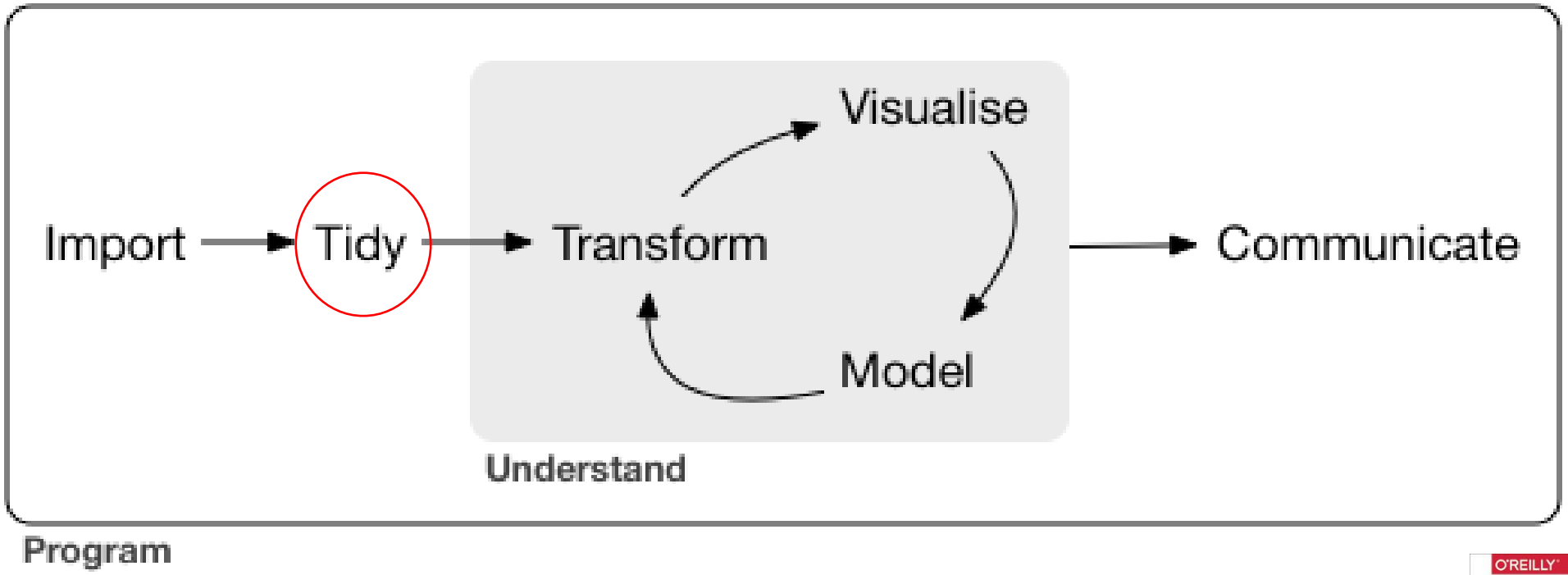
R Intermedio para Ciencias Sociales y Gestión Pública

SESIÓN N° 2:

Tidyverse

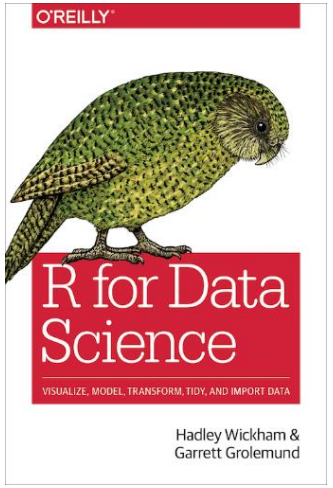
Diciembre 2021

Objetivo final



Tidying your data means storing it in a consistent form that matches the semantics of the dataset with the way it is stored. In brief, when your data is tidy, each column is a variable, and each row is an observation. Tidy data is important because the consistent structure lets you focus your struggle on questions about the data, not fighting to get the data into the right form for different functions.

Lectura recomendada: (2017) Wickham, Hadley & Garrett Grolemund. R for Data Science.

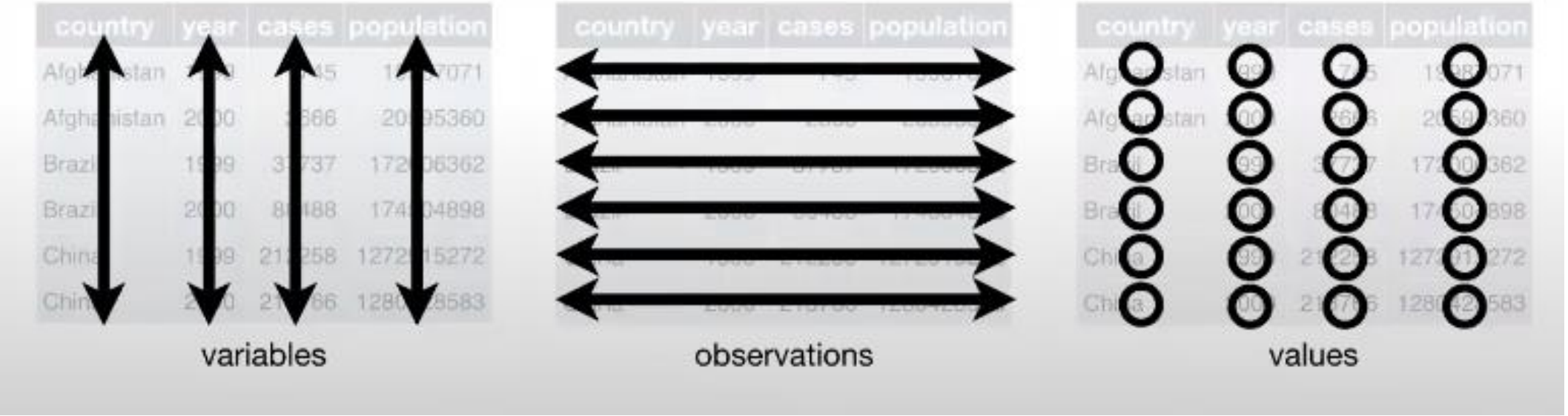


Qué es el Tidyverse?

El tidyverse es una colección de paquetes R diseñados para la ciencia de datos.

Todos los paquetes comparten una filosofía de diseño, una gramática y estructuras de datos subyacentes.

Qué es el Tidyverse?

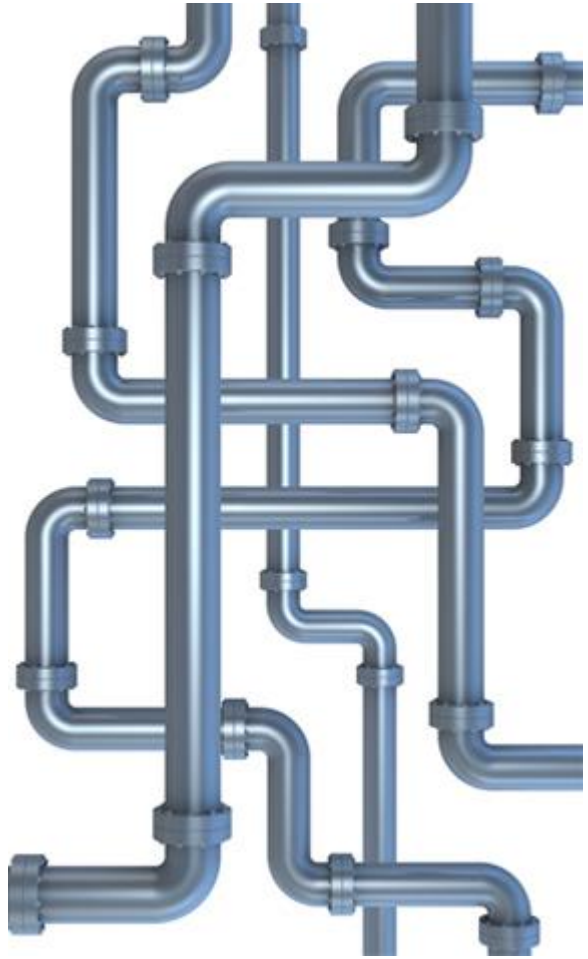


PRINCIPIOS

- Se trabaja con objetos data.frame o tibble (Diferencias, aquí).
- Se trabaja con data ordenada (tidy data):
  - Cada variable está en su propia columna.
  - Cada observación o caso en su propia fila

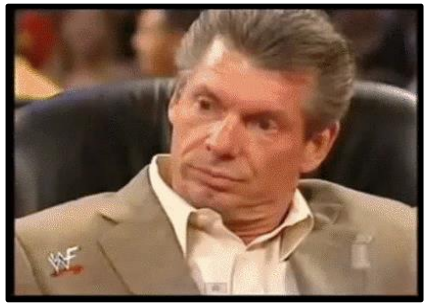
Qué es el operador pipe?

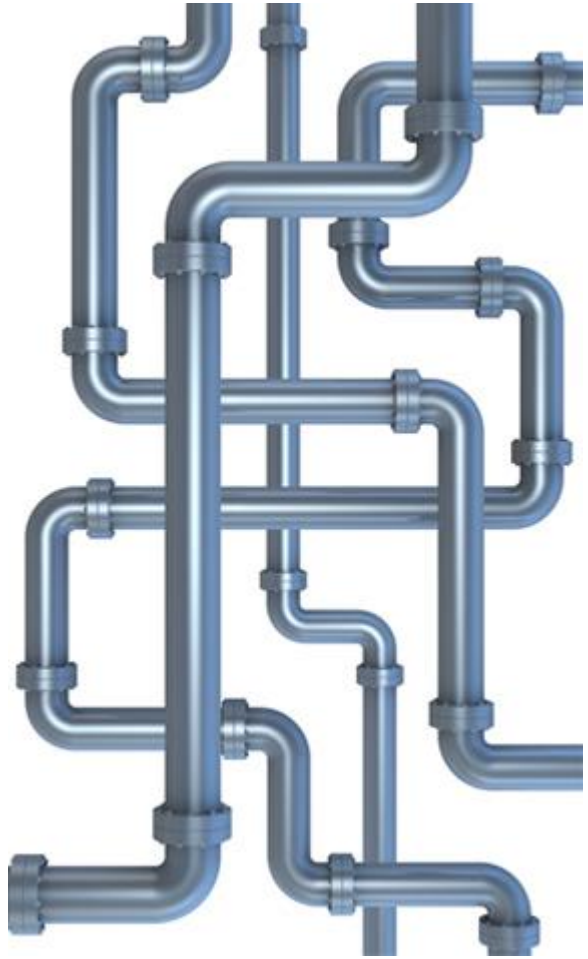
- Operador que permite expresar de forma clara una secuencia de múltiples operaciones.
- Es una sintaxis en cadena (tubería), de forma que el operador coge el output (‘la salida’) de una sentencia de código y la convierte en el input (‘el argumento’) de una nueva sentencia.
  - ✓ Podemos interpretarlo como un “ENTONCES”.
  - ✓ La salida (resultado) del código a la izquierda de |> es argumento de la función de la derecha.
- Recientemente en la versión 4.1.0 de R se insertó la sintaxis del pipe nativo.



Qué es el operador pipe?

```
Resultado1 <- acción1(data)
Resultado2 <- acción2(Resultado1)
Resultadofinal <- acción3(Resultado2)
```





```
Resultadofinal <- data |>
      acción1() |>
      acción2() |>
      acción3()
```

Manipular data con dplyr

- `mutate()` agrega nuevas variables que son funciones de variables existentes
- `select()` elige variables en función de sus nombres.
- `filter()` escoge casos basados en sus valores.
- `summarise()` reduce varios valores a un solo resumen.
- `arrange()` cambia el orden de las filas
- `group_by()` agrupa según una variable de grupo (factor)

Todas las funciones trabajan de forma similar:

- El primer argumento es el data frame
- Los argumentos siguientes describen qué se va a hacer con la data, usando los nombres de las variables.
- El resultado es una nueva data frame

Tienes que recordar las funciones lógicas básicas:

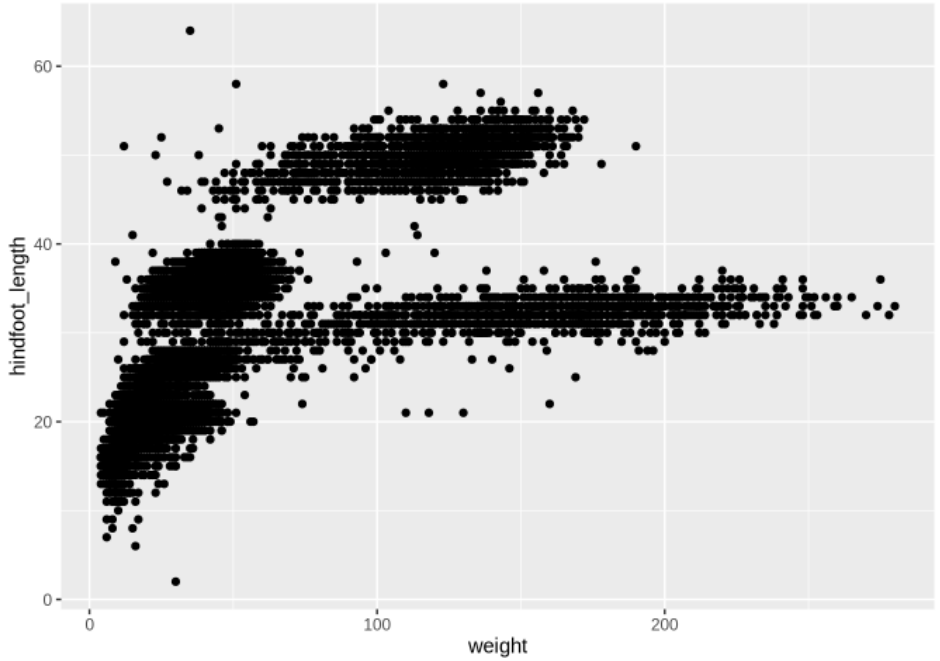
=	Igualdad
==	Equivalencia
|	o
&	y

Creación de gráficos con ggplot2

ESTRUCTURA GENERAL DEL CÓDIGO

```
ggplot(data = <Data>, mapping= aes(<MAPPINGS>)) + <GEOM_FUNCTION>()
```

```
ggplot(data = surveys_complete, aes(x = weight, y = hindfoot_length)) +
  geom_point()
```



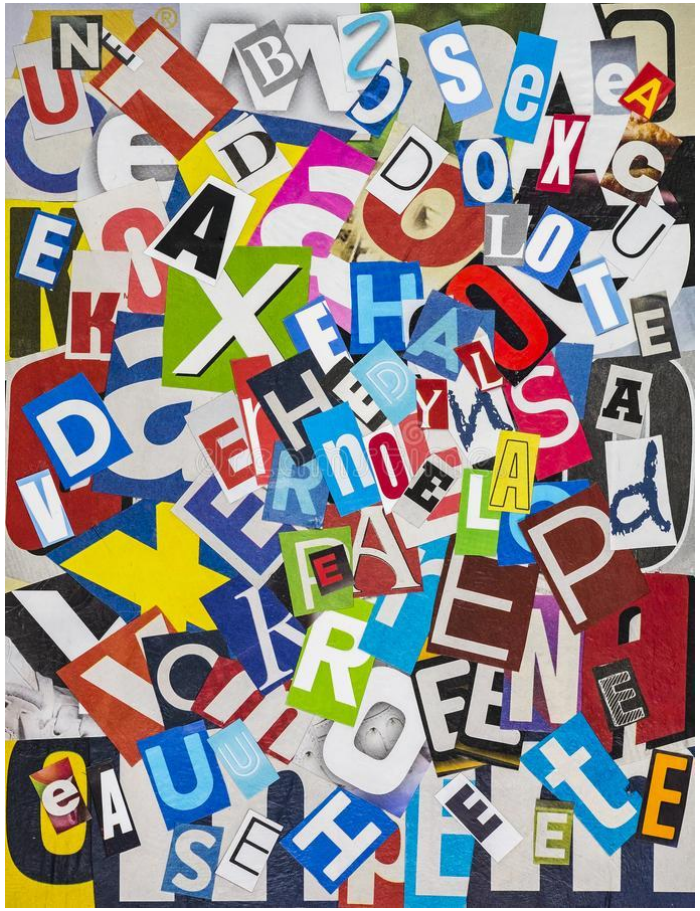
Manipular una cadena de texto con stringr

El paquete stringr proporciona un conjunto cohesivo de funciones diseñadas para hacer que trabajar con strings sea lo más fácil posible.

Las variables string también son llamadas variables alfanuméricas, cadena de texto/palabras/caracteres.

Son variables QUE SON TRATADAS COMO TEXTO (Recordar: un número no siempre es un número).

Lectura recomendada: Capítulo sobre Strings. R for Data Science.



Mg. Christian Chiroque Ruiz | christian.chiroque@pucp.edu.pe

Manipular una cadena de texto con stringr

El trabajar con cadenas de texto no es la parte más “amigable” de la tarea de preprocesamiento. Esto debido a que el principal mecanismo para trabajar con strings es nuestra capacidad para visualizar patrones. Para ello existen las EXPRESIONES REGULARES.

Qué son expresiones regulares? (regex)

En cómputo teórico y teoría de lenguajes formales, una expresión regular, o expresión racional, también son conocidas como regex o regexp, por su contracción de las palabras inglesas regular expression, es una secuencia de caracteres que conforma un patrón de búsqueda.

Se utilizan principalmente para la búsqueda de patrones de cadenas de caracteres u operaciones de sustituciones.

Qué son expresiones regulares? (regex)

Símbolo	Cualquier texto que…
[:alpha:]	… contenga una letra
.x.	… contenga una x entre dos caracteres.
\\.	…contenga un punto. Ojo: Se necesita dos backslash (alt+92 en Windows)
\\\\	…contenga un backslash (Se necesitan 4 backslash)
^x	… inicie con x (^ alt+94)
$x	… termine con x
\d	…contenga dígitos
\s	…contenga espacios
[abc]	…contenga los elementos a, b o c
[^abc]	…contenga cualquier elemento menos a, b o c. Ten cuidado: Recuerda que ^ significa también “al principio”. Sin embargo, si está dentro de [] equivale a negación.
(?<=x)	…esté precedido por x
(?=x)	…. Esté seguido por x

Mg. Christian Chiroque Ruiz | christian.chiroque@pucp.edu.pe

Qué son expresiones regulares? (regex)

Repeticiones: Puedes identificar el patrón de repetición dentro de la cadena. De esa manera:

Símbolo	Significado
?	0 o 1
+	1 o más
*	0 o más

También puedes identificar un número específico de veces:

Símbolo	Significado
{n}	n veces
{n,}	n o más
{,m}	Al menos m
{n,m}	Entre n y m veces

Mg. Christian Chiroque Ruiz | christian.chiroque@pucp.edu.pe

Qué son expresiones regulares? (regex)

Ejemplos:

Patrón	Todos los caracteres que..
“^a”	…comiencen con la letra a
“$a”	…terminen con la letra a
".a."	…tengan dentro del caracter la letra a
"\\.com"	…contenga la frase “.com”
"gr(e|a)y"	…contenga la frase “grey” o “gray
"CC+"	…contengan 1 vez o más veces el frase CC
"C{4}"	…contengan la letra C cuatro veces seguidas
“\s”	…contengan espacios en blanco

Manipular una cadena de texto con stringr

Dentro del paquete strings hay más de 20 funciones, sin embargo, se podría resaltar las siguientes:

Función	Detalle
str_extract	Extrae el fragmento de string indicado.
str_split	Parte la cadena en un patrón determinado
str_subset	Extrae una porción de la cadena de acuerdo a la ubicación de ciertos caracteres.
str_view	Muestra los matches que se realicen con el patrón indicado.
str_to_lower	Cambia todo a minúscula
str_to_upper	Cambia todo a mayúscula

Manipular una cadena de texto con stringr

Practiquemos con la data de titanic, la variable Name:

```
> head(df$Name)
[1] "Braund, Mr. Owen Harris"
[2] "Cumings, Mrs. John Bradley (Florence Briggs Thayer)"
[3] "Heikkinen, Miss. Laina"
[4] "Futrelle, Mrs. Jacques Heath (Lily May Peel)"
[5] "Allen, Mr. William Henry"
[6] "Moran, Mr. James"
```

Para un mayor detalle de las funciones y su utilidad te recomiendo revisar: Cheat Sheet de Stringr

Manipular una cadena de texto con stringr

Práctica:

- Cadena que empiece con cualquier vocal.

"[aeiou]"

- Cadena que consiste sólo en consonantes.

"^[^aeiou]+$"

- Cadena que termine en x.

“x$”