Inferencia e introducción a los modelos estadísticos con R

Curso intersemestral de Invierno - Posgrado de Ciencias Políticas y Sociales

Ana Escoto

1/9/23

Table of contents

Introducción al curso

Objetivo general

Que el estudiantado sea capaz de realizar inferencia estadística y modelado de una variable dependiente utilizando R aplicado a las bases de datos mexicanas.

Temas

1. Revisión de elementos estadísticos básicos desde "tidyverse"

a. Tablas de múltiples entradas

b. Repaso de ggplot2

** 2. Pruebas de hipótesis e intervalos de confianza **

a. De una sola media

b. De dos medias

c. Medias apareadas

d. Proporciones

e. Diferencia de proporciones

f. Chi-cuadrado de independencia

g. Prueba ANOVA de un solo factor

h. Pruebas no parámetricas

3. Factores de expansión y diseño muestral complejo

a. De “survey” a “srvyr”

b. Tabulados

c. Intervalos de confianza para medias y cuantiles

4. Introducción al modelo de regresión lineal

a. Simple

b. Múltiple

c. Evaluación de supuestos

** 5. Introducción a los modelos lineales generalizados **

a. Introducción a la regresión logística

b. Evaluación de supuestos

c. Efectos marginales

d. Interacciones y efectos de más de primer orden

Metodología

La metodología del curso consistirá en lo siguiente:

1. La exposición de la facilitadora. Durante la primera parte de la sesión, se expondrán los comandos necesarios para llevar a cabo cada tema. Se dará una introducción sobre la temática y se buscará dar ejemplos concretos para facilitar el aprendizaje. Se espera que el personal exponga sus dudas o comentarios a lo largo de la explicación.

2. Realización de ejercicios prácticos. Al final de cada sesión, corresponderá a las personas asistentes del curso realizar individualmente o en parejas un ejercicio relacionado con lo visto en la primera parte de la clase.

3. Consulta autónoma de material. Tanto la exposición como los ejercicios serán acompañado de material de consulta realizado ad hoc para el curso y el contenido, de tal manera que el estudiantado pueda volver a los códigos y las explicaciones posteriormente.

Facilitadora

Ana Ruth Escoto Castillo

Doctora en Estudios de Población. Centro de Estudios Demográficos y Urbanos, El Colegio de México.

Semblanza

Profesora de tiempo completo en la Facultad de Ciencias Políticas y Sociales. Investigadora nivel I en elSistema Nacional de Investigadores. Maestra en Población y Desarrollo por la Facultad Latinoamericana de Ciencias Sociales (FLACSO) – Sede México. Posee experiencia en recolección de información estadística, diseño y control de procesos de recolección y su procesamiento. Ha aplicado diversos métodos y herramientas multivariadas, homologación de información y comparabilidad de fuentes en sus investigaciones, así como usa de diversos softwares estadísticos, y ha impartido clases de estadítica aplicada a nivel de licenciatura y posgrado. Es co-coordinadora del Capítulo de CDMX de la iniciativa RLadies.

Instalación de R y Rstudio

Introducción a R

https://youtu.be/YkN5urybh2A Video en YouTube

Instalación en OS

https://youtu.be/icWV8jzYOtA Viedo en YouTube

Instalación en PC

https://youtu.be/TNSQikMfgJI Video en YouTube

Ojo

Pronto RStudio se volverá **"posit"**

Chapter 1

Introducción: {dplyr} y {ggplot2}

1.1 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)
```

1.2 Cargando los datos

Desde STATA

```
tlaxt322<- haven::read_dta("./datos/tlaxt322.dta")
```

Desde Excel:

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

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

1.3 Un poquito de {dplyr} y limpieza

1.3.1 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:

```
tlaxt322|> #pipe nativo, no necesita instalación
  head()
```

```
# A tibble: 6 x 114
  r_def           loc   mun   est est_d~1 est_d~2  ageb t_loc~3 t_loc~4 cd_a
  <dbl+lbl>     <dbl> <dbl> <dbl>   <dbl>   <dbl> <dbl> <dbl+l> <dbl+l> <dbl+lb>
1 0 [Entrevist~    NA    31    20     413      NA     0 2 [Loc~ NA      39 [Tla~
2 0 [Entrevist~    NA    31    20     413      NA     0 2 [Loc~ NA      39 [Tla~
3 0 [Entrevist~    NA    31    20     413      NA     0 2 [Loc~ NA      39 [Tla~
4 0 [Entrevist~    NA    31    20     413      NA     0 2 [Loc~ NA      39 [Tla~
5 0 [Entrevist~    NA    31    20     413      NA     0 2 [Loc~ NA      39 [Tla~
6 0 [Entrevist~    NA    31    20     413      NA     0 2 [Loc~ NA      39 [Tla~
# ... with 104 more variables: 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>, e_con <dbl+lbl>, ...
```

```
tlaxt322 %>%  #pipe de dplyr, necesita instalación de dplyr en tidyverse
  head()
```

```
# A tibble: 6 x 114
  r_def           loc   mun   est est_d~1 est_d~2  ageb t_loc~3 t_loc~4 cd_a
  <dbl+lbl>     <dbl> <dbl> <dbl>   <dbl>   <dbl> <dbl> <dbl+l> <dbl+l> <dbl+lb>
1 0 [Entrevist~    NA    31    20     413      NA     0 2 [Loc~ NA      39 [Tla~
2 0 [Entrevist~    NA    31    20     413      NA     0 2 [Loc~ NA      39 [Tla~
3 0 [Entrevist~    NA    31    20     413      NA     0 2 [Loc~ NA      39 [Tla~
4 0 [Entrevist~    NA    31    20     413      NA     0 2 [Loc~ NA      39 [Tla~
5 0 [Entrevist~    NA    31    20     413      NA     0 2 [Loc~ NA      39 [Tla~
6 0 [Entrevist~    NA    31    20     413      NA     0 2 [Loc~ NA      39 [Tla~
# ... with 104 more variables: 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>, e_con <dbl+lbl>, ...
```

1.3.2 Limpieza de nombres con {janitor}

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(tlaxt322)
```

```
  [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:

```
tlaxt322%<>%
  clean_names()

names(tlaxt322)
```

```
 [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

1.4 `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:

```
tlaxt322%>%
  dplyr::select(sex, eda) %>%
  dplyr::filter(eda>11)
```

```
# A tibble: 9,205 x 2
   sex          eda
   <dbl+lbl>  <dbl>
 1 1 [Hombre]    34
 2 2 [Mujer]     57
 3 1 [Hombre]    71
 4 1 [Hombre]    67
 5 2 [Mujer]     60
 6 2 [Mujer]     35
 7 1 [Hombre]    39
 8 1 [Hombre]    38
 9 2 [Mujer]     33
10 1 [Hombre]    14
# ... with 9,195 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:

```
tlaxt322%<>%
  filter(r_def==0) %>%
```

```
  filter(!c_res==2)
```

1.5 Tabulados con `tabyl()`

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

```
tlaxt322%>%
  dplyr::mutate(sex=sjlabelled::as_label(sex))  %>%
  janitor::tabyl(sex)
```

```
   sex    n   percent
Hombre 5392 0.4767041
 Mujer 5919 0.5232959
```

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":

```
tlaxt322%>%
  mutate(sex=as_label(sex)) %>%
  tabyl(sex) %>%
  adorn_totals() #primer enchulamiento
```

```
   sex     n   percent
Hombre  5392 0.4767041
 Mujer  5919 0.5232959
 Total 11311 1.0000000
```

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

```
tlaxt322%>%
  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  5392   47.7%
```

```
Mujer  5919   52.3%
Total 11311  100.0%
```

Vamos a darle una "ojeada" a esta variable

```
glimpse(tlaxt322$niv_ins)
```

```
dbl+lbl [1:11311] 4, 2, 3, 3, 3, 4, 2, 3, 4, 2, 2, 2, 2, 3, 3, 2, 3, 2, 2,...
@ 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" "Se
```

Hoy hacemos la tabla, con las etiquetas:

```
tlaxt322%>%
  mutate(niv_ins=as_label(niv_ins)) %>% #esto sólo si hay etiquetas declaradas, recuerda
  tabyl(niv_ins)
```

```
                niv_ins    n      percent
              No aplica  714 0.0631243922
     Primaria incompleta 2173 0.1921138715
      Prrimaria completa 2025 0.1790292635
     Secundaria completa 3201 0.2829988507
Medio superior y superior 3190 0.2820263460
         No especificado    8 0.0007072761
```

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

```
tlaxt322%>%
  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   714 0.0631243922
     Primaria incompleta  2173 0.1921138715
      Prrimaria completa  2025 0.1790292635
     Secundaria completa  3201 0.2829988507
Medio superior y superior  3190 0.2820263460
         No especificado     8 0.0007072761
                   Total 11311 1.0000000000
```

1.5.1 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. Básicamente 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:

```
tlaxt322%>%
  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    389   325
     Primaria incompleta   1066  1107
      Prrimaria completa    950  1075
     Secundaria completa   1450  1751
Medio superior y superior  1533  1657
         No especificado      4     4
                   Total   5392  5919
```

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

```
tlaxt322%>%
    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                     10245
2 TRUE                       1066
```

1.5.2 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.

```
tlaxt322%>%
  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    389   325   714
   Primaria incompleta   1066  1107  2173
    Prrimaria completa    950  1075  2025
   Secundaria completa   1450  1751  3201
Medio superior y superior   1533  1657  3190
        No especificado      4     4     8
```

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

```
tlaxt322%>%
  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    389   325   714
   Primaria incompleta   1066  1107  2173
    Prrimaria completa    950  1075  2025
   Secundaria completa   1450  1751  3201
Medio superior y superior   1533  1657  3190
        No especificado      4     4     8
                  Total   5392  5919 11311
```

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:

```
tlaxt322%>%
  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   7.2%   5.5%   6.3%
     Primaria incompleta  19.8%  18.7%  19.2%
      Prrimaria completa  17.6%  18.2%  17.9%
     Secundaria completa  26.9%  29.6%  28.3%
Medio superior y superior  28.4%  28.0%  28.2%
         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:

```
tlaxt322%>%
  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  54.5% 45.5% 100.0%
     Primaria incompleta  49.1% 50.9% 100.0%
      Prrimaria completa  46.9% 53.1% 100.0%
     Secundaria completa  45.3% 54.7% 100.0%
Medio superior y superior  48.1% 51.9% 100.0%
         No especificado  50.0% 50.0% 100.0%
                   Total  47.7% 52.3% 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.

```
tlaxt322%>%
  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   3.4%  2.9%   6.3%
     Primaria incompleta   9.4%  9.8%  19.2%
      Prrimaria completa   8.4%  9.5%  17.9%
     Secundaria completa  12.8% 15.5%  28.3%
Medio superior y superior  13.6% 14.6%  28.2%
         No especificado   0.0%  0.0%   0.1%
                   Total  47.7% 52.3% 100.0%
```

1.6 *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<-tlaxt322%>%
  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	3.4%	2.9%	6.3%
Primaria incompleta	9.4%	9.8%	19.2%
Prrimaria completa	8.4%	9.5%	17.9%
Secundaria completa	12.8%	15.5%	28.3%

Medio superior y superior	13.6%	14.6%	28.2%
No especificado	0.0%	0.0%	0.1%
Total	47.7%	52.3%	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 = "Tlaxcala, trimestre III de 2022"
  )

gt_tabla
```

Distribución del sexo de la población según nivel de escolaridad
Tlaxcala, trimestre III de 2022

niv_ins	Hombre	Mujer	Total
No aplica	3.4%	2.9%	6.3%
Primaria incompleta	9.4%	9.8%	19.2%
Prrimaria completa	8.4%	9.5%	17.9%
Secundaria completa	12.8%	15.5%	28.3%
Medio superior y superior	13.6%	14.6%	28.2%
No especificado	0.0%	0.0%	0.1%
Total	47.7%	52.3%	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
Tlaxcala, trimestre III de 2022

niv_ins	Hombre	Mujer	Total
No aplica	3.4%	2.9%	6.3%
Primaria incompleta	9.4%	9.8%	19.2%
Prrimaria completa	8.4%	9.5%	17.9%

Secundaria completa	12.8%	15.5%	28.3%
Medio superior y superior	13.6%	14.6%	28.2%
No especificado	0.0%	0.0%	0.1%
Total	47.7%	52.3%	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

1.7 Descriptivos para variables cuantitativas

Vamos a empezar a revisar los gráficos para variables cuantitativas.

1.7.1 Medidas numéricas básicas

5 números

```
summary(tlaxt322$ing_x_hrs) ## ingreso por horas
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00    0.00    0.00   12.10   18.75 1356.59
```

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

```
tlaxt322%>%
  summarise(nombre_indicador=mean(ing_x_hrs, na.rm=T))
```

```
# A tibble: 1 x 1
  nombre_indicador
             <dbl>
1             12.1
```