



Analítica de datos

Manipulación y visualización de datos: Tidyverse y ggplot2



Pontificia Universidad
JAVERIANA
Bogotá

Profesor: Nicolás Velásquez

Algunas definiciones

VARIABLE

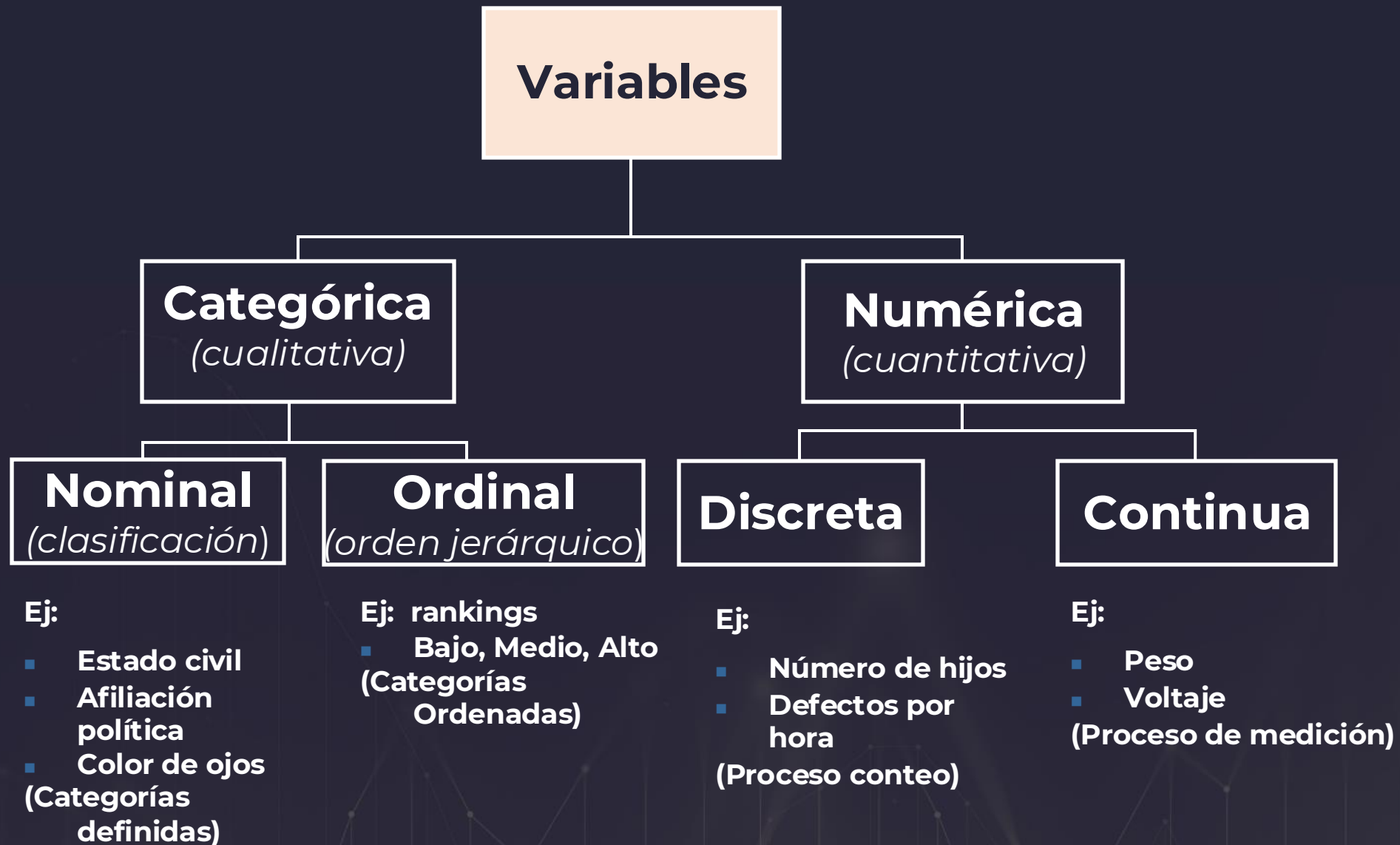
Una característica (género, ventas, PIB) que puede tomar distintos valores para distintas observaciones (personas, firmas, años).

DATOS

El conjunto de observaciones (filas) con sus respectivas realizaciones para las variables (columnas).

Ejemplo: caracterización de la clase.

Tipos de Variables



Ejemplos de tipos de variables

Pregunta	Respuesta	Tipo de Variable
¿Tiene perfil de Facebook?	Sí o No	?
¿Cuántos mensajes de texto ha enviado en los últimos dos días?	-----	?
¿Cuánto tiempo le tomó bajar la aplicación?	-----	?
¿Cómo evaluaría su experiencia en Facebook?	Muy mala, Mala, Regular, Buena, Muy Buena	?

Organización y visualización de datos categóricos

Organización de datos categóricos: Tablas



Organizando datos categóricos:

Tabla Resumen

- La tabla resumen organiza las frecuencias o porcentajes de cada una de las posibles categorías de una variable categórica.

Razones por las que los adultos jóvenes compran en línea

Razón	%
Mejores precios	37%
Evitar multitudes	29%
C conveniencia	18%
Mejores alternativas	13%
Otras	3%

Fuente : “Main Reason Young Adults Shop Online?” *USA Today*, December 5, 2012, p. 1A.

Tabla de Contingencia:

Permite organizar dos o más variables categóricas

Ejemplo:

- Muestra aleatoria de 400 facturas.
- Cada factura se categoriza en monto alto, medio, bajo.
- Cada factura se analiza para buscar errores.
- Se organizan los datos en la tabla de contingencia de la derecha.

	No Errores	Errores	Total
Monto Bajo	170	20	190
Monto Medio	100	40	140
Monto Alto	65	5	70
Total	335	65	400

Lectura: Tabla de Contingencia basada en % del total (o de cada línea)

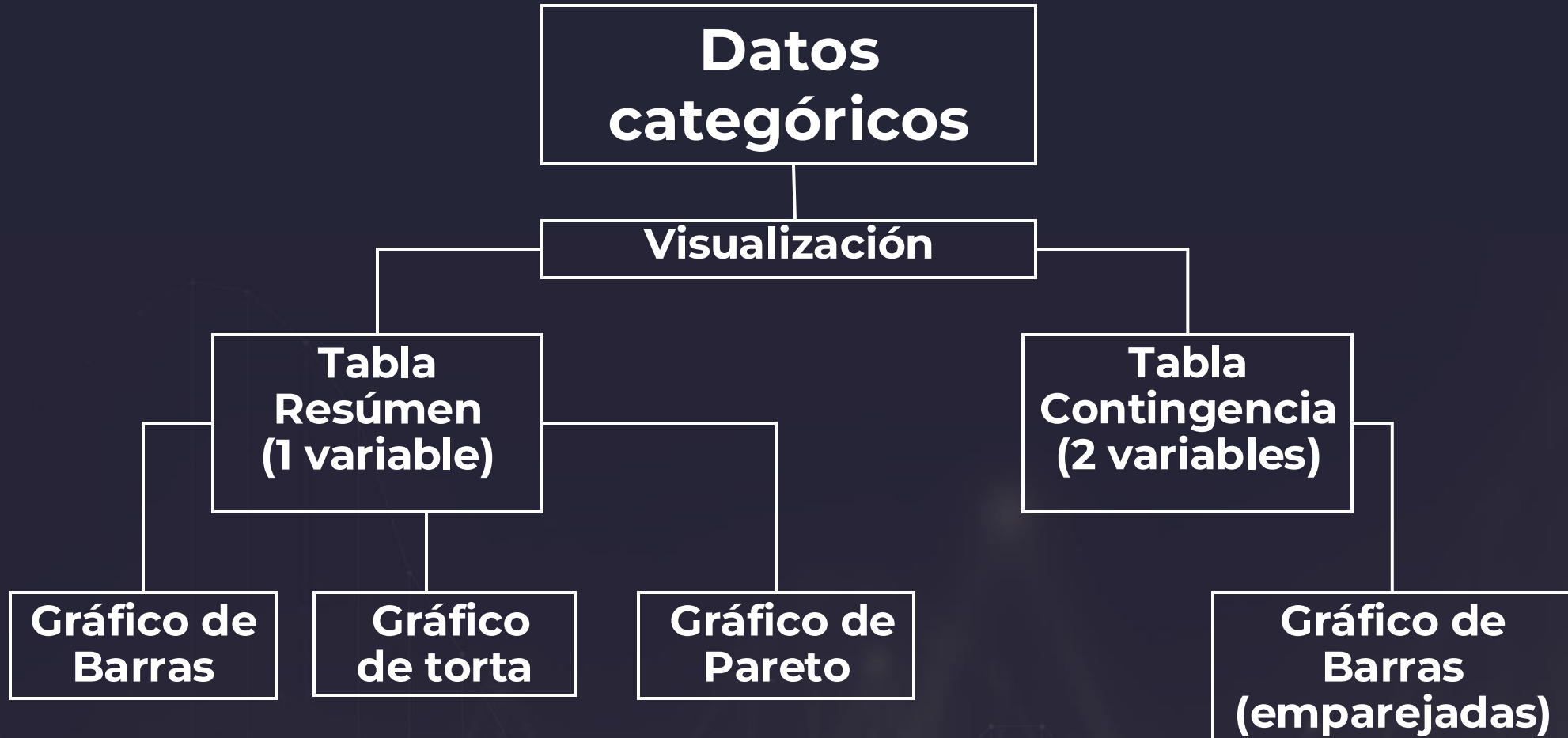
	No Errores	Errores	Total
Monto Bajo	170	20	190
Monto Medio	100	40	140
Monto Alto	65	5	70
Total	335	65	400

$$42.50\% = 170 / 400$$
$$25.00\% = 100 / 400$$
$$16.25\% = 65 / 400$$

	No Errores	Errores	Total
Monto Bajo	42.50%	5.00%	47.50%
Monto Medio	25.00%	10.00%	35.00%
Monto Alto	16.25%	1.25%	17.50%
Total	83.75%	16.25%	100.0%

83.75% de las facturas en la muestra no tienen errores y 47.50% son de monto bajo.

Visualizando datos categóricos



Visualizando datos categóricos:

Gráfico de Barras

- La gráfica de barras nos permite visualizar variables categóricas por medio de una serie de barras. **La longitud de cada barra representa la frecuencia o el % de los valores que caen en cada categoría.**
- Ejemplo:**

Reason For Shopping Online?	Percent
Better Prices	37%
Avoiding holiday crowds or hassles	29%
Convenience	18%
Better selection	13%
Ships directly	3%

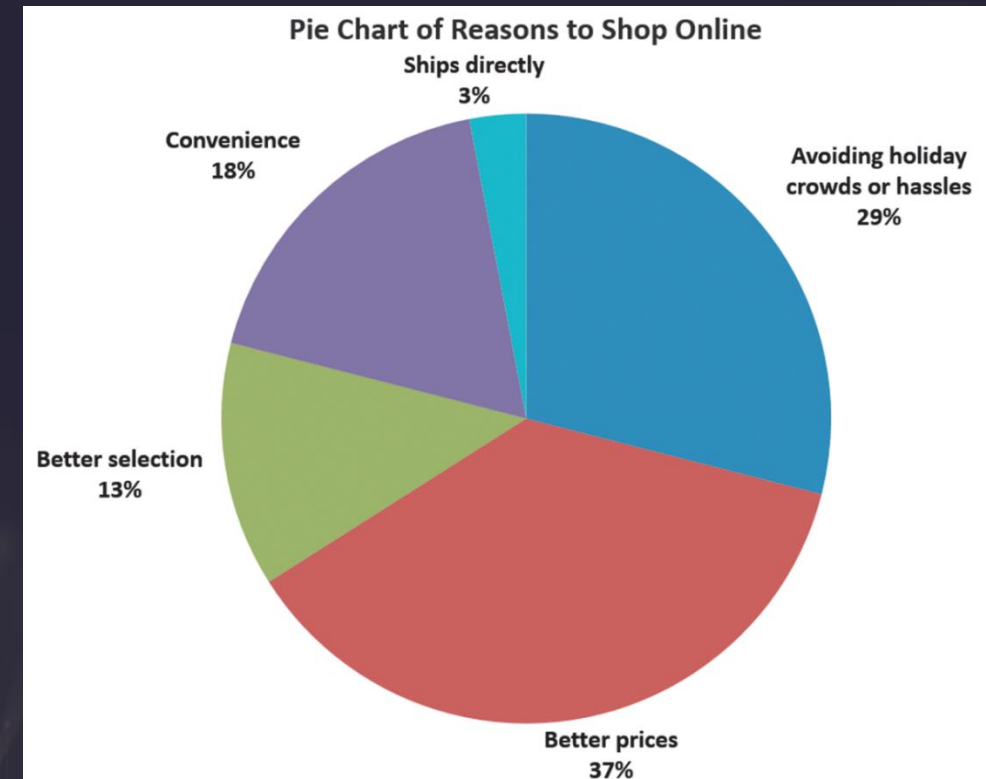


Visualizando datos categóricos:

Gráfico de torta

- Es un círculo cortado en rebanadas que representan las categorías. **El tamaño de cada rebanada varía de acuerdo al porcentaje que representa cada categoría.**
- Ejemplo:

Reason For Shopping Online?	Percent
Better Prices	37%
Avoiding holiday crowds or hassles	29%
Convenience	18%
Better selection	13%
Ships directly	3%

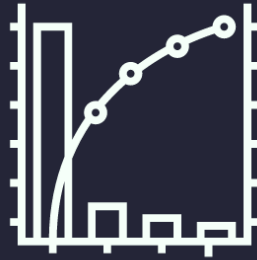


Visualizando datos categóricos:

Gráfico de Pareto



Una gráfica de **barras verticales**, donde las **categorías** se muestran **en orden de frecuencia descendente**.



También se muestra un **polígono de acumulación**.



Utilizado para separar los “pocos importantes” de los “muchos triviales”.

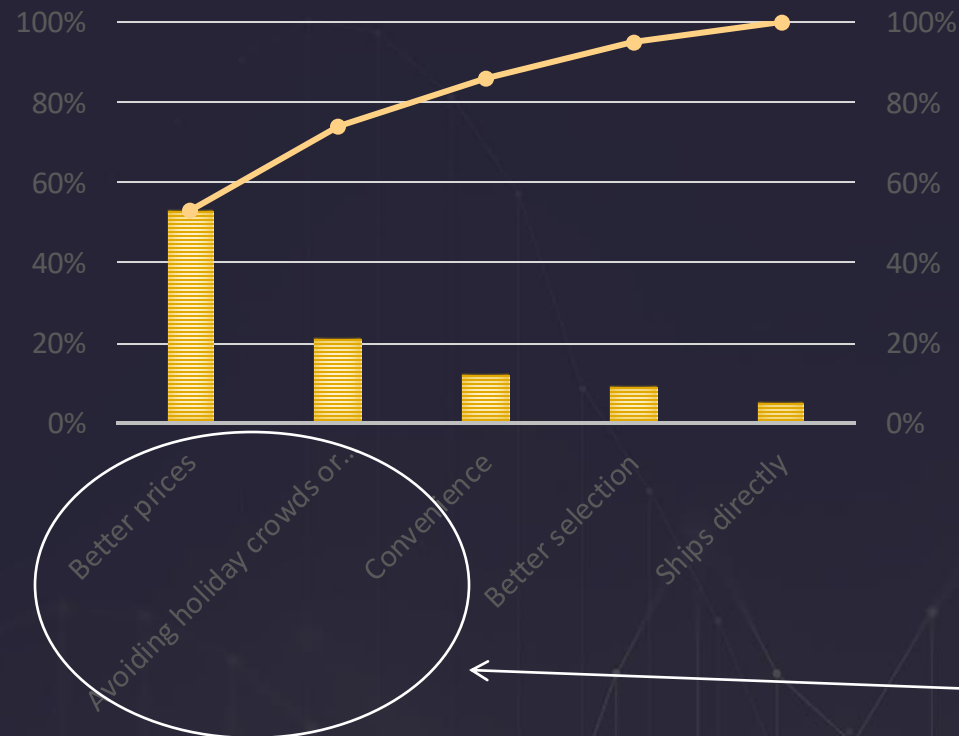


Permite fácilmente leer porcentaje acumulado hasta cierta categoría.

Visualización de 1 variable categórica:

Gráfico de Pareto

**PARETO CHART OF
REASONS TO SHOP ONLINE**



Reason For Shopping Online?	Percent	Cumulative Percent
Better prices	53%	53%
Avoiding holiday crowds or hassles	21%	74%
Convenience	12%	86%
Better selection	9%	95%
Ships directly	5%	100%

“importantes”

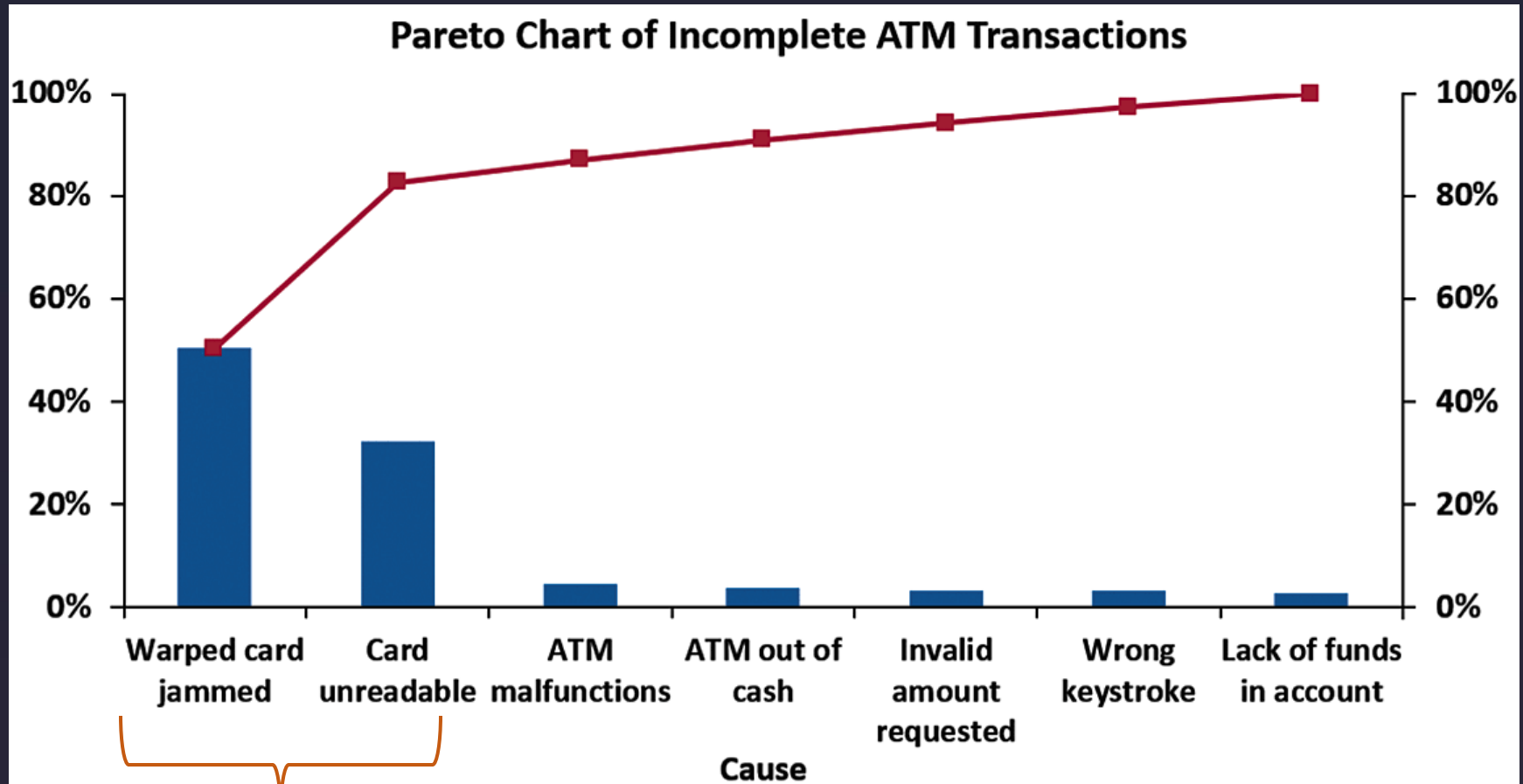
Otro ejemplo...

Ordered Summary Table For Causes Of Incomplete ATM Transactions

Cause	Frequency	Percent	Cumulative
			Percent
Warped card jammed	365	50.41%	50.41%
Card unreadable	234	32.32%	82.73%
ATM malfunctions	32	4.42%	87.15%
ATM out of cash	28	3.87%	91.02%
Invalid amount requested	23	3.18%	94.20%
Wrong keystroke	23	3.18%	97.38%
Lack of funds in account	19	2.62%	100.00%
Total	724	100.00%	

Fuente: A. Bhalla, "Don't Misuse the Pareto Principle," *Six Sigma Forum Magazine*, May 2009, pp. 15-18.

Otro ejemplo...

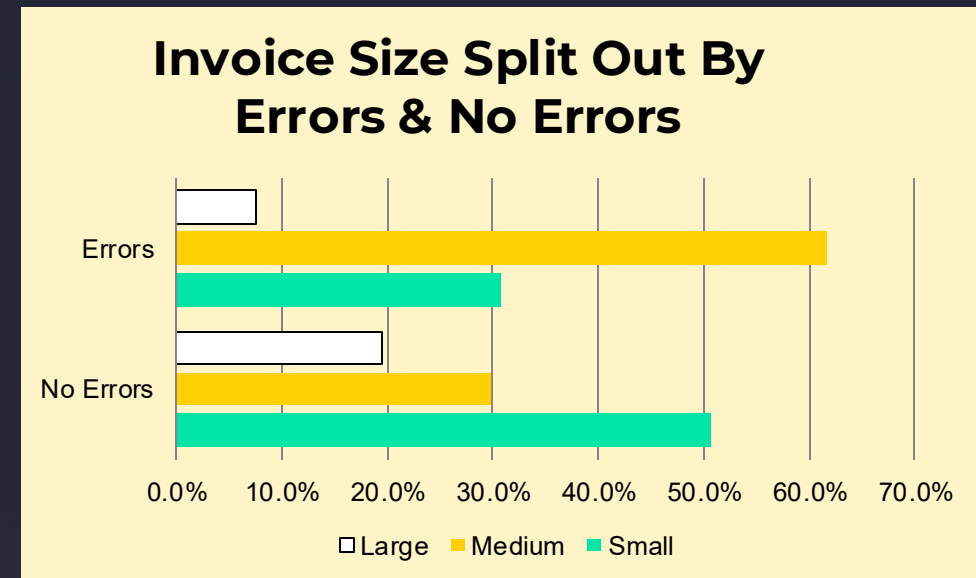


Los que son importantes

Visualizando datos categóricos: Gráfico de Barras Emparejadas

- **Representa** los datos de una **tabla de contingencia** de dos variables categóricas

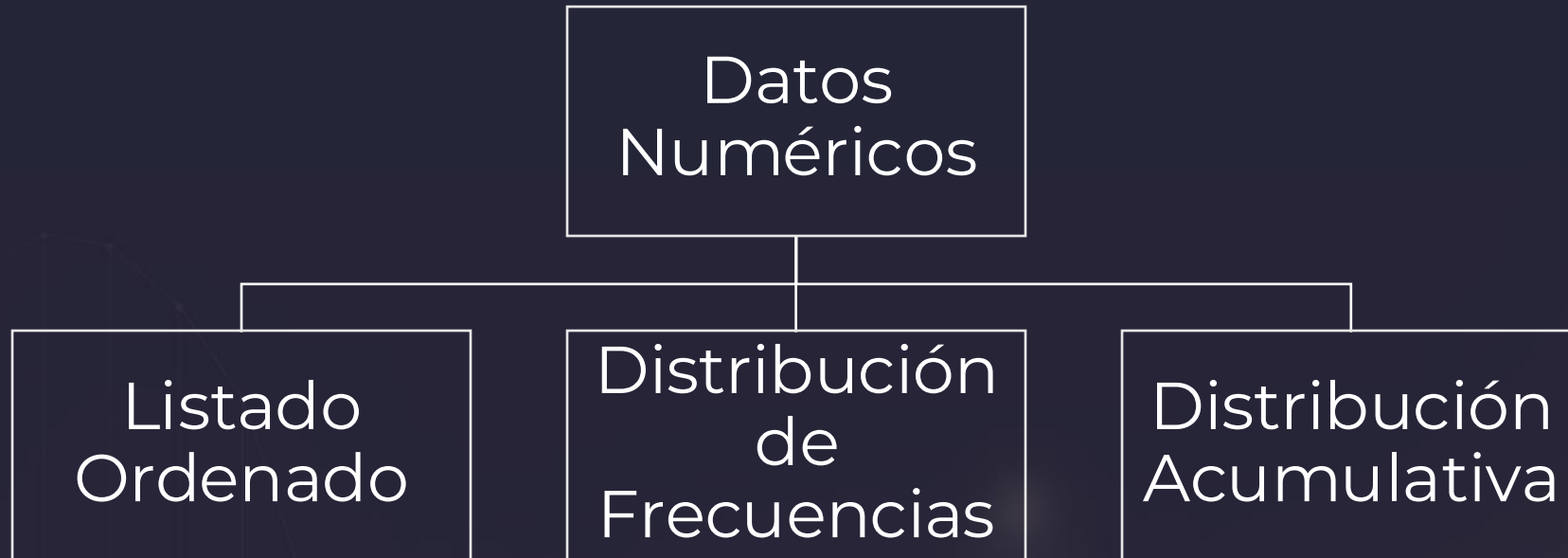
	No Errors	Errors	Total
Small Amount	50.75%	30.77%	47.50%
Medium Amount	29.85%	61.54%	35.00%
Large Amount	19.40%	7.69%	17.50%
Total	100.0%	100.0%	100.0%



Es mucho más probable que las facturas de valores medios presenten mayores errores (61.5% vs 30.8% & 7.7%).

Organización y visualización de datos numéricos

Organización de variables numéricas



Organizando datos numéricos:

Distribuciones de Frecuencia

- La **distribución de frecuencia** es una tabla resumen en la que los datos están organizados en categorías numéricamente ordenadas.
- Se tienen que **definir las categorías adecuadas**, determinando su ancho y sus fronteras.
- El número de categorías depende del rango de valores que toma la variable. Cuando el rango es grande, normalmente hay más categorías. Generalmente se usan entre 5 y 15 categorías.
- Para determinar el **ancho de cada categoría**, se divide el **rango** (valor más alto – valor más bajo) entre el número de categorías deseadas.

Organizando datos numéricos:

Distribuciones de Frecuencia

Ejemplo: una empresa de gas selecciona 20 días durante el invierno y registra la temperatura máxima:

**24, 35, 17, 21, 24, 37, 26, 46, 58, 30, 32, 13, 12, 38, 41, 43, 44,
27, 53, 27**

Organización de variables numéricas: Distribución de Frecuencias (Ejemplo)

12, 13, 17, 21, 24, 24, 26, 27, 27, 30, 32, 35, 37, 38, 41, 43, 44, 46, 53, 58 (en excel)

Clase	Frecuencia	Frec. Relativa	%
10 y menos de 20	3	.15	15%
20 y menos de 30	6	.30	30%
30 y menos de 40	5	.25	25%
40 y menos de 50	4	0.20	20%
50 y menos de 60	2	.10	10%
Total	20	1.00	100%

Frecuencia Relativa= Frecuencia / Total,
ej. $0.10 = 2 / 20$

Podemos añadir columna de frecuencia acumulada

Categoría	Frec,	%	Frec. Acumulada	% Acumulado
10 y menos de 20	3	15%	3	15%
20 y menos de 30	6	30%	9	45%
30 y menos de 40	5	25%	14	70%
40 y menos de 50	4	20%	18	90%
50 y menos de 60	2	10%	20	100%
Total	20	100%	20	100%

% acumulado = Frec. Acumulada / Total * 100

p.e. 45% = 100*9/20

¿Por qué utilizar una distribución de frecuencia?

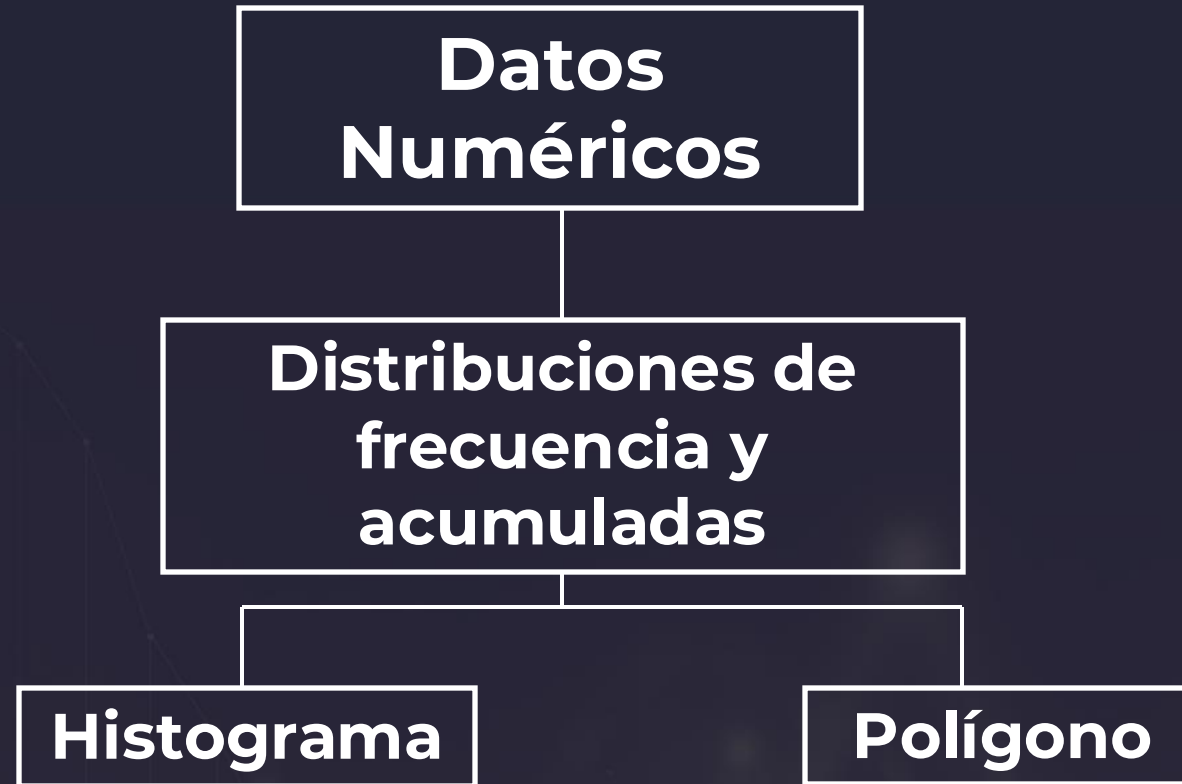
- Condensa datos brutos en una forma más útil.
- Permite una rápida interpretación visual de los datos.
- Permite determinación de ciertas características de los datos, incluyendo dónde están concentrados.

Distribuciones de frecuencia:

Tips

- **Diferentes anchos** de las categorías pueden contar distintas historias sobre dónde están concentrados los datos.
- Cuando se están comparando dos o más grupos con distintos tamaños muestrales, se tiene que utilizar frec. relativa o %.

Visualización de datos numéricos



Visualizando datos numéricos:

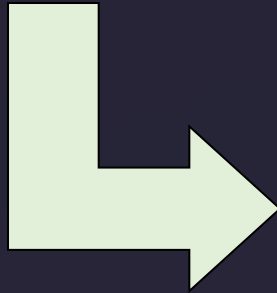
Histograma

- Una gráfica con barras verticales que representa la distribución de frecuencias de los datos.
- No tiene espacios entre las barras contiguas.
- Las **fronteras** de cada categoría o los **puntos medios** se muestran en el eje horizontal.
- El eje vertical son frecuencias, frecuencias relativas o porcentajes.
- El eje vertical son **las categorías**.

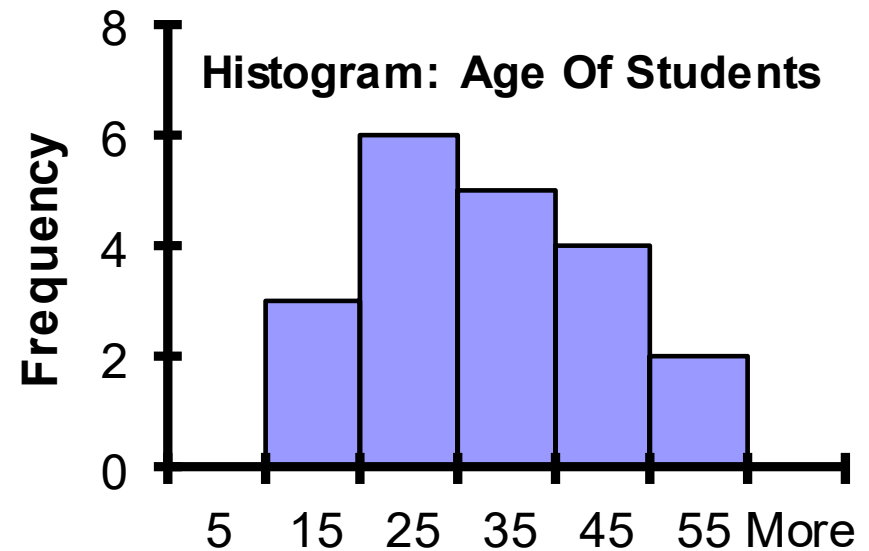
Visualizando datos numéricos:

Histograma

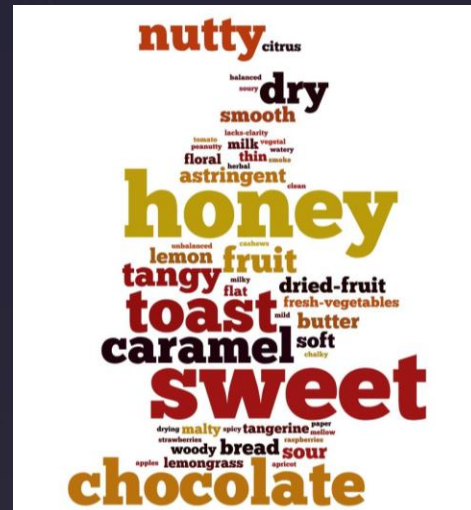
Categoría	Frecuencia	Frec. Relativa	Porcentaje
10 y menos de 20	3	.15	15
20 y menos de 30	6	.30	30
30 y menos de 40	5	.25	25
40 y menos de 50	4	.20	20
50 y menos de 60	2	.10	10
Total	20	1.00	100



Histograma: Temperatura



Visualizando datos : Histograma



- Otros tipos de histogramas para variables categóricas (Word clouds)

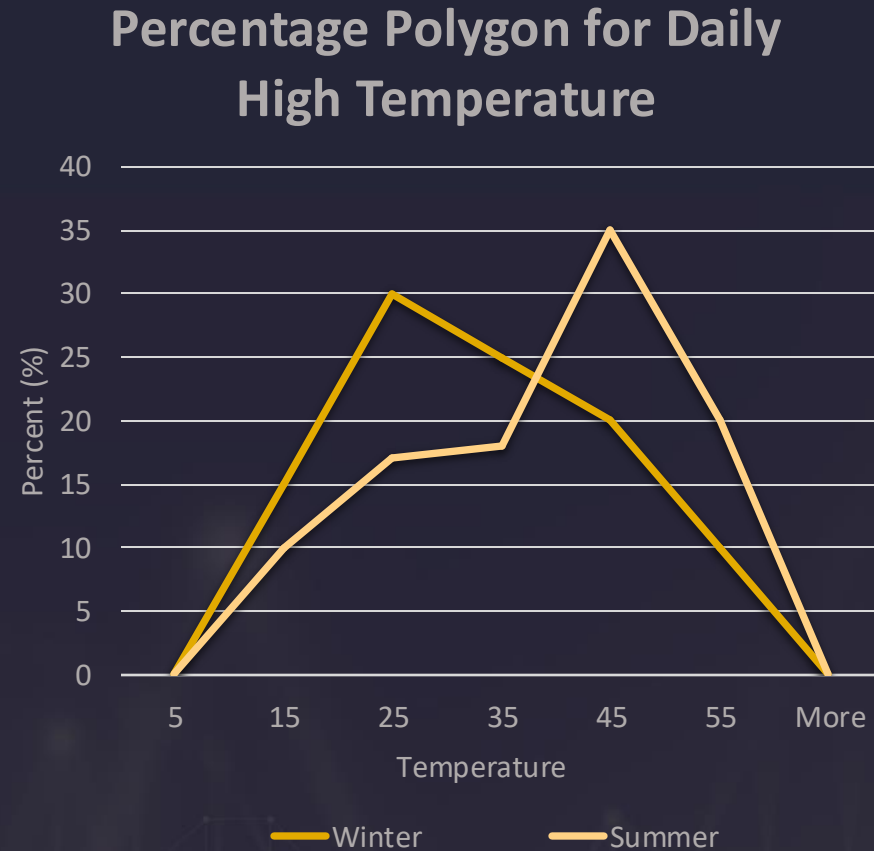


Source:
<http://danielhumphries.typepad.com/coffee/2011/07/word-clouds-for-different-varietals.html>

Visualización de 1 variable numérica:

Polígono de Frecuencias

- Para hacerlo, se utiliza el punto medio de cada clase (barra de histograma), y después se conecta la secuencia de puntos medios con sus respectivos porcentajes.
- Es muy útil cuando se quieren comparar dos o más grupos.



Ejemplo: algoritmo histograma

- Ordene los datos (orden ascendente):
12, 13, 17, 21, 24, 24, 26, 27, 27, 30, 32, 35, 37, 38, 41, 43, 44, 46, 53, **58**.
- Encuentre el rango: **$58 - 12 = 46$**
- Defina el número de categorías: **5**.
- Encuentre el ancho de cada categoría (rango ÷ número de clases): **$46/5$ y aproxime al entero mayor = 10**
- Calcule las fronteras de cada categoría:
 - **Cat 1: 10 y menos de 20**
 - **Cat 2: 20 y menos de 30**
 - **Cat 3: 30 y menos de 40**
 - **Cat 4: 40 y menos de 50**
 - **Cat 5: 50 y menos de 60**
- Cuente el número de observaciones y asigne a cada categoría.

Visualizando dos variables numéricas utilizando gráficas.

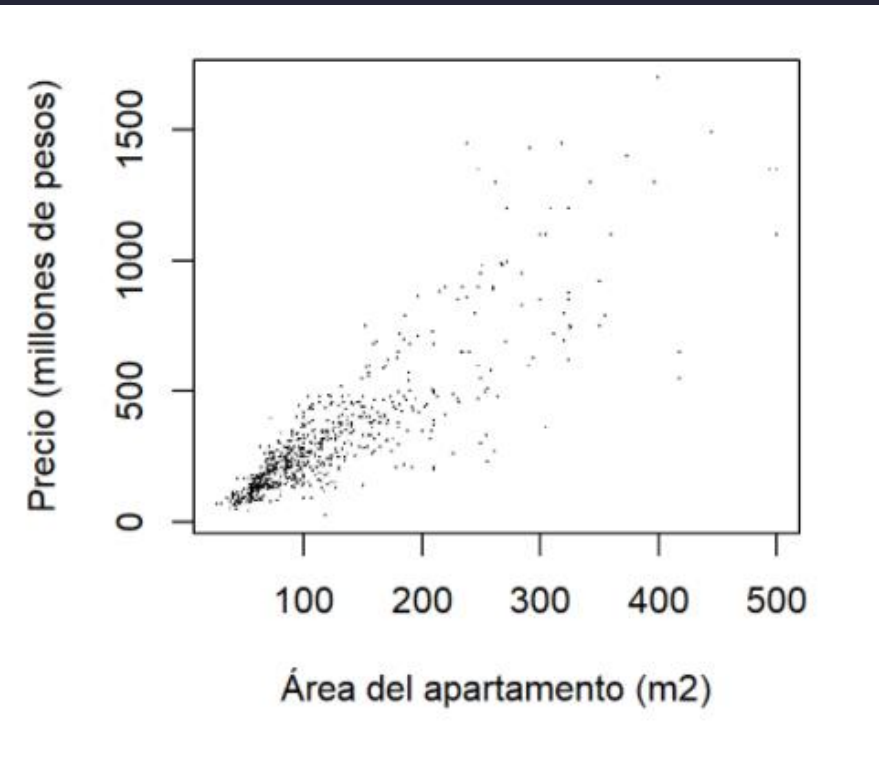


Visualizando dos variables numéricas:

Gráfico de Dispersión

- Se utiliza cuando los datos numéricos contienen observaciones de dos variables numéricas.
- Una variable se mide en el eje horizontal y la otra en el vertical.
- Utilizados para determinar si existe alguna relación entre ambas variables.

Precio de Apartamentos



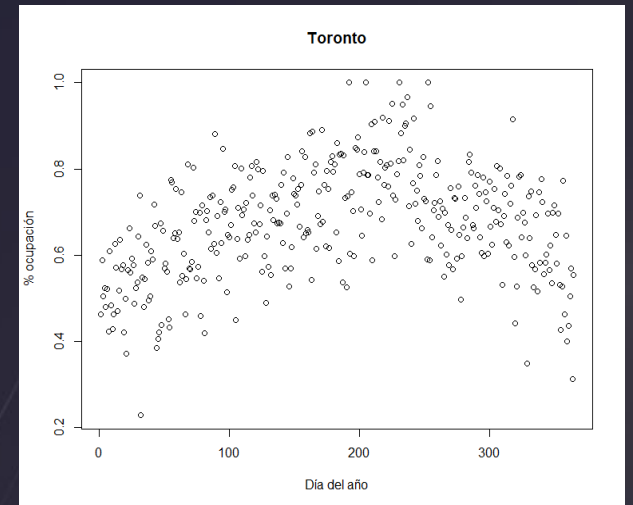
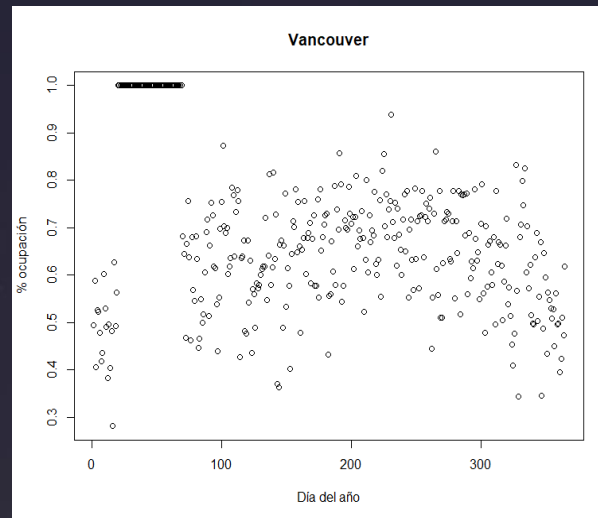
Visualizando dos variables numéricas: Gráfico de Series de Tiempo

- Se utiliza para analizar si hay patrones en el comportamiento de una variable a lo largo del tiempo
- La variable de interés se mide en el eje vertical y la variable “tiempo” en el horizontal.

Precio del petróleo



Hoteles Canadá



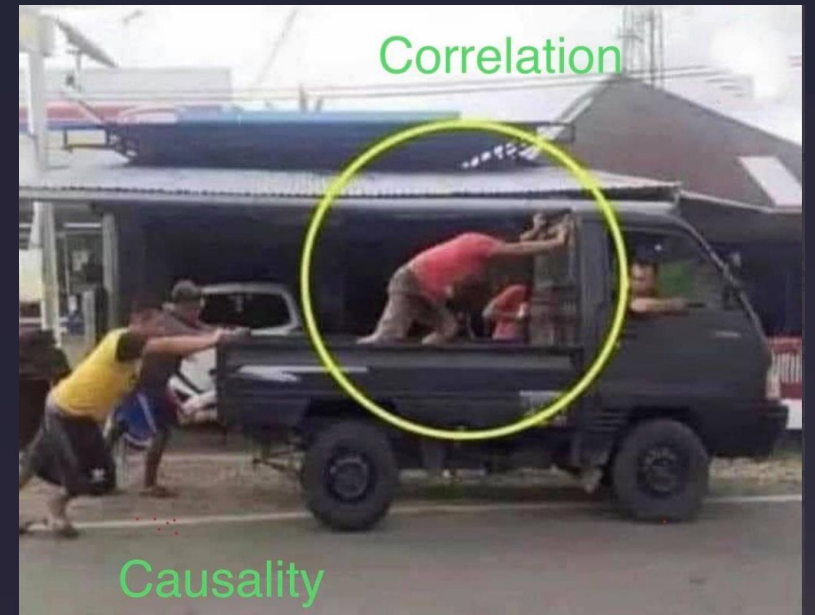
Algunas ideas de correlación

El diagrama de dispersión nos muestra si hay alguna correlación entre dos variables.

La correlación es útil para predecir, no es necesario saber por qué la correlación ocurre.

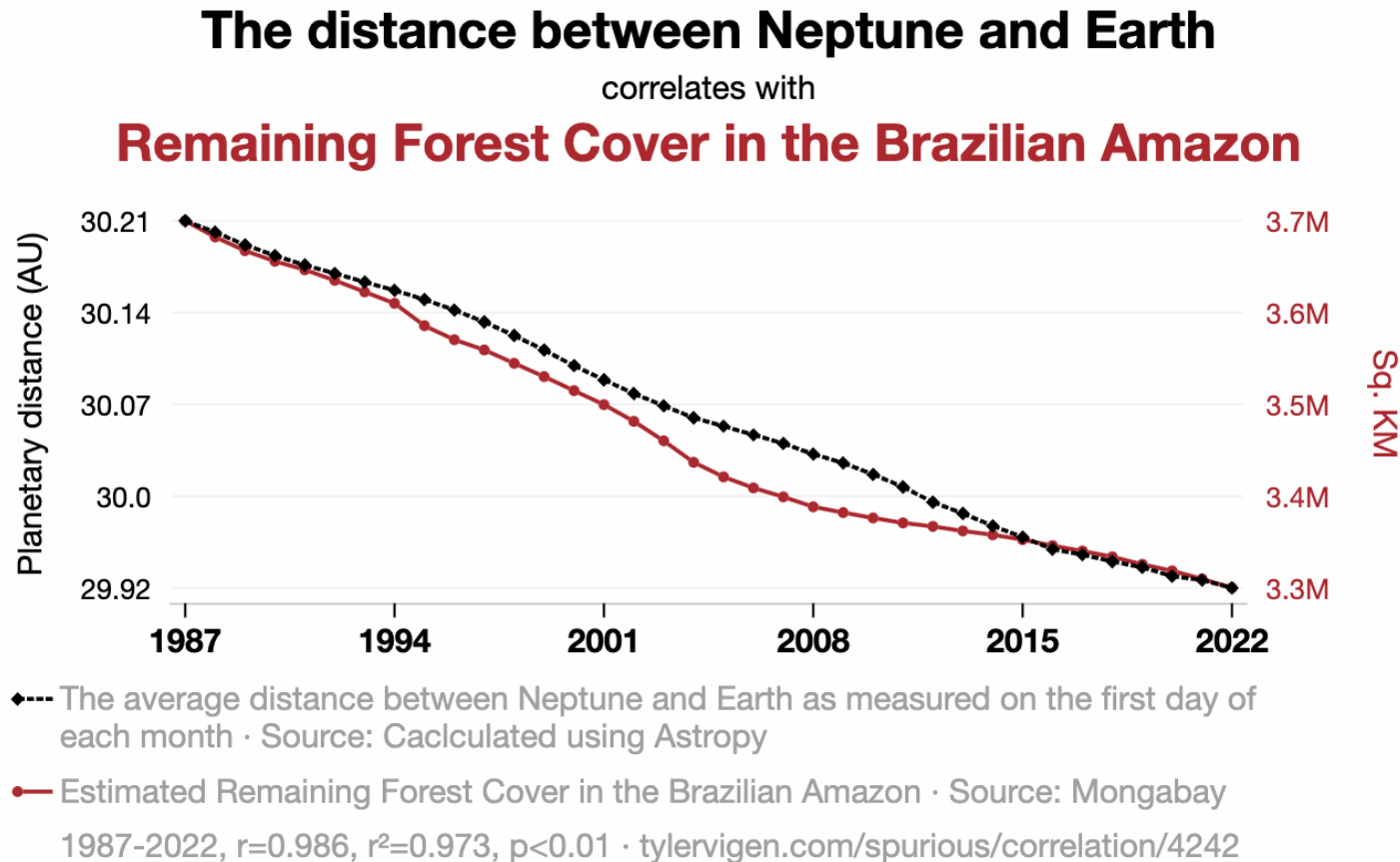
Sin embargo, que dos variables estén correlacionadas no quiere decir que un aumento/disminución en una, cause un aumento/disminución en la otra.

- Si un cambio en una variable causa un cambio en otra, están correlacionadas.
- Si dos variables están correlacionadas, no necesariamente un cambio en una variable cause un cambio en la otra.



<http://www.tylervigen.com/spurious-correlations>

Algunas ideas de correlación



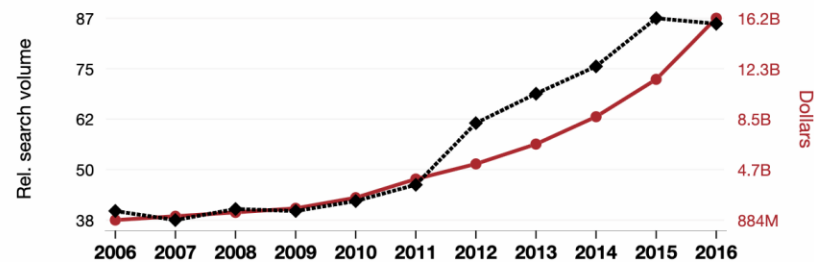
<http://www.tylervigen.com/spurious-correlations>

Algunas ideas de correlación

Google searches for 'cute cats'

correlates with

Amazon's Annual Outbound Shipping Expenditure in Millions



◆ Relative volume of Google searches for 'cute cats' (Worldwide, without quotes) · Source: Google Trends

● Amazon's Annual Outbound Shipping Expenditure in Millions · Source: Statista

2006-2016, $r=0.950$, $r^2=0.902$, $p<0.01$ · tylervigen.com/spurious/correlation/5231

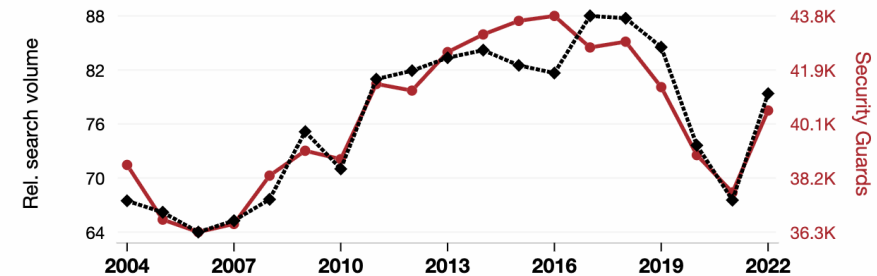
[View details about correlation #5,231](#)

Fur-real Friends: Exploring the Purr-suasive Connection Between 'Cute Cats' Google Searches and Amazon's Annual Outbound Shipping Expenditure

Google searches for 'best schools'

correlates with

The number of security guards in Pennsylvania



◆ Relative volume of Google searches for 'best schools' (United States, without quotes) · Source: Google Trends

● BLS estimate of security guards in Pennsylvania · Source: Bureau of Labor Statistics

2004-2022, $r=0.939$, $r^2=0.881$, $p<0.01$ · tylervigen.com/spurious/correlation/4246

[View details about correlation #4,246](#)

Googling for Schooling: Linking Best Schools Searches to Security Legion in Pennsylvania

[Show GenAI's made-up explanation](#)

<http://www.tylervigen.com/spurious-correlations>

Falsas impresiones

- Seleccionar los datos:
 - Solamente presentar parte de los datos recolectados.
- Gráficas construidas inapropiadamente:

<https://www.youtube.com/watch?v=E91bGT9BjYk>

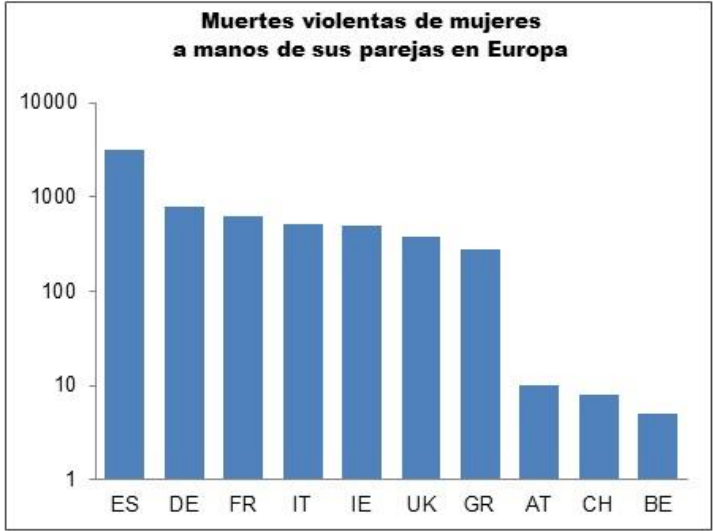
(¡cuidado con los ejes y las escalas!)

A tomar en cuenta...

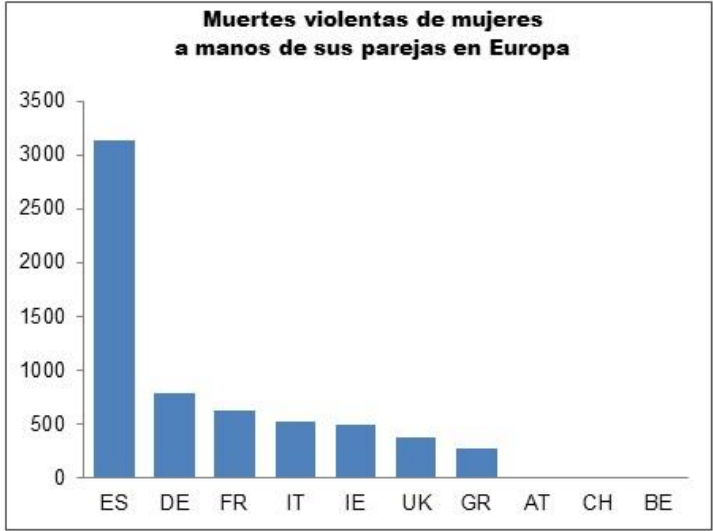
- Use la **visualización más simple de interpretar.**
- Incluya **título y nombre los ejes.**
- Incluya una **escala** para cada eje.
- Use una **escala constante.**
- Evitar 3D o efectos especiales. **No incluir “basura”.**

Ejemplos de gráficos engañosos

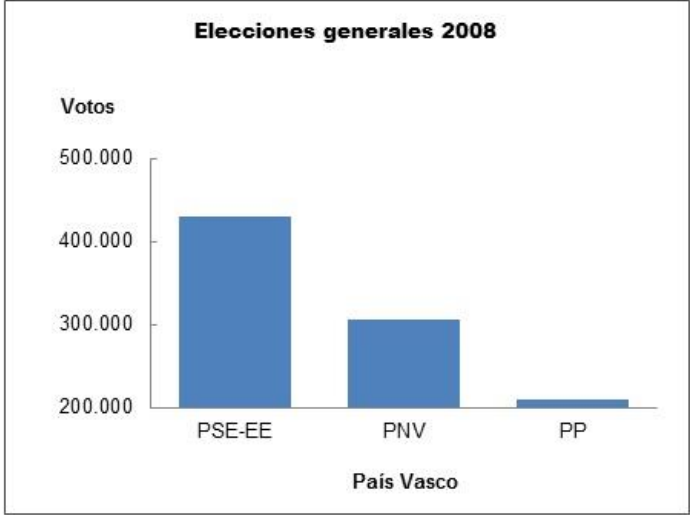
► 1



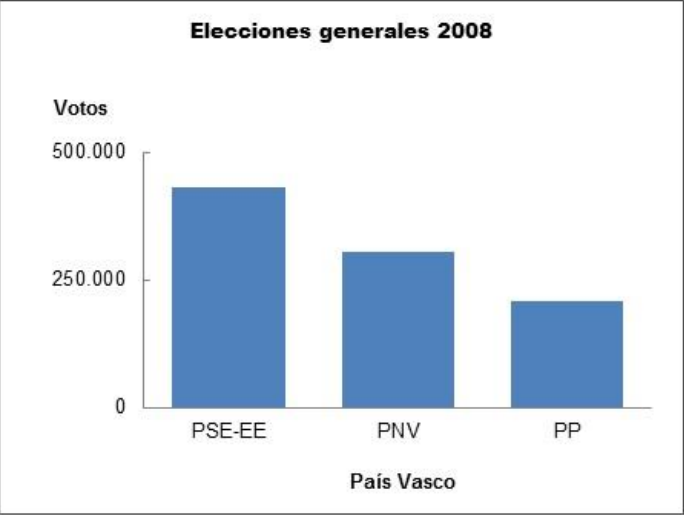
► 2



► 1



► 2



US Today

Ejemplos de gráficos engañosos



La Falacia Del Día
@FalaciaDelDia · Seguir



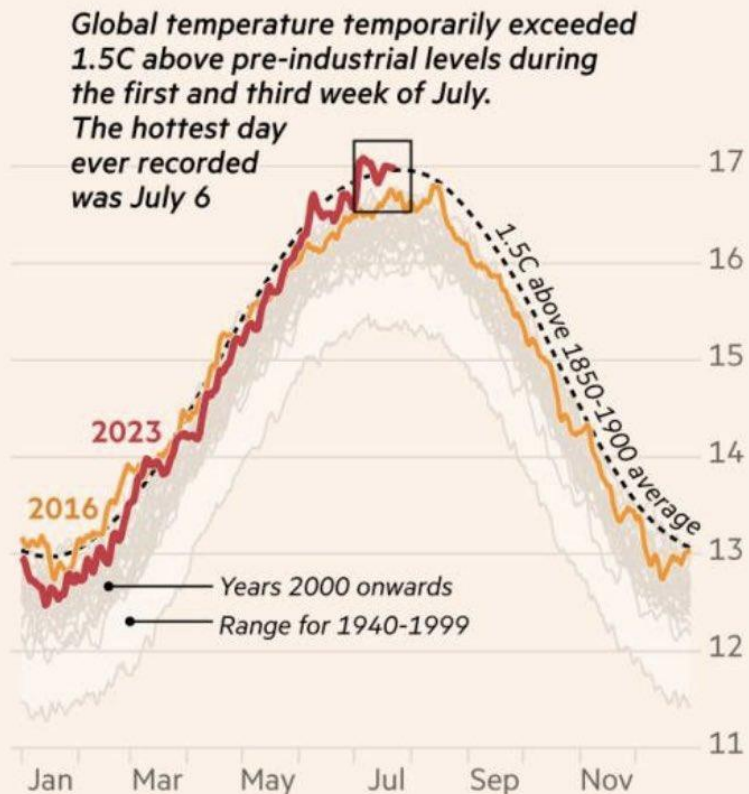
Fiel a su estilo, la revista quincena ha renunciado a tener un poquito de rigor y muestra una infografía no muy fiel con la realidad. Sin embargo, los datos que muestra no son mentira. Entonces ¿Dónde está la trampa? En que incurre en SESGO DE ENCUADRE. Abro hilo para explicarlo.



Ejemplos de gráficos buenos

July is set to be the hottest month on record

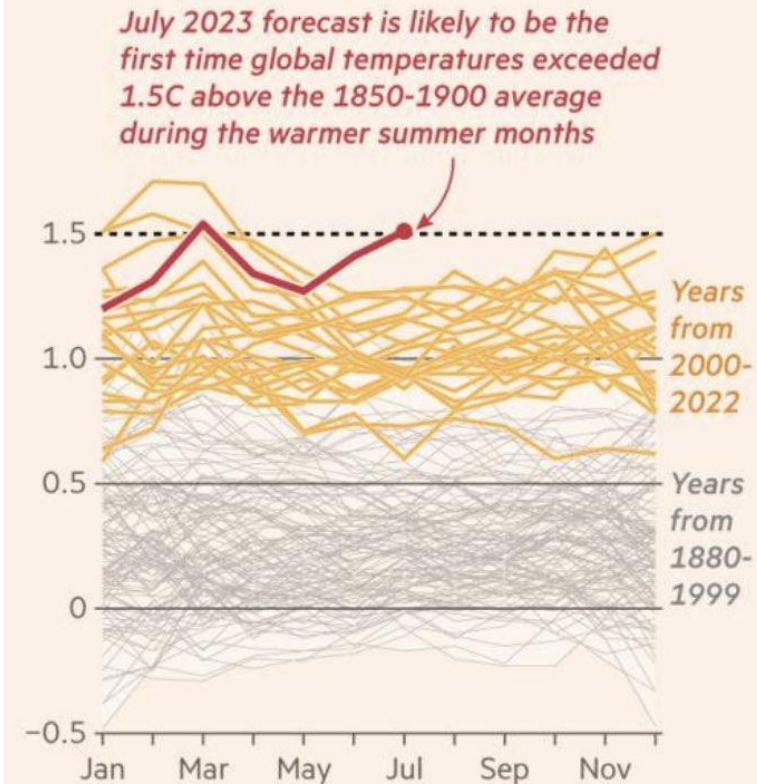
Average daily global 2-metre surface temperature, 1940 to present (C)



Data to July 23 2023 Source: Copernicus/ECMWF
© FT

July is set to be the hottest month on record

Global temperature anomaly compared with 1850-1900 average (C)



July 2023 forecast by Karsten Haustein, Leipzig University
Source: Nasa GISTEMP; Berkeley Earth for 1850-1900 baseline
© FT

Financial
Times

Tips para la Visualización de Datos (principios generales)

Toda la atención de la audiencia se debe enfocar en la idea que estamos comunicando con la visualización

+

La visualización debe ser fácil de interpretar

+

(No usar tablas)

Tips para la Visualización de Datos

- Estos principios son especialmente importantes durante presentaciones: la atención de la audiencia está en nuestra voz (la idea que queremos expresar) y en la visualización (que fundamenta nuestra idea).
- En texto, hay tiempo de regresar a la idea y leerla cuantas veces queramos.
- Más sobre esto en próximas clases.

Tidyverse y GGplot

Recursos

- Introducción a la ciencia de datos (Rafael A. Irizarry)
<http://rafalab.dfci.harvard.edu/dslibro/r-basics.html>

El tidyverse



- Hasta ahora hemos estado manipulando vectores reordenándolos y creando subconjuntos mediante la indexación.
- Sin embargo, una vez comencemos los análisis más avanzados, no usamos normalmente el vector sino el data frame.
- Nos enfocaremos en un formato de datos específico denominado tidy y en una colección específica de paquetes que son particularmente útiles para trabajar con data tidy y que se denomina el tidyverse.
- Podemos cargar todos los paquetes del tidyverse a la vez al instalar y cargar el paquete tidyverse:

```
library(tidyverse)
```


El tidyverse: tidy data

country	year	cases	population
Afghanistan	1999	745	19987071
Afghanistan	2000	2666	20595360
Brazil	1999	37737	172006362
Brazil	2000	80488	174504898
China	1999	212258	127291272
China	2000	216766	128042583

variables

country	year	cases	population
Afghanistan	1999	745	19987071
Afghanistan	2000	2666	20595360
Brazil	1999	37737	172006362
Brazil	2000	80488	174504898
China	1999	212258	127291272
China	2000	216766	128042583

observations

country	year	cases	population
Afghanistan	1999	745	19987071
Afghanistan	2000	2666	20595360
Brazil	1999	37737	172006362
Brazil	2000	80488	174504898
China	1999	212258	127291272
China	2000	216766	128042583

values

Figure 12.1: Seguir tres reglas hace que un conjunto de datos sea *tidy*: las variables están en columnas, las observaciones en filas y los valores en celdas.

fuentes: [Grolemund, Wickham 2017: R for Data Science](#)

tidy data: ¿porqué?

- un conjunto de datos en forma *tidy* facilita la selección de variables, la agrupación, el resumen y la visualización.
- Algunas herramientas y paquetes en el tidyverse como dplyr o ggplot2 requieren que los datos se organicen de esa manera
- **El problema:** la mayoría de los datos que obtenga no estarán "ordenados" desde el principio → tendrá que remodelarlos.

tidy data

- ¿Son estos datos tidy?

```
#>      country    1960  1961  1962
#> 1   Germany    2.41  2.44  2.47
#> 2 South Korea  6.16  5.99  5.79
```

¿Y ahora?

```
#>      country    year fertility
#> 1   Germany    1960     2.41
#> 2 South Korea  1960     6.16
#> 3   Germany    1961     2.44
#> 4 South Korea  1961     5.99
#> 5   Germany    1962     2.47
#> 6 South Korea  1962     5.79
```

Ejercicio

1. Examine el set de datos `titanic.csv`. ¿Cuál de los siguientes es cierto?

```
# Asegúrate de que el archivo 'titanic.csv' esté en tu directorio de trabajo
titanic <- read_csv("titanic.csv")
# 4. Explorar la estructura de los datos
glimpse(titanic)
head(titanic)
```

- a) `titanic` son datos tidy: tiene un pasajero para cada fila.
- b) `co2` no es tidy: necesitamos al menos una columna con un vector de caracteres.
- c) `co2` no es tidy: es una matriz en lugar de un data frame.
- d) `co2` no es tidy: para ser tidy tendríamos que cambiarle la forma (wrangle it en inglés) para tener múltiples columnas, y entonces cada observación de `titanic` tendría una fila

Cómo manipular los data con dplyr

Carguemos el paquete primero:

```
install.packages("dplyr")  
library(dplyr)
```

- El paquete dplyr del tidyverse ofrece funciones que realizan algunas de las operaciones más comunes cuando se trabaja con data frames y usa nombres para estas funciones que son relativamente fáciles de recordar.
- Por ejemplo, para cambiar la tabla de datos agregando una nueva columna, utilizamos *mutate*. Para filtrar la tabla de datos a un subconjunto de filas, utilizamos *filter*. Finalmente, para subdividir los datos seleccionando columnas específicas, usamos *select*.

Cómo añadir una columna con mutate

Queremos que toda la información necesaria para nuestro análisis se incluya en la tabla de datos. Entonces, la primera tarea es añadir si el pasajero es mayor de edad en nuestro data frame

```
# 6. Crear una nueva variable con mutate (edad_grupo)
# Clasifica en "Menor" si Age < 18, de lo contrario "Adulto"
titanic <- mutate(titanic, edad_grupo = if_else(Age < 18, "Menor", "Adulto"))
```

- Recuerden que aquí usamos Age dentro de la función, que son objetos no definidos en nuestro espacio de trabajo. Pero, ¿por qué no recibimos un error?
- Esta es una de las principales características de dplyr. Las funciones en este paquete, como mutate, saben buscar variables en el data frame que el primer argumento les provee.

Cómo crear subconjuntos con filter

Ahora supongan que queremos filtrar la tabla de datos para mostrar solo los pasajeros que sobrevivieron al iceberg. Para hacer esto, usamos la función `filter`, que toma la tabla de datos como primer argumento y luego la declaración condicional como el segundo.

```
# 7. Filtrar pasajeros que sobrevivieron  
sobrevivientes <- filter(titanic, Survived == 1)
```

Cómo seleccionar columnas con select

Aunque nuestra tabla de datos solo tiene seis columnas, algunas tablas de datos incluyen cientos. Si queremos ver solo algunas columnas, podemos usar la función `select` de `dplyr`.

```
# 8. Seleccionar columnas específicas: Name, Sex y Age
datos_basicos <- select(titanic, Name, Sex, Age)
```

Más funciones de dplyr

```
# 10. Crear subconjunto con menores de edad y guardar en nuevo data frame
pasajeros_menores <- select(
  filter(titanic, Age < 18),
  Name, Age, Sex
)
```

```
# 11. Mostrar solo ciertas columnas con select
head(select(titanic, Name, Age, Sex))
```

```
# 12. Filtrar una fila específica (por nombre completo)
filter(titanic, Name == "Allen, Mr. William Henry")
```

```
# 13. Eliminar filas según una condición (!=)
# Quitar pasajeros embarcados en el puerto 'S'
no_S <- filter(titanic, Embarked != "S")
```

```
# 14. Filtrar filas con múltiples condiciones usando %in%
# Pasajeros que embarcaron en C o Q
filter(titanic, Embarked %in% c("C", "Q"))
```


El pipe: `|>` o `%>%`

En R podemos realizar una serie de operaciones, por ejemplo `select` y entonces `filter`, enviando los resultados de una función a otra usando lo que se llama el pipe operator: `|>`. Esta función se hizo disponible a partir de la version 4.1.0 de R, pero antes de esto el tidyverse usaba el operador `%>%` del paquete `magrittr`.

El pipe: `|>` o `%>%`

- Escribimos el código anterior para mostrar tres variables (Name, Age, Pclass) para los pasajeros que son mujeres y sobrevivieron al iceberg. Para hacer esto, definimos el objeto intermedio `mujeres_sobrevivientes`.

```
# 9. Combinar filter y select en un solo pipeline
mujeres_sobrevivientes <- titanic |>
  filter(Sex == "female", Survived == 1) |>
  select(Name, Age, Pclass)
```

- En general, el pipe envía el resultado que se encuentra en el lado izquierdo del pipe para ser el primer argumento de la función en el lado derecho del pipe. Aquí vemos un ejemplo sencillo:

```
16 |> sqrt()
```

Cómo resumir datos **summarize**

- La función **summarize** de **dplyr** ofrece una forma de calcular estadísticas de resumen con código intuitivo y legible.

```
# 11. Calcular estadísticas resumen con summarize
# Promedio y desviación estándar de edad por sexo
resumen_edad <- titanic |>
  filter(!is.na(Age)) |>
  group_by(Sex) |>
  summarize(promedio = mean(Age), desviacion = sd(Age))
```

Cómo resumir datos **summarize**

- Como la tabla resultante almacenada en `s` es un data frame, podemos acceder a los componentes con el operador de acceso `$`:

```
# 12. Acceder a una columna con $  
# Por ejemplo, acceder a la edad promedio de mujeres  
resumen_edad$Sex # Esto te dará NA porque `female` no es  
# Lo correcto es:  
resumen_edad |> filter(Sex == "female") |> pull(promedio)
```

Cómo resumir datos **summarize**

- Sin embargo, observe que los resúmenes se devuelven en una fila cada uno. Para obtener los resultados en diferentes columnas, tenemos que definir una función que devuelva un marco de datos como este:

```
# 1. Crear función personalizada que retorna mediana, mínimo y máximo
resumen_edad <- function(x) {
  qs <- quantile(x, c(0.5, 0, 1), na.rm = TRUE)
  data.frame(mediana = qs[1], minimo = qs[2], maximo = qs[3])
}

# 2. Aplicar la función a las mujeres del Titanic
titanic |>
  filter(Sex == "female") |>
  summarize(resumen_edad(Age))
```

Cómo agrupar y luego resumir con `group_by`

- Una operación común en la exploración de datos es dividir primero los datos en grupos y luego calcular resúmenes para cada grupo.
- Por ejemplo, podemos querer calcular el promedio y la desviación estándar para las alturas de hombres y mujeres por separado.

```
heights |> group_by(Sex)
```

- El resultado no se ve muy diferente de `heights`, excepto que vemos `Groups: sex [2]` cuando imprimimos el objeto.
- Aunque no es inmediatamente obvio por su apariencia, esto ahora es un data frame especial llamado un grouped data frame y las funciones de `dplyr`, en particular `summarize`, se comportarán de manera diferente cuando actúan sobre este objeto.

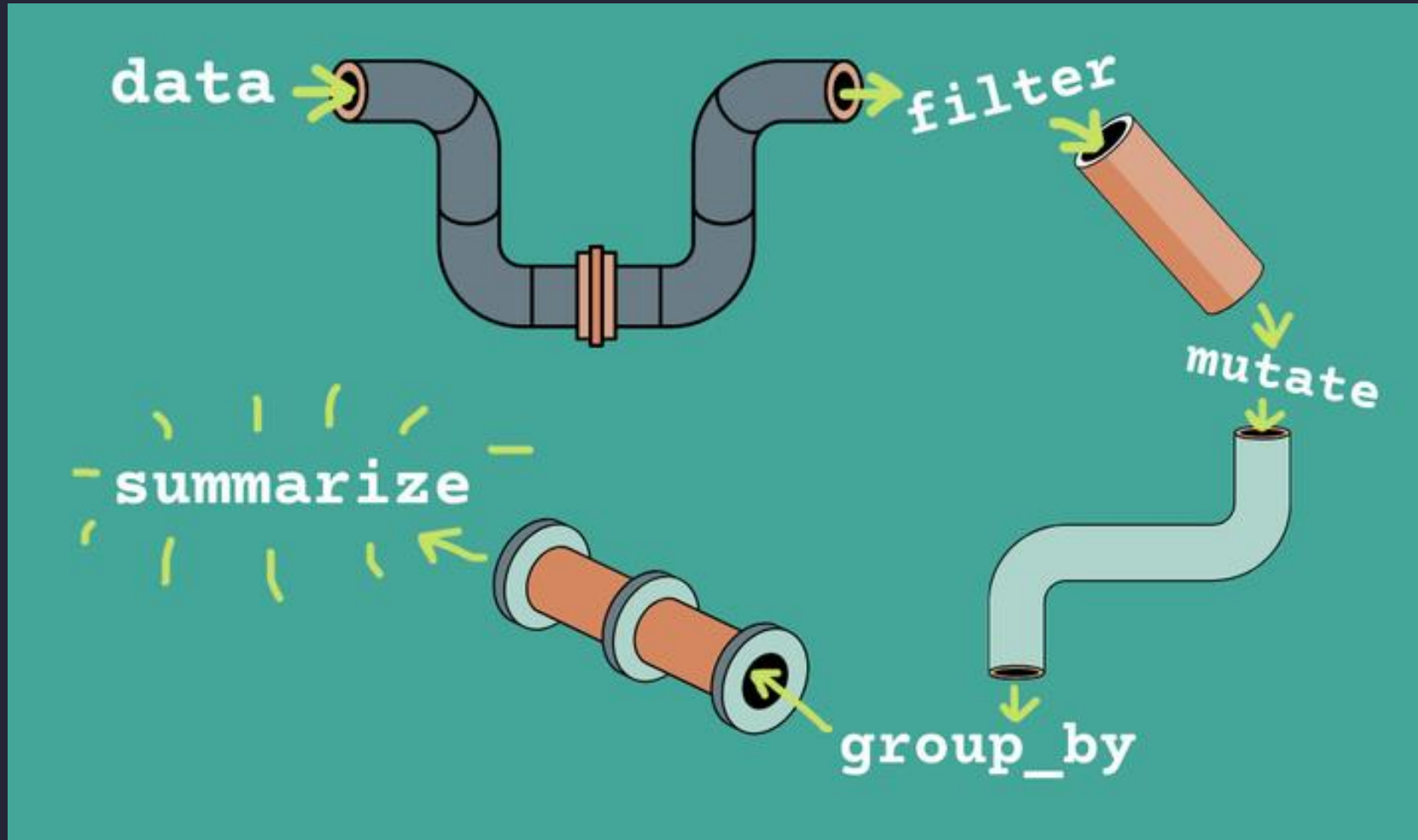
Cómo agrupar y luego resumir con **group_by**

- Cuando resumimos los datos después de la agrupación, esto es lo que sucede:

```
#3. Calcular estadísticas de edad por sexo
titanic |>
  filter(!is.na(Age)) |> # Elimina valores faltantes en edad
  group_by(Sex) |>       # Agrupa por sexo
  summarize(
    promedio_edad = mean(Age),
    mediana_edad = median(Age),
    total_pasajeros = n()
  )
```

- La función `summarize` aplica el resumen a cada grupo por separado.
- Para ver otro ejemplo, calculemos la mediana, el mínimo y máximo de la tasa de asesinatos en las cuatro regiones del país usando la función `promedio`, `mediana` y `total pasajeros` definida anteriormente:

Resumen pipe operator



¿Qué hacer si mis datos no son tidy?

- Carguemos estos datos:

```
library(tidyverse)
library(dslabs)
path <- system.file("extdata", package="dslabs")
filename <- file.path(path, "fertility-two-countries-example.csv")
wide_data <- read_csv(filename)
```

- Una de las funciones más usadas del paquete tidyr es pivot_longer, que nos permite convertir datos anchos (wide data en inglés) en datos tidy.

¿Qué hacer si mis datos no son tidy?

country	year	cases		country	1999	2000
Afghanistan	1999	745	←	Afghanistan	745	2666
Afghanistan	2000	2666	←	Brazil	37737	80488
Brazil	1999	37737	←	China	212258	213766
Brazil	2000	80488	←			
China	1999	212258	←			
China	2000	213766	←			

table4

pivot_longer

- Igual que con la mayoría de las funciones de tidyverse, el primer argumento de la función `pivot_longer` es el data frame que será procesado.
- Aquí queremos cambiar la forma del set de datos `wide_data` para que cada fila represente una observación de fertilidad, que implica que necesitamos tres columnas para almacenar el año, el país y el valor observado.
- A través de los argumentos `names_to` y `values_to`, le daremos a `pivot_longer` los nombres de columna que le queremos asignar a las columnas que contienen los nombres de columna y las observaciones actuales, respectivamente.
- Por defecto, estos nombres son `name` (nombre) y `value` (valor), los cuales son buenas opciones en general. En este caso, una mejor opción para estos dos argumentos serían `year` y `fertility`.

pivot_longer

```
new_tidy_data <- pivot_longer(wide_data, `1960`:`2015`,  
names_to = "year", values_to = "fertility")
```

#También podemos usar el pipe de esta manera:

```
new_tidy_data <- wide_data |>  
pivot_longer(`1960`:`2015`, names_to = "year", values_to = "fertility")
```

#Podemos ver que los datos se han convertido al formato tidy

con columnas year y fertility:

```
head(new_tidy_data)
```

pivot_longer

```
# La función pivot_longer supone que los nombres de columna son caracteres.  
# Así que necesitamos un poco más de wrangling antes de poder graficar.  
# Necesitamos convertir la columna con los años en números.
```

```
new_tidy_data <- wide_data |>  
pivot_longer(-country, names_to = "year", values_to = "fertility") |>  
mutate(year = as.integer(year))
```

- Más información para cambiar formato de datos en:
<http://rafalab.dfci.harvard.edu/dslibro/c%C3%B3mo-cambiar-el-formato-de-datos.html>

Visualización de datos: ¿Qué es ggplot2?

- La visualización de datos exploratorios es quizás la mayor ventaja de R. Uno puede pasar rápidamente de la idea a los datos al gráfico con un equilibrio único de flexibilidad y facilidad. Por ejemplo, Excel puede ser más fácil que R para algunos gráficos, pero no es tan flexible.
- R ofrece varios sistemas para graficar (por ejemplo, base R, lattice o ggplot2)
- usaremos ggplot2 porque
 - es versátil y elegante.
 - emplea una teoría bien fundada para la creación declarativa de gráficos llamada "Gramática de Gráficos" (Wilkinson 2005)
 - forma parte del tidyverse → se complementa bien con las herramientas que ya hemos aprendido.



Visualización de datos con ggplot2

```
library(dplyr)  
library(ggplot2)
```

- Una razón por la cual ggplot2 es generalmente más intuitiva para los principiantes es porque usa una gramática de gráficos, el gg de ggplot2.
- Una limitación de ggplot2 es que está diseñado para trabajar exclusivamente con tablas de datos en formato tidy (donde las filas son observaciones y las columnas son variables).
- Para usar ggplot2, tendrán que aprender varias funciones y argumentos. Estos son difíciles de memorizar, por lo que les recomiendo que tengan a mano la hoja de referencia de ggplot2.

Los componentes de un gráfico en ggplot2

- Los tres componentes principales para considerar son:
 - **Datos:** Se está resumiendo el set de datos de asesinatos con armas de Estados Unidos. Nos referimos a esto como el componente **data**.
 - **Geometría:** Tipo de gráfico. Por ejemplo: diagrama de dispersión diagrama de barras, histograma, densidades suaves (*smooth densities* en inglés), gráfico Q-Q y diagrama de cajas.
 - **Mapeo estético:** El gráfico usa varias señales visuales para representar la información proveída por el set de datos. Las dos señales más importantes en este gráfico son las posiciones de los puntos en el eje-x y el eje-y, que representan el tamaño de la población y el número total de asesinatos, respectivamente. Cada punto representa una observación diferente, y *mapeamos* los datos de estas observaciones y las señales visuales a las escalas x e y. El color es otra señal visual que asignamos a la región. Nos referimos a esto como el componente de **mapeo estético**. La forma en que definimos el mapeo depende de qué **geometría** estamos usando.

Construyamos un gráfico: data

```
library(dslabs)  
data(titanic)
```

```
ggplot(data = titanic)
```

o

```
titanic |> ggplot()
```

El código crea un gráfico, en este caso una pizarra en blanco ya que no se ha definido la geometría. La única opción de estilo que vemos es un fondo gris.

Podemos asignarlo a un objeto y ver su clase.

```
p <- ggplot(data = murders)  
class(p)
```

Construimos un gráfico: Geometrías

En ggplot2 creamos gráficos agregando capas (layers en inglés). Las capas pueden definir geometrías, calcular estadísticas de resumen, definir qué escalas (scales en inglés) usar o incluso cambiar estilos. Para añadir capas, usamos el símbolo +.

```
DATOS|> ggplot() + CAPA 1 + CAPA 2 + ... + CAPA N
```

Usualmente, la primera capa que agregamos define la geometría. Queremos hacer un diagrama de dispersión. ¿Qué geometría debemos utilizar?

Echando un vistazo rápido a la hoja de referencia, vemos que la función utilizada para crear gráficos con esta geometría es `geom_point`.

Construyamos un gráfico: Mapeos estéticos

- Describen cómo las propiedades de los datos se conectan con las características del gráfico, como la distancia a lo largo de un eje, el tamaño o el color.
- La función `aes` conecta los datos con lo que vemos en el gráfico mediante la definición de asignaciones estéticas y, por eso, será una de las funciones que más utilizarán al graficar.

```
# Paso 2: Preparar los datos (limpiar valores faltantes)
datos_filtrados <- titanic |>
  filter(!is.na(Age), !is.na(Fare), !is.na(Sex), !is.na(Pclass))
```

Construyamos un gráfico: más capas

- Podemos entonces agregar muchas más capas.

Paso 3: Crear el gráfico de dispersión

```
ggplot(data = datos_filtrados, aes(x = Age, y = Fare)) +
```

Paso 4: Agregar puntos con estética de color y tamaño

```
geom_point(aes(color = Sex, size = as.factor(Pclass)), alpha = 0.7) +
```

Paso 5: Agregar etiquetas y título

```
labs(
```

```
  title = "Relación entre Edad y Tarifa en el Titanic",
```

```
  subtitle = "Colores por Sexo y Tamaño según Clase",
```

```
  x = "Edad del Pasajero",
```

```
  y = "Tarifa Pagada (Fare)",
```

```
  color = "Sexo",
```

```
  size = "Clase"
```

```
) +
```

Paso 6: Agregar tema visual para mejorar apariencia

```
theme_minimal() +
```

```
theme(
```

```
  plot.title = element_text(size = 16, face = "bold"),
```

```
  plot.subtitle = element_text(size = 12),
```

```
  legend.position = "right"
```

```
)
```

Relación entre Edad y Tarifa en el Titanic

Colores por Sexo y Tamaño según Clase



Otro ejemplo

```
# Paso 2: Filtrar los pasajeros que no sobrevivieron
```

```
muertos <- titanic |>  
  filter(Survived == 0)
```

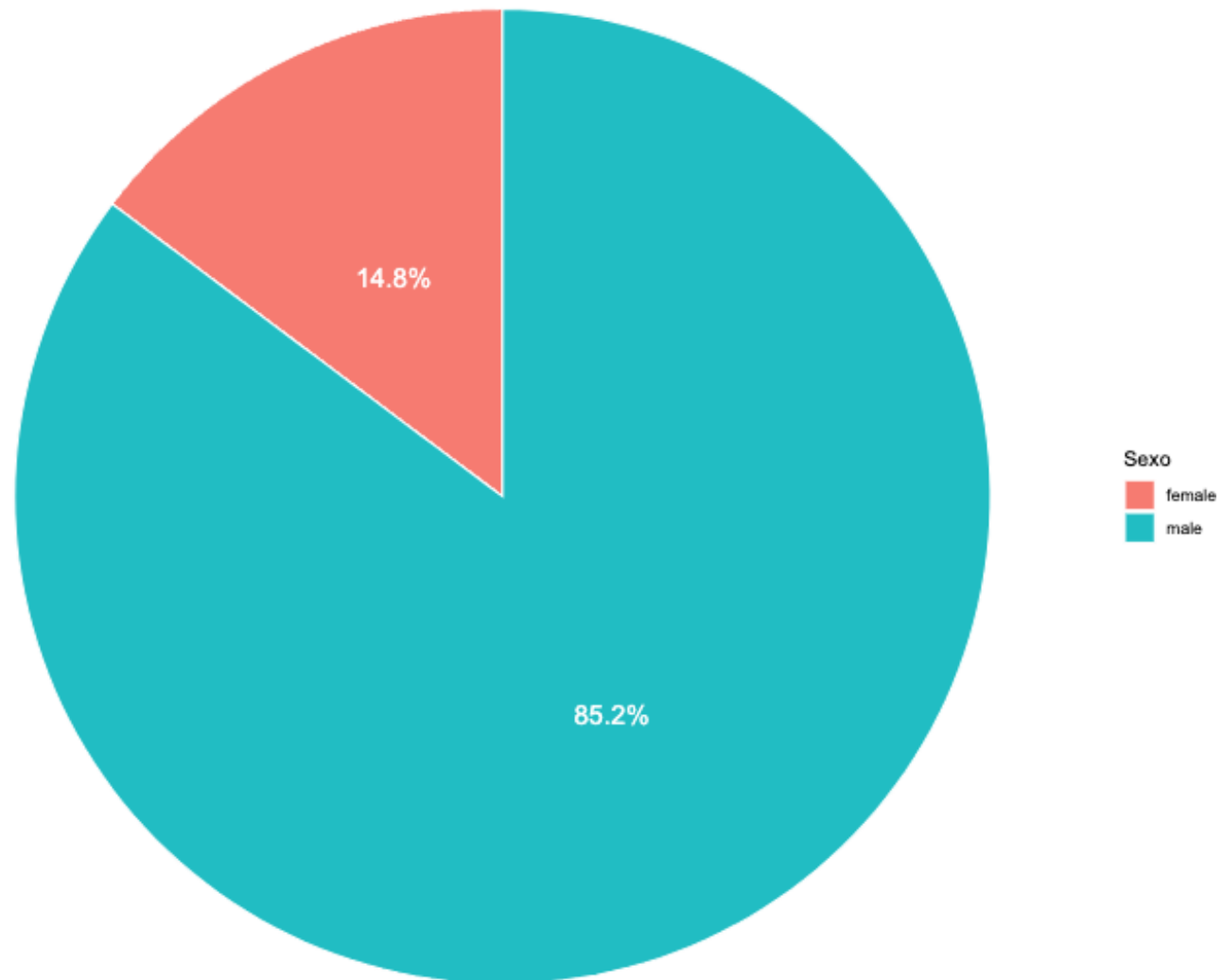
```
# Paso 3: Contar cuántos hombres y mujeres fallecieron
```

```
muertos_por_sexo <- muertos |>  
  count(Sex) |>  
  mutate(porcentaje = n / sum(n) * 100)
```

```
# Paso 4: Crear gráfico de pastel
```

```
ggplot(muertos_por_sexo, aes(x = "", y = porcentaje, fill = Sex)) +  
  geom_col(width = 1, color = "white") +  
  coord_polar("y") + # Convierte el gráfico de barras en circular  
  labs(  
    title = "Distribución de Muertes por Sexo en el Titanic",  
    fill = "Sexo"  
  ) +  
  theme_void() + # Elimina fondo y ejes  
  geom_text(aes(label = paste0(round(porcentaje, 1), "%")),  
    position = position_stack(vjust = 0.5), color = "white", size = 5)
```

Distribución de Muertes por Sexo en el Titanic



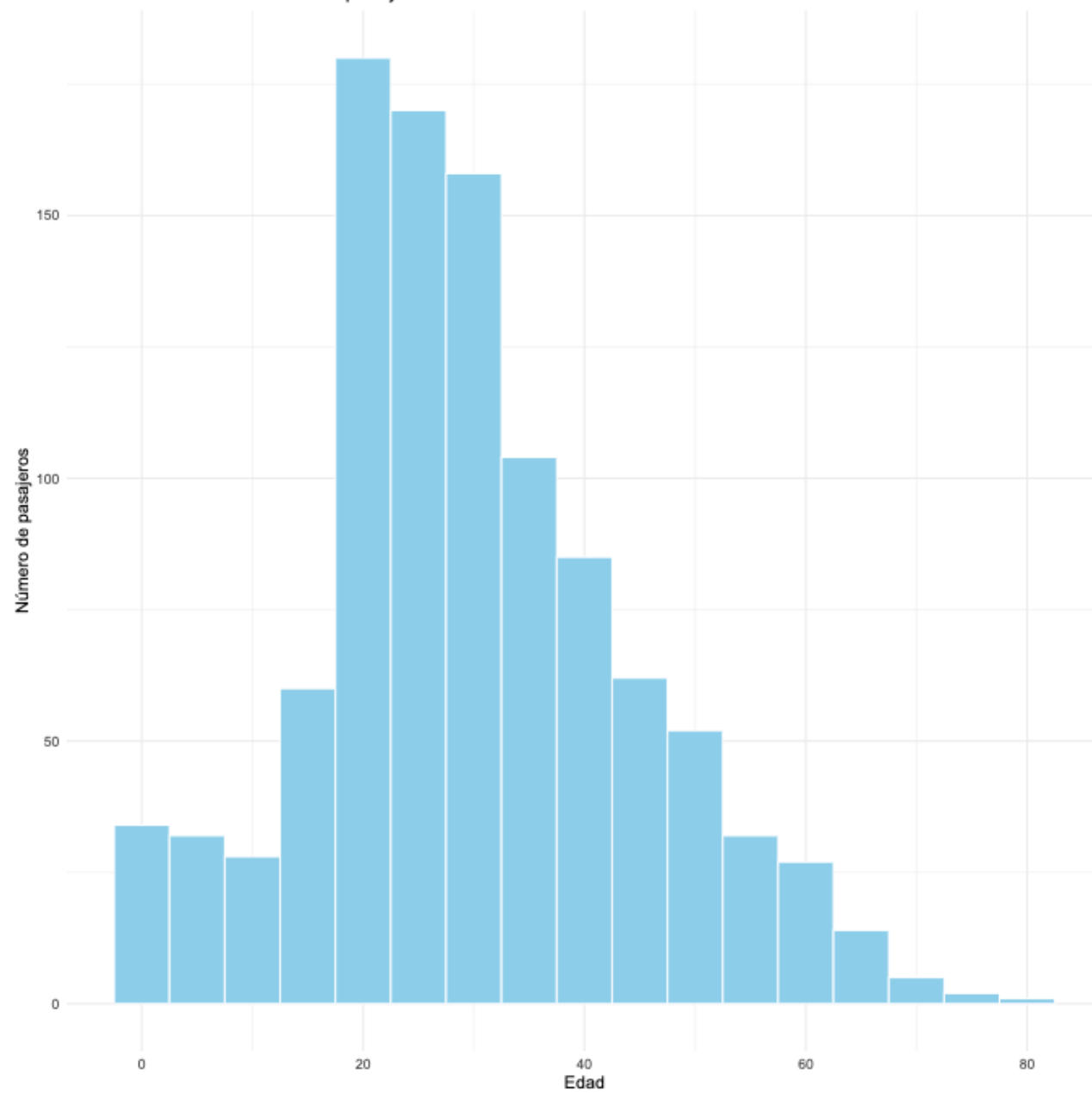
Otro ejemplo

```
# Paso 2: Explorar los datos (opcional)
# Verificar si hay valores NA en Age
sum(is.na(titanic$Age)) # Saber cuántos NA hay

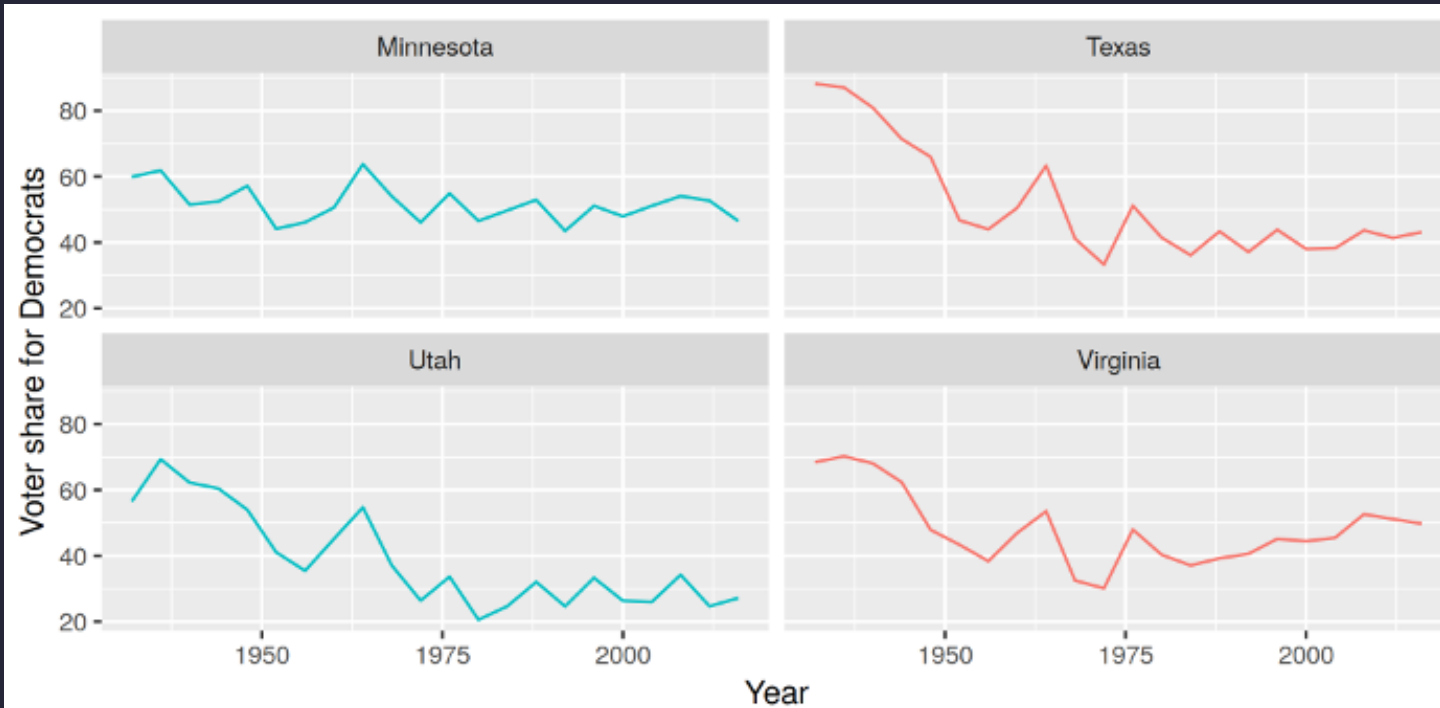
# Paso 3: Eliminar filas con edad NA (opcional pero recomendado para graficar)
titanic_filtrado <- titanic |>
  filter(!is.na(Age))

# Paso 4: Crear el histograma
ggplot(titanic_filtrado, aes(x = Age)) +
  geom_histogram(binwidth = 5, fill = "skyblue", color = "white") +
  labs(
    title = "Distribución de edades de los pasajeros",
    x = "Edad",
    y = "Número de pasajeros"
  ) +
  theme_minimal()
```


Distribución de edades de los pasajeros



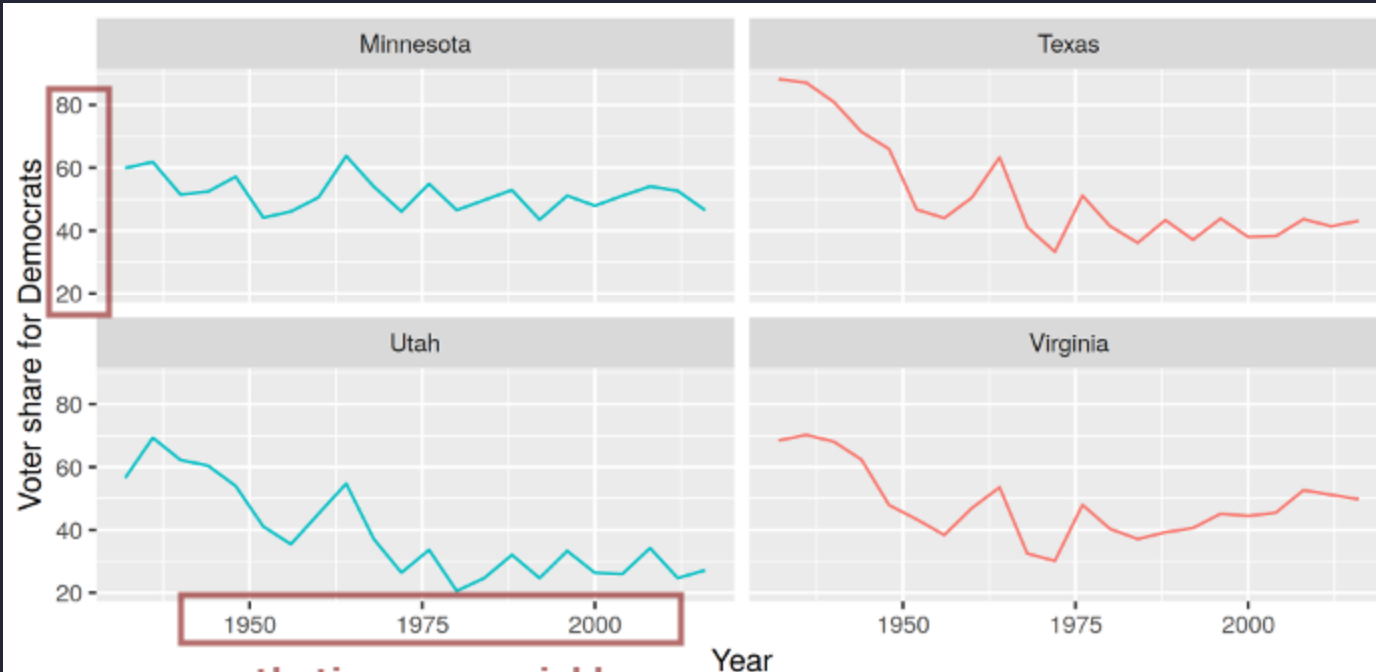
Resumen ggplot



data to plot

```
ggplot(sampled_pres, aes(x = year, y = demVote, color = south)) +  
  geom_line() +  
  facet_wrap(~ state) +  
  scale_color_discrete(guide = guide_legend(title = "Is Southern State"),  
    limits = c(TRUE, FALSE),  
    labels = c('Yes', 'No')) +  
  xlab('Year') +  
  ylab('Voter share for Democrats') +  
  theme(legend.position = 'bottom')
```

Resumen ggplot



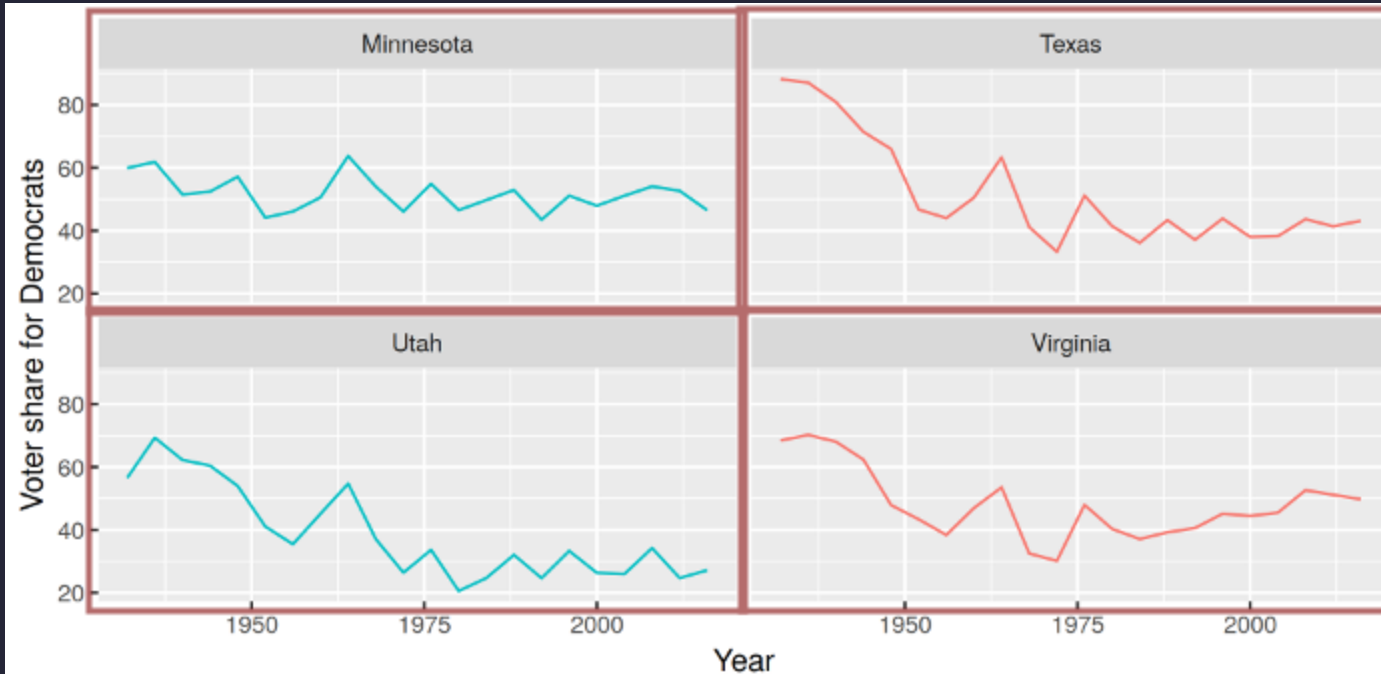
aesthetics map variables
to visual properties

```
ggplot(sampled_pres, aes(x = year, y = demVote, color = south)) +  
  geom_line() +  
  facet_wrap(~ state) +  
  scale_color_discrete(guide = guide_legend(title = "Is Southern State"),  
    limits = c(TRUE, FALSE),  
    labels = c('Yes', 'No')) +  
  xlab('Year') +  
  ylab('Voter share for Democrats') +  
  theme(legend.position = 'bottom')
```

Resumen ggplot



Resumen ggplot



Is Southern State — Yes — No


```
ggplot(sampled_pres, aes(x = year, y = demVote, color = south)) +  
  geom_line() +  
  facet_wrap(~ state) +  
  scale_color_discrete(guide = guide_legend(title = "Is Southern State"),  
    limits = c(TRUE, FALSE),  
    labels = c('Yes', 'No')) +  
  xlab('Year') +  
  ylab('Voter share for Democrats') +  
  theme(legend.position = 'bottom')
```

facets allow to create "small multiples"
for data that is subset by a variable

Visualización de datos con ggplot2

- Un sitio web útil para encontrar inspiración, ejemplos de código y sugerencias es <https://www.data-to-viz.com>.

Ejercicio

-  **Ejercicio 1: Análisis de pasajeros por clase Utilizando la base (titanic.csv)**
- **Instrucciones:**
 1. Construya un gráfico de barras, un gráfico de torta (*pie chart*) y un gráfico de Pareto para mostrar la cantidad de pasajeros por clase (Pclass).
 2. ¿Qué clase tiene más pasajeros?
 3. ¿Qué gráfico considera más adecuado para interpretar esta variable? Justifique.