

BSM

Blockchain School
for Management

Tema 8

Módulo II: R. Visualización de datos I

Nicolás Forteza

2022-11-21

Antes de ponernos a programar, hay que revisar antes cuál es la teoría en la visualización de datos (o data viz).

El data viz es la parte del análisis de datos en la que se grafican los mismos, mapeando los valores de los datos con elementos visuales y cognitivos.

Visualización de datos I

¿Cuántos 7 ves aquí abajo?

134091345713409581475109438510497510984510
4937510943751049571094357109457109475109451'
04581'0435'13457134'058140581458914751'4395814
9051457143'059i14'501435771'340501'4501'4577777
145091'0345910'45910'45910'4395'01495'013457

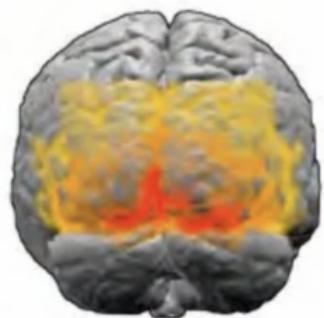
Visualización de datos I

720349656089226535931140790070
322302076958689027429003358787
115045223998424533087922668417
382319480046553364246202505406
711172160430997890121737608183
566145635519888049583302306957
749597705315240714467203496560
892265359311407900703223020769
586890274290033587871150452239
984245330879226684173823194800
465533642462025054067111721604
309978901217376081835661456355

Visualización de datos I



10 Million Bits
Per Second



Visualización de datos I



COLOR HUE



ORIENTATION



TEXTURE



POSITION & ALIGNMENT



COLOR BRIGHTNESS



COLOR SATURATION

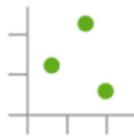


SIZE



SHAPE

Visualización de datos I



Position



Length



Angle/Slope



Area



Volume



Difference



Color hue



Color Saturation



Contrast

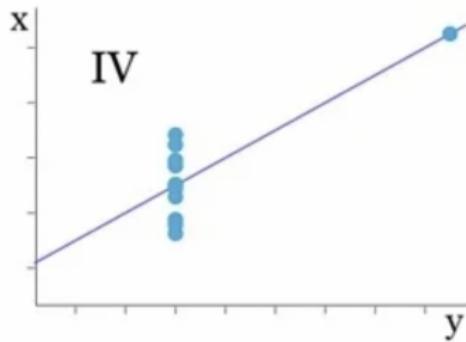
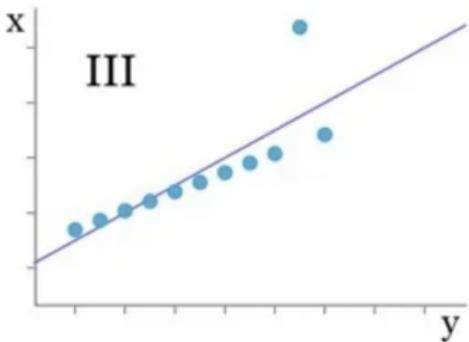
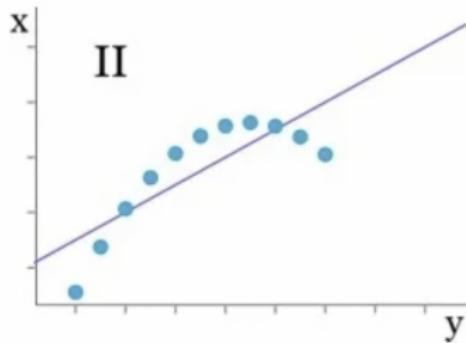
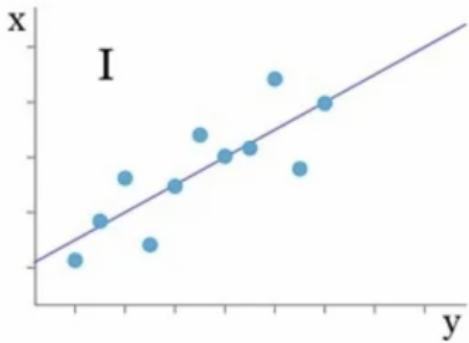


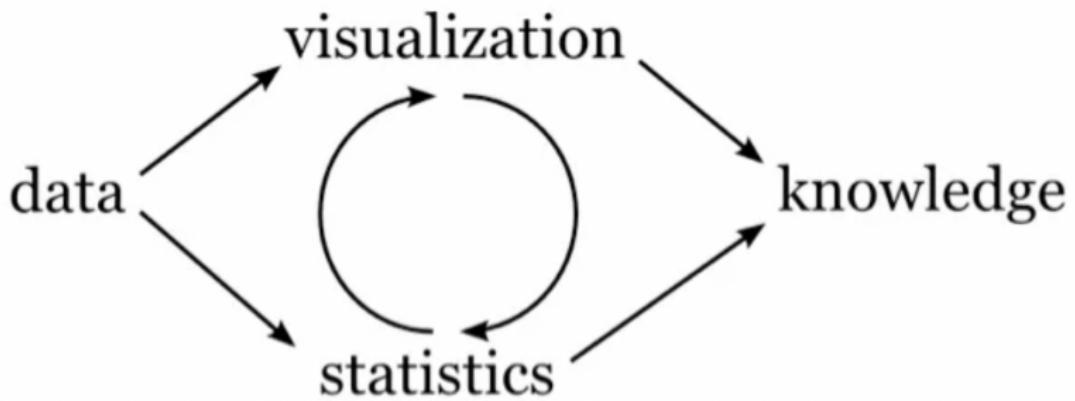
Texture

Visualización de datos I

Example	Encoding	Ordered	Useful values	Quantitative	Ordinal	Categorical	Relational
	position, placement	yes	infinite	Good	Good	Good	Good
1, 2, 3; A, B, C	text labels	optional alpha or num	infinite	Good	Good	Good	Good
	length	yes	many	Good	Good		
	size, area	yes	many	Good	Good		
	angle	yes	medium	Good	Good		
	pattern density	yes	few	Good	Good		
	weight, boldness	yes	few		Good		
	saturation, brightness	yes	few		Good		
	color	no	few (<20)			Good	
	shape, icon	no	medium			Good	
	pattern texture	no	medium			Good	
	enclosure, connection	no	infinite			Good	Good
	line pattern	no	few				Good
	line endings	no	few				Good

Visualización de datos I





Existen un conjunto de normas, reglas y buenas prácticas que tenemos que conocer antes de graficar un conjunto de datos.

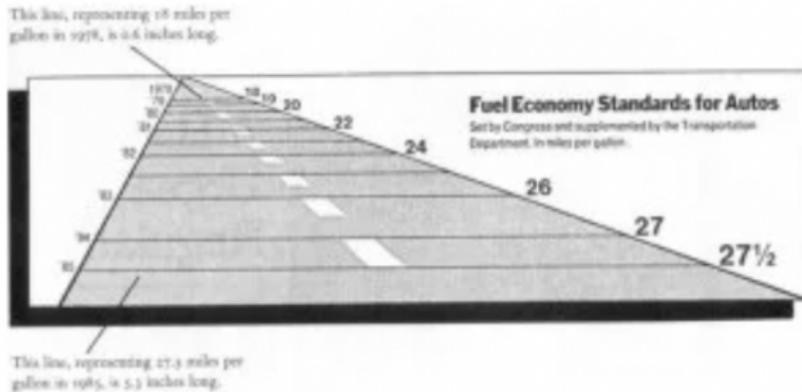
Edward Tufte vue, en el S.XX; el gran impulsor de este movimiento.

1-. Conseguir la integridad gráfica.

Nuestros gráficos deben representar la verdad

Visualización de datos I

El factor “mentira” se calcula dividiendo el tamaño del efecto mostrado en el gráfico entre el tamaño del efecto en los datos.

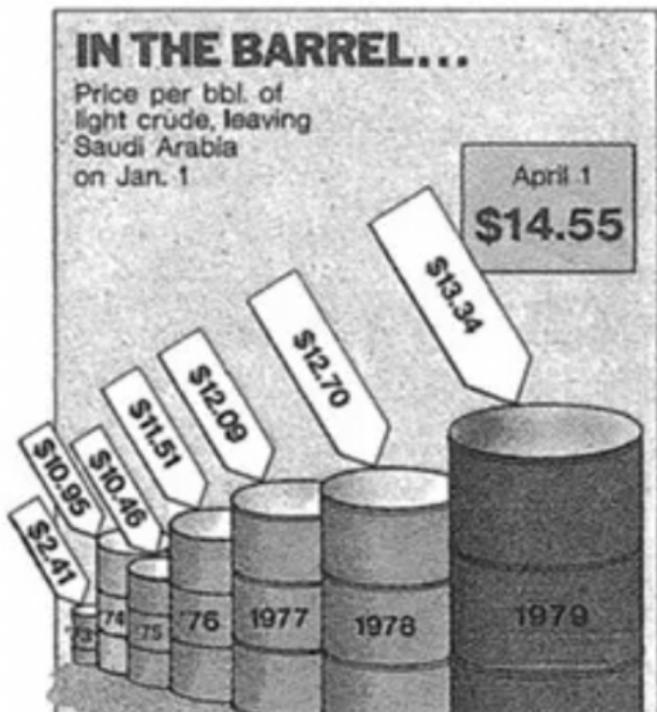


New York Times, August 9, 1978, p. D-2.

According to Tufte the Lie Factor of this graph is 14.8. A numerical change of 53% is represented by a graphical change (size of horizontal lines) of 783%.

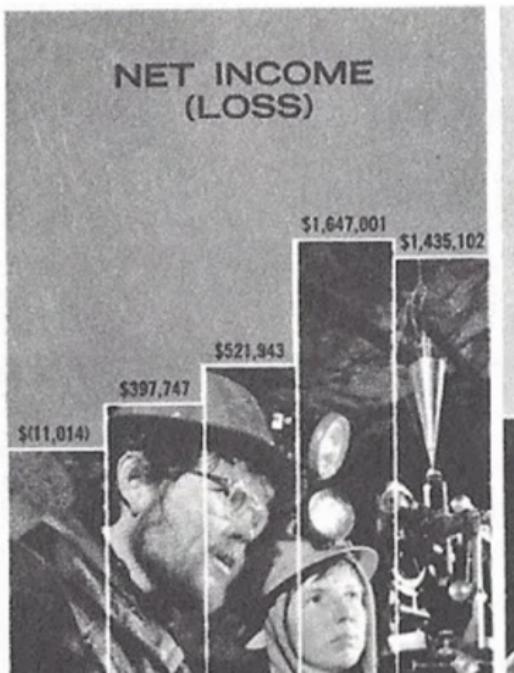
Visualización de datos I

La representación numérica en la superficie debe ser directamente proporcional a las cantidades numéricas de los datos.



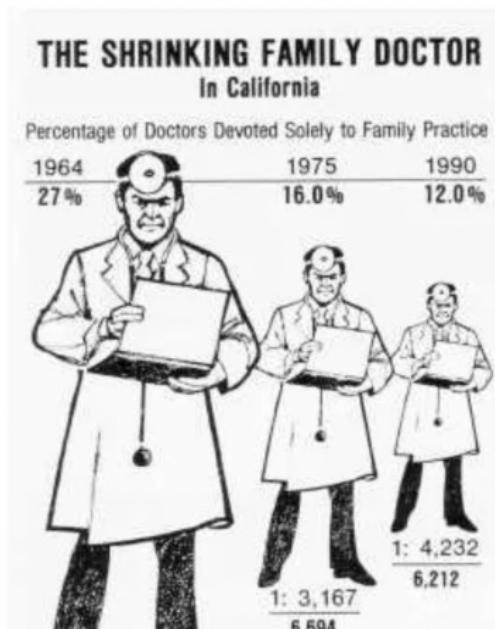
Visualización de datos I

Los *labels* del gráfico tienen que ser claros, detallados y concisos, sin dar pie a ambigüedades. E incluso graficar eventos importantes en los datos.



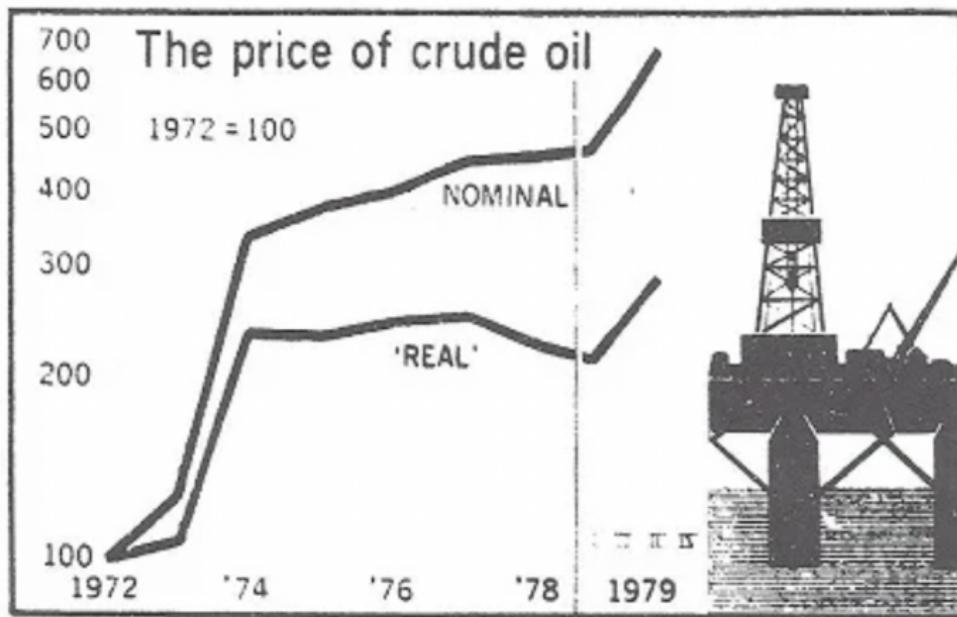
Visualización de datos I

Se tiene que mostrar siempre la variación en los datos, no la variación en tu diseño del gráfico.



Visualización de datos I

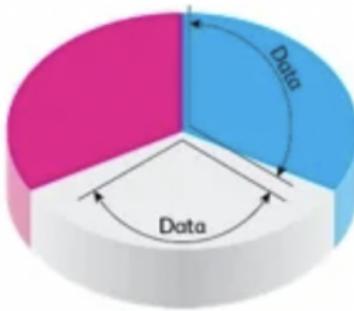
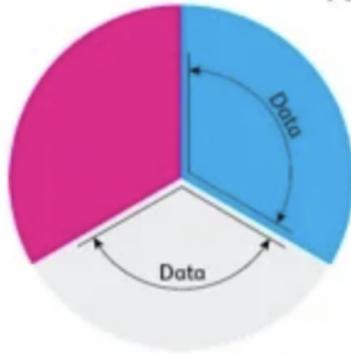
En las series temporales donde se traten unidades monetarias, se debe respetar el deflactor de la serie y estandarizar las unidades (o fijarlas en la misma escala).



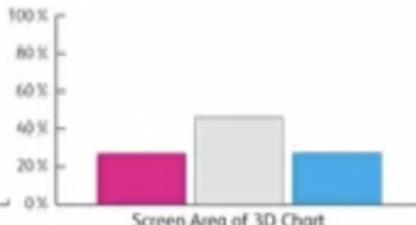
Visualización de datos I

El número de dimensiones en el gráfico nunca debe superar al número de dimensiones que de los datos.

Angle



Area

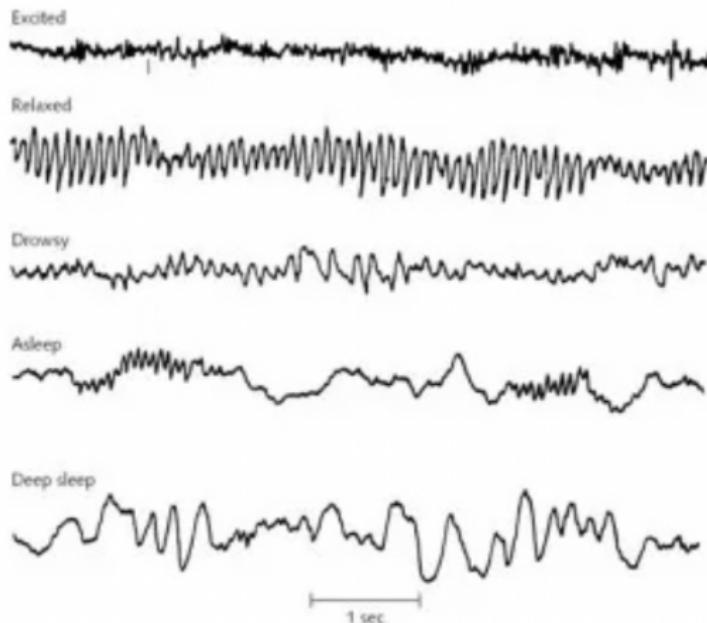


2-. Maximizar la *tinta usada para los datos*

La tinta en un gráfico representa datos. Las buenas representaciones gráficas deben maximizar esto y deben borrar tinta no relacionada con los datos. El ratio de tinta usada para datos vs. tinta usada para otras cosas se calcula como 1 menos la proporción del gráfico que se puede borrar sin que se pierda información

Visualización de datos I

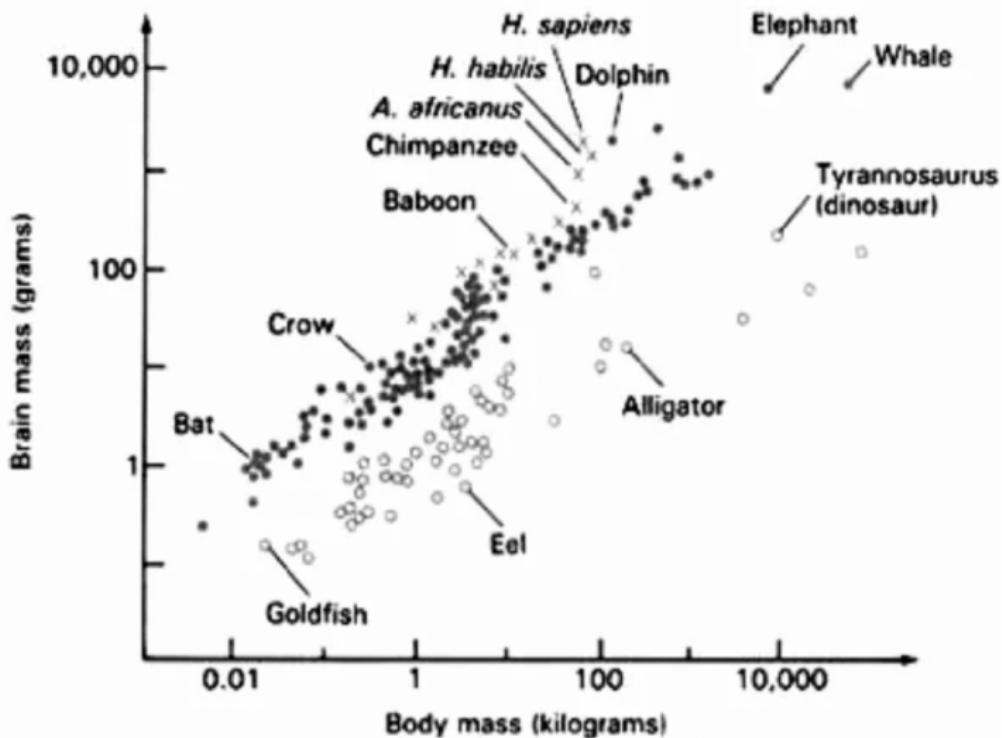
Muestra, por encima de todo, datos.



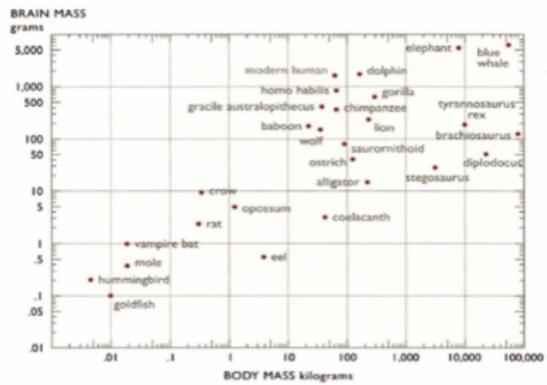
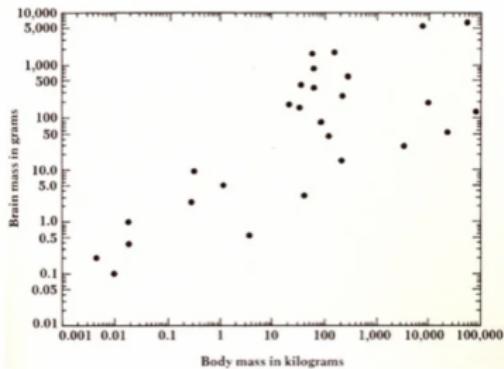
It's an electroencephalogram – a graph that records the electrical activity from the brain. This graph would have a very high data-ink ratio of 1

- Maximiza el data-ink ratio
- Borra tinta no usada para datos
- Deshazte de la tinta data-ink redundante.

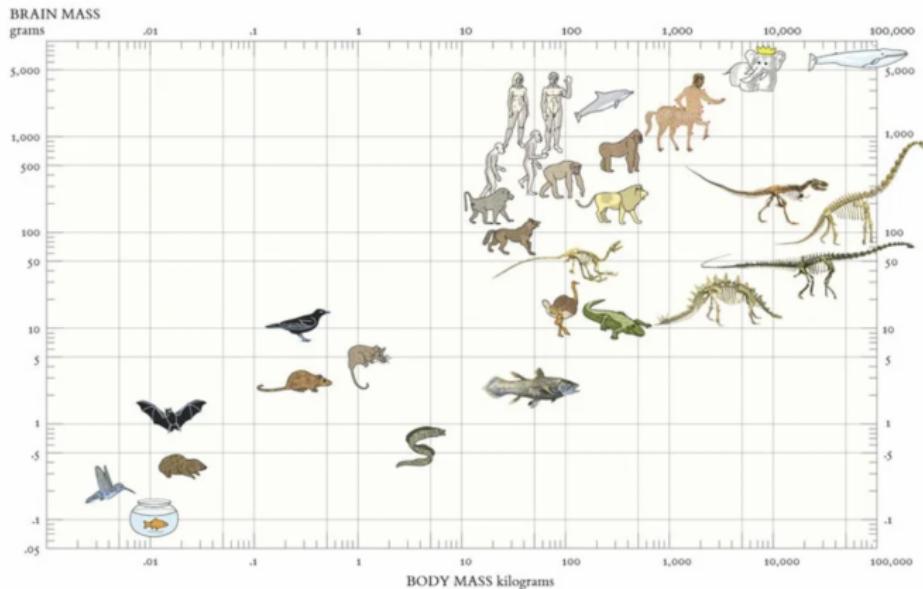
Visualización de datos I



Visualización de datos I



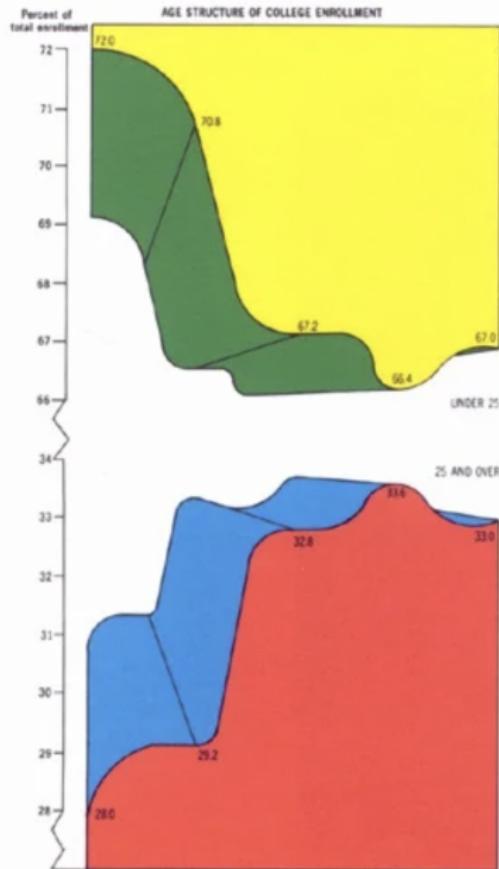
Visualización de datos I



3-. Evita lo sobrerecargado

El uso exceso de elementos innecesarios puede no ser el mejor aliado del ojo humano.

Visualización de datos I



4-. Maximiza la proporción total del gráfico en la que se está graficando datos

5-. Usa soluciones de diseño clásicas

Visualización de datos I

Usa múltiples gráficos que ayudan a la comparación

Change in Home Prices (year over year)

From New York Times Economix blog



Visualización de datos I

Los datos que tienen muchas dimensiones se pueden visualizar de la siguiente manera:



Algunas webs de apoyo:

Data Viz Project

From Data to Viz

Aesthetics y sintaxis.: conceptos clave.

Aesthetics, mapping: es el mapeado o rol que tiene cada variable visualmente. (Ej.: la variable sexo está en el eje x, la variable altura en el eje y...)

Geoms: son objetos geométricos (barras, puntos, líneas...)

Statistics: son agregaciones (medias, conteos..)

Scales: las leyendas visuales.

Facets: agrupaciones de los datos.

Ggplot2

```
library(tidyverse)
mpg %>% glimpse()
```

La base para cualquier gráfico es la función ggplot

```
ggplot(datos, aes)
```

En nuestro caso es el dataset mpg

```
ggplot(mpg, aes)
```

Por ejemplo:

```
ggplot(mpg, aes(class)) + geom_...
```

Crear un gráfico con una geometría compuesta por:

- Dataset
- Mappings
- Una geometría

Para añadir la geometría simplemente hay que añadir la que corresponda, con + geom_.

Probemos otras geometrías, funcionan?

Cada geometría acepta *aes* distintos. Por ejemplo, en un scatter plot (`geom_point`) puedes usar *shape*, pero en un gráfico de barras (`geom_bar`) no se puede.

PERO algunas geometrias comparten *aes*. `geom_bar` y `geom_points` comparten `x!`.

Ir a la documentación de GGplot2,y buscar las funciones geom_bar y geom_points. Compáralas.

Vamos a ver a partir de ahora, cómo podemos construir gráficos para los siguientes tipos de datos:

- Una variable: discreta o continua
- Dos variables continuas y/o discretas
- Dos variables continuas
- Mapas

¿Cuál es la diferencia entre una variable continua y una variable discreta?

Empezamos por uno de los gráficos más clásicos: scatter plot, nube de puntos, etc.

```
ggplot(data = mtcars, aes(x = wt, y = mpg)) +  
  geom_point()
```

Prueba a lanzar éstas geometrías:

```
geom_point()  
geom_smooth()  
geom_quantile()  
geom_rug()  
geom_jitter()  
geom_text()
```

Ggplot2

```
b <- ggplot(mtcars, aes(x = wt, y = mpg))  
b + geom_point()  
b + geom_point(aes(color = cyl, shape = cyl))  
b + geom_point(aes(color = cyl, shape = cyl)) +  
  scale_color_manual(values = c("#999999", "#E69F00", "#56B8D8"))  
b + theme_minimal()
```

Ggplot2

```
b + geom_point() +
  geom_smooth(method = lm, se = FALSE)
b + geom_point() + geom_smooth()
b + geom_point(aes(color=cyl, shape=cyl)) +
  geom_smooth(aes(color=cyl, shape=cyl),
              method=lm, se=FALSE, fullrange=TRUE)
```

Ggplot2

```
ggplot(faithful, aes(x=eruptions, y=waiting)) +  
  geom_point() + geom_rug()
```

Ggplot2

```
sp <- ggplot(faithful, aes(x=eruptions, y=waiting))
sp + geom_density_2d()
sp + geom_point() + geom_density_2d()
sp + geom_point() +
  stat_density_2d(aes(fill = ..level..), geom="polygon")
```

Para cambiar un poco el estilo de los puntos:

```
ggplot(mtcars, aes(x = wt, y = mpg)) +  
  geom_point(size = 2, shape = 23)
```

Se puede especificar el mapeo con strings también:

```
ggplot(mtcars, aes_string(x = "wt", y = "mpg")) +  
  geom_point(size = 2, shape = 23)
```

Gráficos de Densidad (parecidos a los histogramas)

```
ggplot(wdata, aes(x = weight)) + geom_density()  
ggplot(wdata, aes(x = weight)) + stat_density()
```

En qué se diferencian de un histograma? Qué muestran este tipo de gráficos?

Otra forma de crear densidades, o áreas:

```
wdata = data.frame(  
    sex = factor(rep(c("F", "M"), each=200)),  
    weight = c(rnorm(200, 55), rnorm(200, 58)))  
  
ggplot(wdata, aes(x = weight)) + geom_area(stat = "bin")
```

En este caso, qué otra cosa se podría hacer? Pensar en más mapeos ...

Ggplot2

```
a = ggplot(wdata, aes(x = weight))
a + geom_density()
a + geom_density(aes(color = sex))
a + geom_density(aes(fill = sex), alpha=0.4)
```

Ggplot2

```
a + geom_histogram()  
a + geom_histogram(aes(color = sex), fill = "white",  
                    position = "dodge")  
a + geom_histogram(aes(y = ..density..))
```

Otras formas útiles de visualizar distribuciones empíricas:

```
a + stat_ecdf()  
ggplot(mtcars, aes(sample=mpg)) + stat_qq()
```

¿Para qué sirven estos gráficos?

Ahora podemos pasar a ver variables discretas:

```
b <- ggplot(mpg, aes(f1))  
b + geom_bar()  
b + geom_bar(fill = "steelblue", color ="steelblue") +  
  theme_minimal()
```

Ahora podemos pasar a ver variables discretas:

```
g <- ggplot(data=ToothGrowth, aes(x=dose, y=len, fill=supp)
g + geom_bar(stat = "identity")
g + geom_bar(stat="identity", position=position_dodge())
```

Ahora podemos ver cómo graficar series temporales

```
d <- ggplot(economics, aes(x = date, y = unemploy))  
d + geom_area()  
d + geom_line()
```

Y lo realmente interesante: ver variables discretas y continuas en el mismo gráfico:

```
x <- ToothGrowth %>% mutate(dose=as.factor(dose))
e <- x %>%
  ggplot(aes(x = dose, y = len))
```

```
e + geom_boxplot()
e + geom_boxplot(notch = TRUE)
e + geom_boxplot(aes(color = dose))
e + geom_boxplot(aes(fill = dose))
```

Ggplot2

```
x %>% mutate(supp=as.factor(supp)) %>%
ggplot(aes(x=dose, y=len, fill=supp)) +
geom_boxplot()
```

Ggplot2

```
e + geom_violin(trim = FALSE)
e + geom_violin(trim = FALSE) +
  stat_summary(fun.data="mean_sdl",
              fun.args = list(mult=1),
              geom="pointrange", color = "red")
e + geom_violin(trim = FALSE) +
  geom_boxplot(width = 0.2)
e + geom_violin(aes(color = dose), trim = FALSE)
```

Ggplot2

```
e + geom_dotplot(binaxis = "y", stackdir = "center")
e + geom_dotplot(binaxis = "y", stackdir = "center") +
  stat_summary(fun.data="mean_sdl",
               fun.args = list(mult=1),
               geom="pointrange", color = "red")
```

Con los datos mpg, hay que graficar lo siguiente (elige la mejor forma de grafica también!):

- cuántos modelos úncos tiene cada marca.
- cuál es la distribución de la transmisión
- cuál es la distribución de la clase
- La distribución de la variable cty por clase y cyl.
- La distribución de hwy.
- la relación entre displ y hwy por cada tipo de clase

La función `facet_wrap` nos permite ver de una mejor manera los gráficos, añadiendo una categoría o mapeo más a los mismos.

Ggplot2

```
library(ggeversa)
a=ggplot(Anolis, aes(SEX_AGE, fill=SEASON))
a+geom_bar()+
  facet_wrap(~Survey_Site)+
  labs(y="Frecuencia", x="Género y edad")+
  theme(axis.title=element_text(size=10,face="bold"))
```

Ggplot2

```
c=ggplot(Anolis, aes(SVL, HEIGHT))  
c+geom_point() +  
  facet_wrap(~Survey_Site) +  
  labs(y="Altura del muestreo") +  
  theme(axis.title=element_text(size=10, face="bold"))
```

Ggplot2

```
d=ggplot(Anolis, aes(SVL, HEIGHT))
d+geom_point()+
  facet_wrap(~Survey_Site, nrow=1)+
  labs(y="Altitud")+
  theme(axis.title=element_text(size=10,face="bold"))
```

Ggplot2

```
c=ggplot(Anolis, aes(SVL, HEIGHT))  
c+geom_point() +  
  facet_wrap(~Survey_Site, ncol=1) +  
  labs(y="Altitud") +  
  theme(axis.title=element_text(size=10, face="bold"))
```

Ggplot2

```
e=ggplot(Anolis, aes(SVL, HEIGHT))  
e+geom_point() +  
  facet_wrap(~Survey_Site, ncol=1, scales="free") +  
  labs(y="Altitud") +  
  theme(axis.title=element_text(size=10, face="bold"))
```

Ggplot2

```
e=ggplot(Anolis, aes(SVL, HEIGHT))  
e+geom_point() +  
  facet_wrap(~Survey_Site+SEASON) +  
  labs(y="Altitud") +  
  theme(axis.title=element_text(size=10, face="bold"))
```

Ggplot2

```
e=ggplot(Anolis, aes(SVL, HEIGHT))  
e+geom_point() +  
  facet_wrap(~Survey_Site+SEASON, drop=FALSE) +  
  labs(y="Altitud") +  
  theme(axis.title=element_text(size=10, face="bold"))
```

Ggplot2

```
f=ggplot(Anolis, aes(SVL, HEIGHT))
f+geom_point()+
  facet_wrap(~Survey_Site, scales="free")+
  labs(y="Altitud")+
  theme(axis.title=element_text(size=10,face="bold"))
```

Ggplot2

```
ggplot(Anolis, aes(x=SVL, y=HEIGHT, colour=Survey_Site))+  
  geom_point(data = transform(Anolis,  
    Survey_Site = NULL), colour = "grey85") +  
  geom_point() +  
  facet_wrap(~Survey_Site) +  
  labs(y="Altitud") +  
  theme(legend.position="none")
```

Ggplot2

```
ggplot(Anolis, aes(x=SVL, y=HEIGHT, colour=Survey_Site))+  
  geom_point(data = transform(Anolis,  
    Survey_Site = NULL), colour = "grey85") +  
  geom_point() +  
  facet_wrap(~Survey_Site, strip.position = "bottom") +  
  labs(y="Altitud") +  
  theme(legend.position="none")
```

El arte del dato bien contado

Para este ejercicio hay que usar estos datos

Aplicando un estilo o tema propio, graficar:

- Director con más películas
- Distribución de películas por rating
- Por país, la serie temporal de películas añadidas (acumuladas).
- Número de Películas por año y país
- Duración media por rating y por país