Práctica 1

Ana Escoto

1/9/2022

Contents

Paquetes **3**

Cargando los datos **3**

Un poquito de *dplyr* y limpieza **3**

Primero, los pipes . . . 3

Limpieza de nombres . . . 4

select() y *filter()* . . . 9

Tabulados **10**

Tabulados con tabyl() . . . 10

Cálculo de frecuencias . . . 11

Totales y porcentajes . . . 12

Grammar of tables: gt . . . 14

Descriptivos para variables cuantitativas **15**

Medidas numéricas básicas . . . 16

Visualización de datos, un pequeño disclaimer **16**

Gráficos de base . . . 16

Grammar of graphics: ggplot **18**

Un lienzo para dibujar **18**

Gráficos univariados **19**

Para cualitativas . . . 19

Para variables cuantitativas . . . 22

Histograma . . . 23

Intro a dos variables . . . 24

Paquetes

```
if (!require("pacman")) install.packages("pacman")#instala pacman si se requiere
```

```
## Loading required package: pacman
```

```
pacman::p_load(tidyverse,
               readxl,
               writexl,
               haven,
               sjlabelled,
               janitor,
               infer,
               ggpubr,
               magrittr,
               gt,
               GGally)
```

Cargando los datos

Desde STATA

```
ags_t321 <- haven::read_dta("./datos/AGS_SDEMT321.dta", encoding="latin1")
```

Desde Excel:

```
ICI_2021 <- readxl::read_excel("./datos/ICI_2021.xlsx",
                               sheet = "para_importar")
```

```
## New names:
## * `` -> ...2
```

Un poquito de *dplyr* y limpieza

Primero, los pipes

R utiliza dos pipes el nativo "|>" y el pipe que está en *dplyr* "%>%". Algunas de las diferencias las puedes checar acá https://eliocamp.github.io/codigo-r/2021/05/r-pipa-nativa/

En estas prácticas utilizaremos el segundo, pero son muy parecidos y para que esta instructora recicle algunos de sus códigos viejos. Pero funcionan igual:

```
ags_t321 |> #pipe nativo, no necesita instalación
  head()
```

```
## # A tibble: 6 x 114
##            R_DEF   LOC   MUN   EST EST_D_TRI EST_D_MEN  AGEB T_LOC_TRI T_LOC_MEN
##        <dbl+lbl> <dbl> <dbl> <dbl>     <dbl>     <dbl> <dbl>     <dbl>     <dbl>
## 1 0 [Entrevista~    NA    11    20         7         6     0         2         2
## 2 0 [Entrevista~    NA     5    30         6         5     0         3         2
## 3 0 [Entrevista~    NA     1    20        11         9     0         4         4
## 4 0 [Entrevista~    NA     3    20        11         9     0         4         4
## 5 0 [Entrevista~    NA     2    20        11         9     0         4         4
## 6 0 [Entrevista~    NA     1    20         1         1     0         1         1
## # ... with 105 more variables: CD_A <dbl+lbl>, ENT <dbl+lbl>, CON <dbl>,
## #   UPM <dbl>, D_SEM <dbl+lbl>, N_PRO_VIV <dbl>, V_SEL <dbl+lbl>,
## #   N_HOG <dbl+lbl>, H_MUD <dbl+lbl>, N_ENT <dbl+lbl>, PER <dbl+lbl>,
## #   N_REN <dbl+lbl>, C_RES <dbl+lbl>, PAR_C <dbl>, SEX <dbl+lbl>, EDA <dbl>,
## #   NAC_DIA <dbl+lbl>, NAC_MES <dbl+lbl>, NAC_ANIO <dbl>, L_NAC_C <dbl+lbl>,
## #   CS_P12 <dbl+lbl>, CS_P13_1 <dbl+lbl>, CS_P13_2 <dbl+lbl>, CS_P14_C <chr>,
## #   CS_P15 <dbl+lbl>, CS_P16 <dbl+lbl>, CS_P17 <dbl+lbl>, N_HIJ <dbl+lbl>, ...
```

```
ags_t321  %>%  #pipe de dplyr, necesita instalación de dplyr en tidyverse
  head()
```

```
## # A tibble: 6 x 114
##            R_DEF   LOC   MUN   EST EST_D_TRI EST_D_MEN  AGEB T_LOC_TRI T_LOC_MEN
##        <dbl+lbl> <dbl> <dbl> <dbl>     <dbl>     <dbl> <dbl>     <dbl>     <dbl>
## 1 0 [Entrevista~    NA    11    20         7         6     0         2         2
## 2 0 [Entrevista~    NA     5    30         6         5     0         3         2
## 3 0 [Entrevista~    NA     1    20        11         9     0         4         4
## 4 0 [Entrevista~    NA     3    20        11         9     0         4         4
## 5 0 [Entrevista~    NA     2    20        11         9     0         4         4
## 6 0 [Entrevista~    NA     1    20         1         1     0         1         1
## # ... with 105 more variables: CD_A <dbl+lbl>, ENT <dbl+lbl>, CON <dbl>,
## #   UPM <dbl>, D_SEM <dbl+lbl>, N_PRO_VIV <dbl>, V_SEL <dbl+lbl>,
## #   N_HOG <dbl+lbl>, H_MUD <dbl+lbl>, N_ENT <dbl+lbl>, PER <dbl+lbl>,
## #   N_REN <dbl+lbl>, C_RES <dbl+lbl>, PAR_C <dbl>, SEX <dbl+lbl>, EDA <dbl>,
## #   NAC_DIA <dbl+lbl>, NAC_MES <dbl+lbl>, NAC_ANIO <dbl>, L_NAC_C <dbl+lbl>,
## #   CS_P12 <dbl+lbl>, CS_P13_1 <dbl+lbl>, CS_P13_2 <dbl+lbl>, CS_P14_C <chr>,
## #   CS_P15 <dbl+lbl>, CS_P16 <dbl+lbl>, CS_P17 <dbl+lbl>, N_HIJ <dbl+lbl>, ...
```

Limpieza de nombres

Este paso también nos permitirá enseñar otro pipe que está en el paquete *magrittr*.

Los nombres de una base de datos son los nombres de las columnas.

```
names(ags_t321)
```

```
##   [1] "R_DEF"      "LOC"        "MUN"        "EST"        "EST_D_TRI"
##   [6] "EST_D_MEN"  "AGEB"       "T_LOC_TRI"  "T_LOC_MEN"  "CD_A"
##  [11] "ENT"        "CON"        "UPM"        "D_SEM"      "N_PRO_VIV"
##  [16] "V_SEL"      "N_HOG"      "H_MUD"      "N_ENT"      "PER"
##  [21] "N_REN"      "C_RES"      "PAR_C"      "SEX"        "EDA"
##  [26] "NAC_DIA"    "NAC_MES"    "NAC_ANIO"   "L_NAC_C"    "CS_P12"
##  [31] "CS_P13_1"   "CS_P13_2"   "CS_P14_C"   "CS_P15"     "CS_P16"
```

```
##  [36] "CS_P17"     "N_HIJ"      "E_CON"      "CS_P20A_1"  "CS_P20A_C"
##  [41] "CS_P20B_1"  "CS_P20B_C"  "CS_P20C_1"  "CS_AD_MOT"  "CS_P21_DES"
##  [46] "CS_AD_DES"  "CS_NR_MOT"  "CS_P23_DES" "CS_NR_ORI"  "UR"
##  [51] "ZONA"       "SALARIO"    "FAC_TRI"    "FAC_MEN"    "CLASE1"
##  [56] "CLASE2"     "CLASE3"     "POS_OCU"    "SEG_SOC"    "RAMA"
##  [61] "C_OCU11C"   "ING7C"      "DUR9C"      "EMPLE7C"    "MEDICA5C"
##  [66] "BUSCAR5C"   "RAMA_EST1"  "RAMA_EST2"  "DUR_EST"    "AMBITO1"
##  [71] "AMBITO2"    "TUE1"       "TUE2"       "TUE3"       "BUSQUEDA"
##  [76] "D_ANT_LAB"  "D_CEXP_EST" "DUR_DES"    "SUB_O"      "S_CLASIFI"
##  [81] "REMUNE2C"   "PRE_ASA"    "TIP_CON"    "DISPO"      "NODISPO"
##  [86] "C_INAC5C"   "PNEA_EST"   "NIV_INS"    "EDA5C"      "EDA7C"
##  [91] "EDA12C"     "EDA19C"     "HIJ5C"      "DOMESTICO"  "ANIOS_ESC"
##  [96] "HRSOCUP"    "INGOCUP"    "ING_X_HRS"  "TPG_P8A"    "TCCO"
## [101] "CP_ANOC"    "IMSSISSSTE" "MA48ME1SM"  "P14APOYOS"  "SCIAN"
## [106] "T_TRA"      "EMP_PPAL"   "TUE_PPAL"   "TRANS_PPAL" "MH_FIL2"
## [111] "MH_COL"     "SEC_INS"    "TIPO"       "MES_CAL"
```

```
names(ICI_2021)
```

```
##  [1] "País"
##  [2] "...2"
##  [3] "Protección de derechos humanos"
##  [4] "Homicidios dolosos"
##  [5] "Confianza en la policía"
##  [6] "Independencia del poder judicial"
##  [7] "Protección de derechos de propiedad"
##  [8] "Tiempo para resolver quiebras"
##  [9] "Cumplimiento de contratos"
## [10] "Índice de Estado de Derecho"
## [11] "Índice de Paz Global"
## [12] "Contaminación del aire"
## [13] "Emisiones de CO2"
## [14] "Recursos hídricos renovables"
## [15] "Áreas naturales protegidas"
## [16] "Superficie forestal perdida"
## [17] "Uso de pesticidas"
## [18] "Fuentes de energía no contaminantes"
## [19] "Índice de vulnerabilidad a efectos del cambio climático"
## [20] "Índice de Gini"
## [21] "Índice Global de Brecha de Género"
## [22] "Mujeres en la PEA"
## [23] "Dependientes de la PEA"
## [24] "Acceso a agua potable"
## [25] "Acceso a alcantarillado"
## [26] "Analfabetismo"
## [27] "Escolaridad promedio"
## [28] "Calidad educativa"
## [29] "Esperanza de vida"
## [30] "Mortalidad infantil"
## [31] "Cobertura de vacunación"
## [32] "Médicos y médicas"
## [33] "Gasto en salud per cápita"
## [34] "Gasto en salud por cuenta propia"
## [35] "Estabilidad política y ausencia de violencia"
```

```
## [36] "Interferencia militar en el Estado de derecho o en el proceso político"
## [37] "Libertades civiles"
## [38] "Índice de Percepción de Corrupción"
## [39] "Disponibilidad de información pública"
## [40] "Participación electoral"
## [41] "Equidad en los congresos"
## [42] "Índice de efectividad del gobierno"
## [43] "Miembro de la Alianza para el Gobierno Abierto"
## [44] "Índice de desarrollo de Gobierno Electrónico"
## [45] "Facilidad para abrir una empresa"
## [46] "Tiempo para preparar y pagar impuestos"
## [47] "Ingresos fiscales"
## [48] "Finanzas sanas"
## [49] "Carga impositiva"
## [50] "Edad efectiva de retiro"
## [51] "Flexibilidad de las leyes laborales"
## [52] "Productividad media del trabajo"
## [53] "Valor agregado de la industria"
## [54] "Índice de transparencia y regulación de la propiedad privada"
## [55] "Crecimiento del PIB"
## [56] "Crecimiento promedio del PIB"
## [57] "Inflación"
## [58] "Inflación promedio"
## [59] "Desempleo"
## [60] "Deuda externa"
## [61] "Calificación de deuda"
## [62] "Reservas"
## [63] "Libertad económica"
## [64] "Índice Riesgos de seguridad energética"
## [65] "Líneas móviles"
## [66] "Usuarios de internet"
## [67] "Servidores de internet seguros"
## [68] "Flujo de pasajeros aéreos"
## [69] "Índice de desempeño logístico (transporte)"
## [70] "Tráfico portuario de contenedores"
## [71] "Penetración del sistema financiero privado"
## [72] "Capitalización del mercado de valores"
## [73] "Socios comerciales efectivos"
## [74] "Apertura comercial"
## [75] "Diversificación de las exportaciones"
## [76] "Diversificación de las importaciones"
## [77] "Libertad comercial"
## [78] "Inversión extranjera directa (neta)"
## [79] "Inversión Extranjera Directa neta promedio"
## [80] "Ingresos por turismo"
## [81] "Gasto en investigación y desarrollo"
## [82] "Coeficiente de invención"
## [83] "Artículos científicos y técnicos"
## [84] "Exportaciones de alta tecnología"
## [85] "Índice de Complejidad Económica"
## [86] "Empresas ISO 9001"
## [87] "PIB en servicios"
## [88] "0"
## [89] "Inversión (FBCF)"
```

```
## [90] "Talento"
```

Como vemos en las bases hay mayúsculas, caracteres especiales y demás. Esto lo podemos cambiar

```
ICI_2021<-ICI_2021 %>%
  janitor::clean_names()

names(ICI_2021)
```

```
##  [1] "pais"
##  [2] "x2"
##  [3] "proteccion_de_derechos_humanos"
##  [4] "homicidios_dolosos"
##  [5] "confianza_en_la_policia"
##  [6] "independencia_del_poder_judicial"
##  [7] "proteccion_de_derechos_de_propiedad"
##  [8] "tiempo_para_resolver_quiebras"
##  [9] "cumplimiento_de_contratos"
## [10] "indice_de_estado_de_derecho"
## [11] "indice_de_paz_global"
## [12] "contaminacion_del_aire"
## [13] "emisiones_de_co2"
## [14] "recursos_hidricos_renovables"
## [15] "areas_naturales_protegidas"
## [16] "superficie_forestal_perdida"
## [17] "uso_de_pesticidas"
## [18] "fuentes_de_energia_no_contaminantes"
## [19] "indice_de_vulnerabilidad_a_efectos_del_cambio_climatico"
## [20] "indice_de_gini"
## [21] "indice_global_de_brecha_de_genero"
## [22] "mujeres_en_la_pea"
## [23] "dependientes_de_la_pea"
## [24] "acceso_a_agua_potable"
## [25] "acceso_a_alcantarillado"
## [26] "analfabetismo"
## [27] "escolaridad_promedio"
## [28] "calidad_educativa"
## [29] "esperanza_de_vida"
## [30] "mortalidad_infantil"
## [31] "cobertura_de_vacunacion"
## [32] "medicos_y_medicas"
## [33] "gasto_en_salud_per_capita"
## [34] "gasto_en_salud_por_cuenta_propia"
## [35] "estabilidad_politica_y_ausencia_de_violencia"
## [36] "interferencia_militar_en_el_estado_de_derecho_o_en_el_proceso_politico"
## [37] "libertades_civiles"
## [38] "indice_de_percepcion_de_corrupcion"
## [39] "disponibilidad_de_informacion_publica"
## [40] "participacion_electoral"
## [41] "equidad_en_los_congresos"
## [42] "indice_de_efectividad_del_gobierno"
## [43] "miembro_de_la_alianza_para_el_gobierno_abierto"
## [44] "indice_de_desarrollo_de_gobierno_electronico"
```

```
## [45] "facilidad_para_abrir_una_empresa"
## [46] "tiempo_para_preparar_y_pagar_impuestos"
## [47] "ingresos_fiscales"
## [48] "finanzas_sanas"
## [49] "carga_impositiva"
## [50] "edad_efectiva_de_retiro"
## [51] "flexibilidad_de_las_leyes_laborales"
## [52] "productividad_media_del_trabajo"
## [53] "valor_agregado_de_la_industria"
## [54] "indice_de_transparencia_y_regulacion_de_la_propiedad_privada"
## [55] "crecimiento_del_pib"
## [56] "crecimiento_promedio_del_pib"
## [57] "inflacion"
## [58] "inflacion_promedio"
## [59] "desempleo"
## [60] "deuda_externa"
## [61] "calificacion_de_deuda"
## [62] "reservas"
## [63] "libertad_economica"
## [64] "indice_riesgos_de_seguridad_energetica"
## [65] "lineas_moviles"
## [66] "usuarios_de_internet"
## [67] "servidores_de_internet_seguros"
## [68] "flujo_de_pasajeros_aereos"
## [69] "indice_de_desempeno_logistico_transporte"
## [70] "trafico_portuario_de_contenedores"
## [71] "penetracion_del_sistema_financiero_privado"
## [72] "capitalizacion_del_mercado_de_valores"
## [73] "socios_comerciales_efectivos"
## [74] "apertura_comercial"
## [75] "diversificacion_de_las_exportaciones"
## [76] "diversificacion_de_las_importaciones"
## [77] "libertad_comercial"
## [78] "inversion_extranjera_directa_neta"
## [79] "inversion_extranjera_directa_neta_promedio"
## [80] "ingresos_por_turismo"
## [81] "gasto_en_investigacion_y_desarrollo"
## [82] "coeficiente_de_invencion"
## [83] "articulos_cientificos_y_tecnicos"
## [84] "exportaciones_de_alta_tecnologia"
## [85] "indice_de_complejidad_economica"
## [86] "empresas_iso_9001"
## [87] "pib_en_servicios"
## [88] "x0"
## [89] "inversion_fbcf"
## [90] "talento"
```

Si quisiéramos que la acción quedará de un solo, podemos usar un pipe diferente:

```
ags_t321 %<>%
  clean_names()

names(ags_t321)
```

```
##   [1] "r_def"      "loc"        "mun"        "est"        "est_d_tri"
##   [6] "est_d_men"  "ageb"       "t_loc_tri"  "t_loc_men"  "cd_a"
##  [11] "ent"        "con"        "upm"        "d_sem"      "n_pro_viv"
##  [16] "v_sel"      "n_hog"      "h_mud"      "n_ent"      "per"
##  [21] "n_ren"      "c_res"      "par_c"      "sex"        "eda"
##  [26] "nac_dia"    "nac_mes"    "nac_anio"   "l_nac_c"    "cs_p12"
##  [31] "cs_p13_1"   "cs_p13_2"   "cs_p14_c"   "cs_p15"     "cs_p16"
##  [36] "cs_p17"     "n_hij"      "e_con"      "cs_p20a_1"  "cs_p20a_c"
##  [41] "cs_p20b_1"  "cs_p20b_c"  "cs_p20c_1"  "cs_ad_mot"  "cs_p21_des"
##  [46] "cs_ad_des"  "cs_nr_mot"  "cs_p23_des" "cs_nr_ori"  "ur"
##  [51] "zona"       "salario"    "fac_tri"    "fac_men"    "clase1"
##  [56] "clase2"     "clase3"     "pos_ocu"    "seg_soc"    "rama"
##  [61] "c_ocu11c"   "ing7c"      "dur9c"      "emple7c"    "medica5c"
##  [66] "buscar5c"   "rama_est1"  "rama_est2"  "dur_est"    "ambito1"
##  [71] "ambito2"    "tue1"       "tue2"       "tue3"       "busqueda"
##  [76] "d_ant_lab"  "d_cexp_est" "dur_des"    "sub_o"      "s_clasifi"
##  [81] "remune2c"   "pre_asa"    "tip_con"    "dispo"      "nodispo"
##  [86] "c_inac5c"   "pnea_est"   "niv_ins"    "eda5c"      "eda7c"
##  [91] "eda12c"     "eda19c"     "hij5c"      "domestico"  "anios_esc"
##  [96] "hrsocup"    "ingocup"    "ing_x_hrs"  "tpg_p8a"    "tcco"
## [101] "cp_anoc"    "imssissste" "ma48me1sm"  "p14apoyos"  "scian"
## [106] "t_tra"      "emp_ppal"   "tue_ppal"   "trans_ppal" "mh_fil2"
## [111] "mh_col"     "sec_ins"    "tipo"       "mes_cal"
```

Más de otros *pipes* https://r4ds.had.co.nz/pipes.html

select() y *filter()*

Este es un recordatorio de que en dplyr, se filtran CASOS, es decir, líneas o renglones, y se seleccionan VARIABLES.

Por ejemplo:

```
ags_t321 %>%
  dplyr::select(sex, eda) %>%
  dplyr::filter(eda>11)
```

```
## # A tibble: 10,029 x 2
##           sex   eda
##     <dbl+lbl> <dbl>
##  1 2 [Mujer]     26
##  2 1 [Hombre]    34
##  3 2 [Mujer]     29
##  4 1 [Hombre]    56
##  5 1 [Hombre]    34
##  6 2 [Mujer]     18
##  7 2 [Mujer]     27
##  8 2 [Mujer]     51
##  9 1 [Hombre]    20
## 10 2 [Mujer]     28
## # ... with 10,019 more rows
```

En la documentación de la base de datos de la ENOE se nos señala que debemos quedarnos con quienes tienen entrevista completa "r_def==0" y con quiénes son habitante habituales ("c_res!=2")

Hagamos estos cambios:

```
ags_t321 %<>%
  filter(r_def==0) %>%
  filter(!c_res==2)
```

Tabulados

Tabulados con tabyl()

El comando tabyl del paquete janirot nos sirve para hacer tabulados. Para que sean más bonitas, necesitaremos cambiar algunas de nuestras variables a sus datos etiquetados

```
ags_t321 %>%
  dplyr::mutate(sex=sjlabelled::as_label(sex))  %>%
  janitor::tabyl(sex)
```

```
##     sex    n   percent
##  Hombre 6060 0.4827531
##   Mujer 6493 0.5172469
```

Para ver que esto es una distribución de frecuencias sería muy útil ver la proporción total, ello se realiza agregando un elemento más en nuestro código con una "tubería":

```
ags_t321 %>%
  mutate(sex=as_label(sex)) %>%
  tabyl(sex) %>%
  adorn_totals() #primer enchulamiento
```

```
##     sex     n   percent
##  Hombre  6060 0.4827531
##   Mujer  6493 0.5172469
##   Total 12553 1.0000000
```

Ahora, las proporciones son raras, y preferimos por los porcentajes.

```
ags_t321 %>%
  mutate(sex=as_label(sex)) %>% # cambia los valores de la variable a sus etiquetas
  tabyl(sex) %>% # para hacer la tabla
  adorn_totals() %>% # añade totales
  adorn_pct_formatting()  # nos da porcentaje en lugar de proporción
```

```
##     sex     n percent
##  Hombre  6060   48.3%
##   Mujer  6493   51.7%
##   Total 12553  100.0%
```

Vamos a darle una "ojeada" a esta variable

```
glimpse(ags_t321$niv_ins)
```

```
##  dbl+lbl [1:12553] 1, 3, 3, 2, 0, 2, 3, 3, 1, 1, 4, 4, 1, 4, 4, 4, 4, 0, 1,...
##  @ label       : chr "Clasificación de la población ocupada por nivel de instrucción"
##  @ format.stata: chr "%12.0g"
##  @ labels      : Named num [1:6] 0 1 2 3 4 5
##   ..- attr(*, "names")= chr [1:6] "No aplica" "Primaria incompleta" "Prrimaria completa" "Secundaria
```

Hoy hacemos la tabla, con las etiquetas:

```
ags_t321 %>%
  mutate(niv_ins=as_label(niv_ins)) %>% #esto sólo si hay etiquetas declaradas, recuerda
                tabyl(niv_ins)
```

```
##                    niv_ins    n      percent
##                  No aplica  938 0.0747231737
##         Primaria incompleta 2549 0.2030590297
##          Prrimaria completa 2215 0.1764518442
##         Secundaria completa 3391 0.2701346292
##  Medio superior y superior 3449 0.2747550386
##             No especificado   11 0.0008762846
```

Para que no nos salgan las categorías sin datos podemos poner una opción dentro del comando "tabyl()"

```
ags_t321 %>%
  mutate(niv_ins=as_label(niv_ins)) %>%
  tabyl(niv_ins,
        show_missing_levels=F ) %>% # esta opción elimina los valores con 0
  adorn_totals()
```

```
##                    niv_ins     n      percent
##                  No aplica   938 0.0747231737
##         Primaria incompleta  2549 0.2030590297
##          Prrimaria completa  2215 0.1764518442
##         Secundaria completa  3391 0.2701346292
##  Medio superior y superior  3449 0.2747550386
##             No especificado    11 0.0008762846
##                       Total 12553 1.0000000000
```

Cálculo de frecuencias

Las tablas de doble entrada tiene su nombre porque en las columnas entran los valores de una variable categórica, y en las filas de una segunda. Basicamente es como hacer un conteo de todas las combinaciones posibles entre los valores de una variable con la otra.

Por ejemplo, si quisiéramos combinar las dos variables que ya estudiamos lo podemos hacer, con una tabla de doble entrada:

```
ags_t321 %>%
  mutate(niv_ins=as_label(niv_ins)) %>% # para que las lea como factor
  mutate(sex=as_label(sex)) %>% # para que las lea como factor
  tabyl(niv_ins, sex, show_missing_levels=F ) %>% # incluimos aquí
  adorn_totals()
```

```
##                    niv_ins Hombre Mujer
##                  No aplica    474   464
##        Primaria incompleta   1272  1277
##         Prrimaria completa   1045  1170
##        Secundaria completa   1535  1856
##  Medio superior y superior   1730  1719
##            No especificado      4     7
##                      Total   6060  6493
```

Observamos que en cada celda confluyen los casos que comparten las mismas características:

```
ags_t321 %>%
    count(niv_ins==1 & sex==1) # nos da la segunda celda de la izquierda
```

```
## # A tibble: 2 x 2
##   `niv_ins == 1 & sex == 1`     n
##   <lgl>                     <int>
## 1 FALSE                     11281
## 2 TRUE                       1272
```

Totales y porcentajes

De esta manera se colocan todos los datos. Si observa al poner la función "adorn_totals()" lo agregó como una nueva fila de totales, pero también podemos pedirle que agregue una columna de totales.

```
ags_t321 %>%
  mutate(niv_ins=as_label(niv_ins)) %>% # para que las lea como factor
  mutate(sex=as_label(sex)) %>% # para que las lea como factor
  tabyl(niv_ins, sex, show_missing_levels=F ) %>% # incluimos aquí sex
  adorn_totals("col")
```

```
##                    niv_ins Hombre Mujer Total
##                  No aplica    474   464   938
##        Primaria incompleta   1272  1277  2549
##         Prrimaria completa   1045  1170  2215
##        Secundaria completa   1535  1856  3391
##  Medio superior y superior   1730  1719  3449
##            No especificado      4     7    11
```

O bien agregar los dos, introduciendo en el argumento "c("col","row")" un vector de caracteres de las dos opciones requeridas:

```
ags_t321 %>%
  mutate(niv_ins=as_label(niv_ins)) %>% # para que las lea como factor
  mutate(sex=as_label(sex)) %>% # para que las lea como factor
  tabyl(niv_ins, sex, show_missing_levels=F ) %>% # incluimos aquí sexo
  adorn_totals(c("col", "row"))
```

```
##                    niv_ins Hombre Mujer Total
##                  No aplica    474   464   938
##        Primaria incompleta   1272  1277  2549
```

```
##         Prrimaria completa   1045  1170  2215
##        Secundaria completa   1535  1856  3391
##  Medio superior y superior   1730  1719  3449
##            No especificado      4     7    11
##                      Total   6060  6493 12553
```

Del mismo modo, podemos calcular los porcentajes. Pero los podemos calcular de tres formas. Uno es que lo calculemos para los totales calculados para las filas, para las columnas o para el gran total poblacional.

Para columnas tenemos el siguiente código y los siguientes resultados:

```
ags_t321 %>%
  mutate(niv_ins=as_label(niv_ins)) %>% # para que las lea como factor
  mutate(sex=as_label(sex)) %>% # para que las lea como factor
  tabyl(niv_ins, sex, show_missing_levels=F ) %>% # incluimos aquí sexo
  adorn_totals(c("col", "row")) %>%
  adorn_percentages("col") %>% # Divide los valores entre el total de la columna
  adorn_pct_formatting() # lo vuelve porcentaje
```

```
##                   niv_ins Hombre  Mujer  Total
##                 No aplica   7.8%   7.1%   7.5%
##       Primaria incompleta  21.0%  19.7%  20.3%
##        Prrimaria completa  17.2%  18.0%  17.6%
##       Secundaria completa  25.3%  28.6%  27.0%
##  Medio superior y superior  28.5%  26.5%  27.5%
##           No especificado   0.1%   0.1%   0.1%
##                     Total 100.0% 100.0% 100.0%
```

Cuando se hagan cuadros de distribuciones (que todas sus partes suman 100), los porcentajes pueden ser una gran ayuda para la interpretación, sobre todos cuando se comparar poblaciones de categorías de diferente tamaño. Por lo general, queremos que los cuadros nos den información de donde están los totales y su 100%, de esta manera el lector se puede guiar de porcentaje con respecto a qué está leyendo. En este caso, vemos que el 100% es común en la última fila.

Veamos la diferencia de cómo podemos leer la misma celda, pero hoy, hemos calculado los porcentajes a nivel de fila:

```
ags_t321 %>%
  mutate(niv_ins=as_label(niv_ins)) %>% # para que las lea como factor
  mutate(sex=as_label(sex)) %>% # para que las lea como factor
  tabyl(niv_ins, sex, show_missing_levels=F ) %>%
  adorn_totals(c("col", "row")) %>%
  adorn_percentages("row") %>% # Divide los valores entre el total de la fila
  adorn_pct_formatting() # lo vuelve porcentaje
```

```
##                   niv_ins Hombre Mujer  Total
##                 No aplica  50.5% 49.5% 100.0%
##       Primaria incompleta  49.9% 50.1% 100.0%
##        Prrimaria completa  47.2% 52.8% 100.0%
##       Secundaria completa  45.3% 54.7% 100.0%
##  Medio superior y superior  50.2% 49.8% 100.0%
##           No especificado  36.4% 63.6% 100.0%
##                     Total  48.3% 51.7% 100.0%
```

Finalmente, podemos calcular los porcentajes con referencia a la población total en análisis. Es decir la celda en la esquina inferior derecha de nuestra tabla original.

```
ags_t321 %>%
  mutate(niv_ins=as_label(niv_ins)) %>% # para que las lea como factor
  mutate(sex=as_label(sex)) %>% # para que las lea como factor
  tabyl(niv_ins, sex, show_missing_levels=F ) %>% # incluimos aquí sexo
  adorn_totals(c("col", "row")) %>%
   adorn_percentages("all") %>% # Divide los valores entre el total de la población
  adorn_pct_formatting() # lo vuelve porcentaje
```

```
##                    niv_ins Hombre Mujer  Total
##                  No aplica   3.8%  3.7%   7.5%
##         Primaria incompleta  10.1% 10.2%  20.3%
##          Prrimaria completa   8.3%  9.3%  17.6%
##         Secundaria completa  12.2% 14.8%  27.0%
##  Medio superior y superior  13.8% 13.7%  27.5%
##             No especificado   0.0%  0.1%   0.1%
##                       Total  48.3% 51.7% 100.0%
```

Grammar of tables: gt

Es un paquete que nos permite poner nuestras tablas en mejores formatos.

Guardemos un ejemplo anterior en un objeto

```
mi_tabla<-ags_t321 %>%
  mutate(niv_ins=as_label(niv_ins)) %>% # para que las lea como factor
  mutate(sex=as_label(sex)) %>% # para que las lea como factor
  tabyl(niv_ins, sex, show_missing_levels=F ) %>% # incluimos aquí sexo
  adorn_totals(c("col", "row")) %>%
   adorn_percentages("all") %>% # Divide los valores entre el total de la población
  adorn_pct_formatting() # lo vuelve porcentaje
```

Veamos qué pasa con el comando "gt"

```
gt_tabla<-gt(mi_tabla)
gt_tabla
```

niv_ins	Hombre	Mujer	Total
No aplica	3.8%	3.7%	7.5%
Primaria incompleta	10.1%	10.2%	20.3%
Prrimaria completa	8.3%	9.3%	17.6%
Secundaria completa	12.2%	14.8%	27.0%
Medio superior y superior	13.8%	13.7%	27.5%
No especificado	0.0%	0.1%	0.1%
Total	48.3%	51.7%	100.0%

Con este formato será bastante sencillo agregar títulos y demás:

```
gt_tabla<-gt_tabla %>%
  tab_header(
    title = "Distribución del sexo de la población según nivel de escolaridad",
    subtitle = "Aguascalientes, trimestre III de 2021"
  )

gt_tabla
```

Distribución del sexo de la población según nivel de escolaridad
Aguascalientes, trimestre III de 2021

niv_ins	Hombre	Mujer	Total
No aplica	3.8%	3.7%	7.5%
Primaria incompleta	10.1%	10.2%	20.3%
Prrimaria completa	8.3%	9.3%	17.6%
Secundaria completa	12.2%	14.8%	27.0%
Medio superior y superior	13.8%	13.7%	27.5%
No especificado	0.0%	0.1%	0.1%
Total	48.3%	51.7%	100.0%

Agreguemos la fuente a nuestra tabla:

```
gt_tabla<-gt_tabla %>%
  tab_source_note(
    source_note = "Fuente: Cálculos propios con datos de INEGI"
  )

gt_tabla
```

Distribución del sexo de la población según nivel de escolaridad
Aguascalientes, trimestre III de 2021

niv_ins	Hombre	Mujer	Total
No aplica	3.8%	3.7%	7.5%
Primaria incompleta	10.1%	10.2%	20.3%
Prrimaria completa	8.3%	9.3%	17.6%
Secundaria completa	12.2%	14.8%	27.0%
Medio superior y superior	13.8%	13.7%	27.5%
No especificado	0.0%	0.1%	0.1%
Total	48.3%	51.7%	100.0%

Fuente: Cálculos propios con datos de INEGI

Checa más de este paquete por aquí https://gt.rstudio.com/articles/intro-creating-gt-tables.html

Descriptivos para variables cuantitativas

Vamos a empezar a revisar los gráficos para variables cuantitativas.

Medidas numéricas básicas

5 números

```
summary(ags_t321$ing_x_hrs) ## ingreso por horas
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    0.00    0.00   10.95   11.63  739.99
```

Con pipes se pueden crear "indicadores" de nuestras variables es un tibble

```
ags_t321 %>%
  summarise(nombre_indicador=mean(ing_x_hrs, na.rm=T))
```

```
## # A tibble: 1 x 1
##   nombre_indicador
##              <dbl>
## 1             11.0
```

Visualización de datos, un pequeño disclaimer

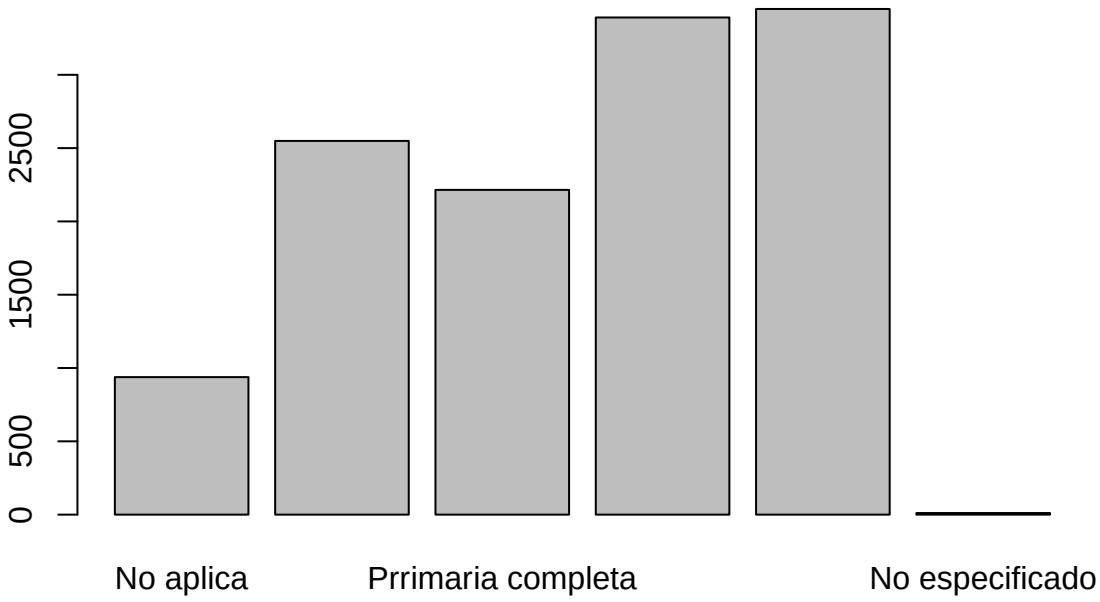
Hay cursos específicos de visualización de datos. Es maravilloso pero también requiere que estudiemos bien qué tipo de datos tenemos y cuáles son nuestros objetivos.

Me gusta mucho este recurso: https://www.data-to-viz.com/

Gráficos de base

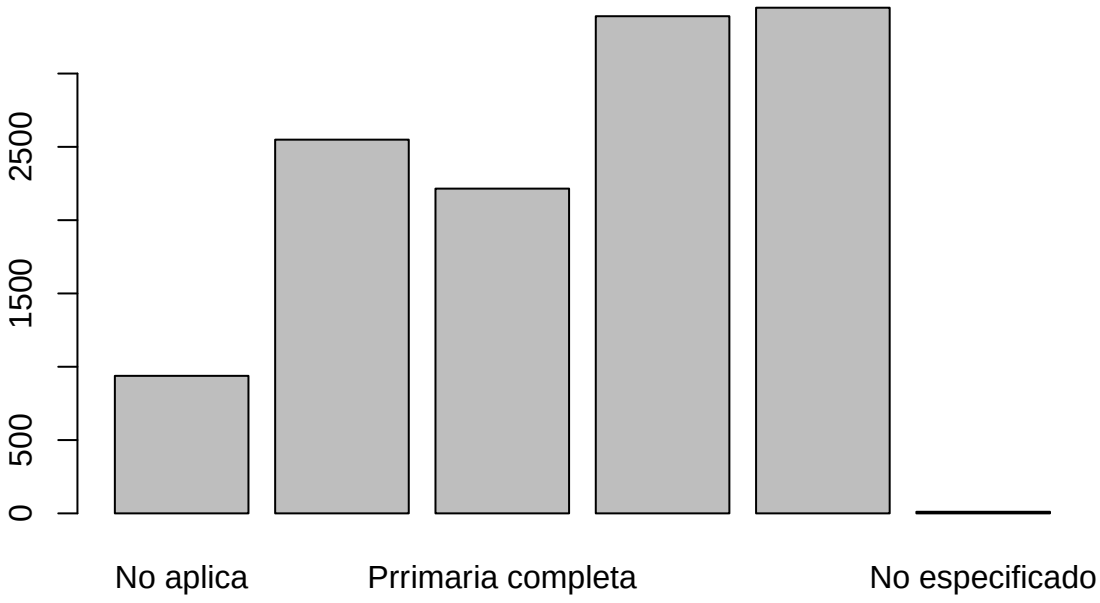
"plot()" Es la función más simple.

```
  plot(as_label(ags_t321$niv_ins))
```



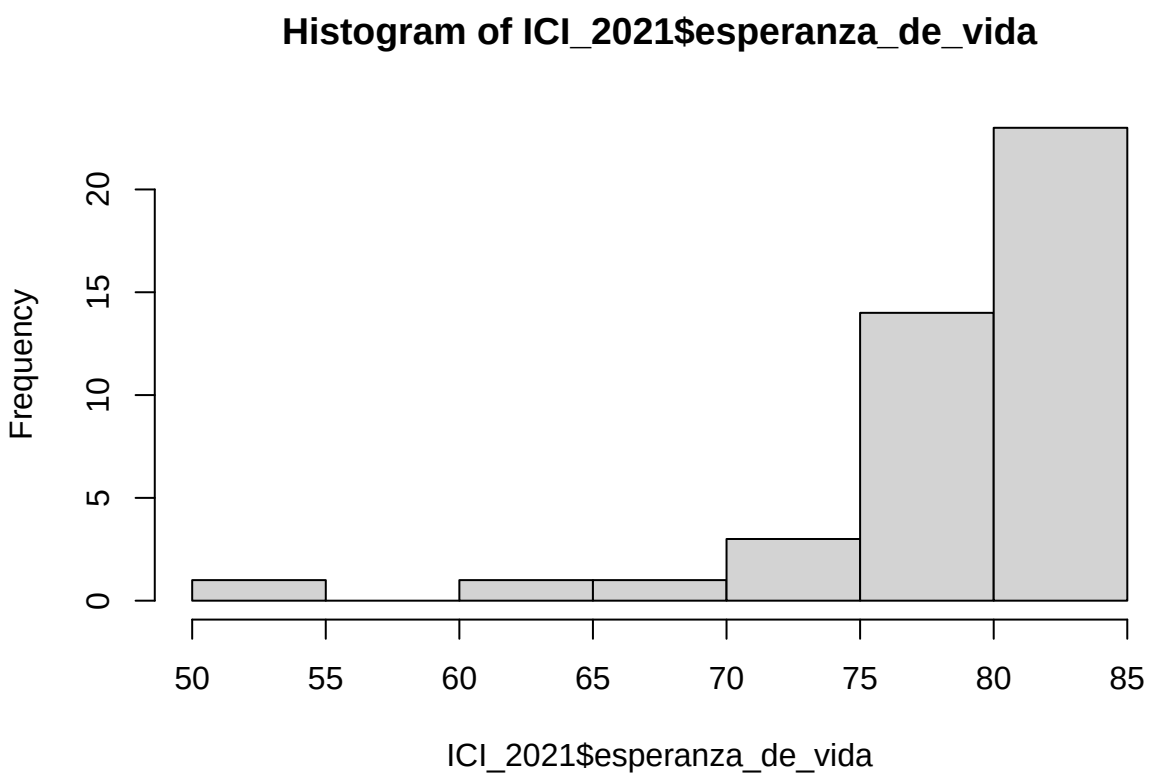
Esto es igual que:

```
barplot(table(as_label(ags_t321$niv_ins)))
```

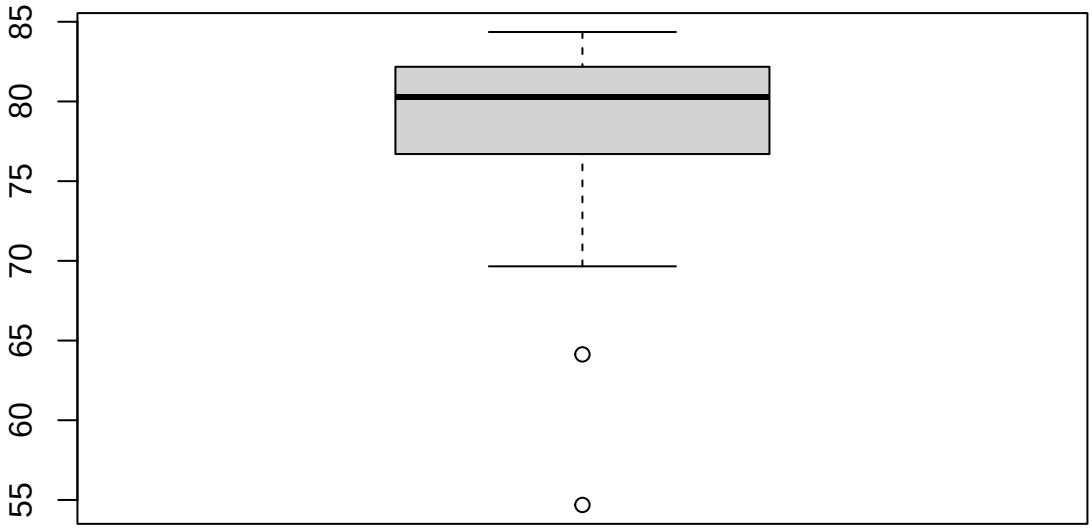


Histograma y el boxplot

```
hist(ICI_2021$esperanza_de_vida)
```




```
boxplot(ICI_2021$esperanza_de_vida)
```



Grammar of graphics: ggplot

Hoy vamos a presentar a un gran paquete ¡Es de los famosos! Y tiene más de diez años.

- https://qz.com/1007328/all-hail-ggplot2-the-code-powering-all-those-excellent-charts-is-10-years-old/

"gg" proviene de "Grammar of Graphics", funciona un poco como sintácticamente, de ahí su nombre.

Algunos recursos para aprender ggplot

- https://ggplot2-book.org/ hecha por el mero mero.
- http://sape.inf.usi.ch/quick-reference/ggplot2
- https://raw.githubusercontent.com/rstudio/cheatsheets/master/data-visualization-2.1.pdf

Vamos a revisar una presentación que es muy interesante

- https://evamaerey.github.io/ggplot2_grammar_guide/ggplot2_grammar_guide.html
- https://huygens.science.uva.nl/ggPlotteR/ Hace gráficos de ggplot con la base de datos de Gapminder

Un lienzo para dibujar

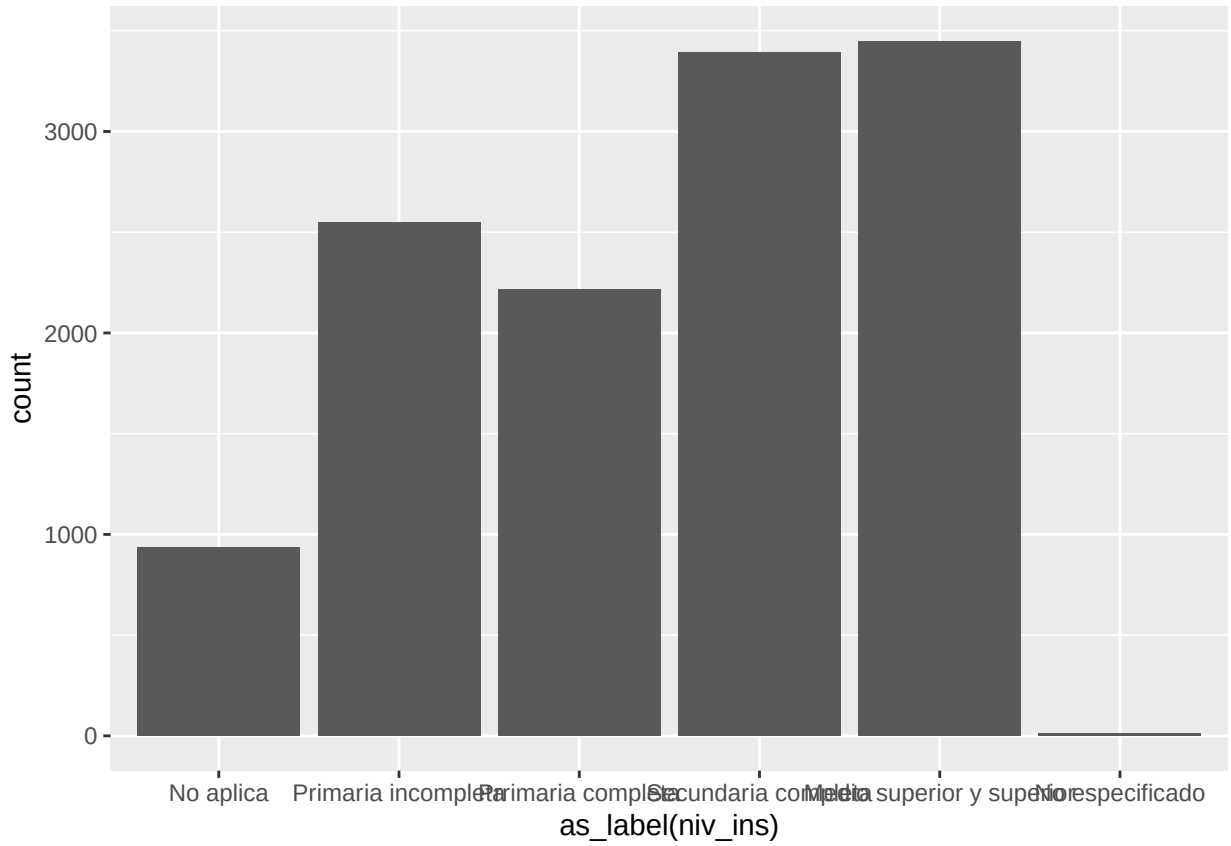
Para hacer un gráfico, ggplot2 tiene el comando "ggplot()". Hacer gráficos con esta función tiene una lógica aditiva. Lo ideal es que iniciemos estableciendo el mapeo estético de nuestro gráfico, con el comando aes()

```
g1<-ags_t321 %>%
  ggplot(aes(as_label(niv_ins)))

g1 # imprime el lienzo
```

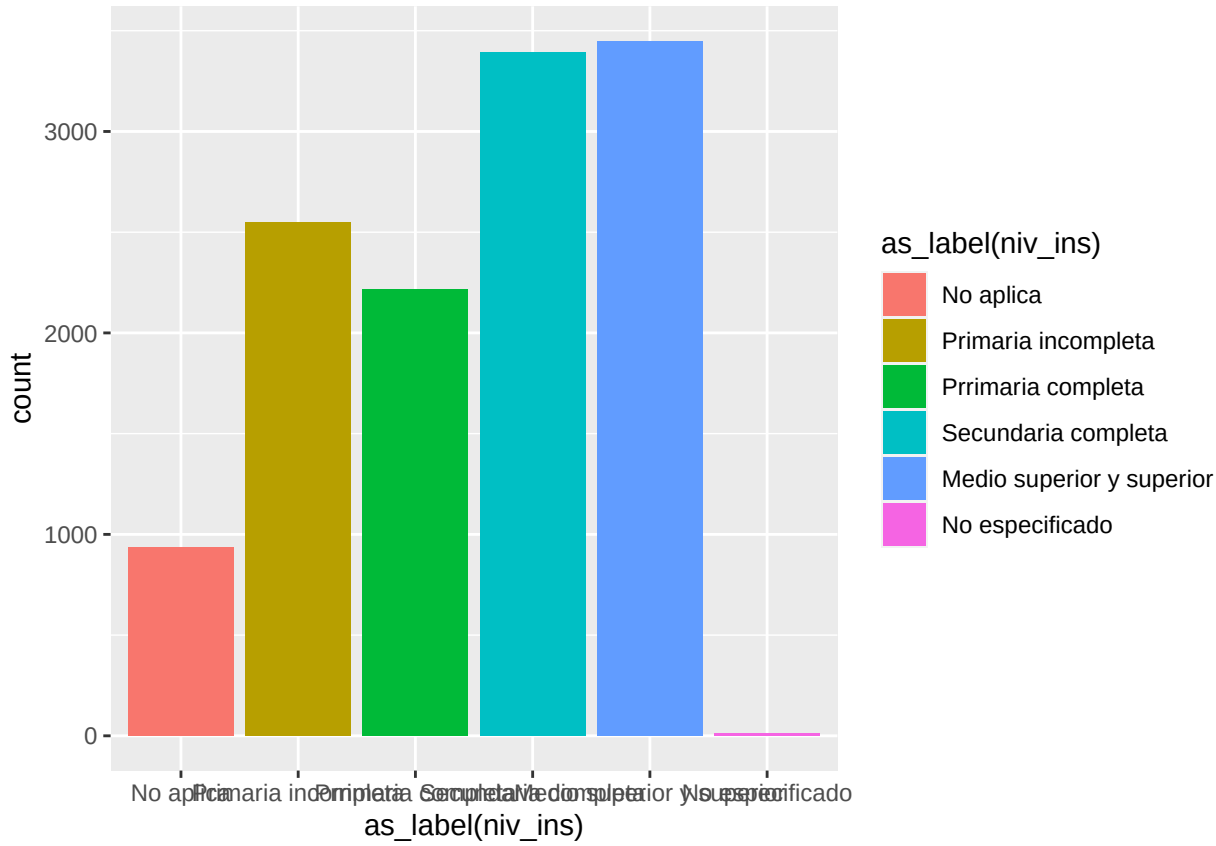
Gráficos univariados

Para cualitativas

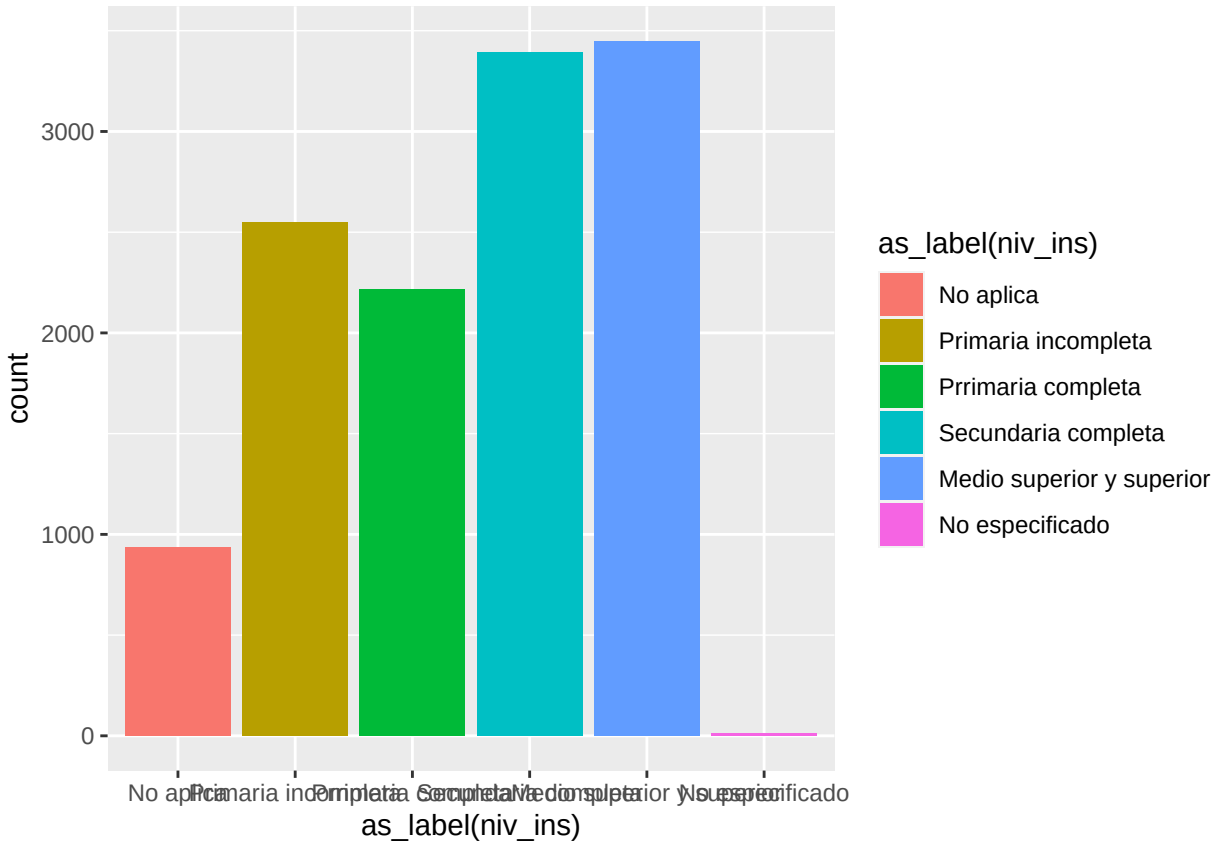
```
g1 + geom_bar()
```




```
g1 +  geom_bar(aes(
  fill = as_label(niv_ins)
  )) # colorea la geometría
```



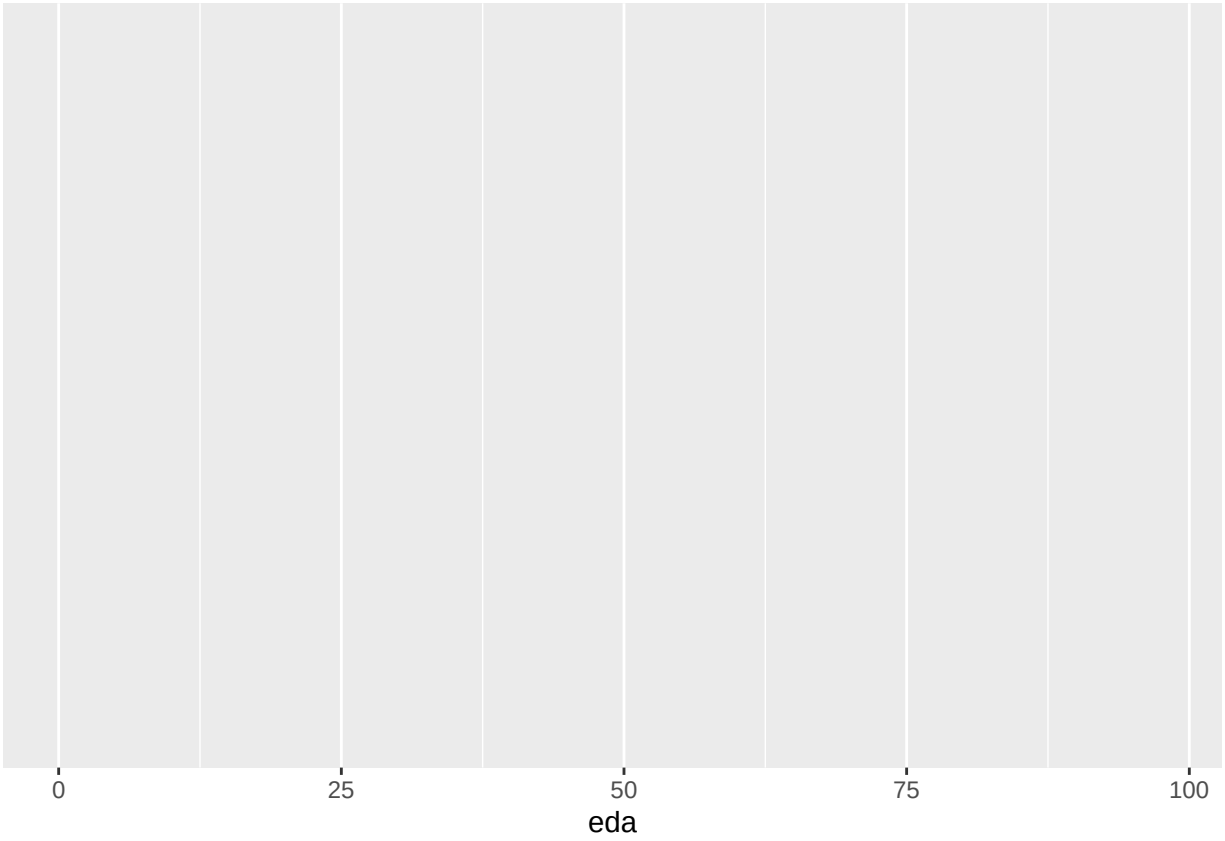

```
# Esto es equivalente

ags_t321 %>%
  ggplot(aes(as_label(niv_ins),
             fill = as_label(niv_ins)
             )
         ) + geom_bar()
```



Para variables cuantitativas

Podemos hacer histogramas y gráficos de densidad, de manera fácil. La idea es agregar en nuestro "lienzo" una geometría, un valor para dibujar en él. Esto se agrega con un "+" y con la figura que se añadirá a nuestro gráfico.

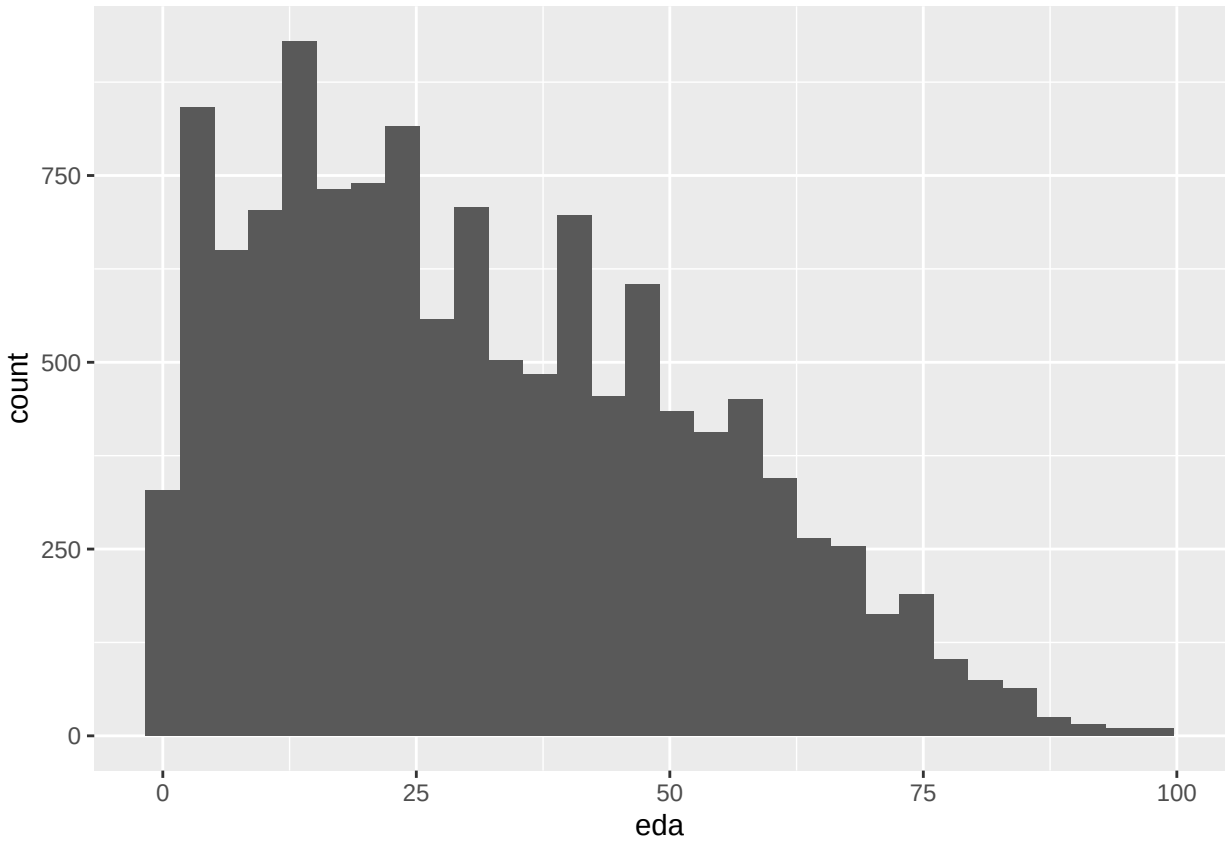
```
g2<-ags_t321 %>%
  ggplot(aes(eda))

g2 # imprime el lienzo
```



Histograma

```
g2 + geom_histogram()
```

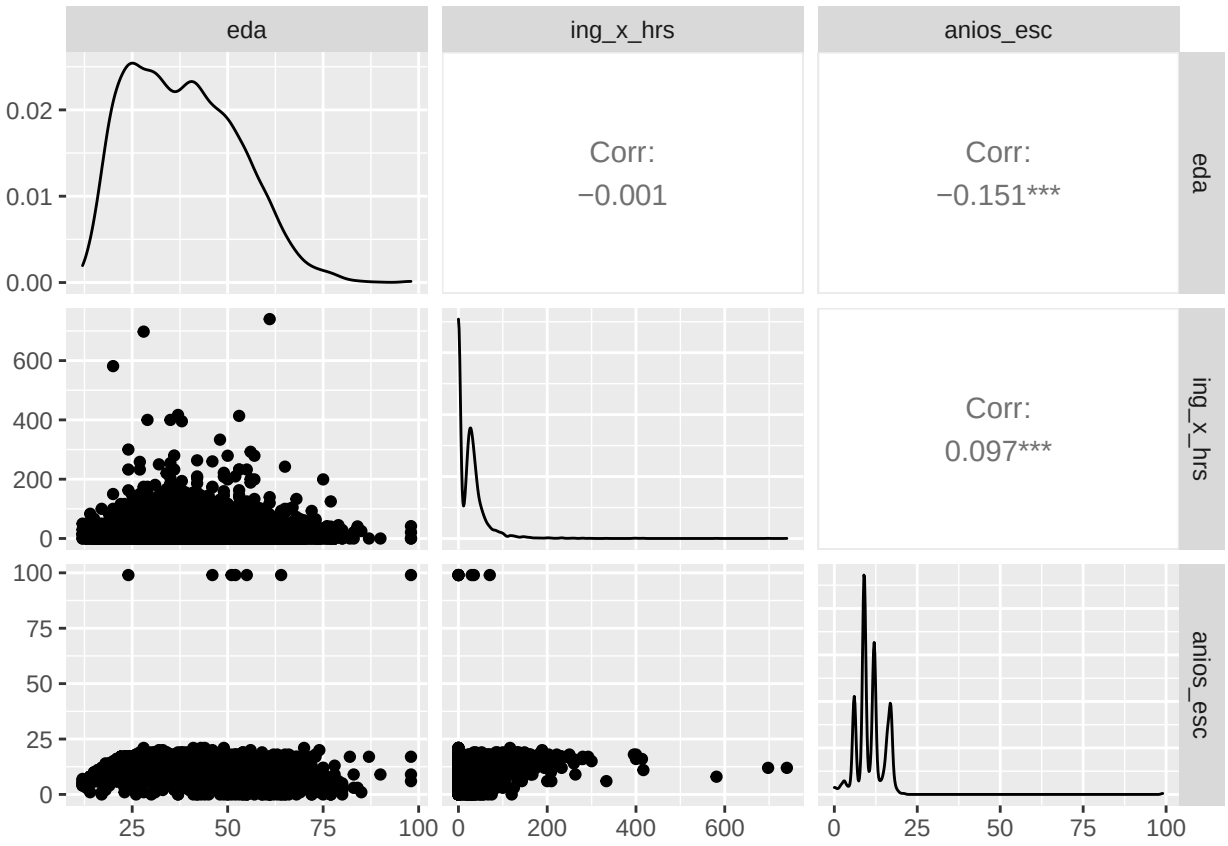
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Intro a dos variables

Vamos a terminar con un código que resume mucho de lo que hemos visto hoy:

```
ags_t321 %>%
  filter(clase2==1) %>%  # nos quedamos sólo con los ocupados
  select(eda, ing_x_hrs, anios_esc) %>%
  GGally::ggpairs()
```


eda
ing_x_hrs
anios_esc
Corr:
−0.001
Corr:
−0.151***
Corr:
0.097***
0.02
0.01
0.00
600
400
200
0
100
75
50
25
0
25
50
75
100
0
200
400
600
0
25
50
75
100