Taller_analisis_de_datos_pesos

Joshua Kock

4/26/2019

Cargar paquetes

```
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.1.1       v purrr   0.3.2
## v tibble  2.1.1       v dplyr   0.8.0.1
## v tidyr   0.8.3       v stringr 1.4.0
## v readr   1.3.1       v forcats 0.4.0

## -- Conflicts ------------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(readxl)
library(infer)
library(broom)
library(pander)

theme_set(theme_light())
```

```
pesos_col <- read_csv("https://raw.githubusercontent.com/vizual-wanderer/6071402_Electiva_II/master/Bas
```

```
## Parsed with column specification:
## cols(
##   sexo = col_double(),
##   peso_nac = col_double(),
##   talla_na = col_double(),
##   numconsu = col_double(),
##   tipo_par = col_double(),
##   mul_part = col_double(),
##   seg_soci = col_double(),
##   edad_mad = col_double(),
##   est_civm = col_double(),
##   niv_edum = col_double(),
##   area_res = col_double(),
##   n_hijosv = col_double(),
##   n_emb = col_double(),
##   edad_pad = col_double(),
##   niv_edup = col_double()
## )
```

verificar numero de datos peridos y el nombre de las variables

```r
map_df(pesos_col, ~sum(is.na(.)))
```

```
## # A tibble: 1 x 15
##    sexo peso_nac talla_na numconsu tipo_par mul_part seg_soci edad_mad
##   <int>    <int>    <int>    <int>    <int>    <int>    <int>    <int>
## 1     0        0        0        0        0        0        0        0
## # ... with 7 more variables: est_civm <int>, niv_edum <int>,
## #   area_res <int>, n_hijosv <int>, n_emb <int>, edad_pad <int>,
## #   niv_edup <int>
```

```r
names(pesos_col)
```

```
##  [1] "sexo"     "peso_nac" "talla_na" "numconsu" "tipo_par" "mul_part"
##  [7] "seg_soci" "edad_mad" "est_civm" "niv_edum" "area_res" "n_hijosv"
## [13] "n_emb"    "edad_pad" "niv_edup"
```

```r
str(pesos_col)
```

```
## Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame': 2998 obs. of  15 variables:
##  $ sexo    : num  1 1 1 2 2 1 1 1 2 1 ...
##  $ peso_nac: num  2450 2390 1640 2160 2400 ...
##  $ talla_na: num  47 47 43 45 47 44 39 44 48 46 ...
##  $ numconsu: num  8 3 2 5 2 6 6 6 4 8 ...
##  $ tipo_par: num  2 2 2 2 1 2 2 2 1 2 ...
##  $ mul_part: num  2 1 1 2 1 1 1 1 1 1 ...
##  $ seg_soci: num  1 1 1 2 1 3 1 4 2 1 ...
##  $ edad_mad: num  36 33 33 25 36 23 26 22 21 37 ...
##  $ est_civm: num  2 4 4 4 2 1 4 2 4 2 ...
##  $ niv_edum: num  2 4 2 3 6 4 5 4 2 5 ...
##  $ area_res: num  3 1 1 3 3 1 1 3 3 1 ...
##  $ n_hijosv: num  4 1 2 4 2 1 1 1 1 3 ...
##  $ n_emb   : num  3 1 2 4 2 1 2 1 1 4 ...
##  $ edad_pad: num  42 67 35 32 36 34 33 29 20 37 ...
##  $ niv_edup: num  4 2 3 2 5 2 4 2 2 4 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   sexo = col_double(),
##   ..   peso_nac = col_double(),
##   ..   talla_na = col_double(),
##   ..   numconsu = col_double(),
##   ..   tipo_par = col_double(),
##   ..   mul_part = col_double(),
##   ..   seg_soci = col_double(),
##   ..   edad_mad = col_double(),
##   ..   est_civm = col_double(),
##   ..   niv_edum = col_double(),
##   ..   area_res = col_double(),
##   ..   n_hijosv = col_double(),
##   ..   n_emb = col_double(),
##   ..   edad_pad = col_double(),
##   ..   niv_edup = col_double()
##   .. )
```

Si verificamos el codebook vemos que tenemos 7 de las 15 variables como categoricas las cuales las podemos converitr en factores para hacer los analisis en R.

- Tipo de parto: `tipo_par`
- Multiplicidad del parto: `mul_part`
- Regimen de seguridad social: `seg_soci`
- Escolaridad Padre y madre: `niv_edum` y `niv_edup`
- Estado civil: `est_civm`
- Area de residencia: `area_res`

Hay que verificar cuales son los valores numerico que tienen cada variable y asi podemos proceder a convertir en factores.

```
vars_fact <- c("sexo", "tipo_par", "mul_part","seg_soci", "niv_edum", "niv_edup","est_civm", "area_res"

pesos_col %>%
  select(vars_fact) %>%
  mutate_if(is.numeric, as.factor) %>%
  tbl_df() %>%
  summary()
```

```
##  sexo     tipo_par mul_part seg_soci    niv_edum      niv_edup   est_civm
##  1:1499   1:1981   1:2845   1:1093   5      :902   2      :725   1: 470
##  2:1499   2: 949   2: 150   2:1136   2      :796   5      :663   2: 760
##           3:  49   3:   3   3: 695   4      :608   4      :590   3:   9
##           4:   2            4:  74   3      :422   3      :438   4:1678
##           9:  17                      6      :135   9      :328   5:  18
##                                       7      : 99   6      :118   9:  63
##                                       (Other): 36   (Other):136
##  area_res
##  1:1989
##  2: 281
##  3: 728
##
##
##
##
```

Conversion de las variables a factores no quiero borrar la base original asi que asigno este cambio a un nuevo objeto.

```
base_pesos_col <- pesos_col %>%
  mutate(
    sexo = recode_factor(sexo, `1` = "Femenino", `2` = "Masculino"),
    tipo_par = recode_factor(tipo_par, `1` = "Espontaneo", `2` = "Cesarea", `3` = "Instrumentado", `4` =
                             `9` = "Ignorado"),
    mul_part = recode_factor(mul_part, `1` = "Simple", `2` = "Doble", `3` = "Triple", `4` = "Cuadruple_
    seg_soci = recode_factor(seg_soci, `1` = "Contributivo", `2` = "Subsidiado", `3` = "Excepcion", `4`
                         `5` = "No_asegurado", .default = NULL),
    est_civm = recode_factor(est_civm, `1` = "Union_libre_mas_dos", `2` = "Union_libre_menos_dos", `3` =
                         `5` = "Soltera", `6` = "Casada", `9` = "Ignorado", .default = NULL),
    niv_edum = recode_factor(niv_edum, `1` = "Preescolar", `2` = "Basica_primaria", `3` = "Basica_secun
```

```
                        `5` =   "Media_tecnica", `6` = "Normalista", `7` = "Tecnica_profesional", `8` =
                        `9` = "Profesional", `10` = "Especializacion", `11` = "Maestria", `12` = "Docto
                        `99` =  "Sin_informacion", .default = NULL),
    niv_edup = recode_factor(niv_edup, `1` = "Preescolar", `2` = "Basica_primaria", `3` = "Basica_secun
                        `5` =   "Media_tecnica", `6` = "Normalista", `7` = "Tecnica_profesional", `8` =
                        `9` = "Profesional", `10` = "Especializacion", `11` = "Maestria", `12` = "Docto
                        `99` =  "Sin_informacion", .default = NULL),
    area_res = recode_factor(area_res, `1` = "Cabecera_municipal", `2` = "Centro_poblado", `3` =    "Ru
    )

#ver la base de datos
base_pesos_col %>%
  view()

#ver datos perdidos:
map_df(base_pesos_col, ~sum(is.na(.)))
```

```
## # A tibble: 1 x 15
##    sexo peso_nac talla_na numconsu tipo_par mul_part seg_soci edad_mad
##   <int>    <int>    <int>    <int>    <int>    <int>    <int>    <int>
## 1     0        0        0        0        0        0        0        0
## # ... with 7 more variables: est_civm <int>, niv_edum <int>,
## #   area_res <int>, n_hijosv <int>, n_emb <int>, edad_pad <int>,
## #   niv_edup <int>
```

Inicio de analisis

Preguntas a responder

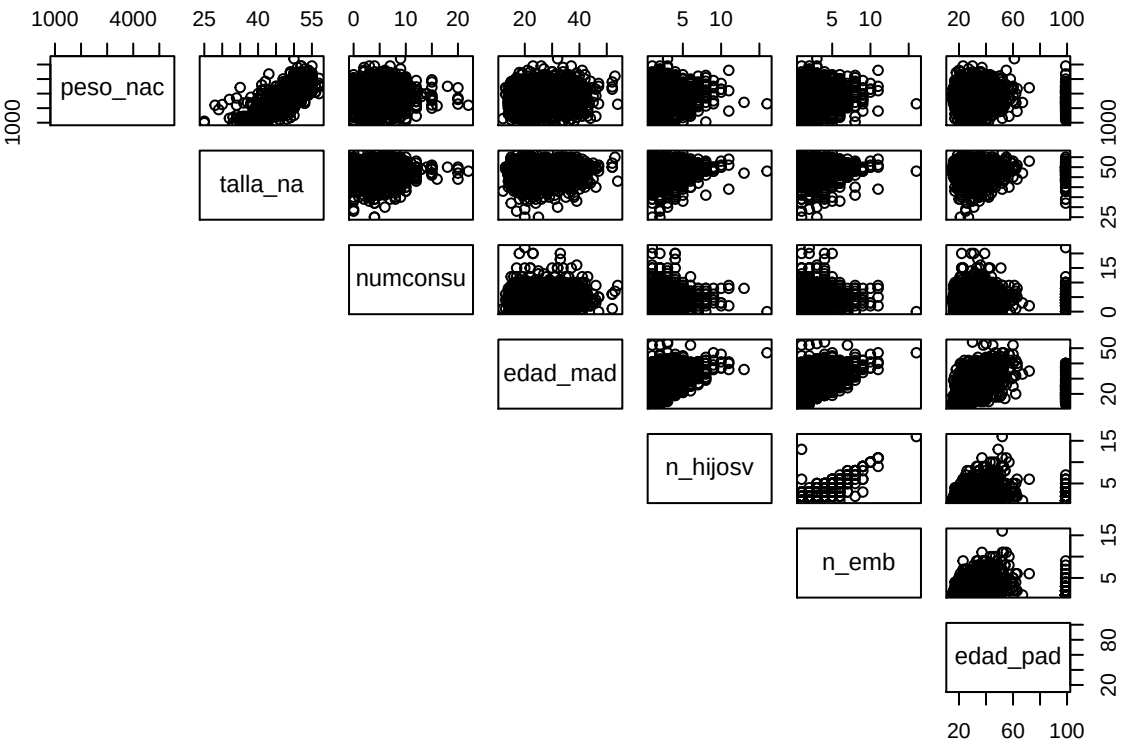
```
summary(base_pesos_col)
```

```
##         sexo         peso_nac       talla_na        numconsu
##  Femenino :1499   Min.   :1000   Min.   :25.00   Min.   : 0.000
##  Masculino:1499   1st Qu.:2400   1st Qu.:47.00   1st Qu.: 3.000
##                   Median :2900   Median :49.00   Median : 5.000
##                   Mean   :2842   Mean   :48.26   Mean   : 5.014
##                   3rd Qu.:3300   3rd Qu.:50.00   3rd Qu.: 7.000
##                   Max.   :5400   Max.   :57.00   Max.   :22.000
##
##           tipo_par      mul_part            seg_soci       edad_mad
##  Espontaneo   :1981   Simple:2845   Contributivo:1093   Min.   :12.00
##  Cesarea      : 949   Doble : 150   Subsidiado  :1136   1st Qu.:20.00
##  Instrumentado:  49   Triple:   3   Excepcion   : 695   Median :24.00
##  Ignorado     :  19                 Especial    :  74   Mean   :25.31
##                                                         3rd Qu.:30.00
##                                                         Max.   :54.00
##
##                   est_civm                   niv_edum
##  Union_libre_mas_dos  : 470   Media_tecnica      :902
##  Union_libre_menos_dos: 760   Basica_primaria    :796
##  Separada             :   9   Media_academica    :608
##  Viuda                :1678   Basica_secundaria  :422
##  Soltera              :  18   Normalista         :135
```

```
##  Ignorado             :  63   Tecnica_profesional: 99
##                               (Other)            : 36
##                area_res       n_hijosv        n_emb           edad_pad
##  Cabecera_municipal:1989   Min.   : 1.0   Min.   : 1.000   Min.   :14.0
##  Centro_poblado    : 281   1st Qu.: 1.0   1st Qu.: 1.000   1st Qu.:24.0
##  Rural_disperso    : 728   Median : 2.0   Median : 2.000   Median :29.0
##                            Mean   : 2.3   Mean   : 2.431   Mean   :33.1
##                            3rd Qu.: 3.0   3rd Qu.: 3.000   3rd Qu.:36.0
##                            Max.   :16.0   Max.   :16.000   Max.   :99.0
##
##               niv_edup
##  Basica_primaria  :725
##  Media_tecnica    :663
##  Media_academica  :590
##  Basica_secundaria:438
##  Profesional      :328
##  Normalista       :118
##  (Other)          :136
```

Ver la relacion que tienen las variables numericas. selecionando unicamente las variables que son numericas.

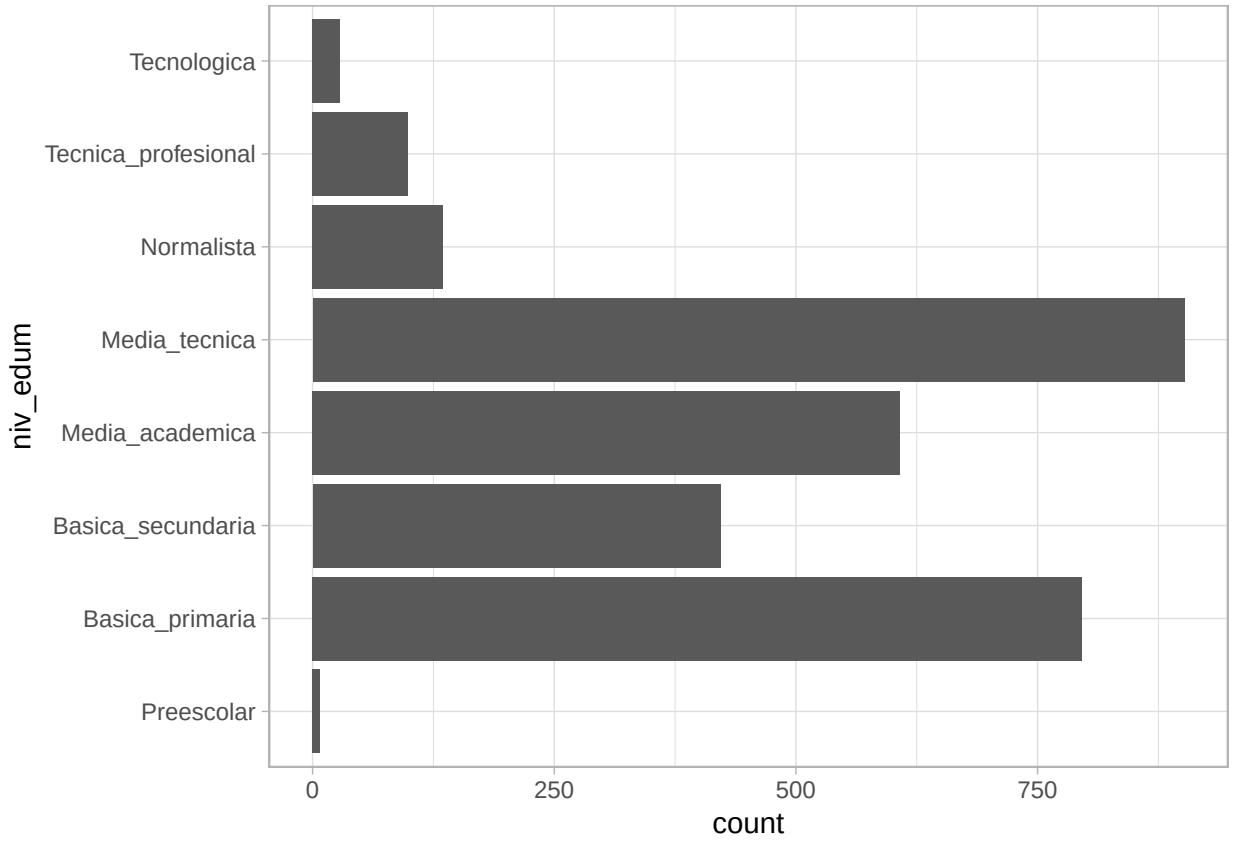
```
base_pesos_col %>%
  select_if(is.numeric) %>%
  pairs(lower.panel = NULL)
```



podemos visualizar el nivel educativo

```
base_pesos_col %>%
  ggplot(aes(niv_edum)) +
```

```
  geom_bar() +
  coord_flip()
```



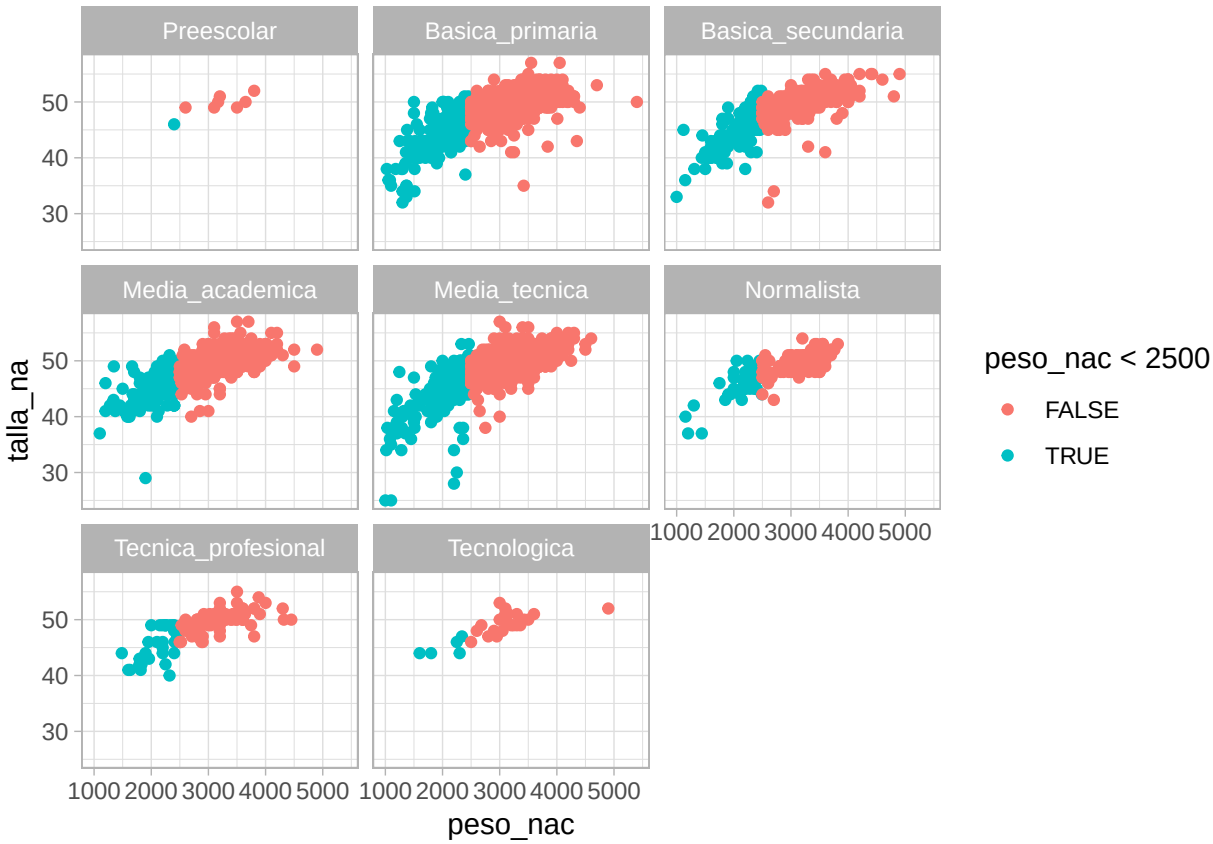
porcentajes de conteos: seguridad social de las madres.

```
base_pesos_col %>%
  count(seg_soci) %>%
  mutate(
    prop = prop.table(n),
    porcentaje = prop*100)
```

```
## # A tibble: 4 x 4
##   seg_soci         n   prop porcentaje
##   <fct>        <int>  <dbl>      <dbl>
## 1 Contributivo  1093 0.365       36.5
## 2 Subsidiado    1136 0.379       37.9
## 3 Excepcion      695 0.232       23.2
## 4 Especial        74 0.0247       2.47
```

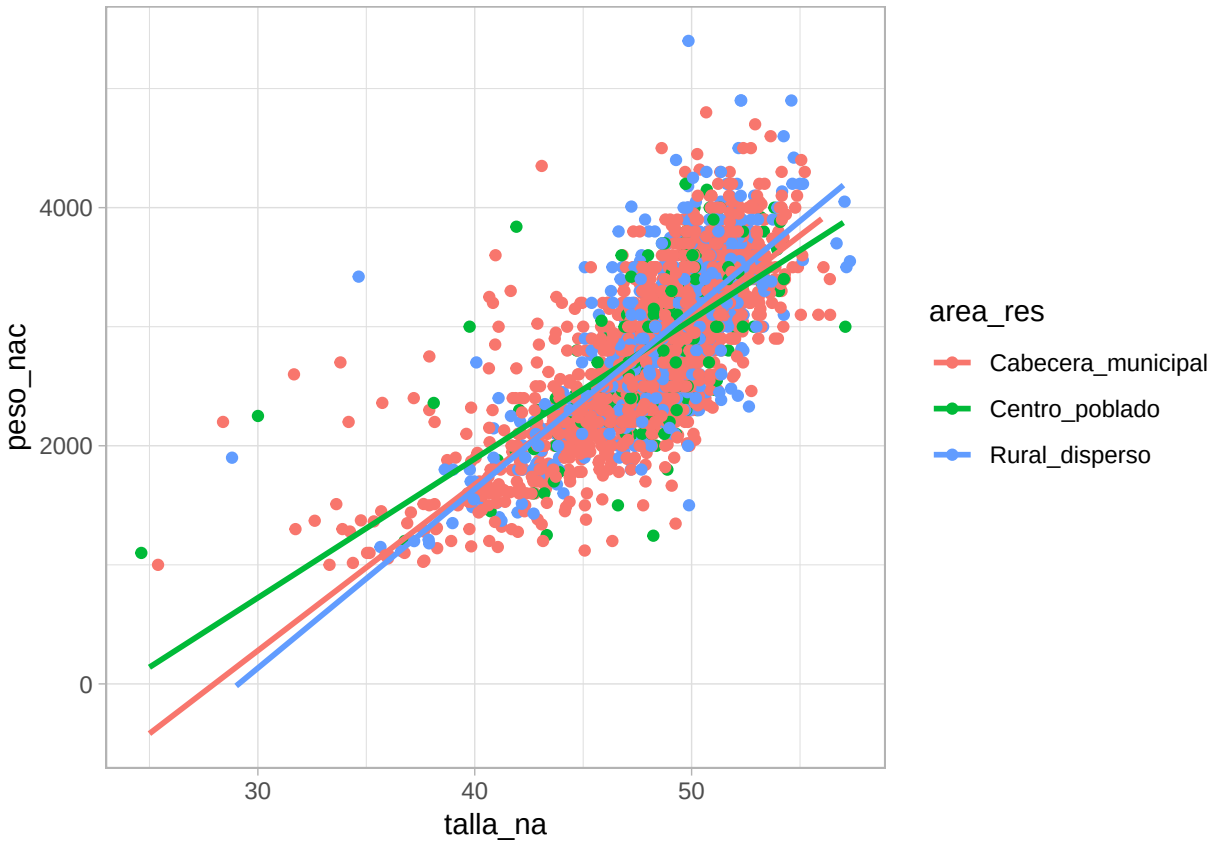
Me gustaria visualizar la distribucion de pesos y talla segun escolaridad de madre identificando si hay recien nacidos con mayor bajo peso al nacer

```
base_pesos_col %>%
  ggplot(aes(x = peso_nac, talla_na, color = peso_nac < 2500)) +
  geom_point() +
  facet_wrap(~niv_edum)
```



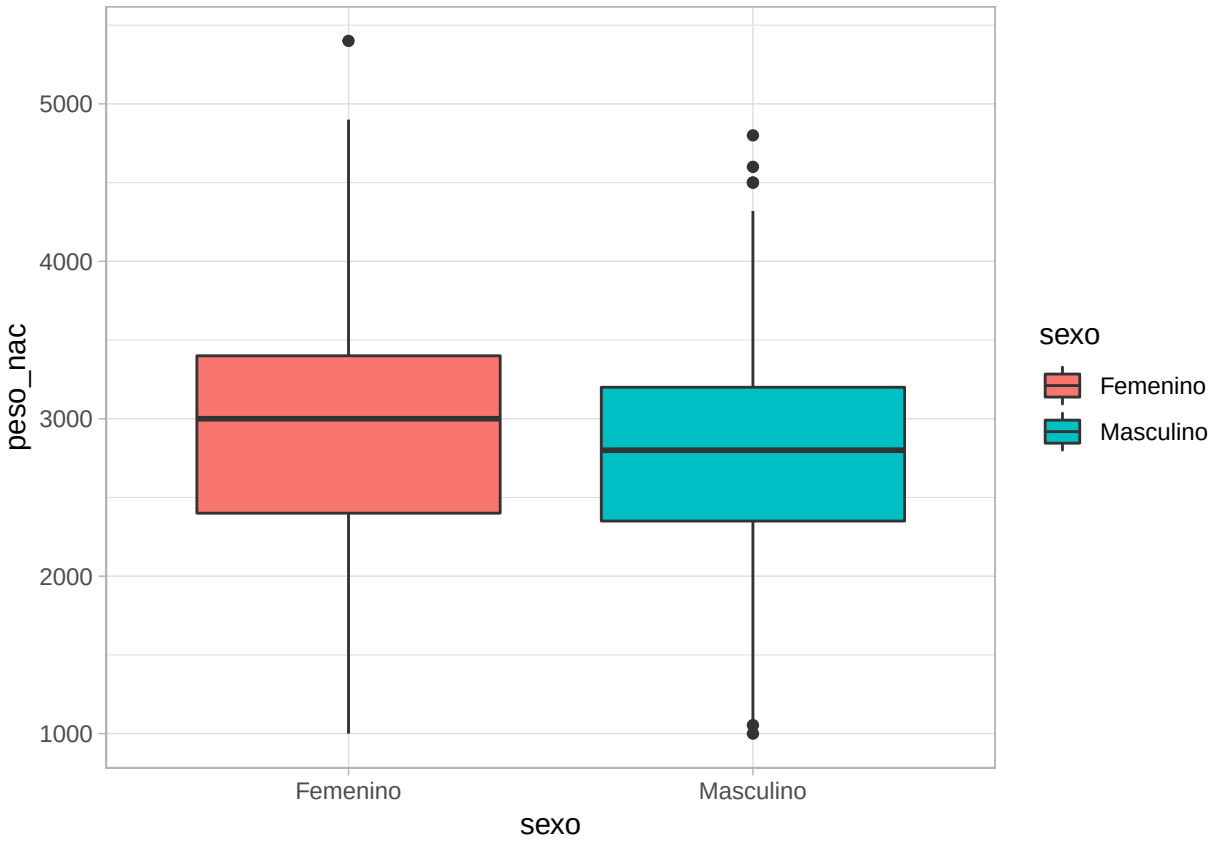
Quiero ver la talla y peso al nacer por area geografica

```
base_pesos_col %>%
  ggplot(aes(talla_na, peso_nac, color = area_res)) +
  geom_jitter() +
  geom_smooth(method = "lm", se = FALSE)
```



Diferencia de peso al nacer comparado con el sexo biologico del recien nacido

```
base_pesos_col %>%
  ggplot(aes(sexo, peso_nac,fill = sexo)) +
  geom_boxplot()
```



quiero ver diferecias entre el peso de los recien nacido por sexo al nacer.

```
t_test(base_pesos_col, peso_nac ~ sexo, order = c("Femenino", "Masculino"), alternative = "two_sided")
```

```
## # A tibble: 1 x 6
##   statistic  t_df       p_value alternative lower_ci upper_ci
##       <dbl> <dbl>         <dbl> <chr>          <dbl>    <dbl>
## 1      5.80 2969. 0.00000000750 two.sided       88.4     179.
```

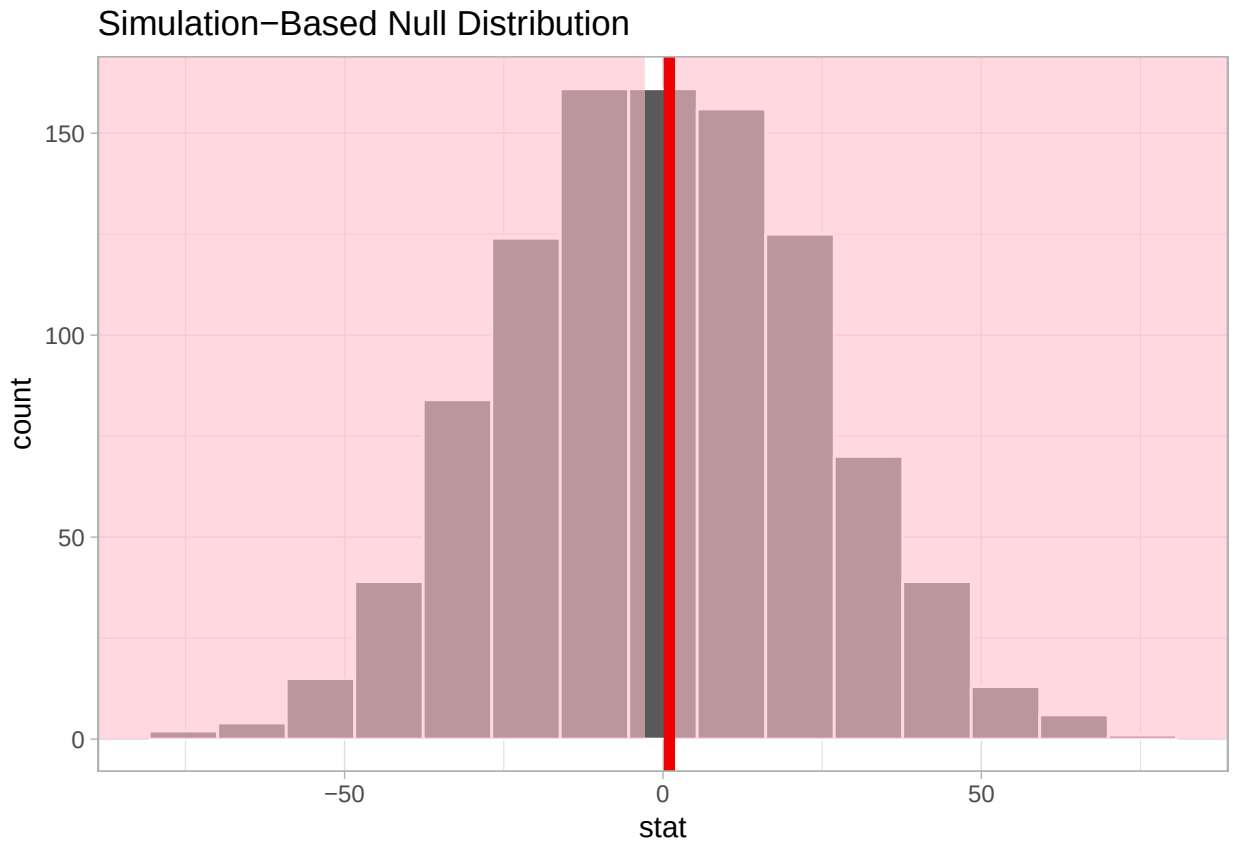
```
#metodo computacional:
est_obs_peso <- base_pesos_col %>%
  specify(peso_nac ~ sexo) %>%
  calculate(stat = "diff in means", order = c("Femenino", "Masculino")) %>%
  pull()

dist_nula <- base_pesos_col %>%
  specify(peso_nac ~ sexo) %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%
  calculate(stat = "diff in means", order = c("Femenino", "Masculino"))

dist_nula %>%
  get_p_value(est_obs_peso, direction = "two_sided")
```

```
## # A tibble: 1 x 1
##   p_value
```

```
##      <dbl>
## 1        0
```

```
visualize(dist_nula) +
  shade_p_value(dist_nula, direction = "two_sided")
```

```
## Warning: The first row and first column value of the given `obs_stat` will
## be used.
```




```
#intervalo de confianza:
ic_peso <- base_pesos_col %>%
  specify(peso_nac ~ sexo) %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "diff in means", order = c("Femenino", "Masculino"))

get_confidence_interval(ic_peso)
```

```
## # A tibble: 1 x 2
##   `2.5%` `97.5%`
##    <dbl>   <dbl>
## 1   88.1    179.
```

El tipo de parto esta relacionado con el numero de consultas pre-natales?

```r
tabla_med_con_par <- base_pesos_col %>%
  group_by(tipo_par) %>%
  summarize(
    cons_pro = mean(numconsu),
    cons_ds = sd(numconsu),
    cons_max = max(numconsu),
    cons_n = n())

tabla_med_con_par
```

```
## # A tibble: 4 x 5
##   tipo_par      cons_pro cons_ds cons_max cons_n
##   <fct>            <dbl>   <dbl>    <dbl>  <int>
## 1 Espontaneo        4.86    2.64       20   1981
## 2 Cesarea           5.36    2.81       22    949
## 3 Instrumentado     4.90    2.23       10     49
## 4 Ignorado          4.21    2.30        8     19
```

Prueba estadistica a aplicar para determinar diferencia de medias (variable continua y categorica con > 2 categorias) Podemos contestar la pregunta anterior con la prueba de ANOVA (Analysis of variance), el cual determina si la diferencia en medias de mi poblacion son "ciertas" o son por variablidad de muestreo. Sacoms esto con el calculo del estaditico F.

```r
estat_F <- base_pesos_col %>%
  specify(numconsu ~ tipo_par) %>%
  calculate(stat = "F") %>%
  pull()

dist_nula_F <- base_pesos_col %>%
  specify(numconsu ~ tipo_par) %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%
  calculate(stat = "F")

dist_nula_F %>%
  get_p_value(estat_F, direction = "greater")
```

```
## # A tibble: 1 x 1
##   p_value
##     <dbl>
## 1       0
```

```r
#la prueba de anova se have con la funcion de aov no la anova, la anova solo toma como argumentos objet
aov_con_par <- aov(numconsu ~ tipo_par, data = base_pesos_col)
summary(aov_con_par)
```

```
##               Df Sum Sq Mean Sq F value  Pr(>F)
## tipo_par       3    177   58.87   8.153 2.1e-05 ***
## Residuals   2994  21617    7.22
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Podemos ver que el estadistico F es de 8.15 el mismo que identificamos en el ejercicio anterior.

Recordar como se lleva a cabo la prueba de hipotesis de anova.

Vemos que nuestro valor p es menor al punto de corte de 0.05 por lo que rechazamos la hipotesis nula, sabemos que hay una diferencia entre medias de los grupos pero no sabemos cual.

Podemos aplicar varias pruebas el ejemplo lo hacemo con el test de Tuley HSD (Tukey Honest Significant Differences) el cual ejecuta una comparacion multiple por pares (pairwise-comparison) entre las medias de cada grupo.

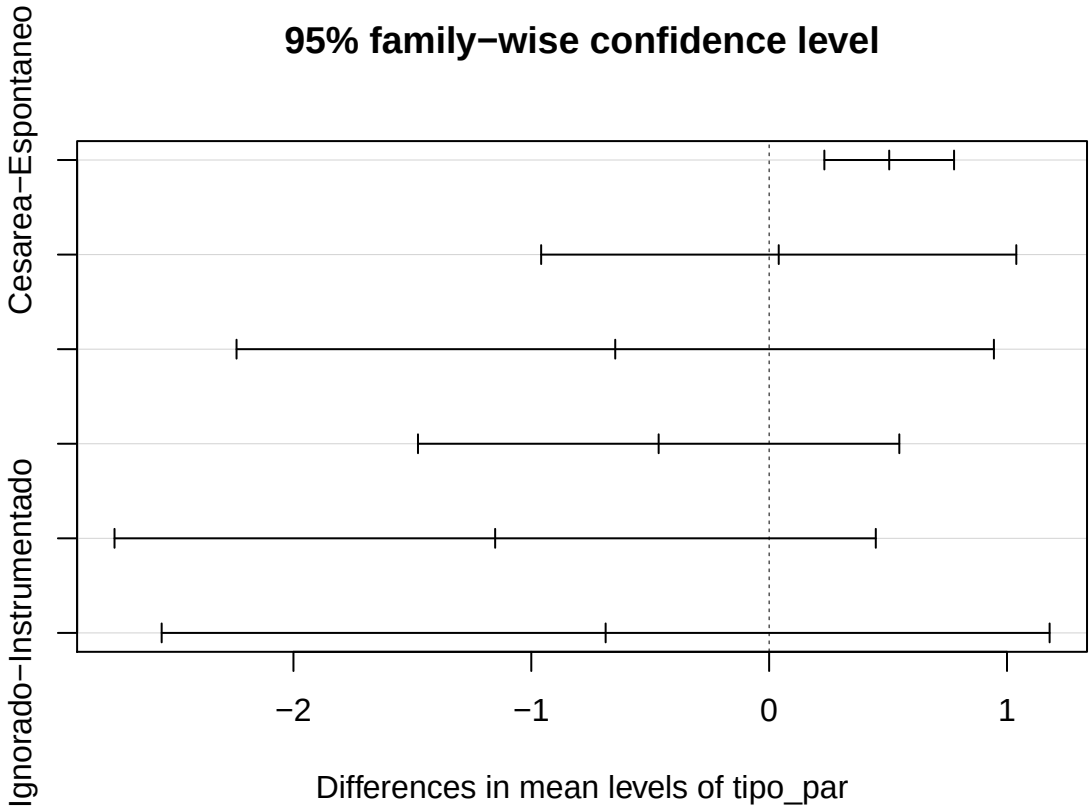
```
TukeyHSD(aov_con_par)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = numconsu ~ tipo_par, data = base_pesos_col)
##
## $tipo_par
##                                 diff        lwr       upr     p adj
## Cesarea-Espontaneo         0.50483918  0.2321664 0.7775119 0.0000121
## Instrumentado-Espontaneo   0.04031153 -0.9585172 1.0391402 0.9996005
## Ignorado-Espontaneo       -0.64712134 -2.2392539 0.9450112 0.7230649
## Instrumentado-Cesarea     -0.46452764 -1.4763806 0.5473253 0.6395197
## Ignorado-Cesarea          -1.15196051 -2.7522960 0.4483750 0.2500638
## Ignorado-Instrumentado    -0.68743287 -2.5540828 1.1792171 0.7796245
```

con los resultado vemos que la unica diferencia de medias es entre los grupos de cesaria y espontaneo.

De igual forma podemos ver esta tabla de forma grafica aplicando la funcion `plot()` al objeto.

```
plot(TukeyHSD(aov_con_par))
```



Quiero ver si el area de residencia esta asociado con el tipo de parto, mi hipotesis: No hay diferencias del tipo de parto segun zona geografica del parto.

```
base_pesos_col %>%
  count(tipo_par, area_res) %>%
  spread(area_res, n)
```

```
## # A tibble: 4 x 4
##   tipo_par      Cabecera_municipal Centro_poblado Rural_disperso
##   <fct>                      <int>          <int>          <int>
## 1 Espontaneo                  1259            200            522
## 2 Cesarea                      680             72            197
## 3 Instrumentado                 37              7              5
## 4 Ignorado                      13              2              4
```

Podemos ver en la tabla de contingencia que hay ciertos valores que son menores a 5 por lo que la prueba de chi cuadrado nos puede dar error al estimar el esperado para cada categoria.

```
base_pesos_col %>%
  chisq_test(tipo_par ~ area_res)
```

```
## Warning in stats::chisq.test(table(df), ...): Chi-squared approximation may
## be incorrect
```

```
## # A tibble: 1 x 3
##   statistic chisq_df  p_value
##       <dbl>    <int>    <dbl>
## 1      25.0        6 0.000336
```

Intentaremos hacer el metodo computacional

```
ji_estad <- base_pesos_col %>%
  specify(tipo_par ~ area_res) %>%
  calculate(stat = "Chisq") %>%
  pull()

ji_dist_nula <- base_pesos_col %>%
  specify(tipo_par ~ area_res) %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1500, type = "permute") %>%
  calculate(stat = "Chisq")

ji_dist_nula %>%
  get_p_value(ji_estad, direction = "greater")
```

```
## # A tibble: 1 x 1
##    p_value
##      <dbl>
## 1 0.000667
```

Varios modelos lineales simples. Nos gustaria determinar como se comporta el peso al nacer al compararla en un analisis bivariado por regresion lineal simple

```
tabla_coef_lm <- base_pesos_col %>%
  map(~ lm(base_pesos_col$peso_nac ~ .x)) %>%
  map_df(tidy, .id = ".x", conf.int = TRUE)

tabla_coef_lm[3:8] <- round(tabla_coef_lm[3:8], digits = 3)

summary(lm(peso_nac ~ sexo, data = base_pesos_col))
```

```
##
## Call:
## lm(formula = peso_nac ~ sexo, data = base_pesos_col)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1908.69  -475.11    74.89   441.31  2491.31
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    2908.69      16.30 178.480  < 2e-16 ***
## sexoMasculino  -133.59      23.05  -5.796 7.49e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 631 on 2996 degrees of freedom
## Multiple R-squared:  0.01109,    Adjusted R-squared:  0.01076
## F-statistic:  33.6 on 1 and 2996 DF,  p-value: 7.493e-09
```