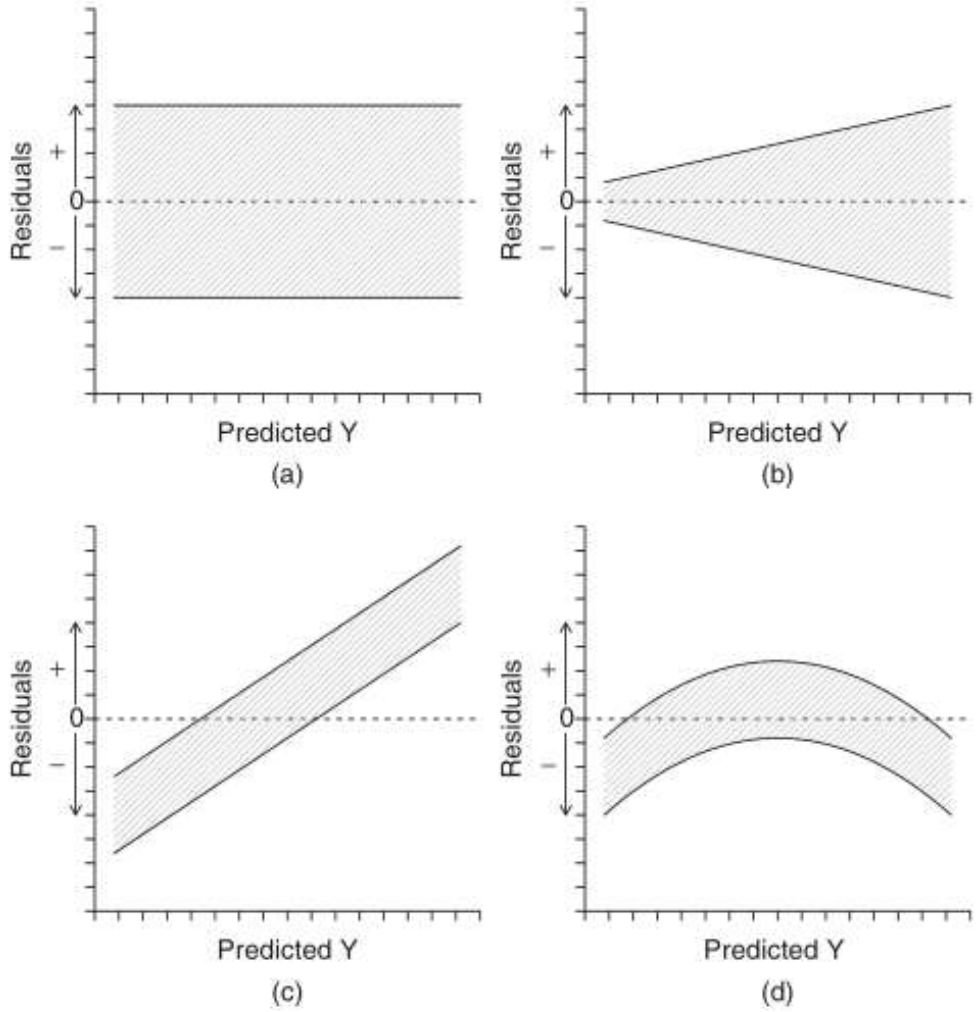
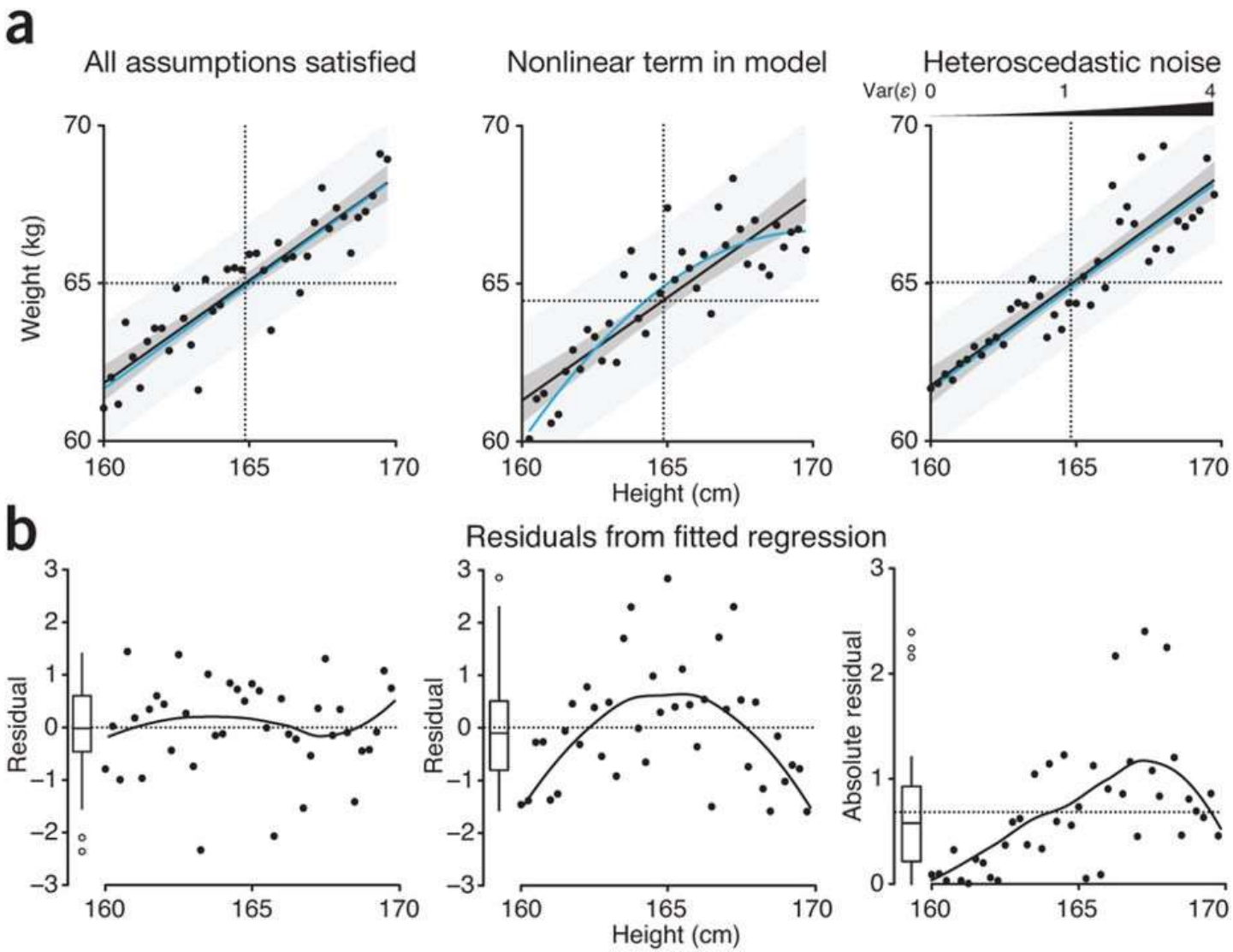


Fig 8.5 Stylised residual plots depicting characteristic patterns of residuals (a) random scatter of points - homogeneity of variance and linearity met (b) "wedge-shaped" - homogeneity of variance not met (c) linear pattern remaining - erroneously calculated residuals or additional variable(s) required and (d) curved pattern remaining - linear function applied to a curvilinear relationship. Modified from Zar (1999).

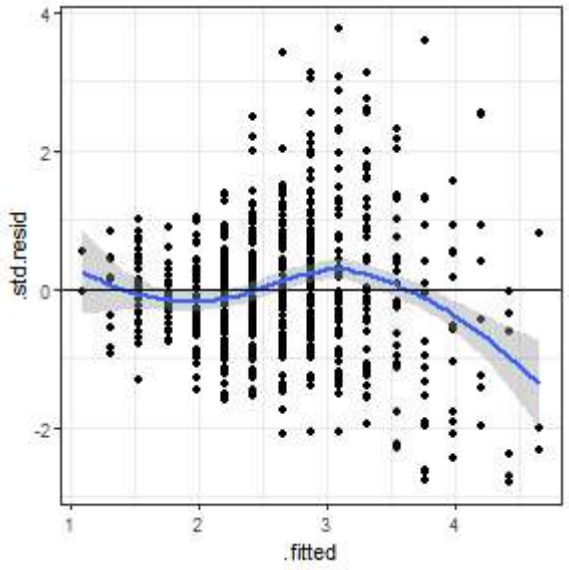
RLinS: supuesto **#4** homoscedasticidad



RLinS: supuesto #4 homoscedasticidad

- **META:** distribución idéntica a ambos lados de la línea

```
wm1 %>%
  augment() %>%
  ggplot(aes(.fitted,.std.resid)) +
  geom_point() +
  geom_smooth() +
  geom_hline(yintercept = c(0))
```

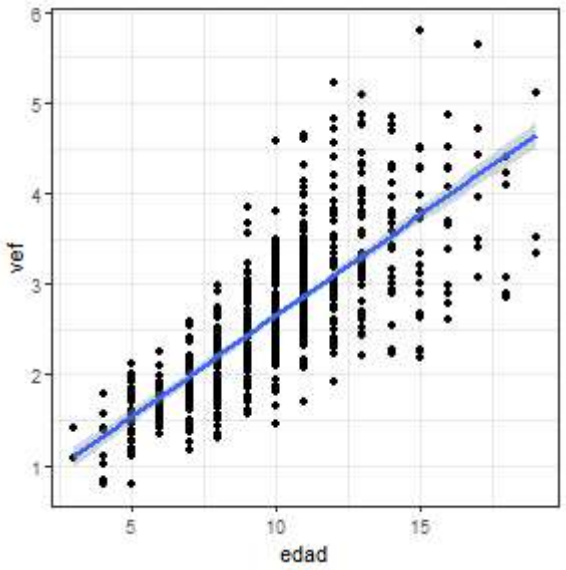


RLinS: ¿cómo se ve el modelo?

$$Y = \beta_0 + \beta_1 X_1 + \epsilon$$

$$VEF = 0.60 + 0.22(edad) + \epsilon$$

```
espir %>%
  ggplot(aes(edad,vef)) +
  geom_point() +
  geom_smooth(method = "lm")
```



RLinS: retroalimentación

- R^2 indica el porcentaje de *variabilidad* del desenlace (var. dependiente) explicada por la exposición (var. independiente).
- los **coeficientes** permiten **cuantificar** el *tamaño del efecto* de la exposición en el desenlace en base a un modelo estadístico.
- los **supuestos** permiten evaluar qué tan adecuado es el ajuste de los datos al modelo.

Práctica 3

- **¿Existe una relación lineal entre talla y VEF?**

```
espir %>%
  ggplot(aes(x = talla,y = vef)) +
  geom____()
```

- identificar coeficientes y R^2

```
# recordar: y ~ x
wm1 <- lm(_____ ~ _____, data = _____)
wm1 %>% g_____
wm1 %>% t_____
wm1 %>% c_____
```

- evaluar supuestos: normalidad y homoscedasticidad

```
wm1 %>% augment() %>%
  ggplot() + _____

wm1 %>% augment() %>%
  ggplot(aes(.fitted,.std.resid)) + __________
```

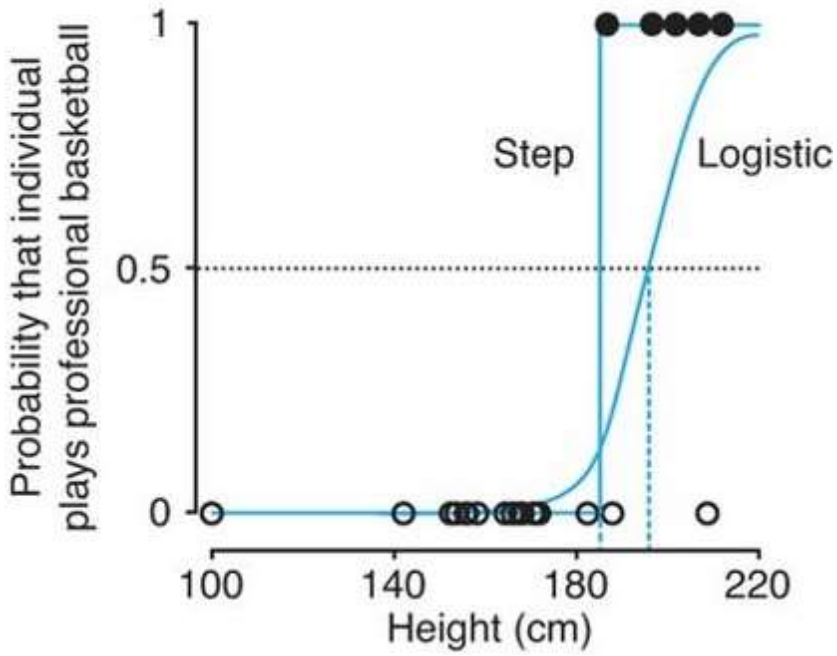
Regresión Logística Simple

características

- **una** variable independiente (simple)
- **una** variable dependiente (univariada)
- la variable *dependiente* debe ser **categórica dicotómica**

objetivos

- ajustar datos a una recta
- interpretar coeficientes



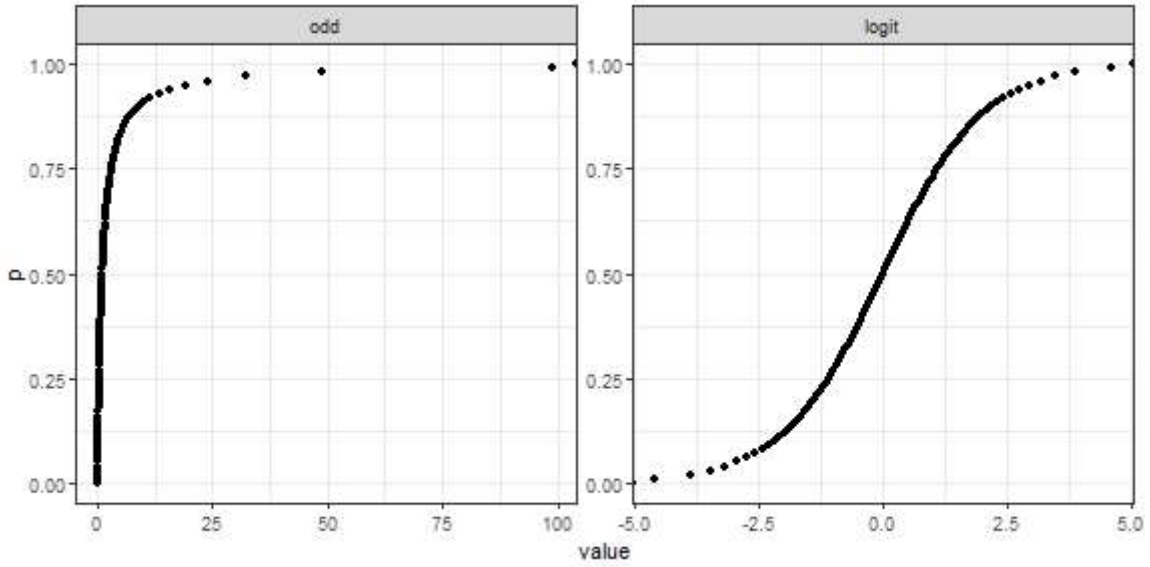
$$Y = \beta_0 + \beta_1 X_1 + \epsilon$$

RlogS: Ecuación y coeficiente

- p posee distribución *binomial*, linealizada $[-\infty, \infty]$ por la función *logit*

$$logit(p) = log(odds) = log\left(\frac{p}{1-p}\right)$$

$$logit(p) = \beta_0 + \beta_1 X_1 + \ldots + \beta_n X_n + \epsilon \;\; ; \;\; y \sim Binomial$$



RlogS: Ecuación y coeficiente

- p posee distribución *binomial*, linealizada $[-\infty, \infty]$ por la función *logit*

$$logit(p) = log(odds) = log\left(\frac{p}{1-p}\right)$$

$$logit(p) = \beta_0 + \beta_1 X_1 + \ldots + \beta_n X_n + \epsilon \;\; ; \;\; y \sim Binomial$$

- El valor exponenciado de los coeficientes se pueden interpretar como **Odds Ratio (OR)**

$$Y = \beta_0 + \beta_1 X_1$$

$$\begin{cases} Y_{x=1} = log(odds_{x=1}) = \beta_0 + \beta_1(1) \\ Y_{x=0} = log(odds_{x=0}) = \beta_0 + \beta_1(0) \end{cases}$$

$$Y_{x=1} - Y_{x=0} = \beta_1$$

$$log(odds_{x=1}) - log(odds_{x=0}) = \beta_1$$

$$log\left(\frac{odds_{x=1}}{odds_{x=0}}\right) = \beta_1$$

$$OR = exp(\beta_1)$$

RlogS: Modelos Lineales Generalizados

GLM ajusta modelos lineales de $g(y)$ con covaribles x

$$g(Y) = \beta_0 + \sum_{n=1}^{n} \beta_n X_n \ , \ y \sim F$$

Donde:

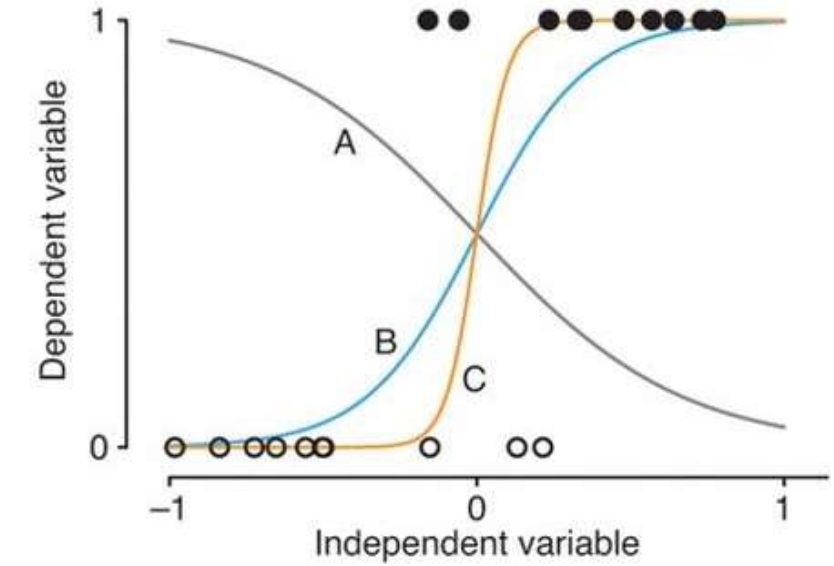
- F es la familia de distribución
- $g(\ )$ es la función de enlace

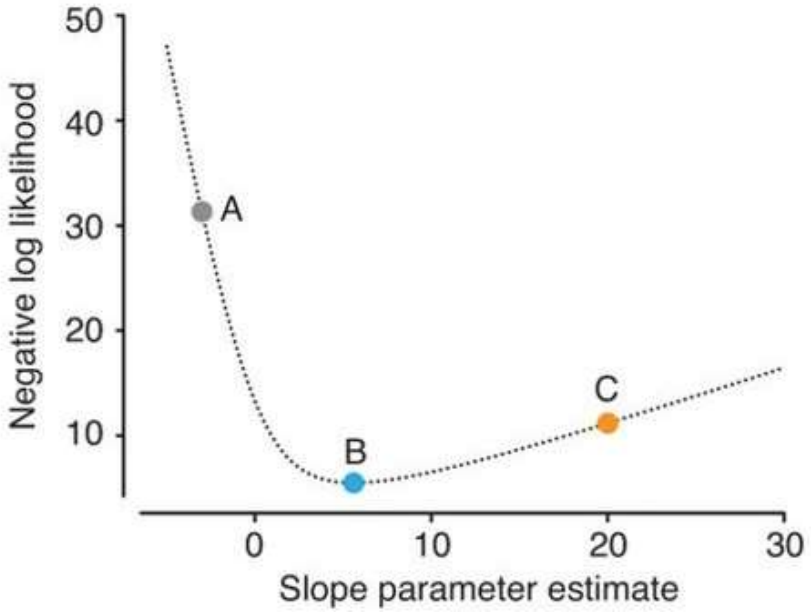
Table 17.1 Common generalized linear models and associated canonical link-distribution pairs.

Model	Response variable	Predictor variable(s)	Residual distribution	Link
Linear regression[a]	Continuous	Continuous/ Categorical	Gaussian (normal)	Identity $g(\mu) = \mu$
Logistic regression	Binary	Continuous/ Categorical	Binomial	Logit $g(\mu) = log_e \frac{\mu}{1-\mu}$
Log-linear models	Counts	Categorical	Poisson	Log $g(\mu) = log_e \mu$

[a]Includes the standard ANOVA and ANCOVA designs.

GLM: máxima verosimilitud





- La estimación de parámetros se da por un proceso de optimización llamado **máxima verosimilitud** (o *likelihood*).
- un estimado por *máxima verosimilitud* es aquel que maximiza la verosimilitud de obtener las actuales observaciones dado el modelo elegido.
- estimación numérica mediante *proceso iterativo* hasta la convergencia.

RlogS: Ejemplo

En personas VIH+, ¿el polimorfismo CCR5 está *asociado* con el desarrollo del SIDA?

- base: `aidsdb.dta`
- exposición: `ccr5`
- desenlace: `aids`

studyid	ttoaidsorexit	aids	age	race	loghivrna	cd4	ccr2	ccr5	sdf1
101750	1019	Yes	22	race1	11.786481	434	No	Yes	No
101780	2809	No	25	race2	12.916421	391	Yes	Yes	No
103328	1717	Yes	32	race1	13.169676	819	No	No	No
104463	2315	Yes	31	race1	10.649203	763	No	No	No
104525	3764	No	39	race1	10.834726	520	No	Yes	Yes
107858	2643	Yes	39	race1	6.790097	NA	No	No	No

RlogS: Interpretar variables categóricas

```
wm1 <- glm(aids ~ ccr5, data = aidsdb,
           family = binomial(link = "logit"))
```

```
## # A tibble: 2 x 7
##   term        log.or    se    or conf.low conf.high p.value
##   <chr>        <dbl> <dbl> <dbl>    <dbl>     <dbl>   <dbl>
## 1 (Intercept) -0.487 0.106 0.614    0.499     0.754   0
## 2 ccr5Yes      0.151 0.245 1.16     0.715     1.87    0.539
```

INTERPRETACIÓN

- β_0
  - En la población, el **odds** de tener sida dado que **no poseen CCR5** es **0.61**,
  - con un intervalo de confianza al 95% de 0.5 a 0.75.

RlogS: Interpretar variables categóricas

```
wm1 <- glm(aids ~ ccr5, data = aidsdb,
           family = binomial(link = "logit"))
```

```
## # A tibble: 2 x 7
##   term        log.or    se    or conf.low conf.high p.value
##   <chr>        <dbl> <dbl> <dbl>    <dbl>     <dbl>   <dbl>
## 1 (Intercept) -0.487 0.106 0.614    0.499     0.754   0
## 2 ccr5Yes      0.151 0.245 1.16     0.715     1.87    0.539
```

INTERPRETACIÓN

- $\beta_{CCR2:Yes}$
  - En la población, el **odds** de tener sida dado que **sí poseen CCR5** es
  - **1.16** veces,
  - el **odds** de tener depresión dado que **no poseen CCR5**,
  - con un intervalo de confianza al 95% de 0.72 a 1.87.
  - Este resultado no es estadísticamente significativo con un valor **p** = **0.539**

RlogS: Interpretar variables contínuas

```
wm1 <- glm(aids ~ loghivrna, data = aidsdb,
           family = binomial(link = "logit"))
```

```
## # A tibble: 2 x 7
##   term        log.or     se     or conf.low conf.high p.value
##   <chr>        <dbl>  <dbl>  <dbl>    <dbl>     <dbl>   <dbl>
## 1 (Intercept) -2.52  0.593  0.0808   0.0246     0.252 0.00002
## 2 loghivrna    0.216 0.0575 1.24     1.11       1.39  0.00017
```

INTERPRETACIÓN

- $\beta_{loghivrna}$
  - En la población, por cada incremento en una unidad del **log de RNA viral**,
  - el **odds** de sida *cambia* en **1.24**, con un intervalo de confianza al 95% de 1.11 a 1.39.
  - Este resultado es estadísticamente significativo con un valor **p < 0.001**

Práctica 4

- **¿Qué otros marcadores están asociados con el desarrollo de SIDA?**
- **Intenta ajustar por otras covariables como edad, raza o sexo**

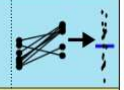
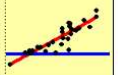
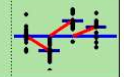
```
# recordar: y ~ x
wm1 <- glm(_____ ~ _____, data=_____, family= _____(link="_____"))

wm1 %>% g_____
wm1 %>% t_____
wm1 %>% c_____
wm1 %>% a_____

wm1 %>% avallecam::epi_tidymodel_or()
```

Relación con Prueba de Hipótesis

Common statistical tests are linear models

Last updated: 02 April, 2019

See worked examples and more details at the accompanying notebook: https://lindeloev.github.io/tests-as-linear

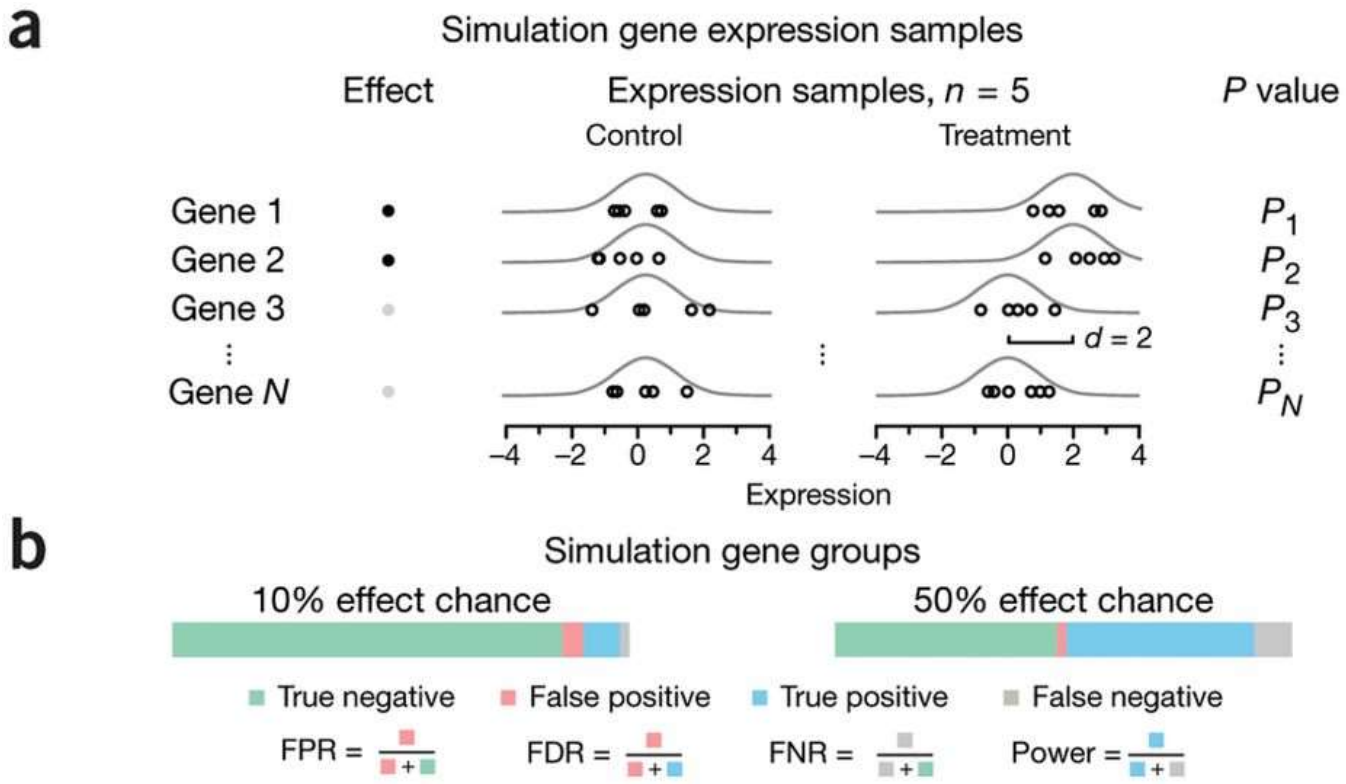
	Common name	Built-in function in R	Equivalent linear model in R	Exact?	The linear model in words	Icon
Simple regression: lm(y ~ 1 + x)	**y is independent of x** P: One-sample t-test N: Wilcoxon signed-rank	t.test(y) wilcox.test(y)	lm(y ~ 1) lm(signed_rank(y) ~ 1)	✓ for N >14	One number (intercept, i.e., the mean) predicts **y**. - (Same, but it predicts the *signed rank* of **y**.)	
	P: Paired-sample t-test N: Wilcoxon matched pairs	t.test(y_1, y_2, paired=TRUE) wilcox.test(y_1, y_2, paired=TRUE)	lm(y_2 - y_1 ~ 1) lm(signed_rank(y_2 - y_1) ~ 1)	✓ for N >14	One intercept predicts the pairwise $\mathbf{y_2}$-$\mathbf{y_1}$ differences. - (Same, but it predicts the *signed rank* of $\mathbf{y_2}$-$\mathbf{y_1}$.)	
	y ~ continuous x P: Pearson correlation N: Spearman correlation	cor.test(x, y, method='Pearson') cor.test(x, y, method='Spearman')	lm(y ~ 1 + x) lm(rank(y) ~ 1 + rank(x))	✓ for N >10	One intercept plus **x** multiplied by a number (slope) predicts **y**. - (Same, but with *ranked* **x** and **y**)	
	y ~ discrete x P: Two-sample t-test P: Welch's t-test N: Mann-Whitney U	t.test(y_1, y_2, var.equal=TRUE) t.test(y_1, y_2, var.equal=FALSE) wilcox.test(y_1, y_2)	lm(y ~ 1 + G_2)[A] gls(y ~ 1 + G_2, weights=…[B])[A] lm(signed_rank(y) ~ 1 + G_2)[A]	✓ ✓ for N >11	An intercept for **group 1** (plus a difference if **group 2**) predicts **y**. - (Same, but with one variance *per group* instead of one common.) - (Same, but it predicts the *signed rank* of **y**.)	
Multiple regression: lm(y ~ 1 + x_1 + x_2 +…)	P: One-way ANOVA N: Kruskal-Wallis	aov(y ~ group) kruskal.test(y ~ group)	lm(y ~ 1 + G_2 + G_3 +…+ G_N)[A] lm(rank(y) ~ 1 + G_2 + G_3 +…+ G_N)[A]	✓ for N >11	An intercept for **group 1** (plus a difference if group ≠ 1) predicts **y**. - (Same, but it predicts the *rank* of **y**.)	
	P: One-way ANCOVA	aov(y ~ group + x)	lm(y ~ 1 + G_2 + G_3 +…+ G_N + x)[A]	✓	- (Same, but plus a slope on **x**.) *Note: this is discrete AND continuous. ANCOVAs are ANOVAs with a continuous x.*	
	P: Two-way ANOVA	aov(y ~ group * sex)	lm(y ~ 1 + G_2 + G_3 + … + G_N + S_2 + S_3+ … + S_K + G_2*S_2+G_3*S_3+…+G_N*S_K)	✓	Interaction term: changing **sex** changes the **y ~ group** parameters. *Note: $\mathbf{G_{2\,to\,N}}$ is an indicator (0 or 1) for each non-intercept levels of the **group** variable. Similarly for $\mathbf{S_{2\,to\,K}}$ for sex. The first line (with G_i) is main effect of group, the second (with S_i) for sex and the third is the **group × sex** interaction. For two levels (e.g. male/female), line 2 would just be "S_2" and line 3 would be S_2 multiplied with each G_i.*	*[Coming]*
	Counts ~ discrete x N: Chi-square test	chisq.test(groupXsex_table)	**Equivalent log-linear model** glm(y ~ 1 + G_2 + G_3 + … + G_N + S_2 + S_3+ … + S_K + G_2*S_2+G_3*S_3+…+G_N*S_K, family=…)[A]	✓	Interaction term: (Same as Two-way ANOVA.) *Note: Run glm using the following arguments: `glm(model, family=poisson())`* *As linear-model, the Chi-square test is $\log(y_i) = \log(N) + \log(\alpha_i) + \log(\beta_j) + \log(\alpha_i\beta_j)$ where α_i and β_j are proportions. See more info in the accompanying notebook.*	*Same as Two-way ANOVA*
	N: Goodness of fit	chisq.test(y)	glm(y ~ 1 + G_2 + G_3 +…+ G_N, family=…)[A]	✓	(Same as One-way ANOVA and see Chi-Square note.)	*1W-ANOVA*

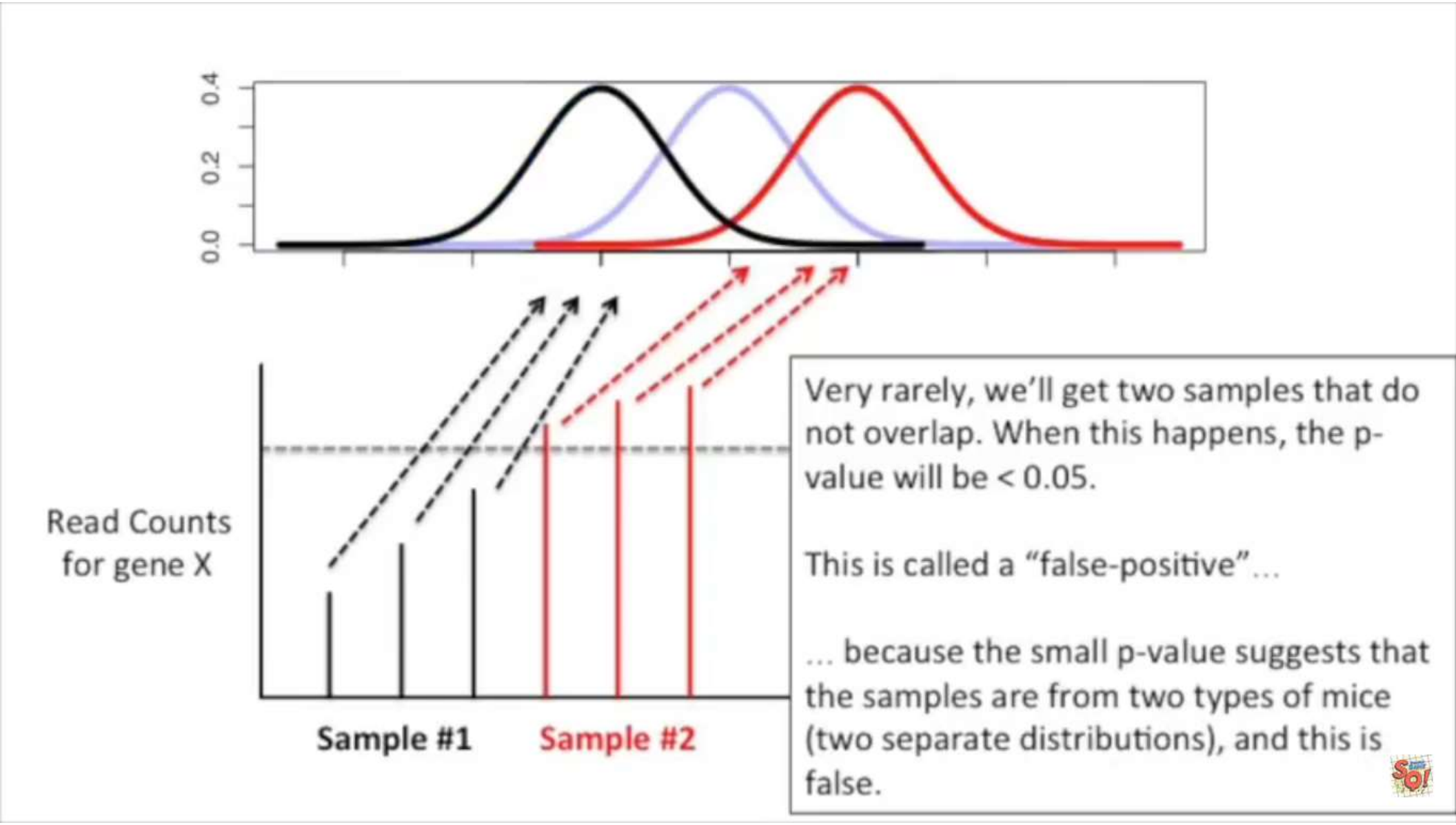
[1] Jonas Kristoffer Lindeløv

Comparaciones múltiples

Problema: *multiple hypothesis testing*

- Cuando monitoreamos un gran número de resultados experimentales, esperamos observar descubrimientos que ocurren al azar.
- Los métodos desarrollados para ajustar o reinterpretar los valores p son llamados *métodos de corrección por múltiples comparaciones*.
- p.e.: experimentos ómicos:

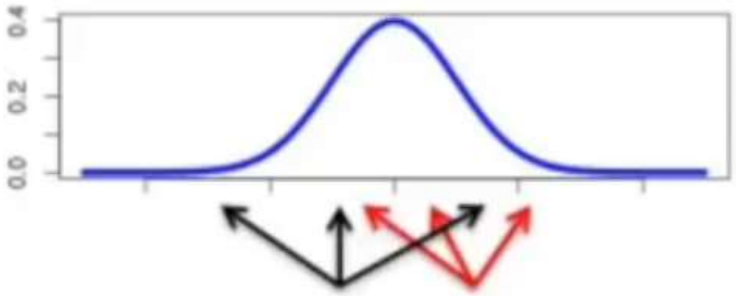




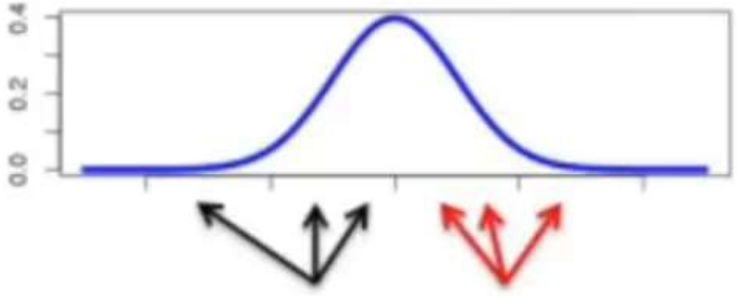
Stamer J. StatQuest

Normally, false positives are rare

95% of the time the samples will overlap.



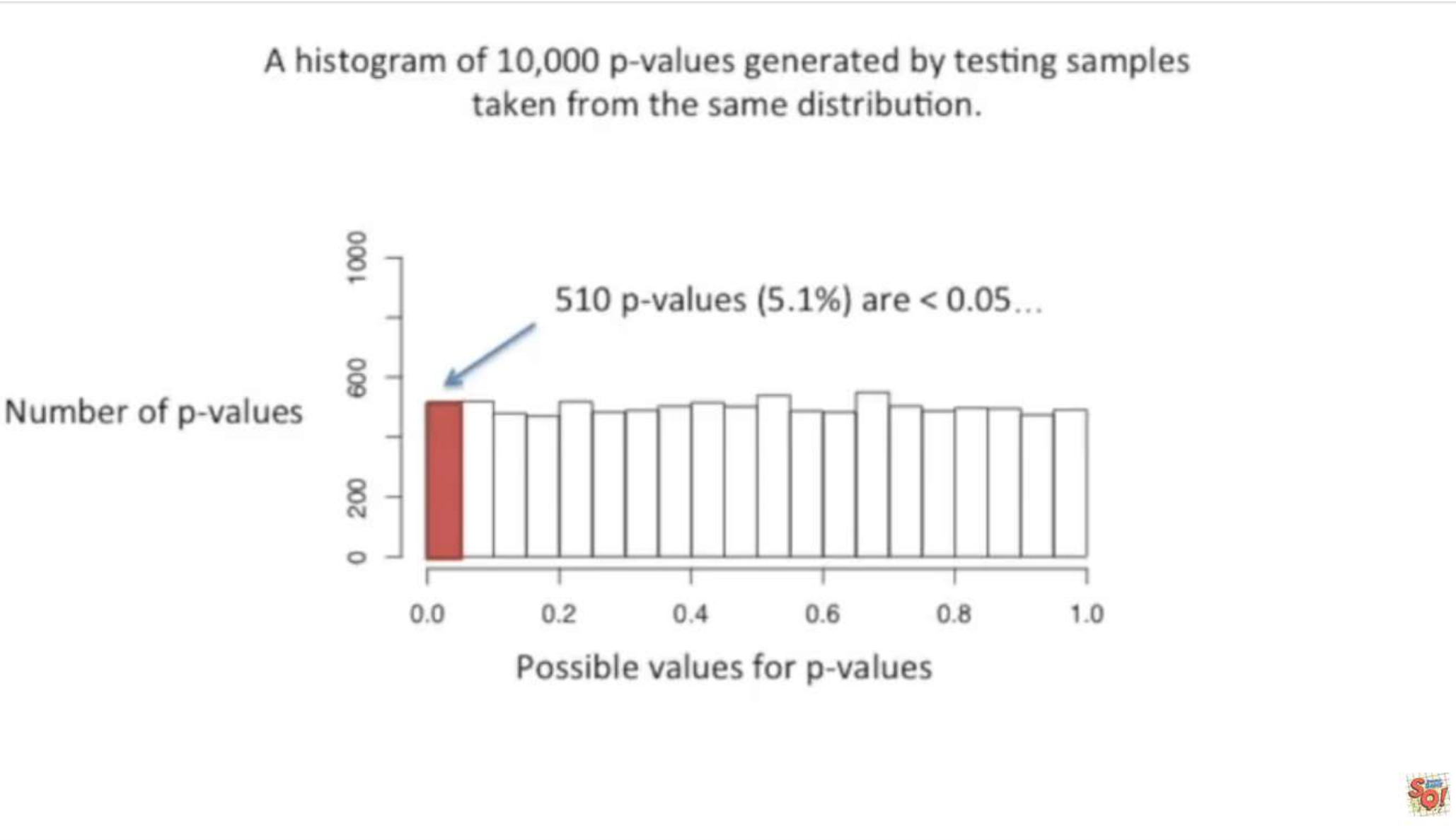
5% of the time they don't.



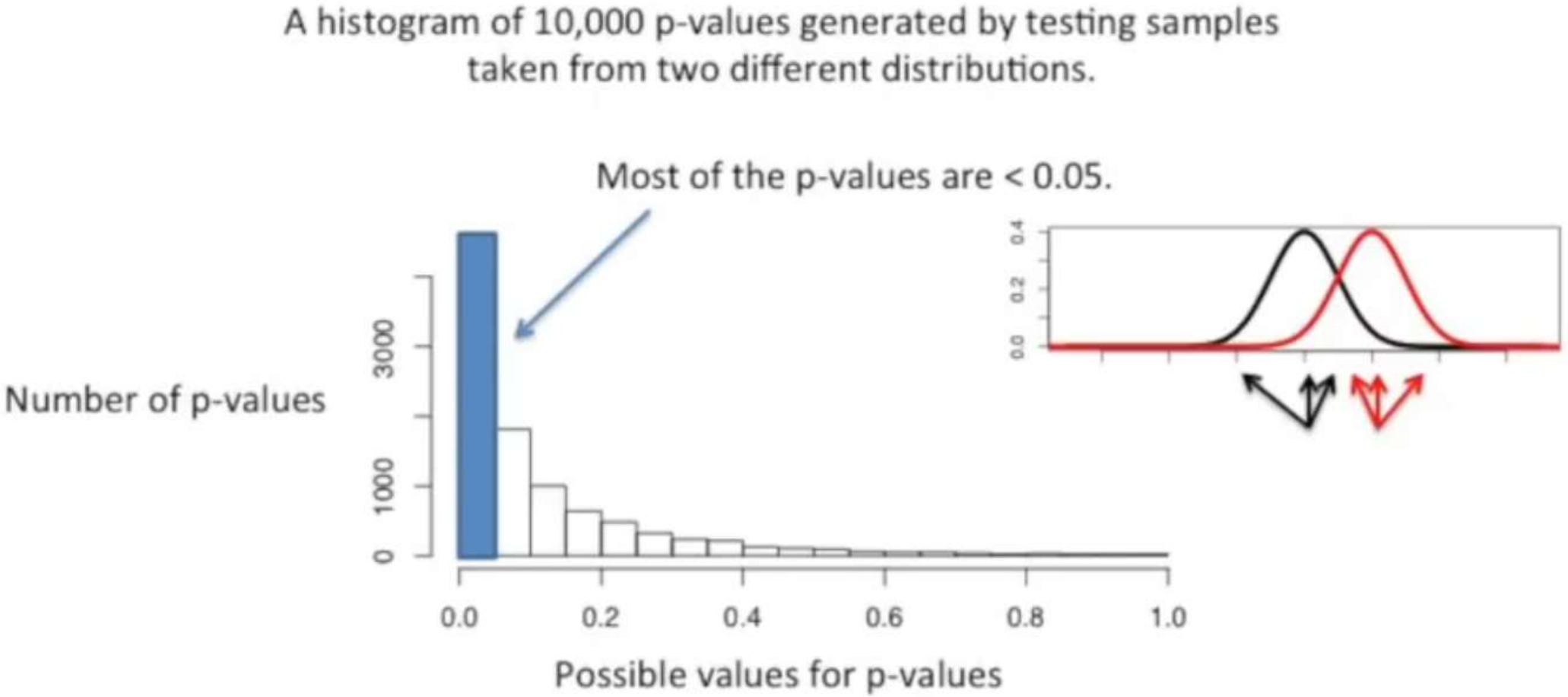
But human and mouse cells have at least 10,000 transcribed genes. If we took two samples from the same type of mice and compared all 10,000 genes…

5% of 10,000 = 500 false positives – 500 genes that appear interesting, even when they are not.

Stamer J. StatQuest



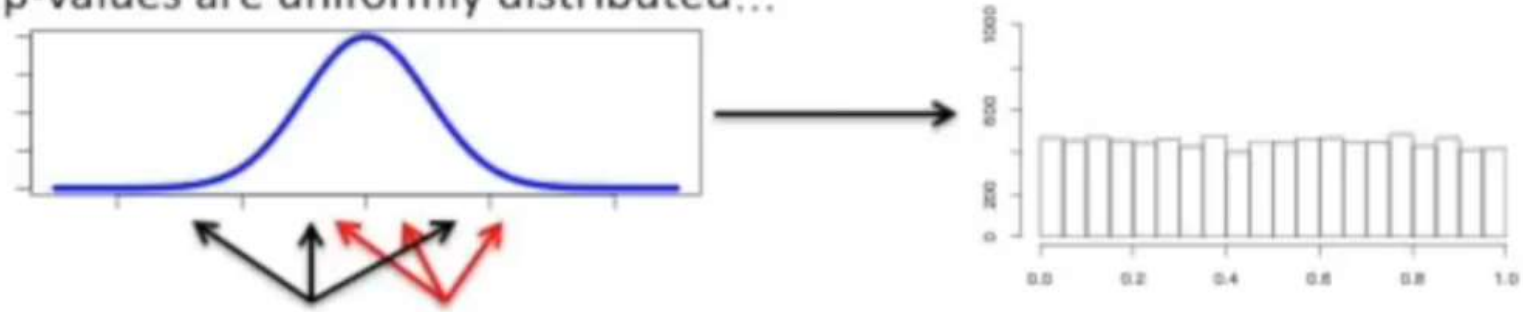
Stamer J. StatQuest



Stamer J. StatQuest

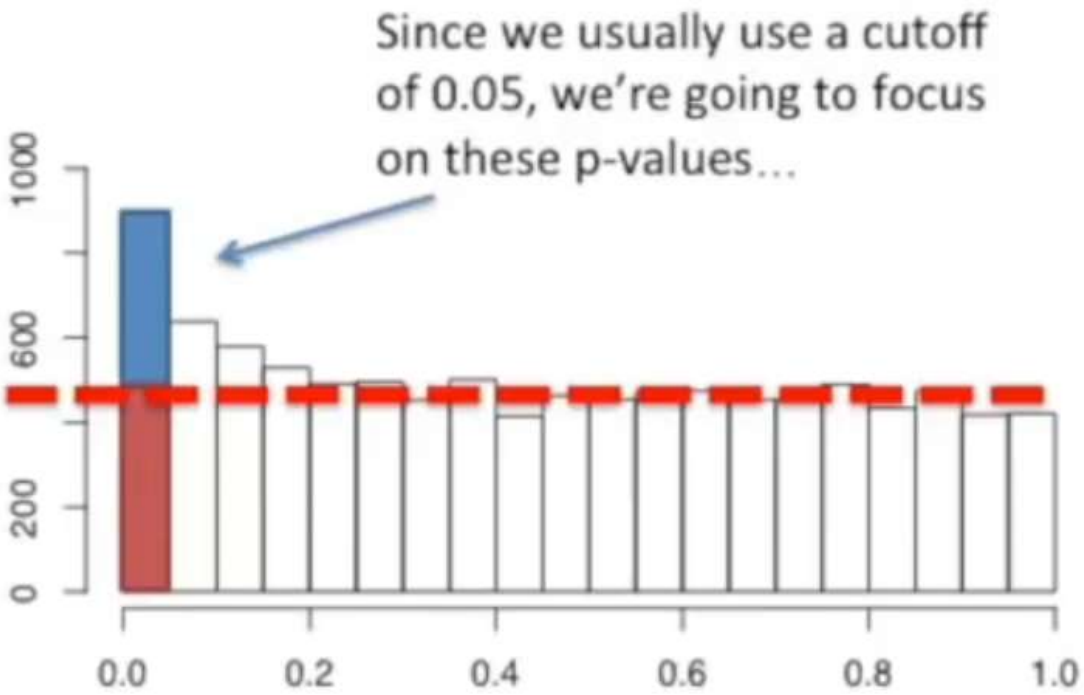
To summarize what we know so far…

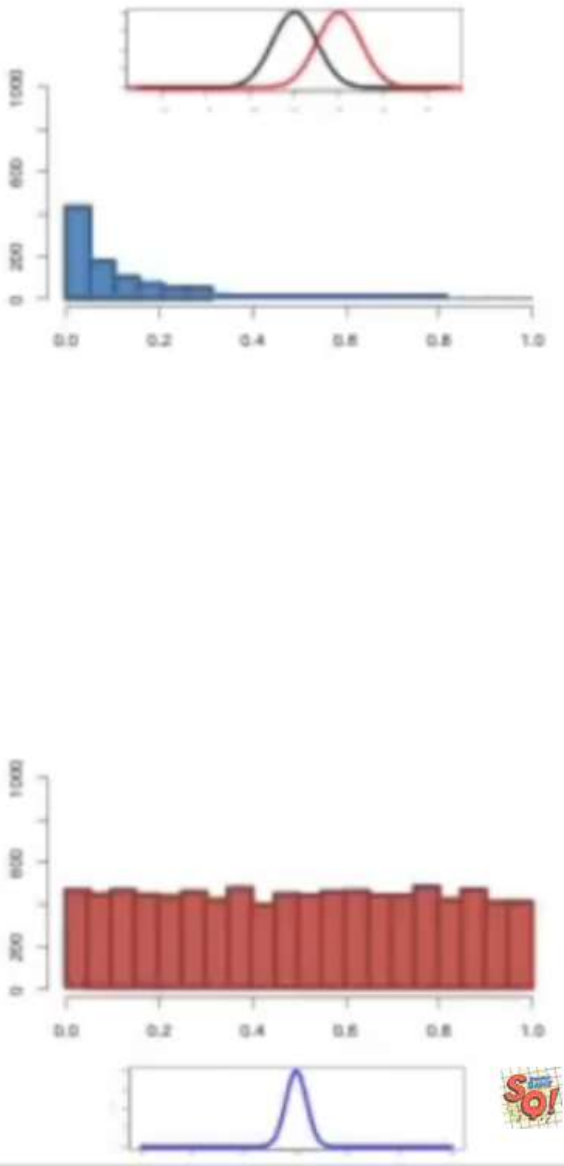
When samples come from the same distribution, the p-values are uniformly distributed…



When samples come from different distributions, the p-values are heavily skewed and closer to 0…

Stamer J. StatQuest





Stamer J. StatQuest

Método Benjamini Hochberg

1. ordenar ascendentemente y rankear valores p
2. el mayor valor p tiene el mismo valor de p ajustado
3. el siguiente es el menor entre: (i) el previo o (ii) la solución:

$$p_{adjusted} = p_{value} * \left(\frac{N_{p_{value}}}{rank_{p_{value}}} \right)$$

```
tibble(
  p=seq(from = 0.01,
        to = 0.91,
        by = 0.1),
  r=1:10)
```

```
## # A tibble: 10 x 2
##         p     r
##     <dbl> <int>
##  1  0.01      1
##  2  0.11      2
##  3  0.21      3
##  4  0.31      4
##  5  0.41      5
##  6  0.51      6
##  7  0.61      7
##  8  0.71      8
##  9  0.81      9
## 10  0.91     10
```

Método Benjamini Hochberg

1. ordenar ascendentemente y rankear valores p
2. el mayor valor p tiene el mismo valor de p ajustado
3. el siguiente es el menor entre: (i) el previo o (ii) la solución:

$$p_{adjusted} = p_{value} * \left(\frac{N_{p_{value}}}{rank_{p_{value}}} \right)$$

```
tibble(
  p=seq(from = 0.01,
        to = 0.91,
        by = 0.1),
  r=1:10) %>%
  mutate(
    p.adjust=
      p.adjust(
        p = p,
        method = "BH"))
```

```
## # A tibble: 10 x 2
##         p     r
##     <dbl> <int>
##  1  0.01      1
##  2  0.11      2
##  3  0.21      3
##  4  0.31      4
##  5  0.41      5
##  6  0.51      6
##  7  0.61      7
##  8  0.71      8
##  9  0.81      9
## 10  0.91     10
```

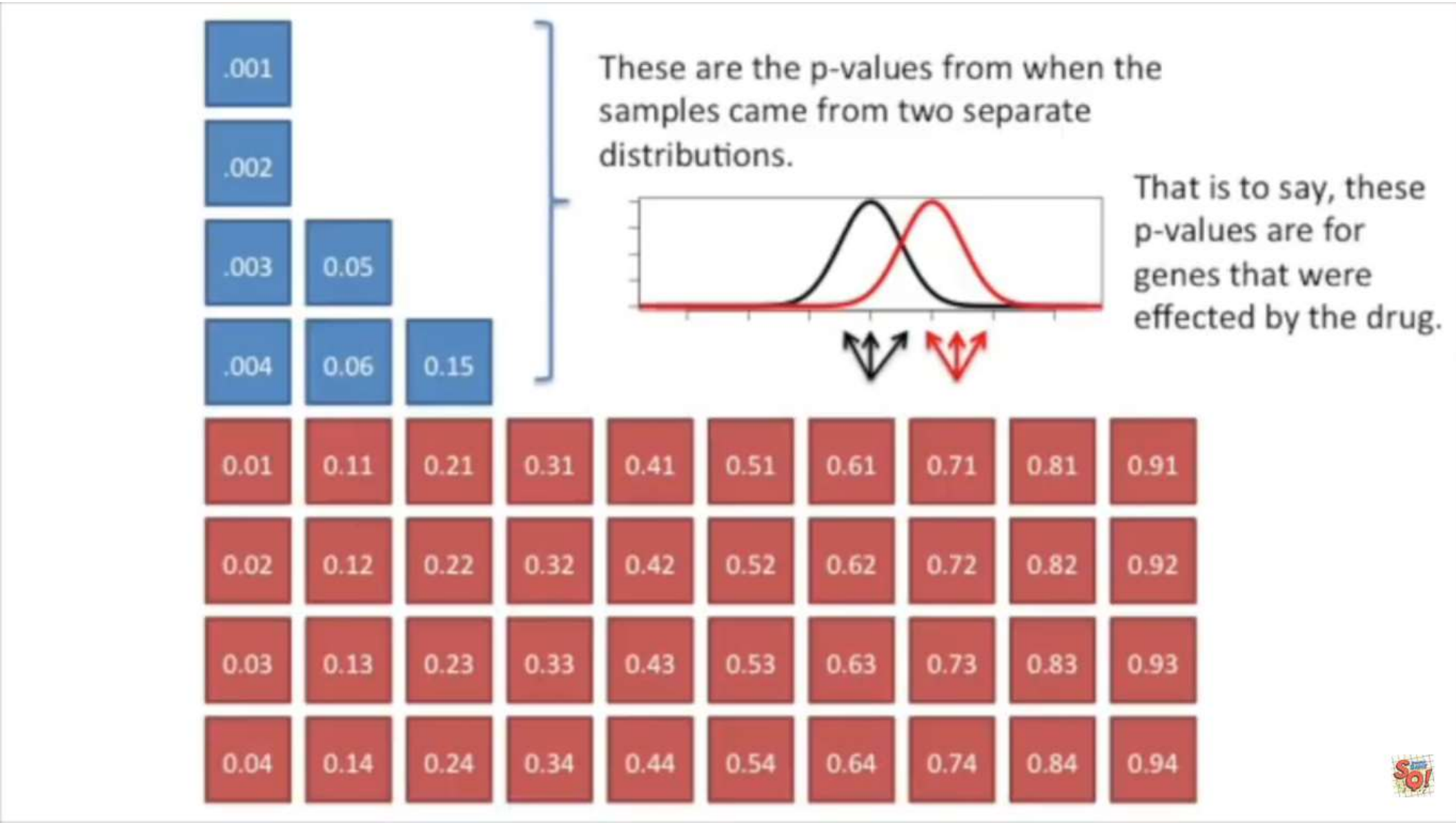
Método Benjamini Hochberg

1. ordenar ascendentemente y rankear valores p
2. el mayor valor p tiene el mismo valor de p ajustado
3. el siguiente es el menor entre: (i) el previo o (ii) la solución:

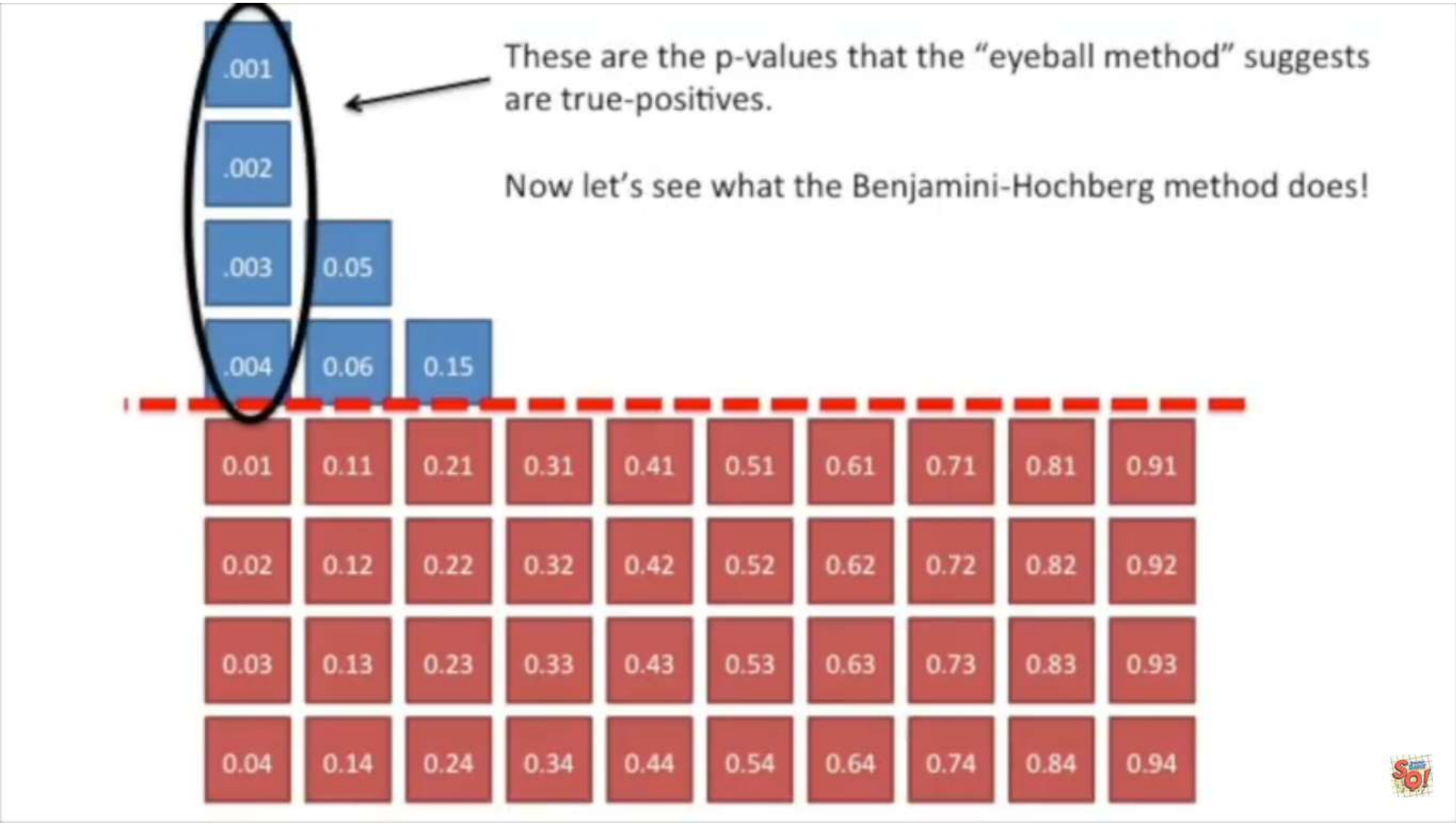
$$p_{adjusted} = p_{value} * \left(\frac{N_{p_{value}}}{rank_{p_{value}}} \right)$$

```
tibble(
  p=seq(from = 0.01,
        to = 0.91,
        by = 0.1),
  r=1:10) %>%
  mutate(
    p.adjust=
      p.adjust(
        p = p,
        method = "BH"))
```

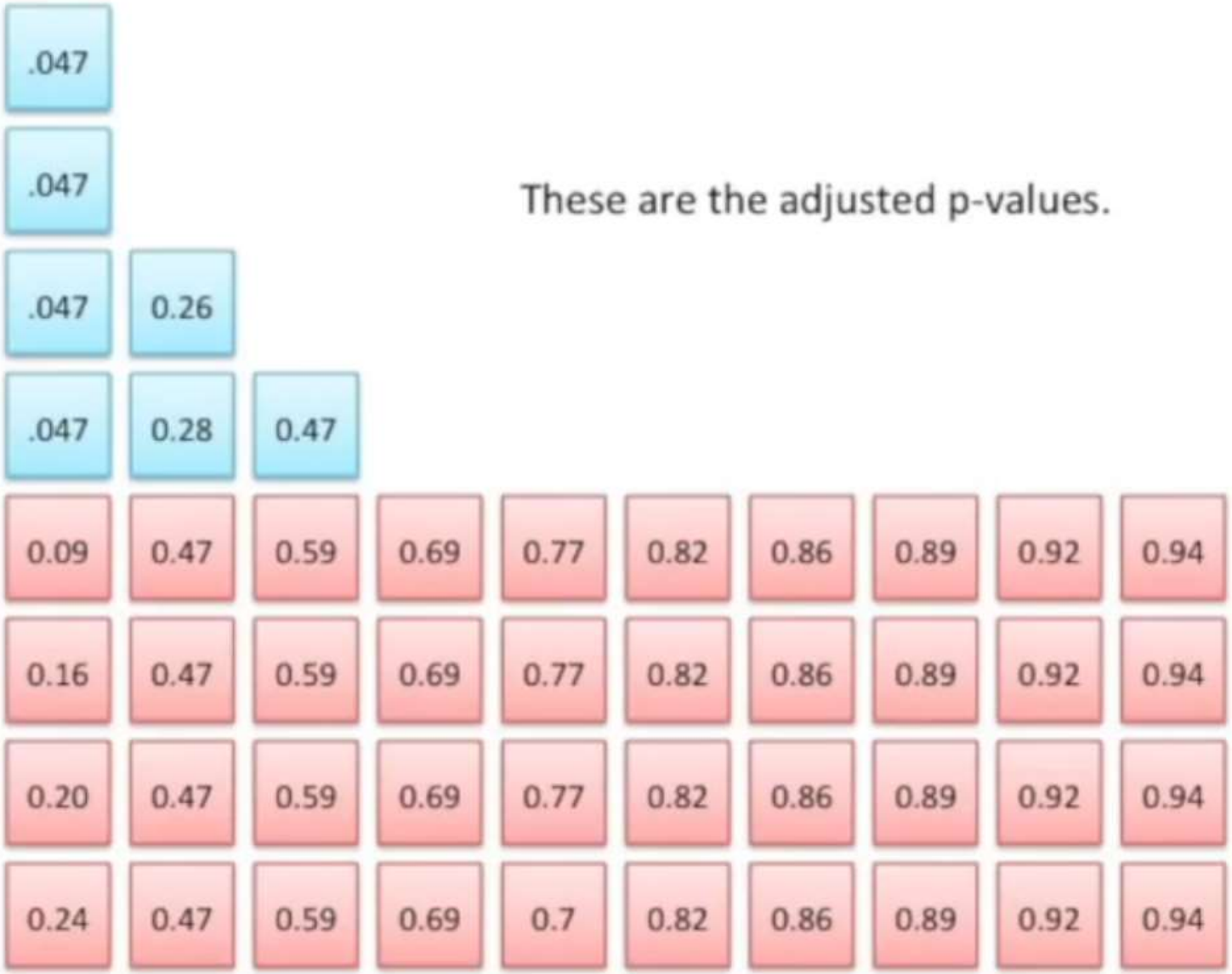
```
## # A tibble: 10 x 3
##         p     r p.adjust
##     <dbl> <int>    <dbl>
##  1  0.01      1    0.1
##  2  0.11      2    0.55
##  3  0.21      3    0.7
##  4  0.31      4    0.775
##  5  0.41      5    0.82
##  6  0.51      6    0.85
##  7  0.61      7    0.871
##  8  0.71      8    0.888
##  9  0.81      9    0.9
## 10  0.91     10    0.91
```



Stamer J. StatQuest



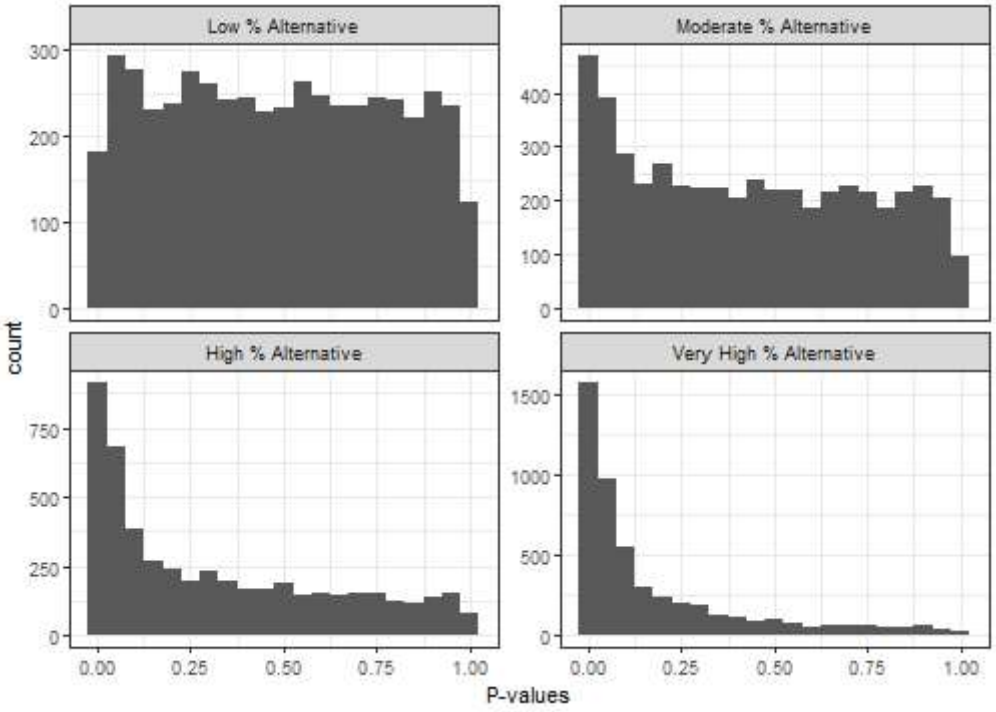
Stamer J. StatQuest



Stamer J. StatQuest

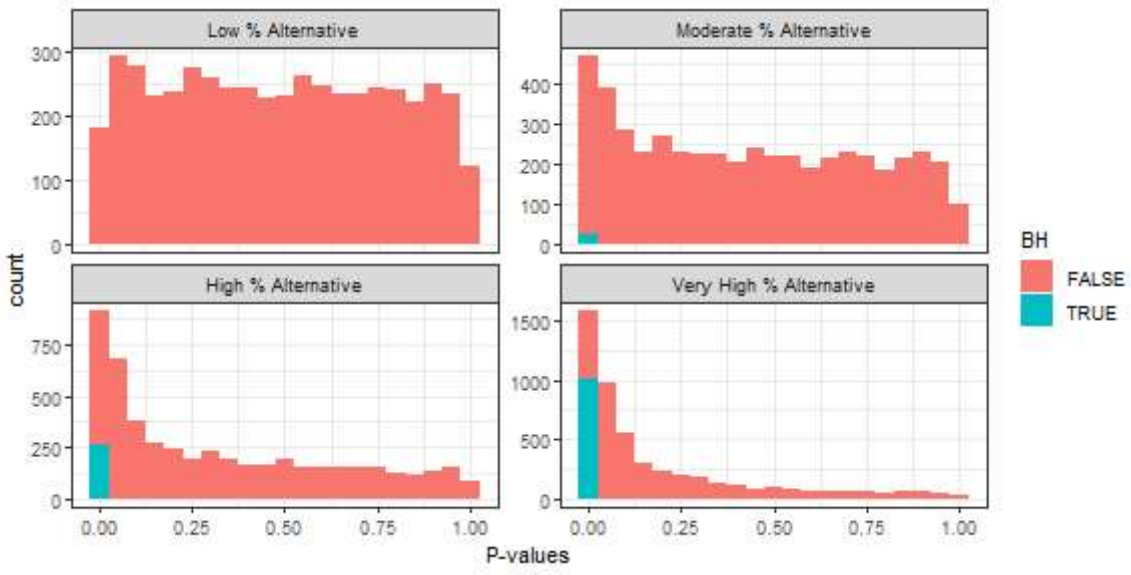
Ejemplo: método BH

```
multpv %>%
  ggplot(aes(p.value)) +
  geom_histogram(binwidth=.05) +
  facet_wrap(~ type, scale="free_y", nrow=2) +
  labs(x="P-values")
```



Ejemplo: método BH

```
multpv %>%
  group_by(type) %>%
  mutate(p.adjust=p.adjust(p = p.value,method = "BH"),
         p.adjust_pass=if_else(p.adjust<0.05,"TRUE","FALSE")) %>%
  ungroup() %>%
  ggplot(aes(p.value,fill=p.adjust_pass)) + geom_histogram(binwidth=.05) +
  facet_wrap(~ type, scale="free_y", nrow=2) +labs(x="P-values",fill="BH")
```



Aplicación: Microarray

- Diseño:
  - Medición de la expresión en inanición (Brauer, 2008)
  - Cultivos de *S. cerevisiae* expuestos a seis concentraciones de nutrientes, suministrado en seis tasas o *rates* por un quimiostato.
  - Un *gene expression microarray* mide qué tanto un gen se expresa en determinada condición.

```
cleaned_data <- read_rds("data-raw/microarraydata.rds")
```

```
cleaned_data %>%
  count(nutrient)
```

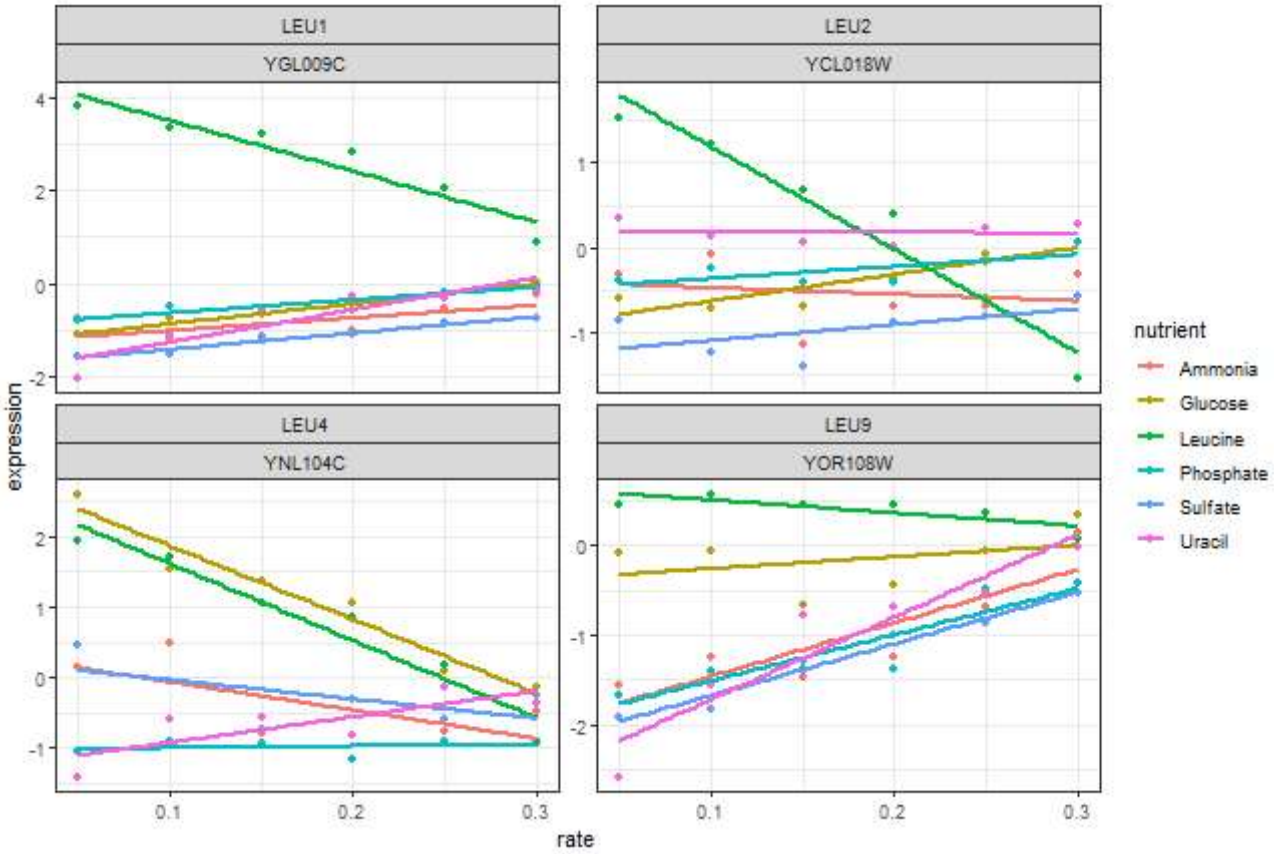
```
## # A tibble: 6 x 2
##   nutrient      n
##   <chr>     <int>
## 1 Ammonia   33141
## 2 Glucose   33138
## 3 Leucine   33178
## 4 Phosphate 33068
## 5 Sulfate   32897
## 6 Uracil    33008
```

```
cleaned_data %>%
  count(rate)
```

```
## # A tibble: 6 x 2
##    rate     n
##   <dbl> <int>
## 1  0.05 32741
## 2  0.1  33132
## 3  0.15 33145
## 4  0.2  33121
## 5  0.25 33177
## 6  0.3  33114
```

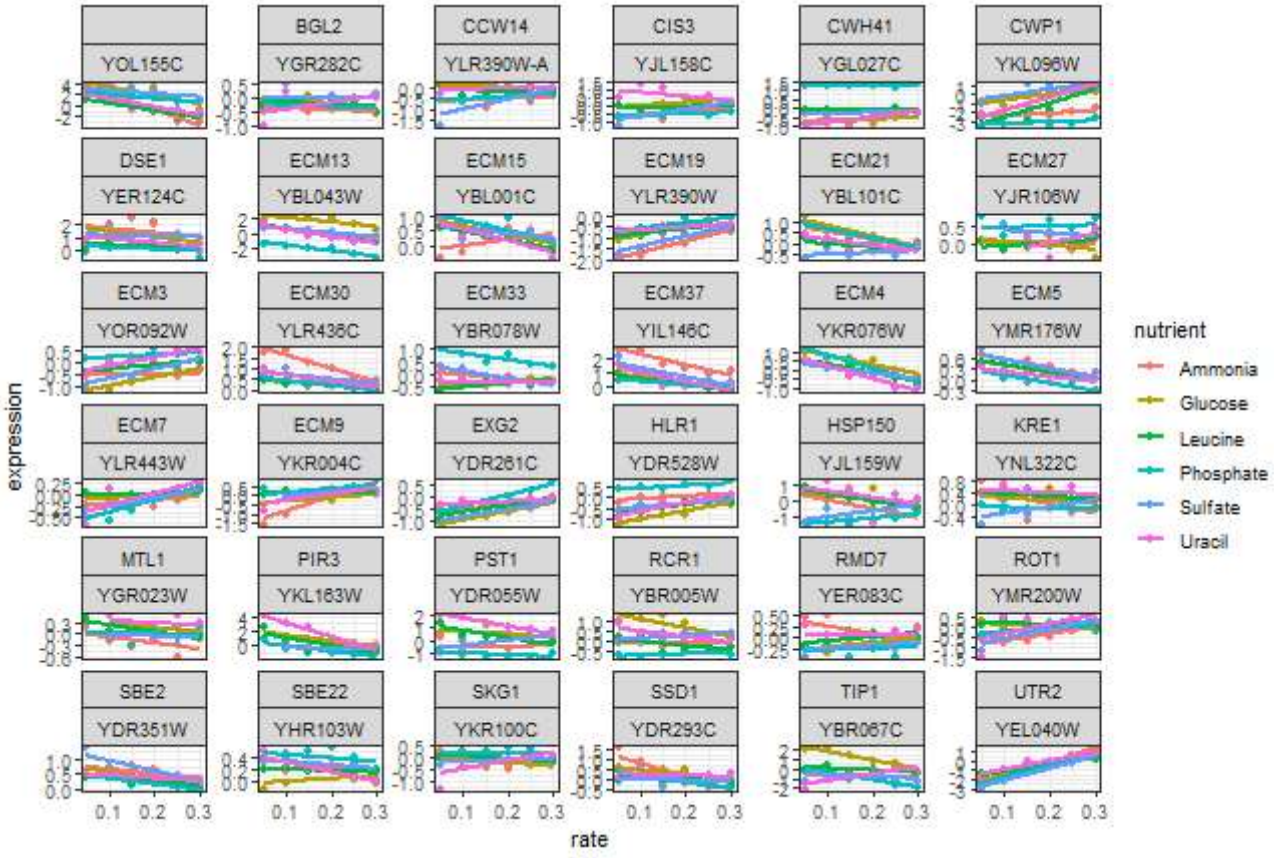
Microarray: visualización

```
cleaned_data %>%
  filter(BP == "leucine biosynthesis") %>%
  plot_expression_data() #función detallada en: practica-05.R
```



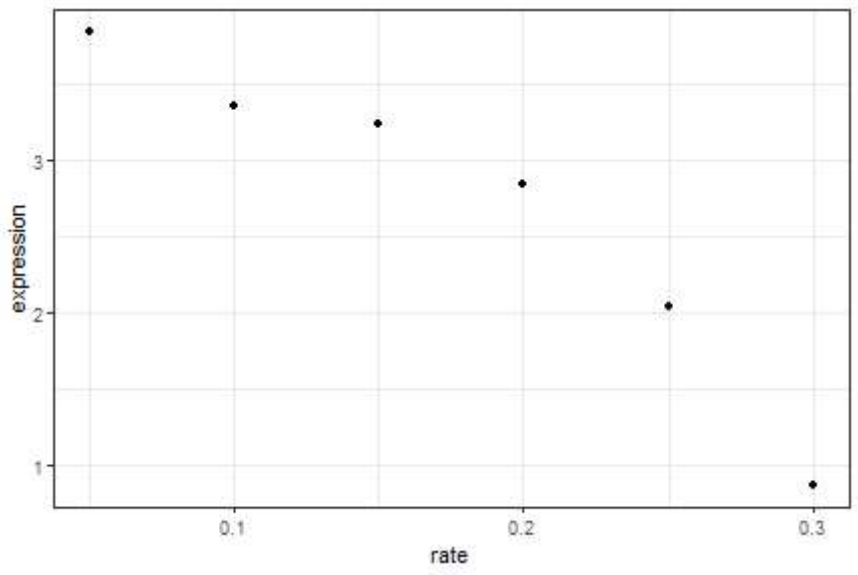
Microarray: visualización

```
cleaned_data %>%
  filter(BP == "cell wall organization and biogenesis") %>%
  plot_expression_data()
```



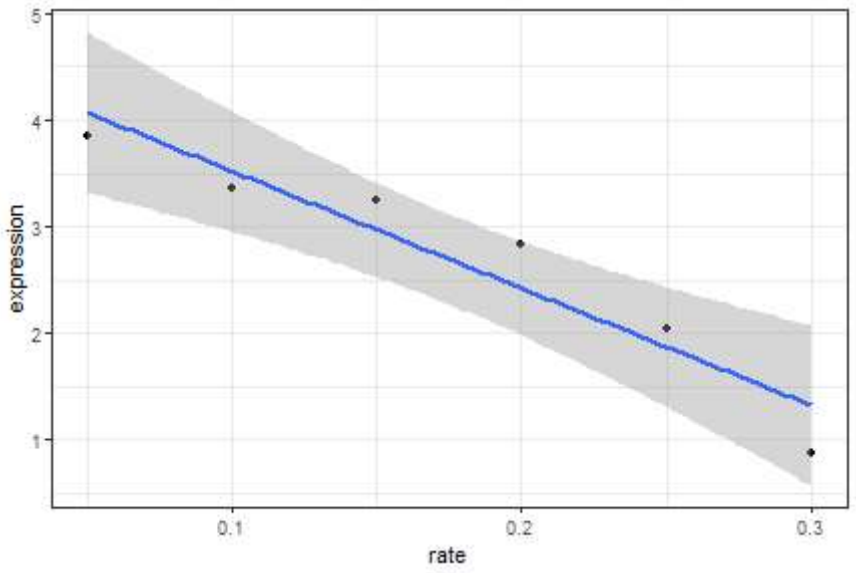
Microarray: Regresión Lineal Simple

```
cleaned_data %>%
  #elegimos 01 gen y 01 nutriente
  filter(name == "LEU1", nutrient == "Leucine") %>%
  #graficamos la relación Y: expresión ~ X: rate
  ggplot(aes(rate, expression)) +
  #empleamos la geometría punto
  geom_point()
```



Microarray: Regresión Lineal Simple

```
cleaned_data %>%
  #elegimos 01 gen y 01 nutriente
  filter(name == "LEU1", nutrient == "Leucine") %>%
  #graficamos la relación Y: expresión ~ X: rate
  ggplot(aes(rate, expression)) +
  #empleamos la geometría punto
  geom_point() + geom_smooth(method = "lm")
```



Microarray: Regresión Lineal Simple

```
cleaned_data %>%
  #elegimos 01 gen y 01 nutriente
  filter(name == "LEU1", nutrient == "Leucine") %>%
  #ajustamos una regresión lineal
  #dado que data no es el primer argumento
  #necesitamos especificarlo en data con "."
  lm(expression ~ rate, data = .) %>%
  #visualizamos tabla con estimados
  tidy()
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)     4.62     0.348     13.3  0.000186
## 2 rate          -11.0      1.79      -6.14 0.00356
```

Microarray: a todas los pares gen-nutriente

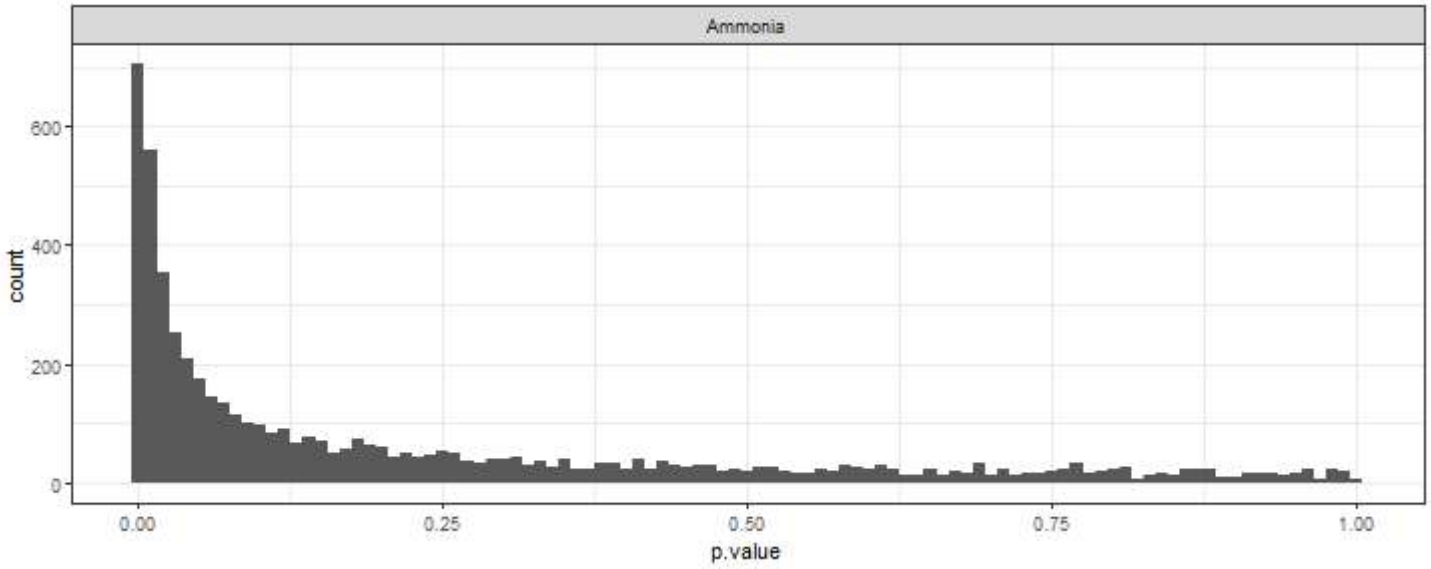
```
linear_models <- cleaned_data %>%
  filter(nutrient=="Ammonia") %>% #filtramos por nutriente
  group_by(name, systematic_name, nutrient) %>% #agrupamos por gen
  nest() %>% #anidamos los datos en una columna lista de df
  # ajustamos un modelo lineal a cada fila -ver paquete purrr::map-
  mutate(model = map(data, ~ lm(expression ~ rate, data = .x)),
         tidym = map(model,tidy))
```

```
linear_models
```

```
## # A tibble: 5,536 x 6
## # Groups:   name, systematic_name, nutrient [5,536]
##    name  systematic_name nutrient           data model  tidym
##    <chr> <chr>           <chr>    <list<df[,4]>> <list> <list>
##  1 SFB2  YNL049C         Ammonia         [6 x 4] <lm>   <tibble [2 x 5]>
##  2 ""    YNL095C         Ammonia         [6 x 4] <lm>   <tibble [2 x 5]>
##  3 QRI7  YDL104C         Ammonia         [6 x 4] <lm>   <tibble [2 x 5]>
##  4 CFT2  YLR115W         Ammonia         [6 x 4] <lm>   <tibble [2 x 5]>
##  5 SSO2  YMR183C         Ammonia         [6 x 4] <lm>   <tibble [2 x 5]>
##  6 PSP2  YML017W         Ammonia         [6 x 4] <lm>   <tibble [2 x 5]>
##  7 RIB2  YOL066C         Ammonia         [6 x 4] <lm>   <tibble [2 x 5]>
##  8 VMA13 YPR036W         Ammonia         [6 x 4] <lm>   <tibble [2 x 5]>
##  9 EDC3  YEL015W         Ammonia         [6 x 4] <lm>   <tibble [2 x 5]>
## 10 VPS5  YOR069W         Ammonia         [6 x 4] <lm>   <tibble [2 x 5]>
## # ... with 5,526 more rows
```

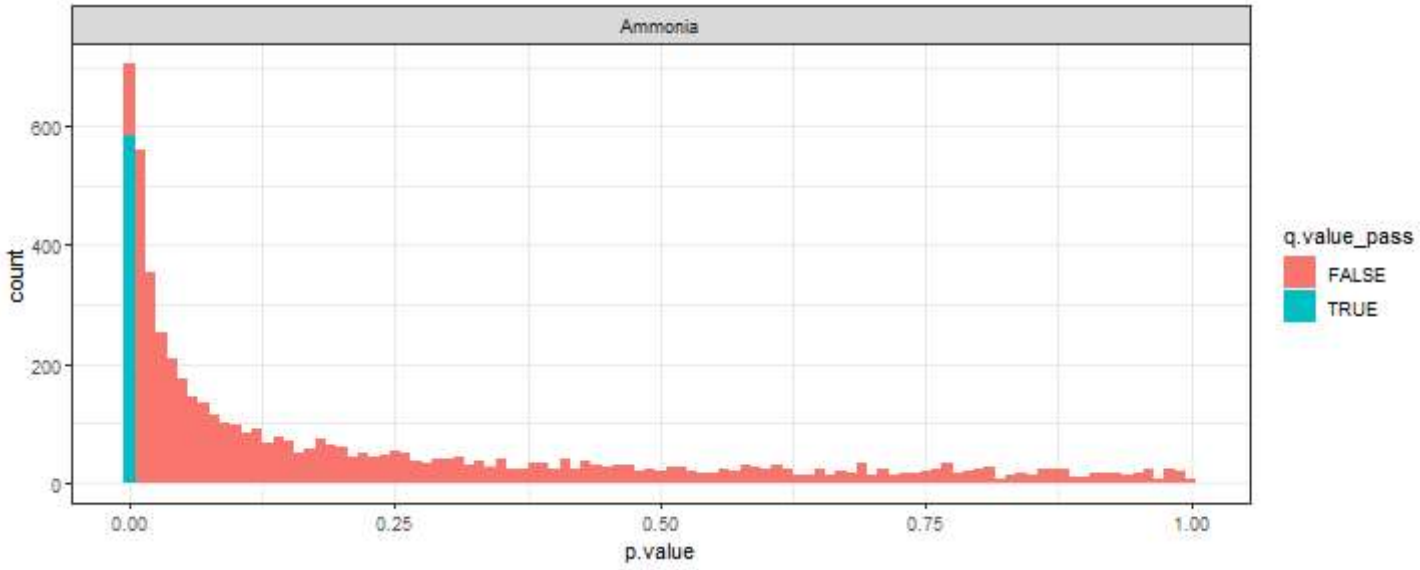
Microarray: conservamos las pendientes

```
slope_terms <- linear_models %>%
  unnest(cols = c(tidym)) %>%
  ungroup() %>%
  filter(term=="rate" & !is.na(p.value))

slope_terms %>%
  ggplot(aes(p.value)) +
  geom_histogram(binwidth = .01) +
  facet_wrap(~nutrient)
```



Microarray: corregimos valor p

```
slope_terms_adj <- slope_terms %>%
  mutate(q.value = qvalue(p.value)$qvalues,
         q.value_pass=if_else(q.value < .01,"TRUE","FALSE"))

slope_terms_adj %>%
  ggplot(aes(p.value,
             fill=q.value_pass)) +
  geom_histogram(binwidth = .01) +
  facet_wrap(~nutrient)
```



Microarray: Lista de genes de interés

```
slope_terms_adj %>%
  filter(q.value_pass=="TRUE") %>%
  arrange(q.value) %>%
  select(systematic_name,nutrient,term,estimate,p.value,q.value)
```

```
## # A tibble: 585 x 6
##    systematic_name nutrient term  estimate     p.value  q.value
##    <chr>           <chr>    <chr>    <dbl>       <dbl>    <dbl>
##  1 YBR291C         Ammonia  rate      11.6  0.000000582 0.000951
##  2 YLR174W         Ammonia  rate     -14.0  0.00000162  0.00132
##  3 YLL003W         Ammonia  rate      -4.22 0.00000346  0.00189
##  4 YMR053C         Ammonia  rate      -7.25 0.00000656  0.00268
##  5 YEL001C         Ammonia  rate       6.61 0.0000155   0.00276
##  6 YHR068W         Ammonia  rate      10.0  0.0000166   0.00276
##  7 YDR069C         Ammonia  rate      -4.49 0.00000939  0.00276
##  8 YML128C         Ammonia  rate     -15.0  0.0000169   0.00276
##  9 YPR049C         Ammonia  rate      -5.90 0.0000128   0.00276
## 10 YGR244C         Ammonia  rate      -5.44 0.0000135   0.00276
## # ... with 575 more rows
```

Práctica 5

- reproducir el proceso para los demás nutrientes
- explorar paquete `limma`
- explorar la documentación del paquete `qvalue` link

```
library(qvalue)

#datos microarray
qmicro <- qvalue(p = slope_terms$p.value)
summary(qmicro)
hist(qmicro)
plot(qmicro)
```

Referencias

- Logan M. (2011). Biostatistical design and analysis using R: a practical guide. John Wiley & Sons. Library Genesis
- Akehurst H. (2016) Bioestadística con R. Curso dictado en UPCH.
- Statistics for Biologist: Points of Significance. Nature Collection
- Ho J, et al. (2019). Moving beyond P values: data analysis with estimation graphics. Nature Methods
- Stamer J. StatQuest. False Discovery Rates, FDR, clearly explained. YouTube channel
- Ding J, et al. (2018). Association of gene polymorphism of SDF1(CXCR12) with susceptibility to HIV-1 infection and AIDS disease progression: A meta-analysis. PLoS One
- Estrada-Aguirre JA, et al. (2013). Protective effect of CCR5 Delta-32 allele against HIV-1 in Mexican women. Curr HIV Res
- Lindeløv J. (2019) Common statistical tests are linear models (or: how to teach stats) Github personal webpage
- Robinson D. (2014). Modeling gene expression with broom: a case study in tidy analysis. Variance explained blog

Material complementario

- Rstudio Education: Learn
    - beginner, intermediate, expert tracks
    - transform table here
    - tidy explain *_join verbs here
- Rstudio Cheat sheets
    - data import
    - data transformation
    - data visualization

¡Gracias!

Andree Valle Campos

@avallecam

avallecam@gmail.com

Slides created via the R package **xaringan**.