



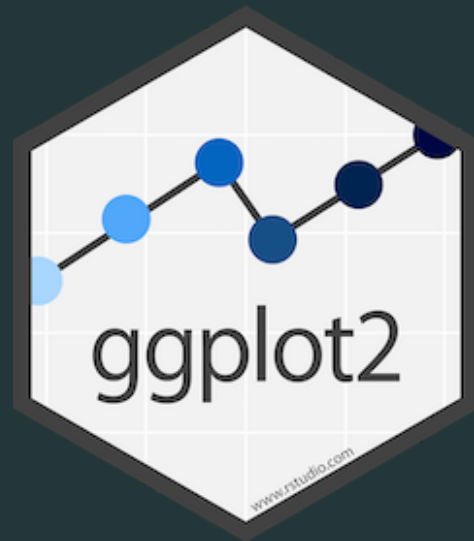
Visualizando datos con el paquete ggplot2

Big Data e Inteligencia Territorial

¿Por qué visualizar?

¿Por qué visualizar?

- *"La visualización es el proceso de hacer visibles los contrastes, ritmos y eventos que los datos expresan, que no podemos percibir cuando vienen en forma de áridas listas de números y categorías."* ¹
- Interpretar / decodificar la información de forma visual



Una forma de visualizar

¿Qué es **ggplot2**?

¿Qué es ggplot2?

- Una implementación del sistema **Grammar of graphics**

¿Qué es ggplot2?

- Una implementación del sistema **Grammar of graphics**
- Un esquema pensado en capas (datos --> plano (ejes **x** e **y**) --> geometrías)

¿Qué es ggplot2?

- Una implementación del sistema **Grammar of graphics**
- Un esquema pensado en capas (datos --> plano (ejes **x** e **y**) --> geometrías)
- Un paquete de funciones de aplicación intuitiva.

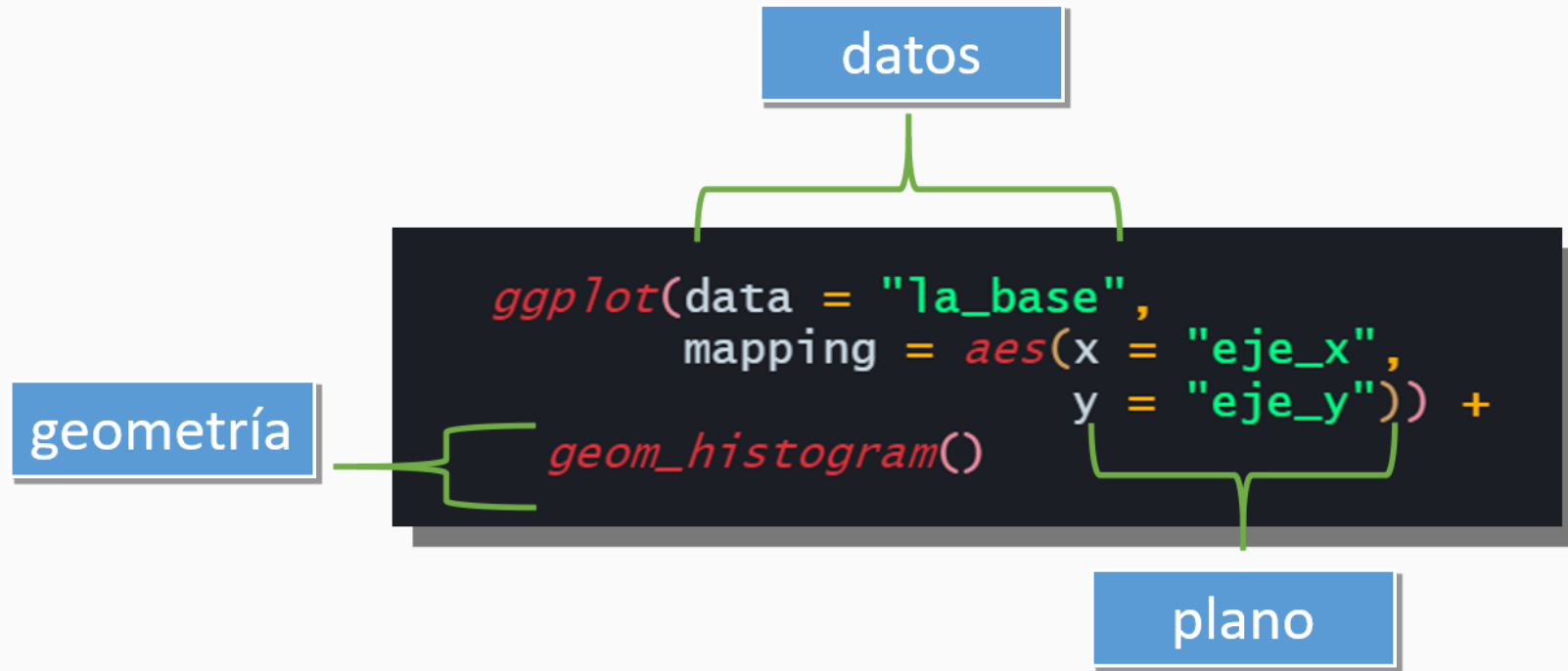
Hoja de ruta

✓ ggplot

- ✓ operador + (nos va a permitir incorporar capas -como lo hacía el pipe %>%-)
 - ✓ `geom_histogram()`
 - ✓ `geom_boxplot()`
 - ✓ `geom_col()`
 - ✓ chapa y pintura (Atributos)
-

```
base_covid ← read.table("entradas/base_covid_muestra.txt",  
                        sep = ",",  
                        header = T,  
                        fileEncoding = "UTF-8")
```

ggplot()



geom_histogram()

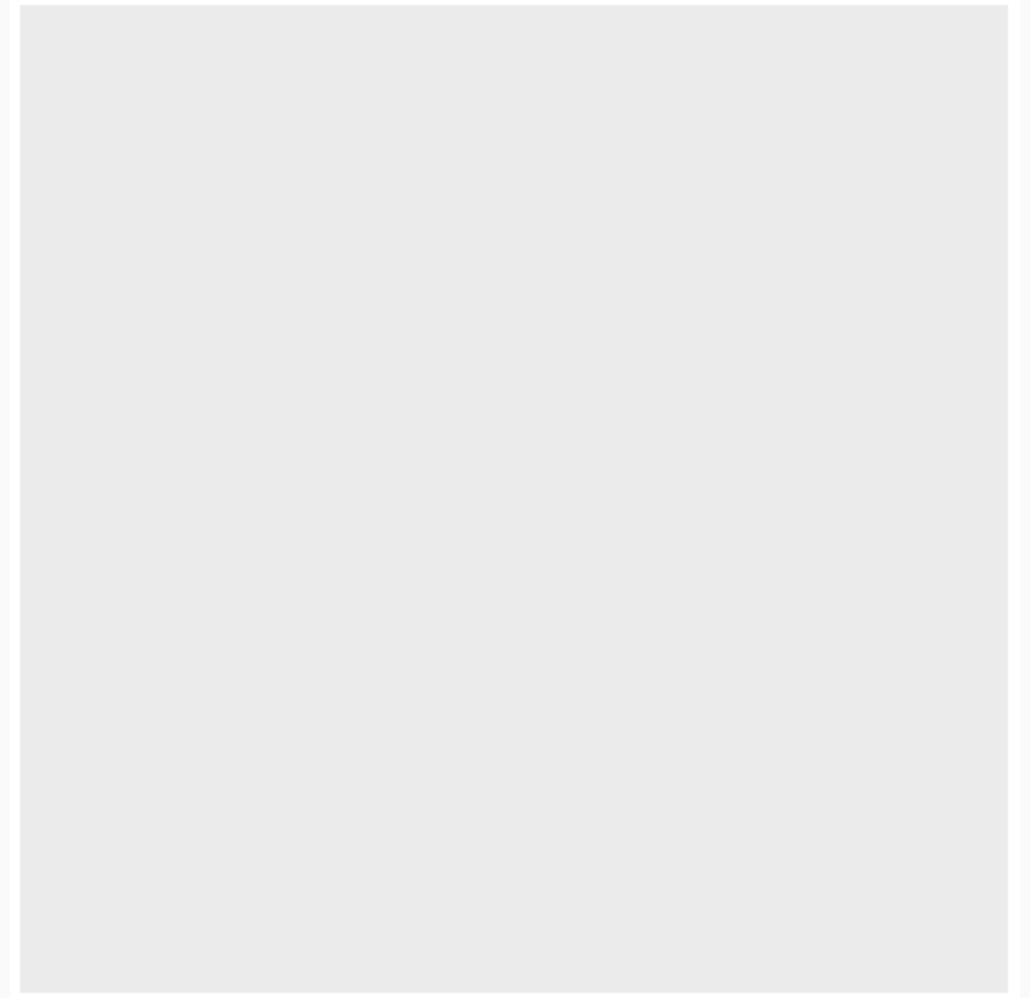
geom_histogram()

- **Caso:** Quiero conocer la distribución de la población según su edad:

Capas del gráfico - insumo(dataframe)

```
ggplot(data = base_covid)
```

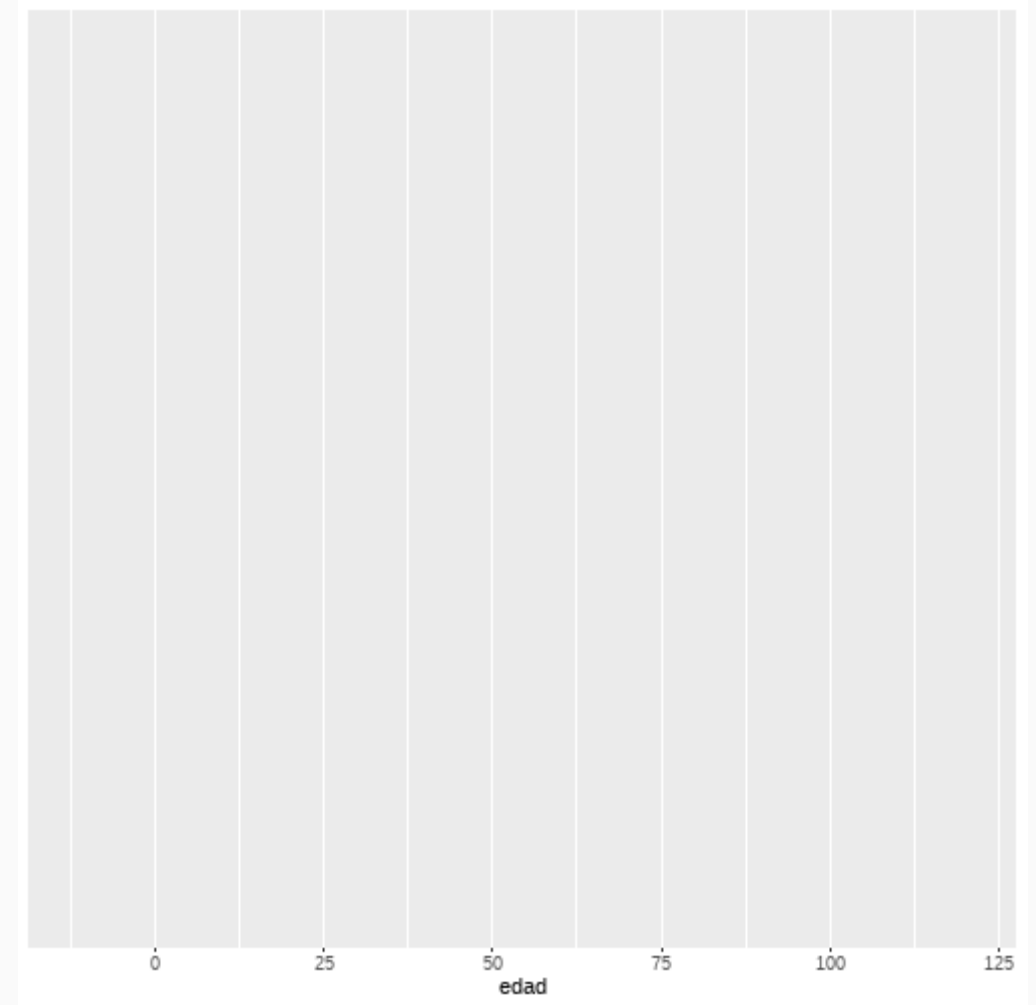
- **¿De donde saco las variables?**



Capas del gráfico - coordenadas

```
ggplot(data = base_covid,  
       mapping = aes(x = edad))
```

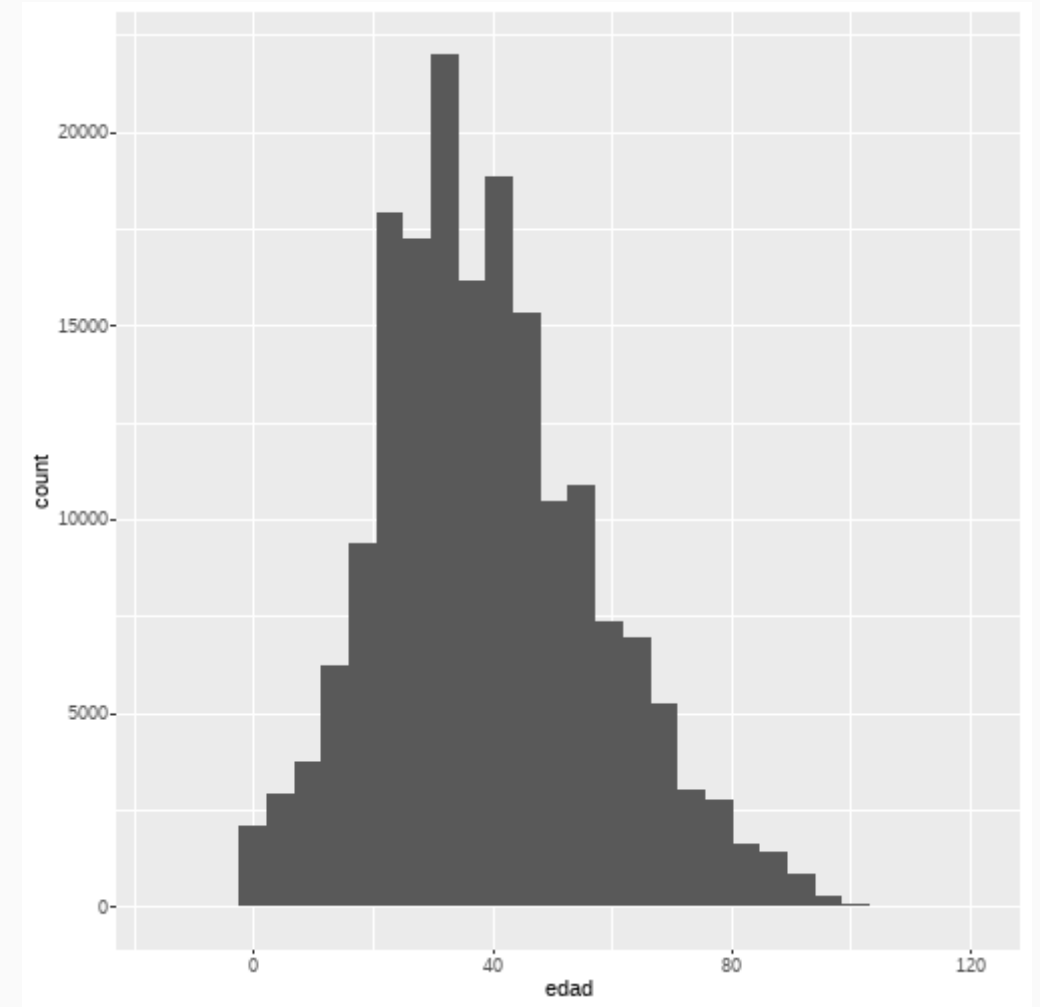
- **¿Qué variables? ¿Cómo me las imagino en el gráfico?**



Capas del gráfico - geometría

```
ggplot(data = base_covid,  
       mapping = aes(x = edad)) +  
  geom_histogram()
```

- ¿Qué tipo de gráfico quiero hacer?

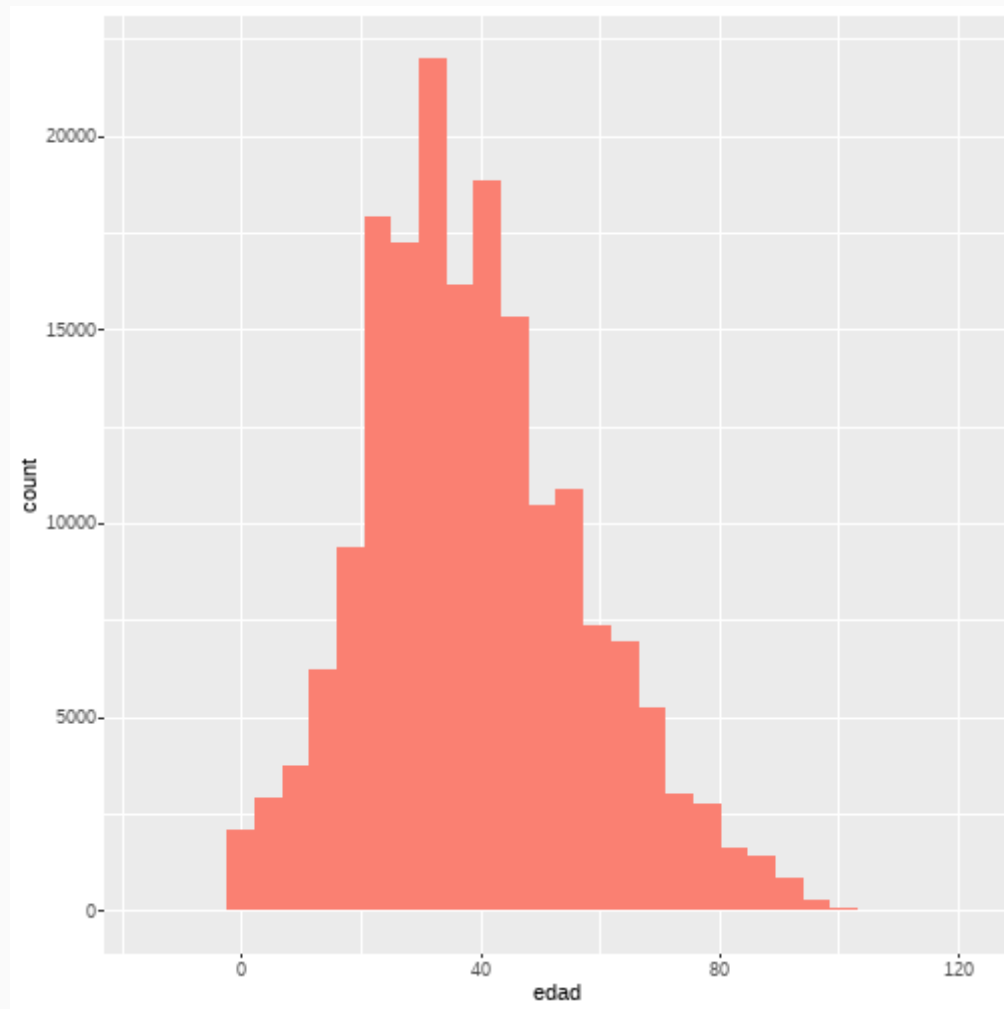


Chapa y pintura (Atributos)

- ✓ `fill()` (*Rellenar*)
- ✓ `colour()` (*colorear*)
- ✓ `alpha` (*transparencia*)

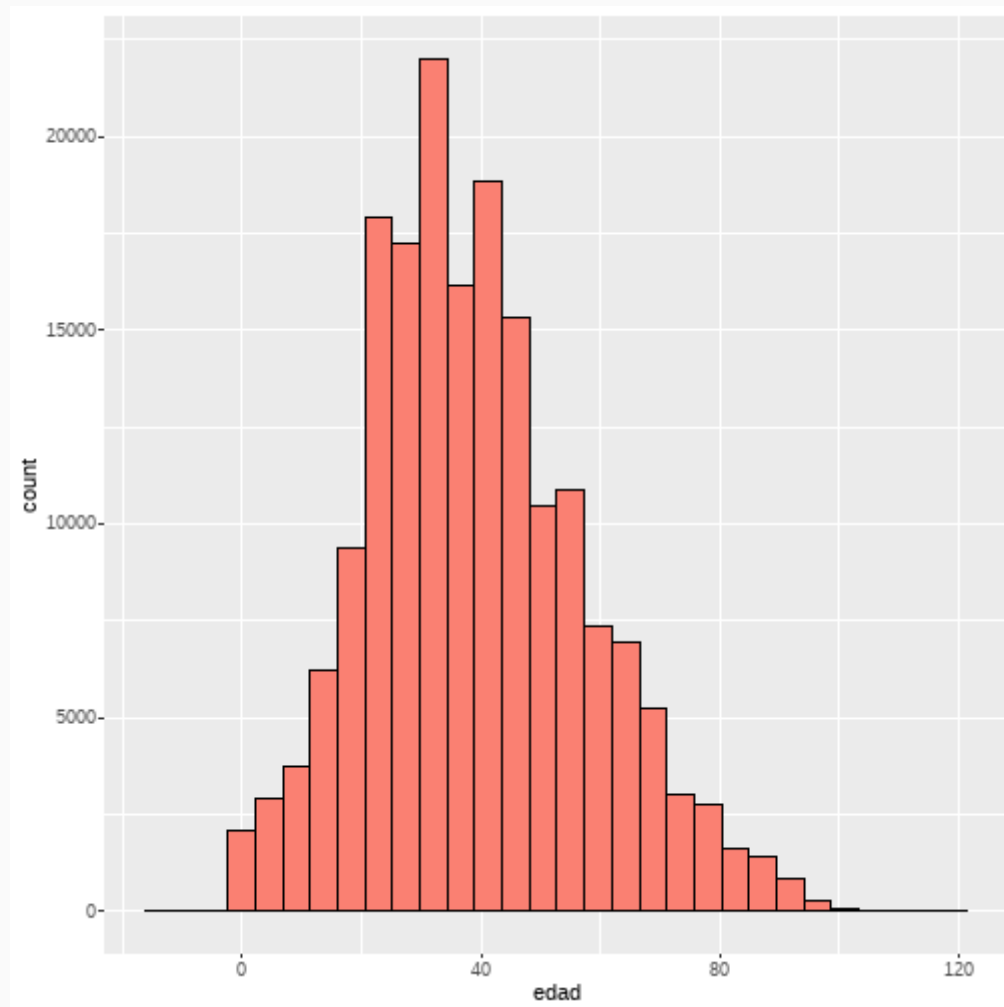
```
ggplot(data = base_covid,  
       mapping = aes(x = edad)) +  
  geom_histogram(fill = "salmon")
```

- **Relleno**



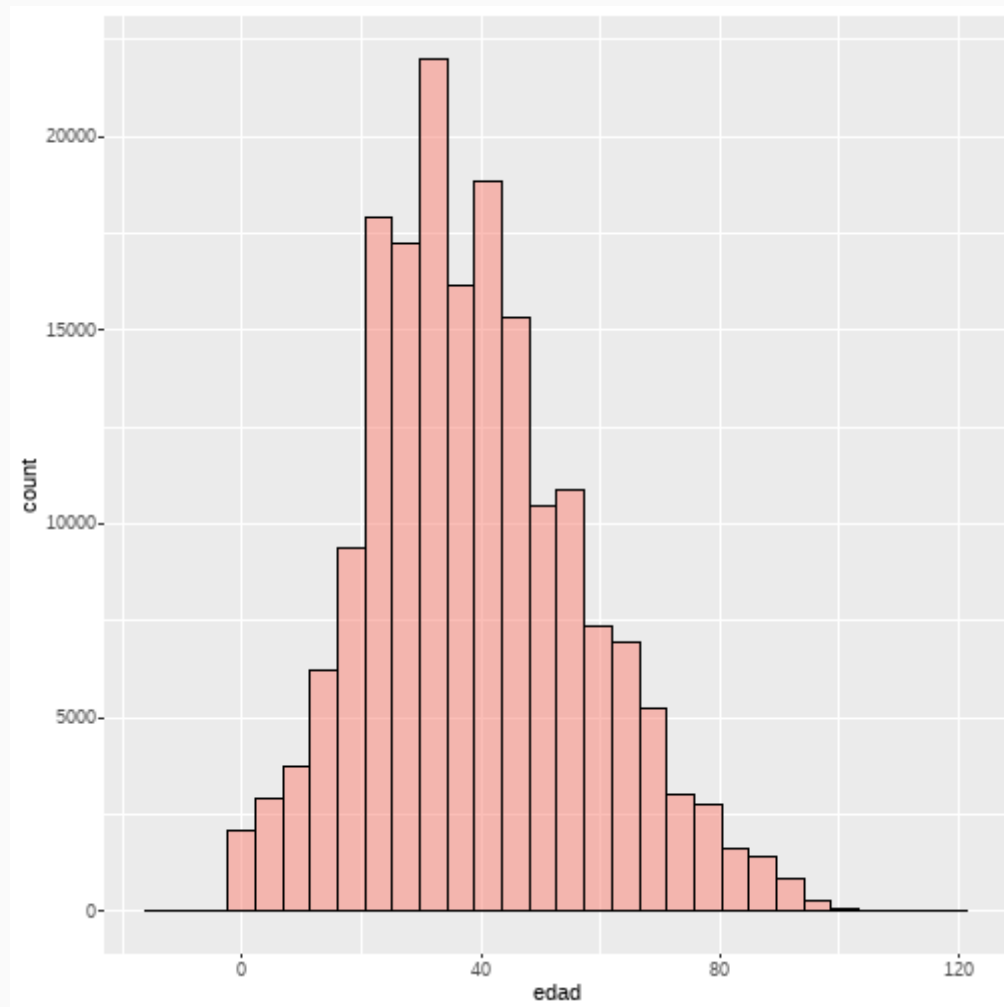
```
ggplot(data = base_covid,  
       mapping = aes(x = edad)) +  
  geom_histogram(fill = "salmon",  
                 colour = "black")
```

- **Contorno**



```
ggplot(data = base_covid,  
       mapping = aes(x = edad)) +  
  geom_histogram(fill = "salmon",  
                 colour = "black",  
                 alpha = 0.5)
```

- **Transparencia**



PRÁCTICA

Práctica

1) Importar la base covid_muestra.

2) El siguiente código debería realizar un histograma. Para ello hace falta completar aquellos espacios con `__` y ejecutarlo:

```
ggplot(data = _____,  
       mapping = aes(x = edad)) _____  
  geom_____ (fill = _____,  
              colour = "black",  
              alpha = 0.5)
```

3) Replicar el código anterior y modificar los valores de los atributos a gusto

geom_col()

geom_col()

Caso: Quiero visualizar la diferencia entre la edad media de aquellas personas que recibieron asistencia respiratoria y aquellas que no:

geom_col()

Caso: Quiero visualizar la diferencia entre la edad media de aquellas personas que recibieron asistencia respiratoria y aquellas que no:

```
summary(base_covid$edad)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
-12.00	26.00	37.00	39.08	50.00	121.00	92

geom_col()

Caso: Quiero visualizar la diferencia entre la edad media de aquellas personas que recibieron asistencia respiratoria y aquellas que no:

```
summary(base_covid$edad)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
-12.00	26.00	37.00	39.08	50.00	121.00	92

```
base_covid %>%  
  filter(edad == 0) %>%  
  count()
```

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

	n
	<int>
1	2

geom_col()

- **Paso 2.** *Limpiar* la variable:

```
tabla_edad <- base_covid %>%  
  rename(asistencia_mec = asistencia_respiratoria_mecanica) %>%  
  filter(!is.na(asistencia_mec) & edad ≥ 0) %>%  
  group_by(asistencia_mec) %>%  
  summarise(edad_media = mean(edad, na.rm = TRUE)) %>%  
  mutate(edaD_media = round(edad_media, digits = 0))
```

geom_col()

- **Paso 2.** *Limpiar* la variable:

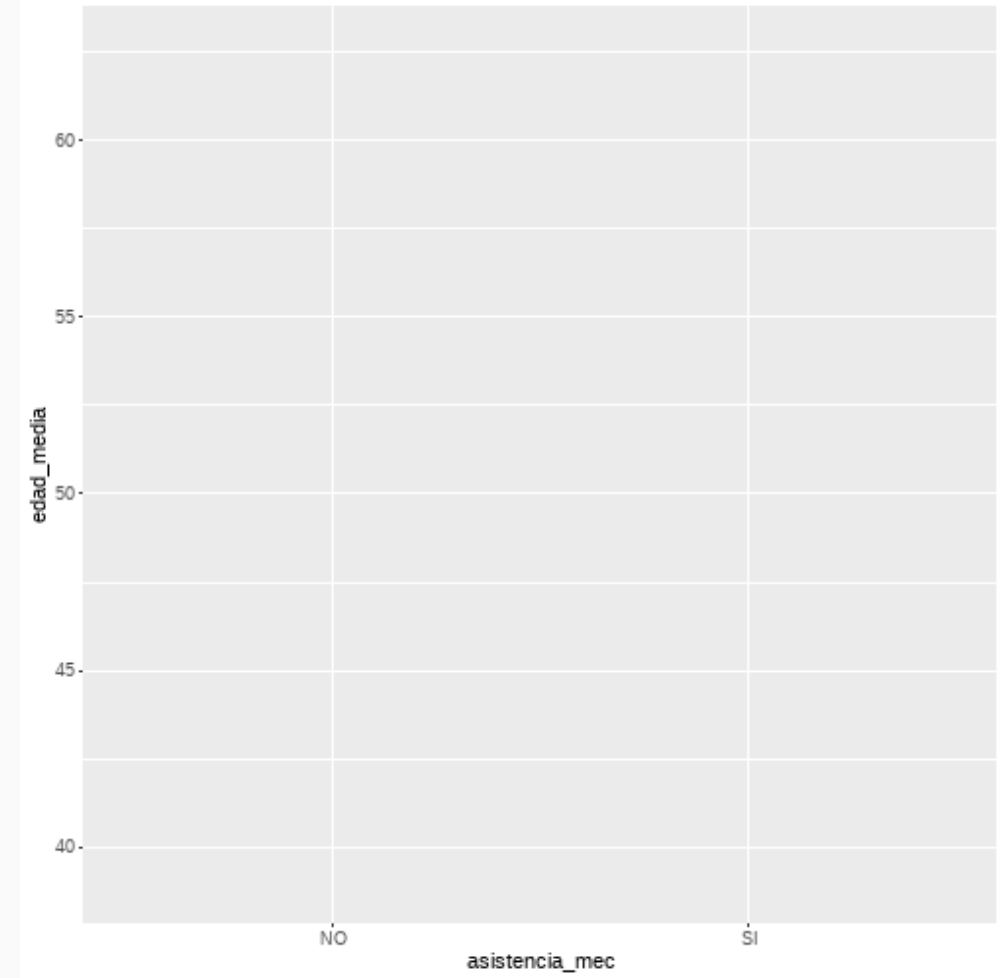
```
tabla_edad <- base_covid %>%  
  rename(asistencia_mec = asistencia_respiratoria_mecanica) %>%  
  filter(!is.na(asistencia_mec) & edad ≥ 0) %>%  
  group_by(asistencia_mec) %>%  
  summarise(edad_media = mean(edad, na.rm = TRUE)) %>%  
  mutate(edaD_media = round(edad_media, digits = 0))
```

A tibble: 2 x 3

	asistencia_mec	edad_media	edaD_media
	<chr>	<dbl>	<dbl>
1	NO	39.0	39
2	SI	62.6	63

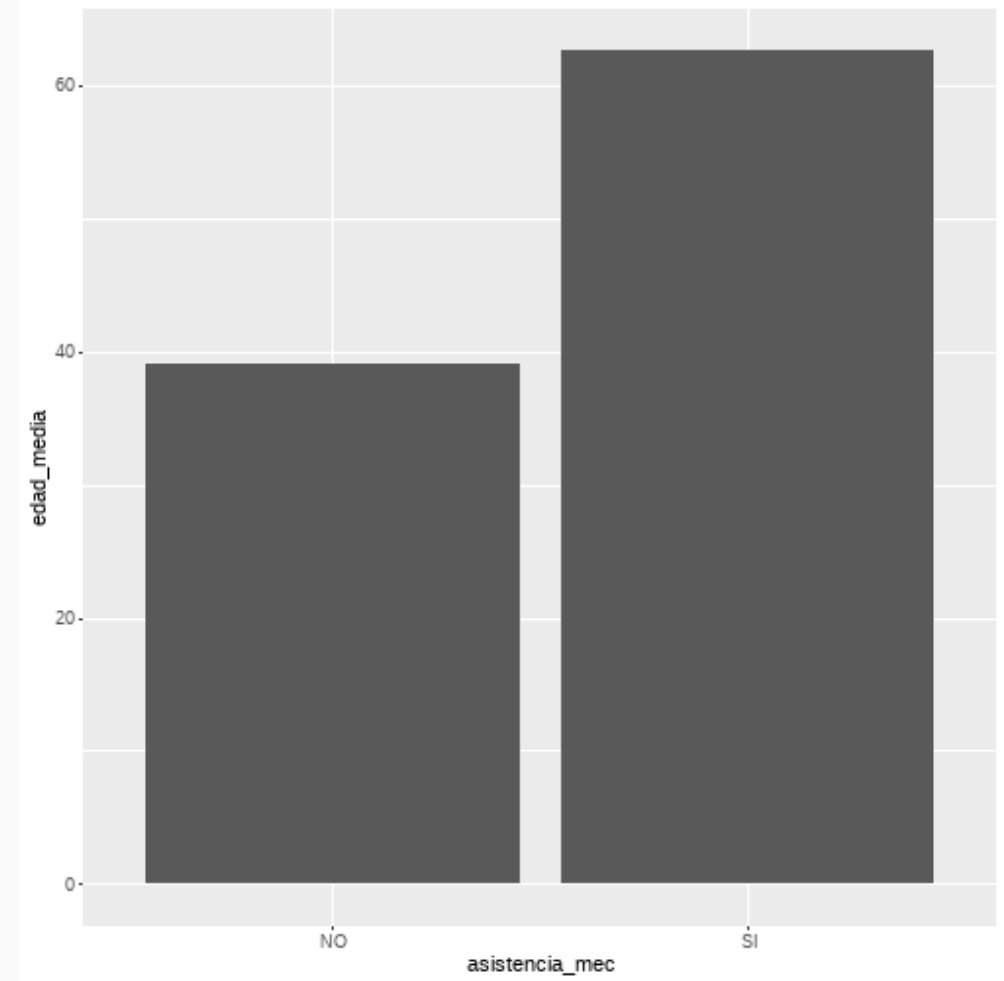
geom_col

```
ggplot(data = tabla_edad,  
       mapping = aes(x = asistencia_mec,  
                     y = edad_media))
```



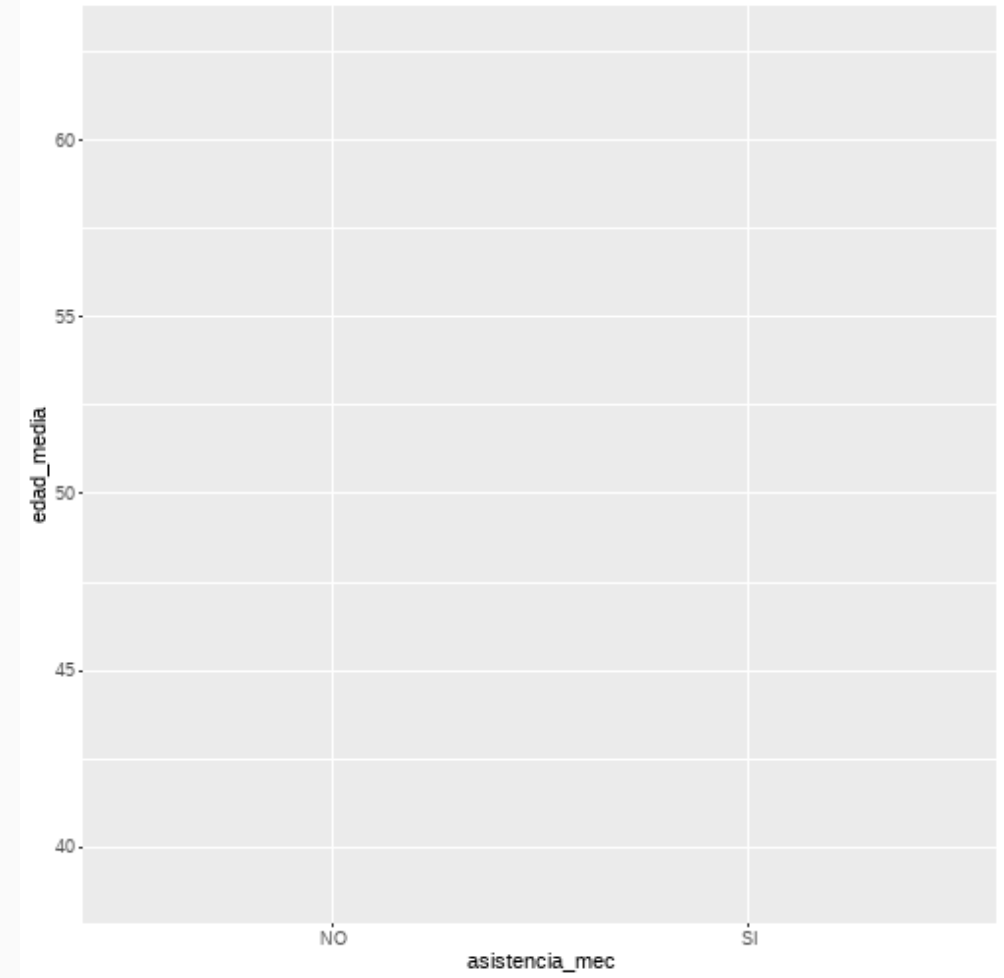
geom_col

```
ggplot(data = tabla_edad,  
       mapping = aes(x = asistencia_mec,  
                     y = edad_media)) +  
  geom_col()
```



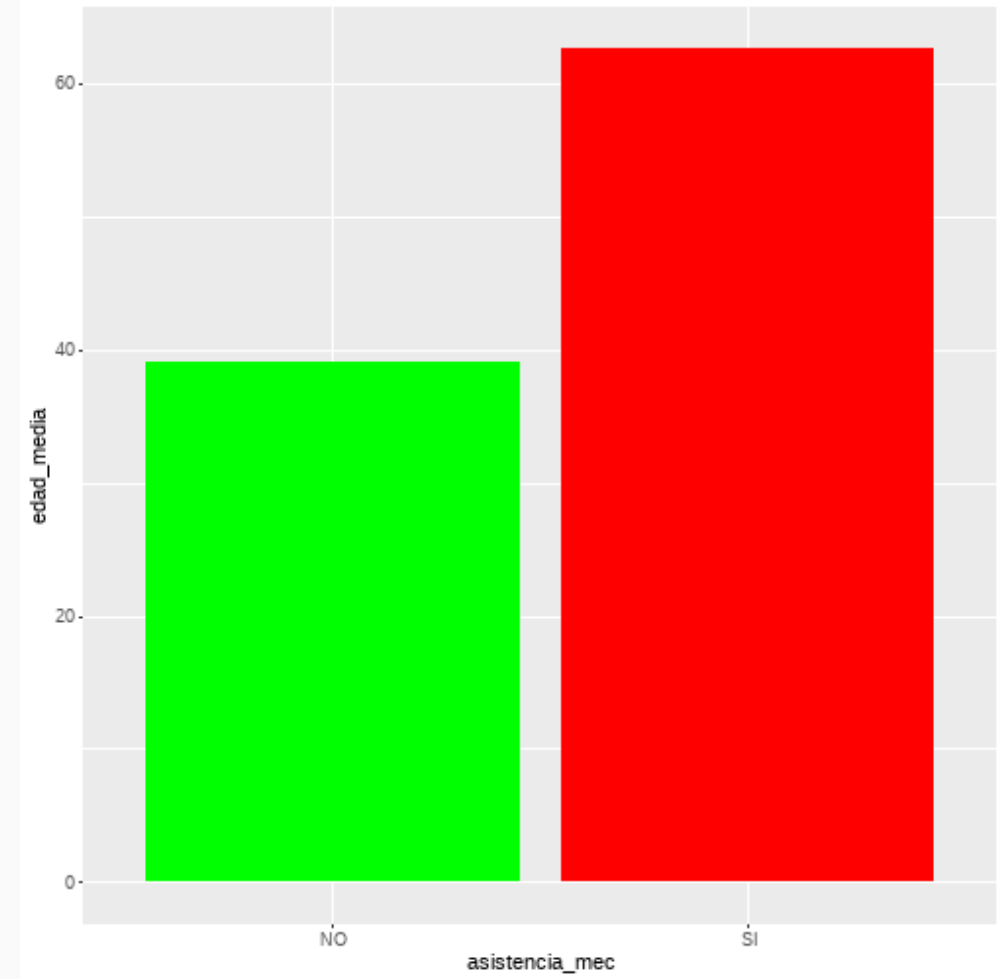
geom_col - relleno

```
ggplot(data = tabla_edad,  
       mapping = aes(x = asistencia_mec,  
                     y = edad_media))
```



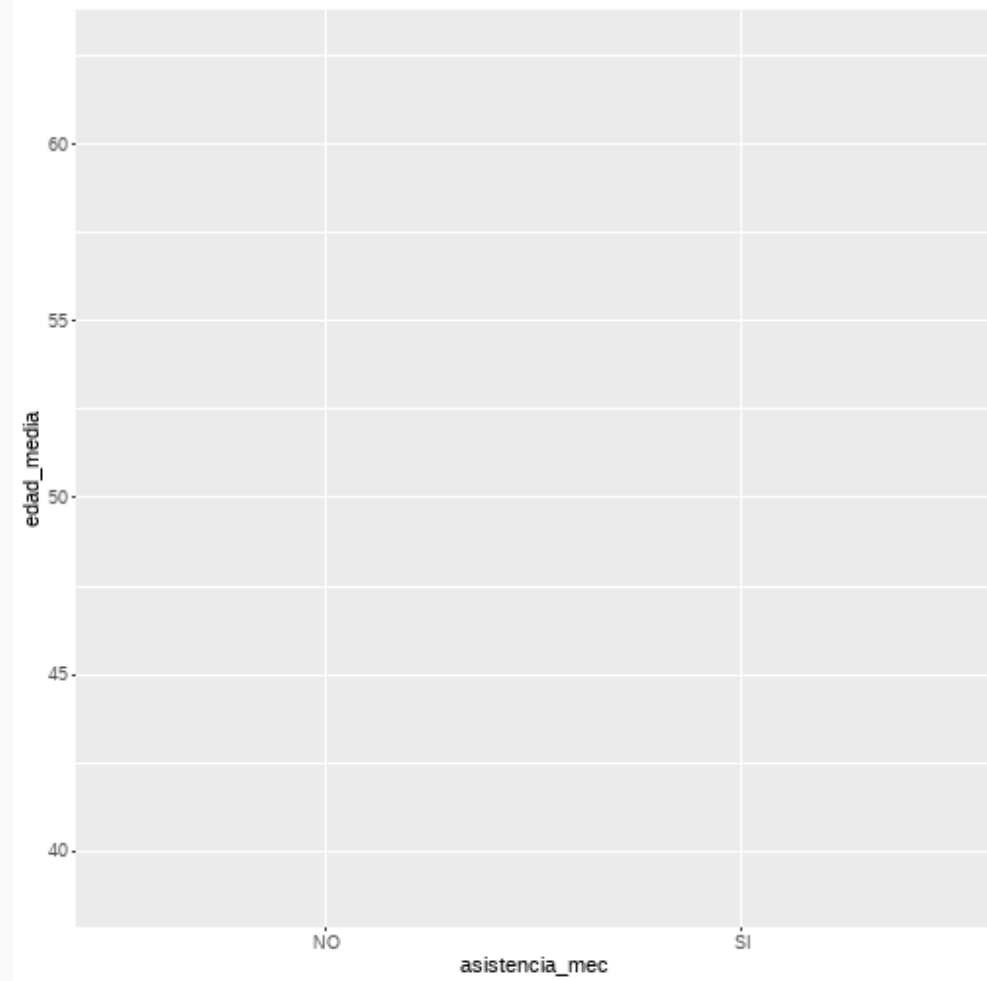
geom_col - relleno

```
ggplot(data = tabla_edad,  
       mapping = aes(x = asistencia_mec,  
                     y = edad_media)) +  
  geom_col(fill = c("green", "red"))
```



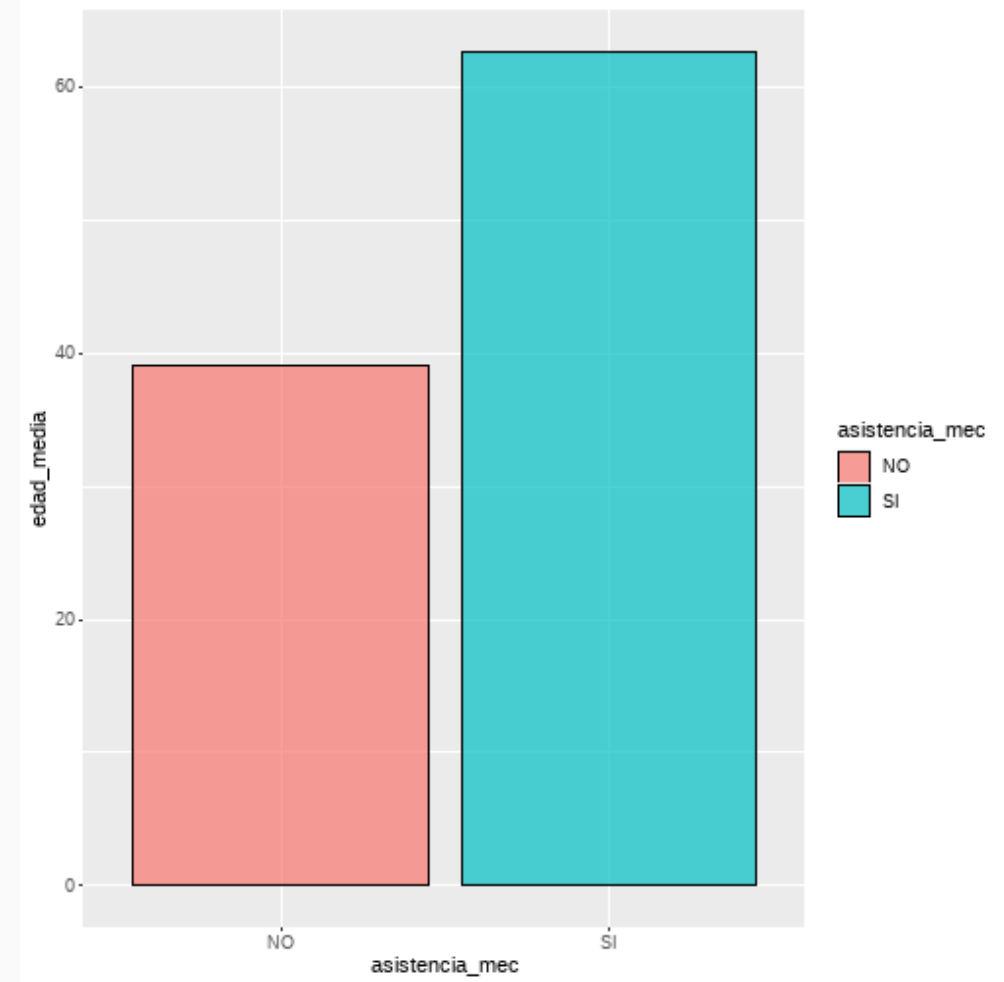
geom_col - Contorno + Transparencia

```
ggplot(data = tabla_edad,  
       mapping = aes(x = asistencia_mec,  
                     y = edad_media))
```



geom_col - Contorno + Transparencia

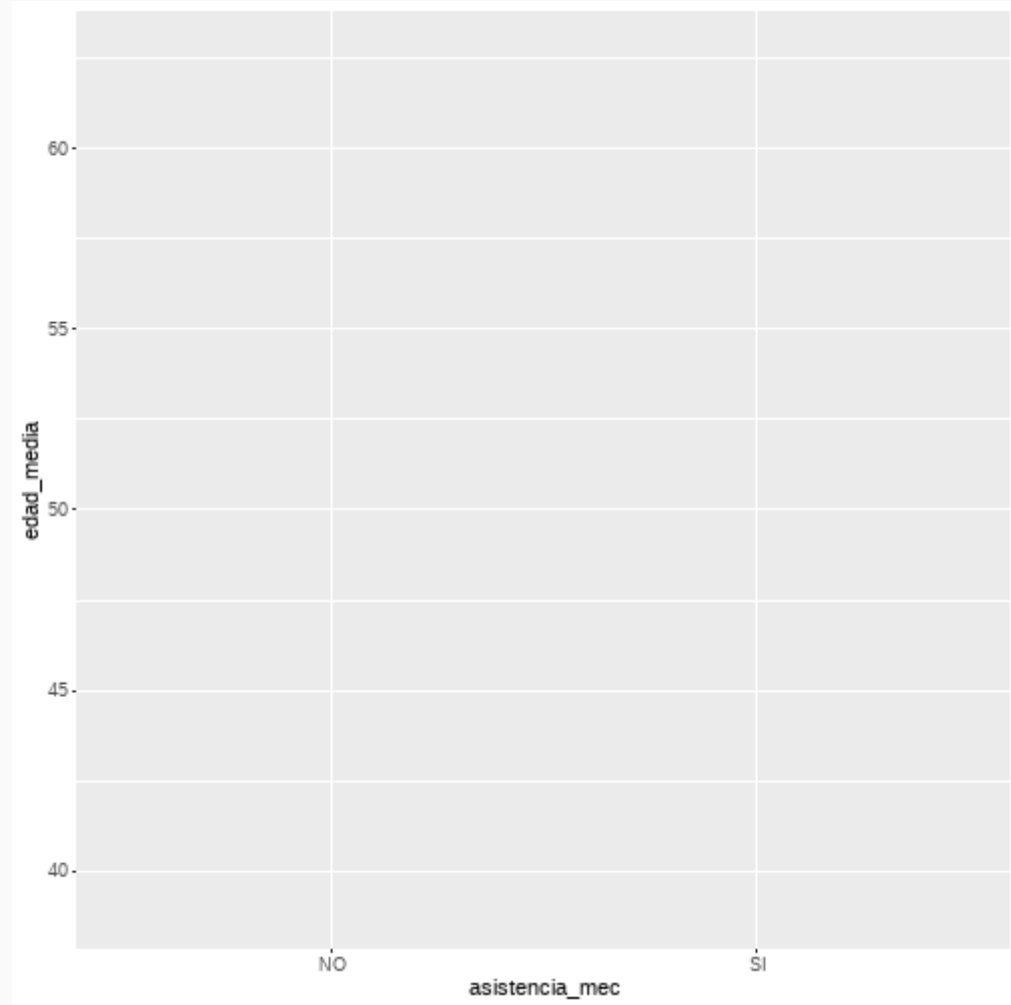
```
ggplot(data = tabla_edad,  
       mapping = aes(x = asistencia_mec,  
                     y = edad_media)) +  
  geom_col(aes(fill = asistencia_mec),  
          colour = "black",  
          alpha = 0.7)
```



chapa y pintura

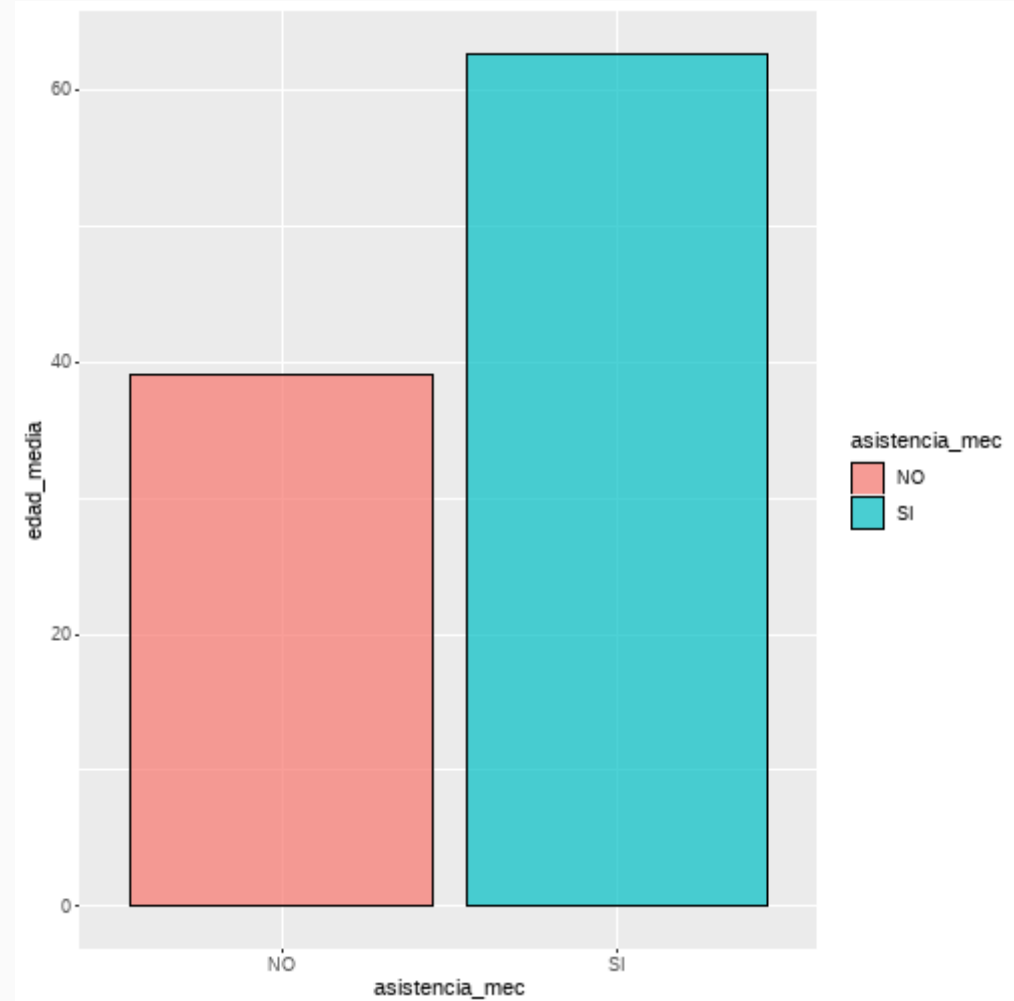
geom_col

```
ggplot(data = tabla_edad,  
       mapping = aes(x = asistencia_mec,  
                     y = edad_media))
```



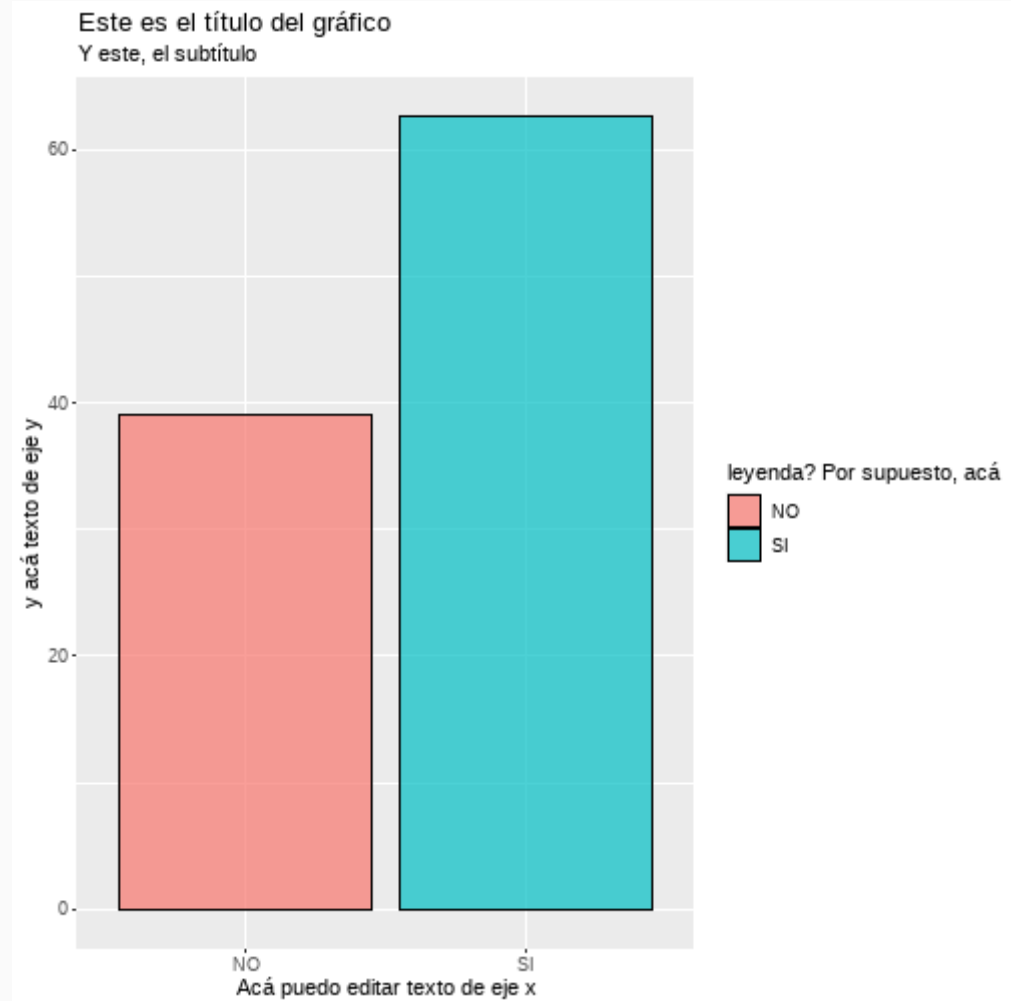
geom_col

```
ggplot(data = tabla_edad,  
       mapping = aes(x = asistencia_mec,  
                     y = edad_media)) +  
  geom_col(aes(fill = asistencia_mec),  
          colour = "black",  
          alpha = 0.7)
```



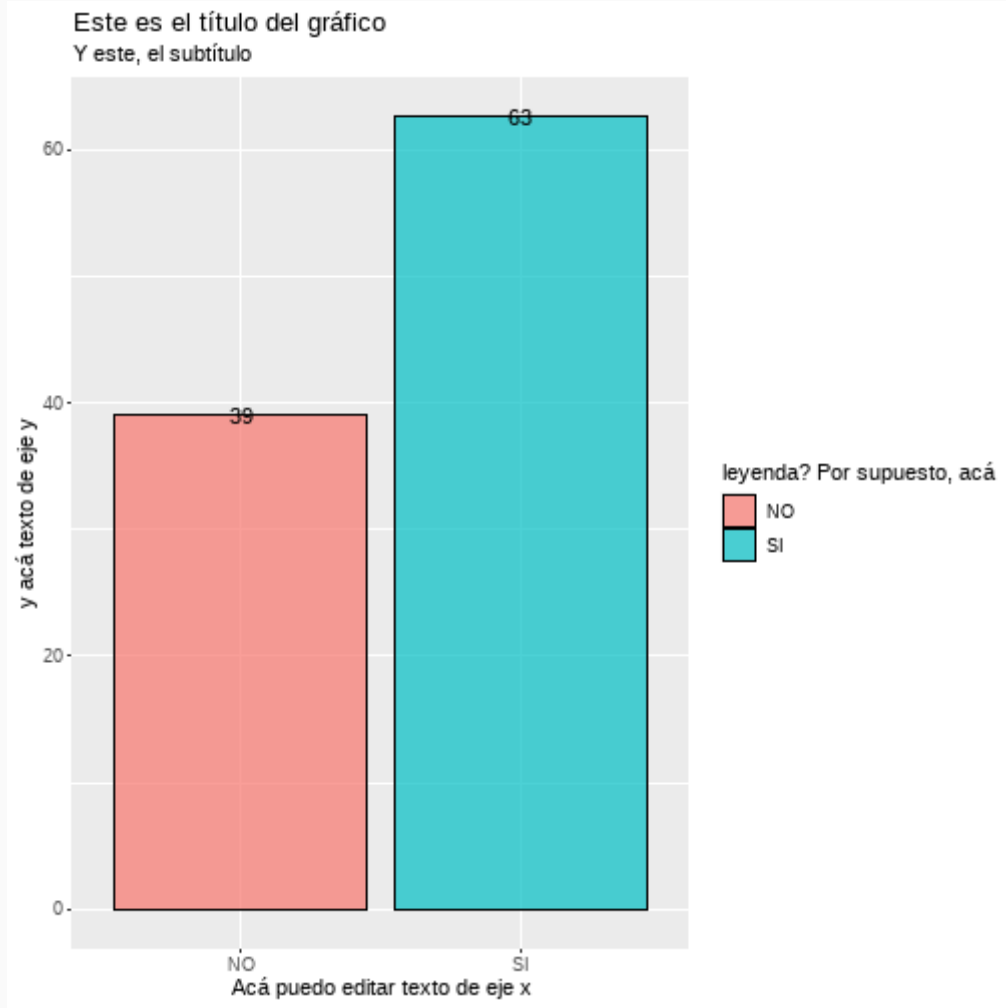
geom_col

```
ggplot(data = tabla_edad,  
       mapping = aes(x = asistencia_mec,  
                     y = edad_media)) +  
  geom_col(aes(fill = asistencia_mec),  
           colour = "black",  
           alpha = 0.7) +  
  labs(title = "Este es el título del gráfico",  
        subtitle = "Y este, el subtítulo",  
        x = "Acá puedo editar texto de eje x",  
        y = "y acá texto de eje y",  
        fill = "leyenda? Por supuesto, acá")
```



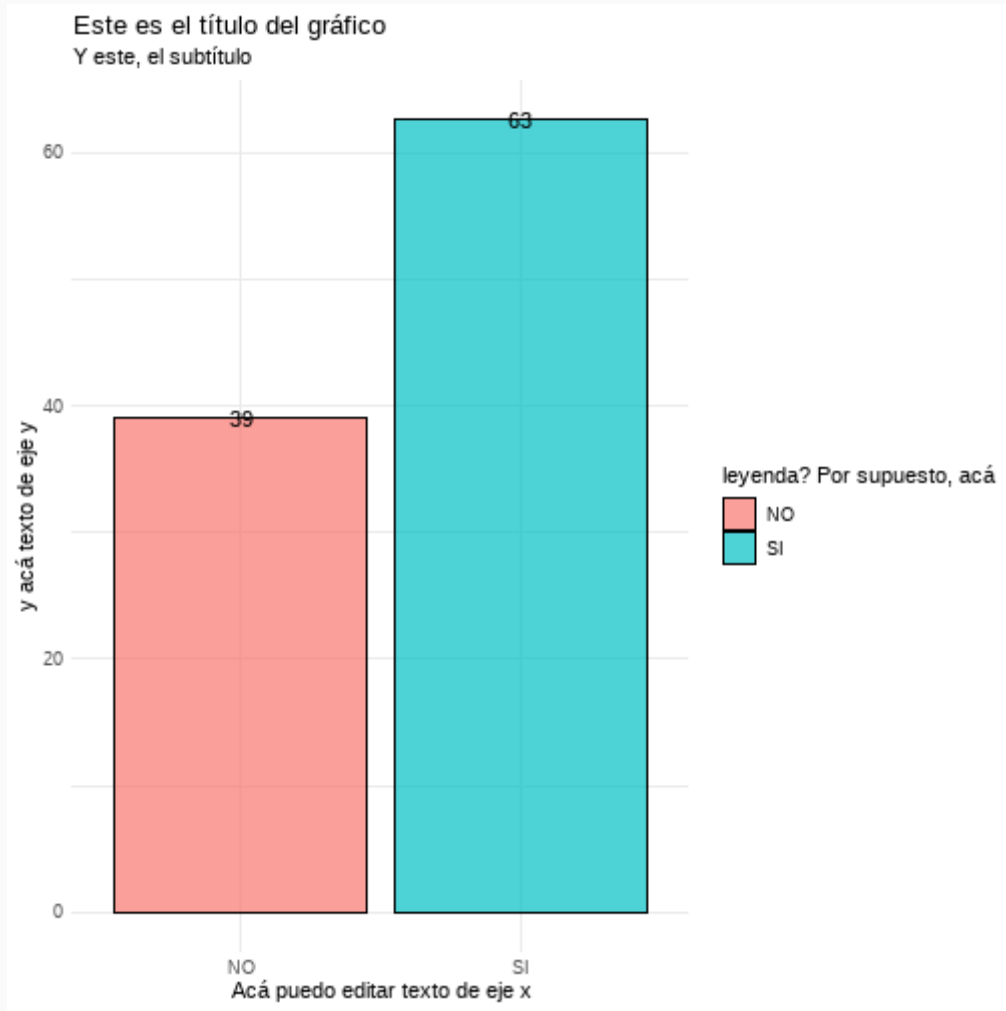
geom_col

```
ggplot(data = tabla_edad,  
       mapping = aes(x = asistencia_mec,  
                     y = edad_media)) +  
  geom_col(aes(fill = asistencia_mec),  
           colour = "black",  
           alpha = 0.7) +  
  labs(title = "Este es el título del gráfico",  
       subtitle = "Y este, el subtítulo",  
       x = "Acá puedo editar texto de eje x",  
       y = "y acá texto de eje y",  
       fill = "leyenda? Por supuesto, acá") +  
  geom_text(aes(label = round(edad_media)))
```



geom_col

```
ggplot(data = tabla_edad,  
       mapping = aes(x = asistencia_mec,  
                     y = edad_media)) +  
  geom_col(aes(fill = asistencia_mec),  
          colour = "black",  
          alpha = 0.7) +  
  labs(title = "Este es el título del gráfico",  
       subtitle = "Y este, el subtítulo",  
       x = "Acá puedo editar texto de eje x",  
       y = "y acá texto de eje y",  
       fill = "leyenda? Por supuesto, acá") +  
  geom_text(aes(label = round(edad_media))) +  
  theme_minimal()
```



Orden de las categorías

- Para darle orden a las categorías en un eje podemos recurrir al tipo de variable **factor**.

Orden de las categorías

- Para darle orden a las categorías en un eje podemos recurrir al tipo de variable **factor**.
- Los factores en R son variables categóricas (o llamadas "cualitativas").

Orden de las categorías

- Para darle orden a las categorías en un eje podemos recurrir al tipo de variable **factor**.
- Los factores en R son variables categóricas (o llamadas "cualitativas").
- Además, otra propiedad de las variables de tipo **factor** es la ordinalidad (no es indistinto el orden de aparición). Por ejemplo, variables como *nivel educativo* o *grado de satisfacción* son variables que (números o texto), son categóricas y ordinales.

Orden de las categorías

```
edad ← c(24,54,75,25,65,12)
sexo_num ← c(1, 2, 2, 2, 1, 2)
sexo_text ← c("hombre", "mujer", "mujer", "mujer", "hombre", "mujer")

base ← data.frame(edad, sexo_num, sexo_text)
```

```
class(base$sexo_num)
```

```
[1] "numeric"
```

Orden de las categorías

```
edad ← c(24,54,75,25,65,12)
sexo_num ← c(1, 2, 2, 2, 1, 2)
sexo_text ← c("hombre", "mujer", "mujer", "mujer", "hombre", "mujer")

base ← data.frame(edad, sexo_num, sexo_text)
```

```
class(base$sexo_num)
```

```
[1] "numeric"
```

```
class(base$sexo_text)
```

```
[1] "character"
```

Orden de las categorías

- Convertimos en factor a la variable *sexo_num*

```
base <- base %>%  
  mutate(sexo_factor = factor(sexo_num,  
                               levels = c(1, 2)))
```

```
base %>%  
  count(sexo_factor)
```

	sexo_factor	n
1	1	2
2	2	4

Orden de las categorías

- Convertimos en factor a la variable `sexo_num`

```
base <- base %>%  
  mutate(sexo_factor = factor(sexo_num,  
                               levels = c(1, 2)))
```

```
base %>%  
  count(sexo_factor)
```

	sexo_factor	n
1	1	2
2	2	4

- ¿Qué pasa si invierto el orden de los niveles (categorías) en el parámetro `level = ?`

geom_col + factor

```
tabla_edad
```

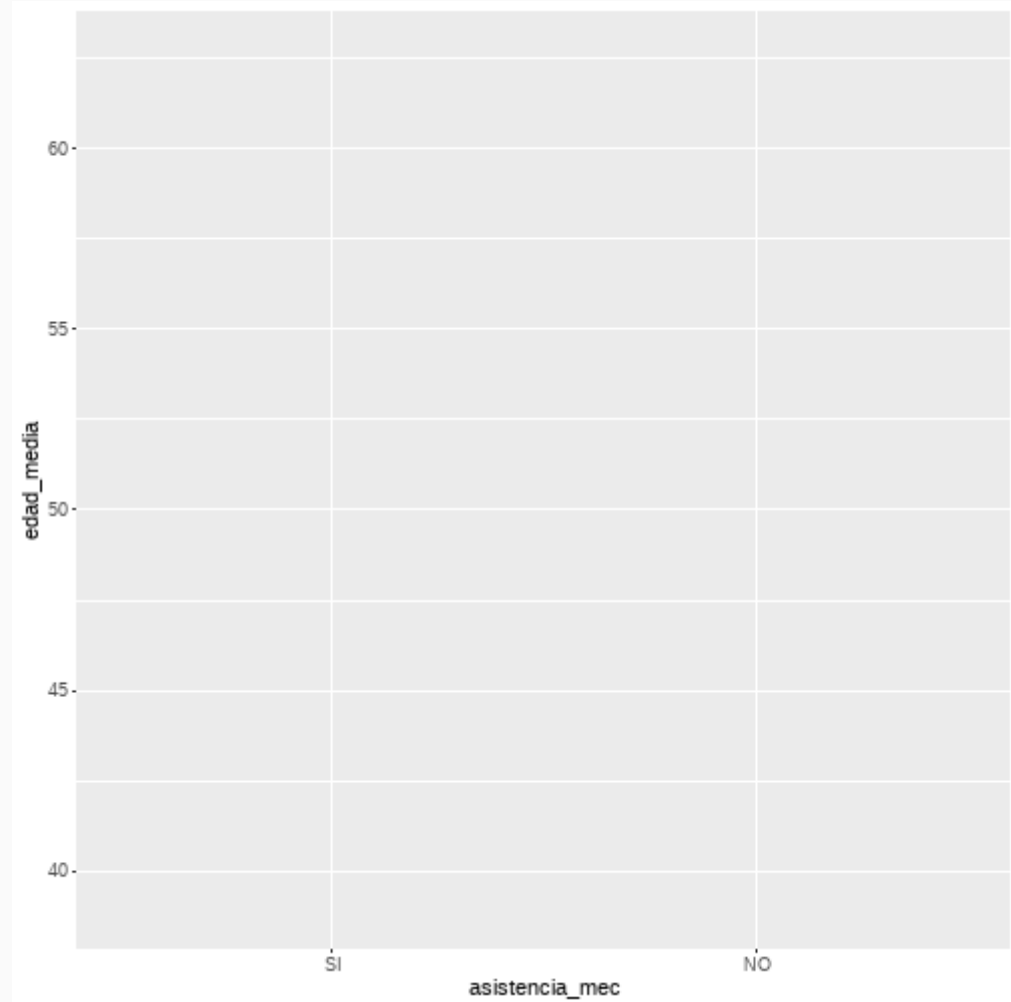
```
# A tibble: 2 x 2  
  asistencia_mec edad_media  
    <chr>          <dbl>  
1 NO              39.0  
2 SI              62.6
```


geom_col + factor

```
tabla_edad %>%  
  mutate(asistencia_mec = factor(asistencia_mec,  
                                levels = c("SI", "NO")))  
# A tibble: 2 x 2  
  asistencia_mec edad_media  
    <fct>          <dbl>  
1 NO             39.0  
2 SI             62.6
```

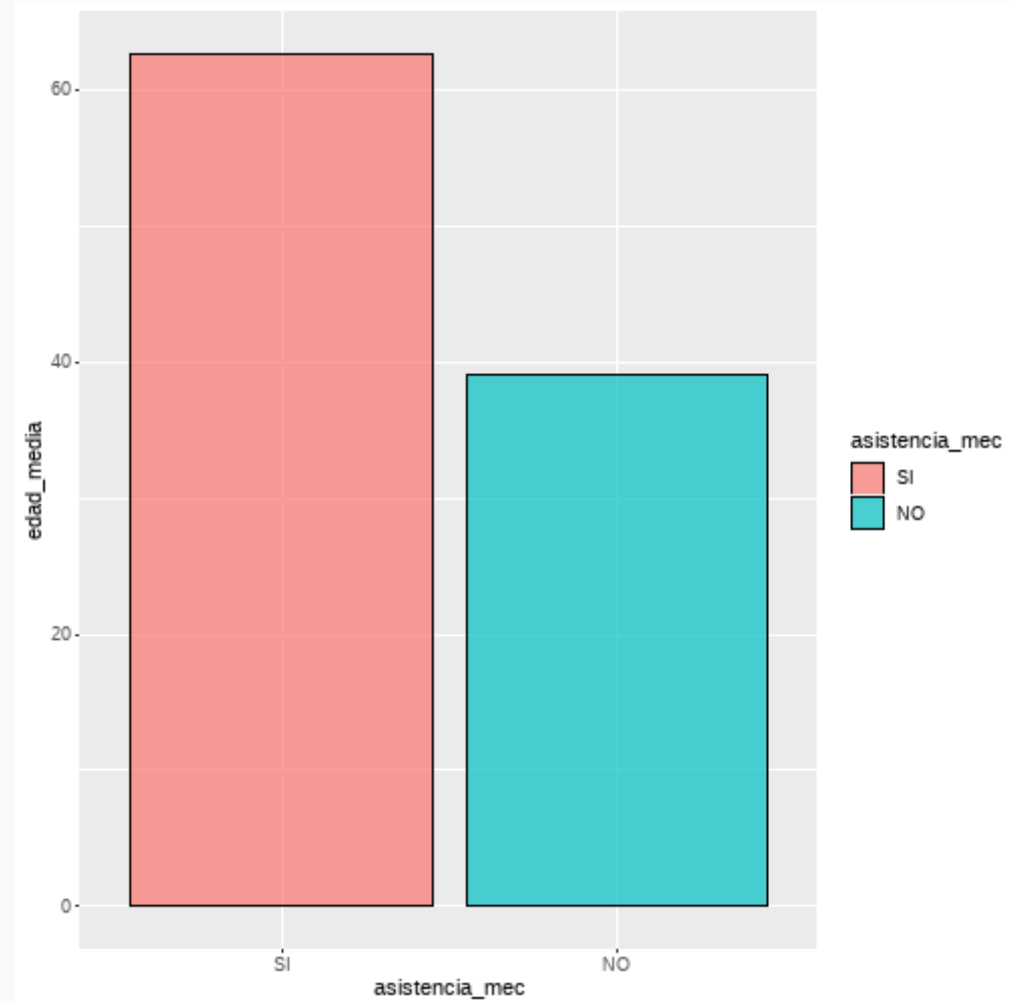
geom_col + factor

```
tabla_edad %>%  
  mutate(asistencia_mec = factor(asistencia_mec,  
                                levels = c("SI", "NO"),  
                                ordered = TRUE),  
  ggplot(mapping = aes(x = asistencia_mec,  
                        y = edad_media))
```



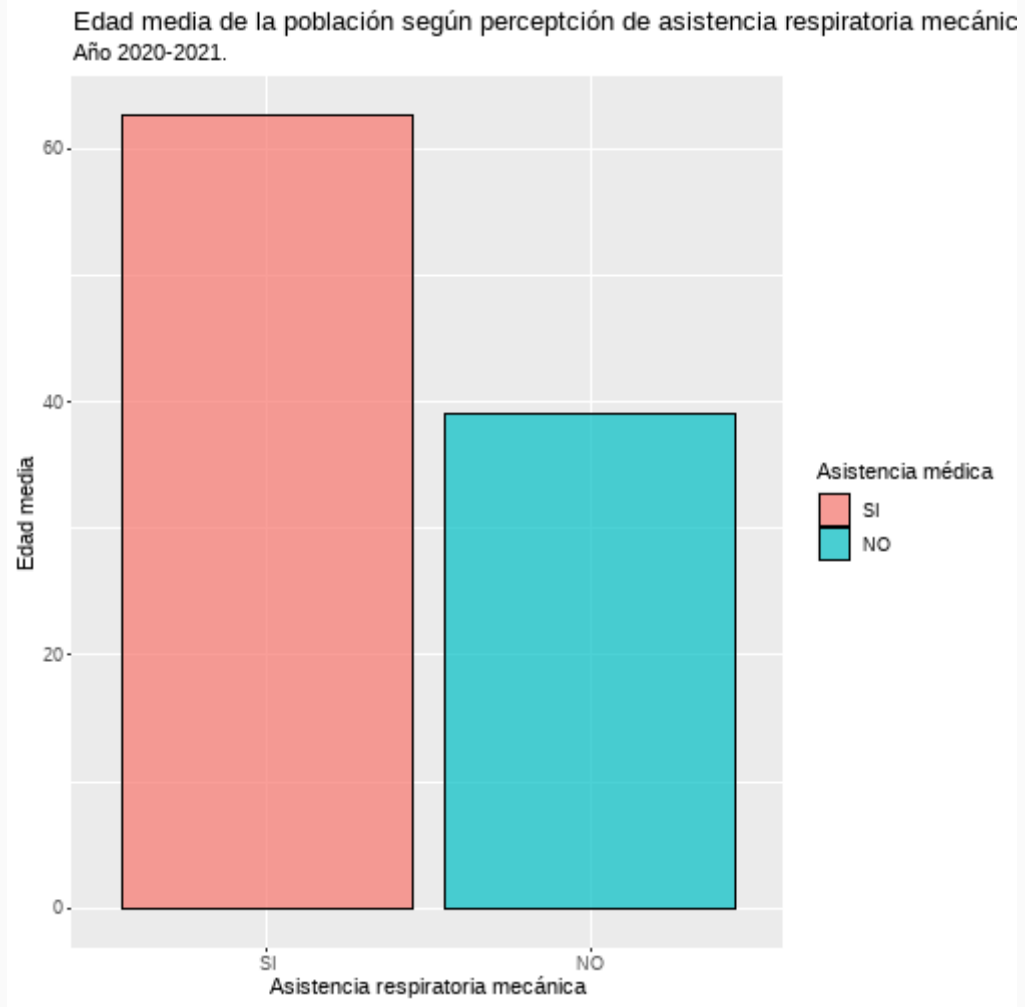
geom_col + factor

```
tabla_edad %>%  
  mutate(asistencia_mec = factor(asistencia_mec,  
                                  levels = c("SI", "NO"),  
                                  ordered = TRUE))  
  ggplot(mapping = aes(x = asistencia_mec,  
                        y = edad_media)) +  
  geom_col(aes(fill = asistencia_mec),  
           colour = "black",  
           alpha = 0.7)
```



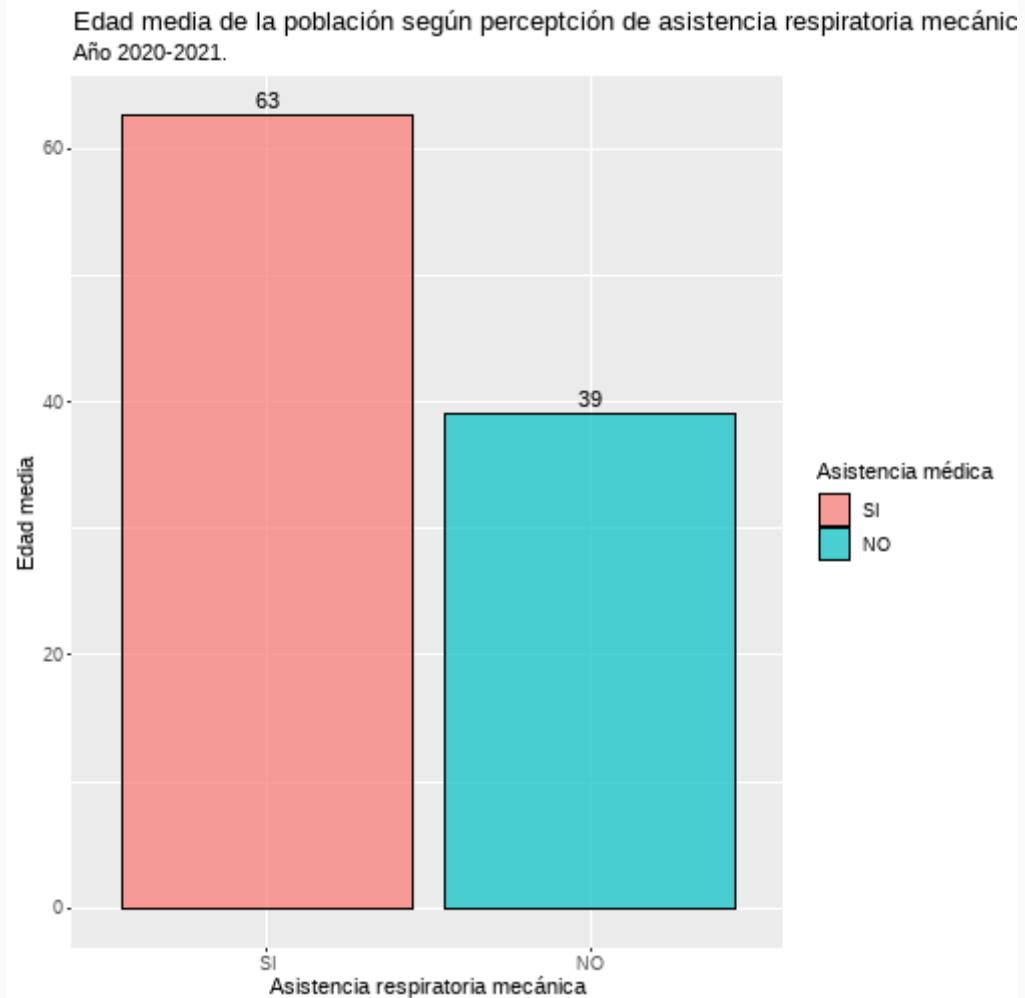
geom_col + factor

```
tabla_edad %>%  
  mutate(asistencia_mec = factor(asistencia_mec,  
                                  levels = c("SI", "NO"),  
                                  ordered = TRUE)) +  
  ggplot(mapping = aes(x = asistencia_mec,  
                        y = edad_media)) +  
  geom_col(aes(fill = asistencia_mec),  
           colour = "black",  
           alpha = 0.7) +  
  labs(title = "Edad media de la población según percepción de asistencia respiratoria mecánica",  
        subtitle = "Año 2020-2021.",  
        x = "Asistencia respiratoria mecánica",  
        y = "Edad media",  
        fill = "Asistencia médica")
```



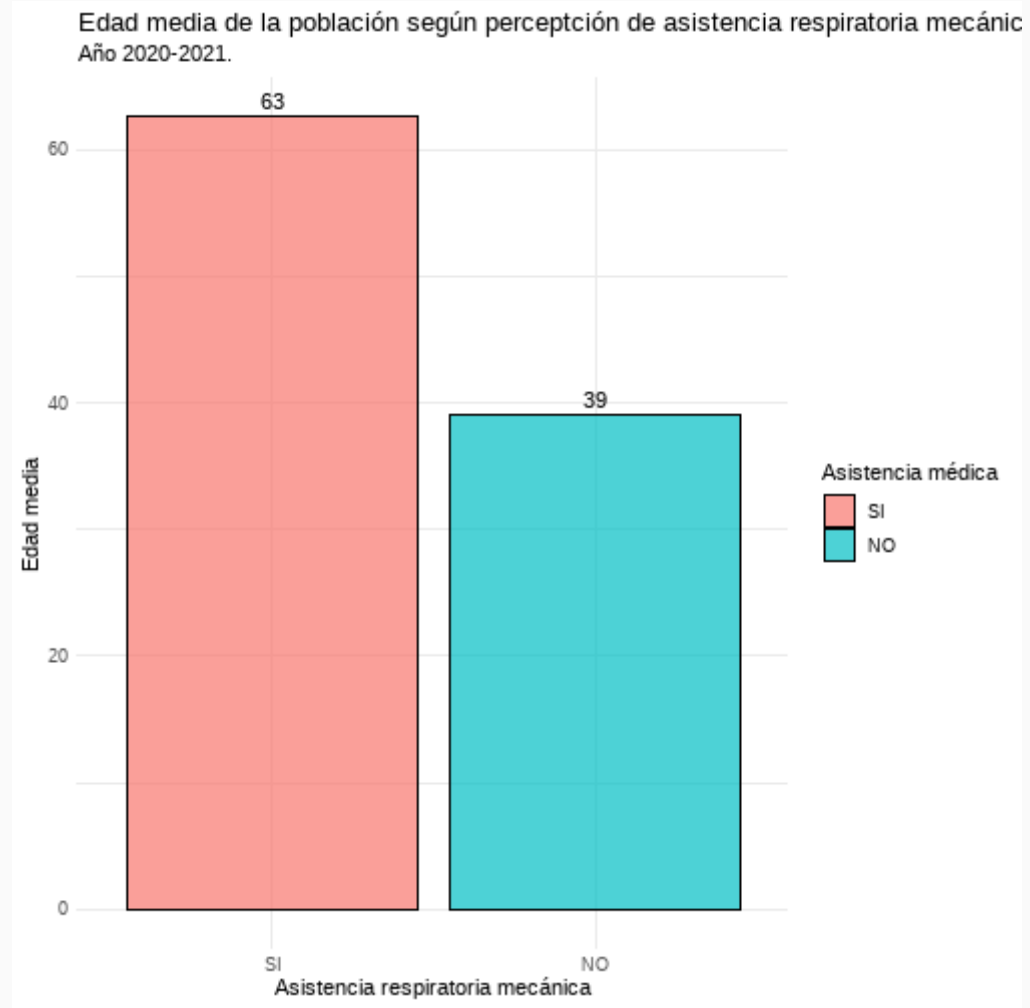
geom_col + factor

```
tabla_edad %>%  
  mutate(asistencia_mec = factor(asistencia_mec,  
                                  levels = c("SI", "NO"),  
                                  ordered = TRUE)) +  
  ggplot(mapping = aes(x = asistencia_mec,  
                        y = edad_media)) +  
  geom_col(aes(fill = asistencia_mec),  
           colour = "black",  
           alpha = 0.7) +  
  labs(title = "Edad media de la población según percepción de asistencia respiratoria mecánica",  
        subtitle = "Año 2020-2021.",  
        x = "Asistencia respiratoria mecánica",  
        y = "Edad media",  
        fill = "Asistencia médica") +  
  geom_text(aes(label = round(edad_media)),  
            vjust = -0.5)
```



geom_col + factor

```
tabla_edad %>%  
  mutate(asistencia_mec = factor(asistencia_mec,  
                                  levels = c("SI", "NO"),  
                                  ordered = TRUE)) +  
  ggplot(mapping = aes(x = asistencia_mec,  
                        y = edad_media)) +  
  geom_col(aes(fill = asistencia_mec),  
           colour = "black",  
           alpha = 0.7) +  
  labs(title = "Edad media de la población según percepción de asistencia respiratoria mecánica",  
        subtitle = "Año 2020-2021.",  
        x = "Asistencia respiratoria mecánica",  
        y = "Edad media",  
        fill = "Asistencia médica") +  
  geom_text(aes(label = round(edad_media)),  
            vjust = -0.5) +  
  theme_minimal()
```



PRÁCTICA

Práctica

- Dado el siguiente código que devuelve la cantidad de casos para las provincias seleccionadas, responder **visualmente** a la pregunta: ¿Cuál es la provincia que registra más casos?:

```
tabla_provincia <- base_covid %>%  
  rename(provincia = residencia_provincia_nombre) %>%  
  filter(provincia %in% c("San Luis", "San Juan", "Santiago del Estero")) %>%  
  count(provincia)
```

1. Hacer un gráfico de barras que muestre la cantidad de casos por provincia
2. Agregarle **color** a las barras en base a la variable **provincia**
3. Agregarle los textos de: **Título, Ejes x e y, Leyenda.**

geom_col + faceteado + flípeo

base_covid

```
# A tibble: 182,680 x 25
```

	id_evento_caso	sexo	edad	edad_años_meses	residencia_pais_~	reside
	<dbl>	<chr>	<dbl>	<chr>	<chr>	<chr>
1	748361	NR	23	Años	Líbano	SIN ES
2	748780	F	53	Años	Argentina	CABA
3	751658	M	44	Años	Argentina	CABA
4	755897	F	29	Años	Argentina	CABA
5	756503	M	54	Años	Argentina	CABA
6	758578	M	2	Años	Argentina	CABA
7	762704	M	41	Años	Argentina	CABA
8	763097	M	53	Años	Argentina	CABA
9	764087	F	70	Años	Argentina	CABA
10	765127	M	30	Años	Argentina	CABA

```
# ... with 182,670 more rows, and 19 more variables:
```

```
# residencia_departamento_nombre <chr>, carga_provincia_nombre <chr>
# fecha_inicio_sintomas <date>, fecha_apertura <date>, sepi_apertura
# fecha_internacion <date>, cuidado_intensivo <chr>,
# fecha_cui_intensivo <lgl>, fallecido <chr>, fecha_fallecimiento <c
# asistencia_respiratoria_mecanica <chr>, carga_provincia_id <chr>,
# origen_financiamiento <chr>, clasificacion <chr>,
# clasificacion_resumen <chr>, residencia_provincia_id <chr>,
# fecha_diagnostico <date>, residencia_departamento_id <chr>,
# ultima_actualizacion <date>
```

geom_col + faceteado + flípeo

```
base_covid %>%
```

```
  rename(provincia = residencia_provincia_nombre)
```

```
# A tibble: 182,680 x 25
```

	id_evento_caso	sexo	edad	edad_años_meses	residencia_pais_nom~	pro
	<dbl>	<chr>	<dbl>	<chr>	<chr>	<ch
1	748361	NR	23	Años	Líbano	SIM
2	748780	F	53	Años	Argentina	CAB
3	751658	M	44	Años	Argentina	CAB
4	755897	F	29	Años	Argentina	CAB
5	756503	M	54	Años	Argentina	CAB
6	758578	M	2	Años	Argentina	CAB
7	762704	M	41	Años	Argentina	CAB
8	763097	M	53	Años	Argentina	CAB
9	764087	F	70	Años	Argentina	CAB
10	765127	M	30	Años	Argentina	CAB

```
# ... with 182,670 more rows, and 19 more variables:
```

```
#   residencia_departamento_nombre <chr>, carga_provincia_nombre <chr>
#   fecha_inicio_sintomas <date>, fecha_apertura <date>, sepi_apertura
#   fecha_internacion <date>, cuidado_intensivo <chr>,
#   fecha_cui_intensivo <lgl>, fallecido <chr>, fecha_fallecimiento <d
#   asistencia_respiratoria_mecanica <chr>, carga_provincia_id <chr>,
#   origen_financiamiento <chr>, clasificacion <chr>,
#   clasificacion_resumen <chr>, residencia_provincia_id <chr>,
#   fecha_diagnostico <date>, residencia_departamento_id <chr>,
#   ultima_actualizacion <date>
```

geom_col + faceteado + flípeo

```
base_covid %>%
```

```
  rename(provincia = residencia_provincia_nombre) %>%
```

```
  filter(provincia %in% c("San Luis", "San Juan", 'San
```

```
# A tibble: 8,302 x 25
```

	id_evento_caso	sexo	edad	edad_años_meses	residencia_pais_no~	prov
	<dbl>	<chr>	<dbl>	<chr>	<chr>	<chr>
1	923273	M	46	Años	Argentina	San
2	2719838	M	46	Años	Argentina	San
3	2932651	M	38	Años	Argentina	San
4	2955548	M	72	Años	Argentina	San
5	2957124	M	72	Años	Argentina	San
6	3099994	M	26	Años	Argentina	San
7	3132693	M	33	Años	Argentina	San
8	7603084	F	21	Años	Argentina	Sant
9	9961441	F	25	Años	Argentina	San
10	807392	M	34	Años	Argentina	San

```
# ... with 8,292 more rows, and 19 more variables:
```

```
#   residencia_departamento_nombre <chr>, carga_provincia_nombre <chr>
#   fecha_inicio_sintomas <date>, fecha_apertura <date>, sepi_apertura
#   fecha_internacion <date>, cuidado_intensivo <chr>,
#   fecha_cui_intensivo <lgl>, fallecido <chr>, fecha_fallecimiento <d
#   asistencia_respiratoria_mecanica <chr>, carga_provincia_id <chr>,
#   origen_financiamiento <chr>, clasificacion <chr>,
#   clasificacion_resumen <chr>, residencia_provincia_id <chr>,
#   fecha_diagnostico <date>, residencia_departamento_id <chr>,
#   ultima_actualizacion <date>
```

geom_col + faceteado + flipeo

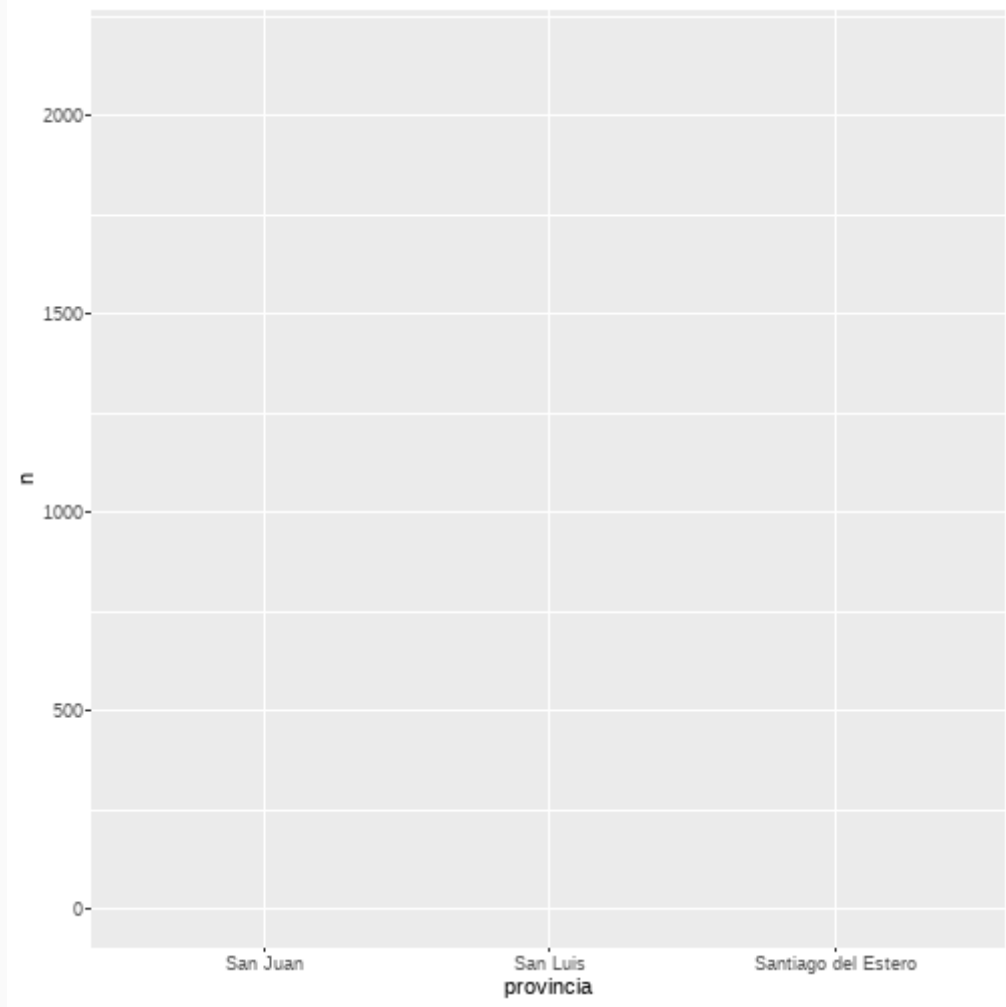
```
base_covid %>%  
  rename(provincia = residencia_provincia_nombre) %  
  filter(provincia %in% c("San Luis", "San Juan", '  
  count(provincia, sexo)
```

A tibble: 9 x 3

	provincia	sexo	n
	<chr>	<chr>	<int>
1	San Juan	F	787
2	San Juan	M	807
3	San Juan	NR	15
4	San Luis	F	2156
5	San Luis	M	1926
6	San Luis	NR	9
7	Santiago del Estero	F	1239
8	Santiago del Estero	M	1317
9	Santiago del Estero	NR	46

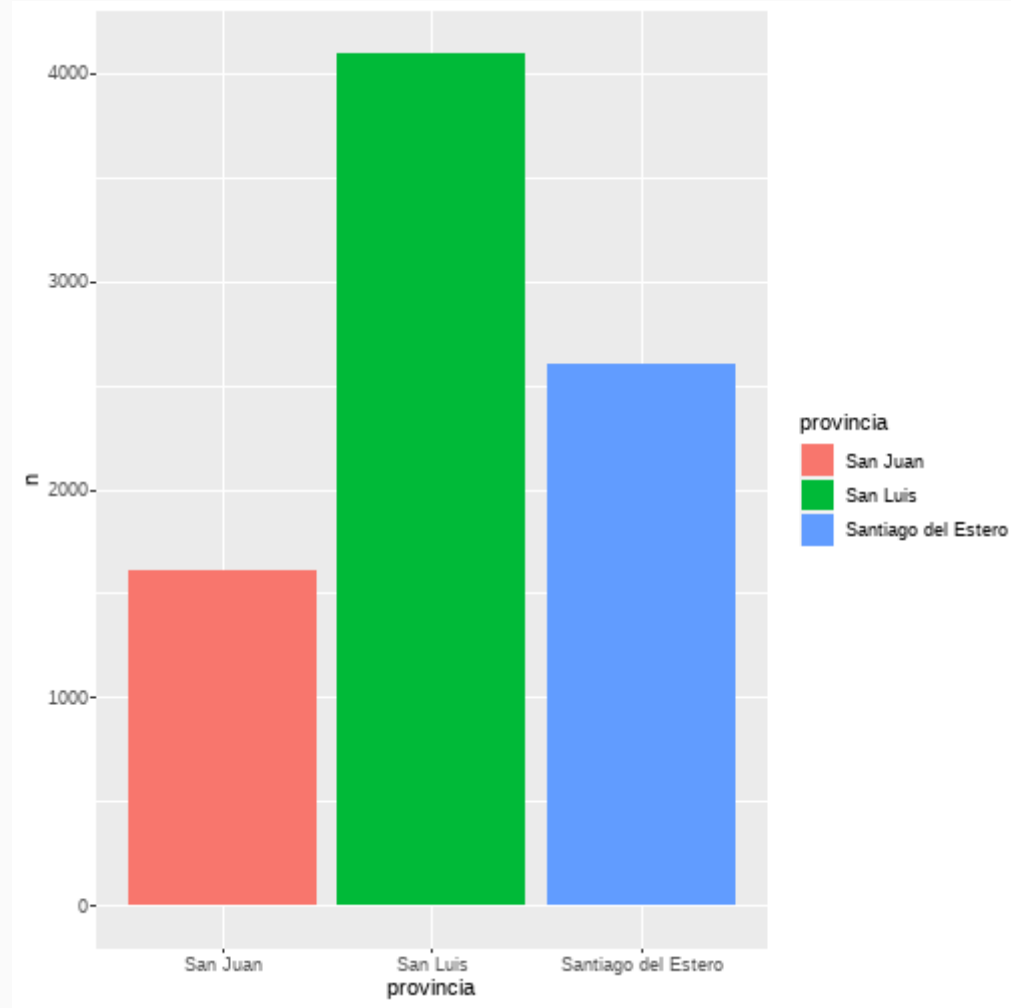
geom_col + faceteado + flipeo

```
base_covid %>%  
  rename(provincia = residencia_provincia_nombre) %  
  filter(provincia %in% c("San Luis", "San Juan", '  
  count(provincia, sexo) %>%  
  ggplot(mapping = aes(x = provincia,  
                        y = n))
```



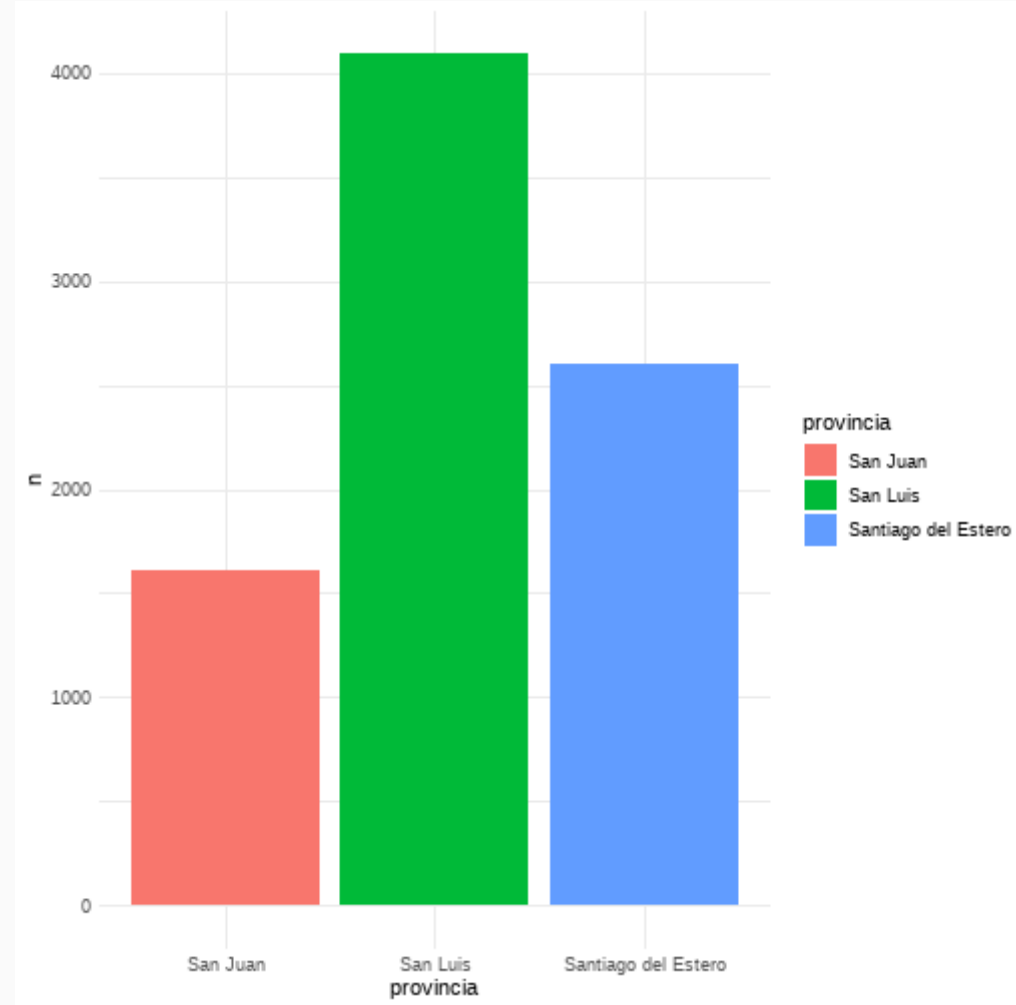
geom_col + faceteado + flípeo

```
base_covid %>%
  rename(provincia = residencia_provincia_nombre) %
  filter(provincia %in% c("San Luis", "San Juan", '
  count(provincia, sexo) %>%
  ggplot(mapping = aes(x = provincia,
                        y = n)) +
  geom_col(aes(fill = provincia))
```



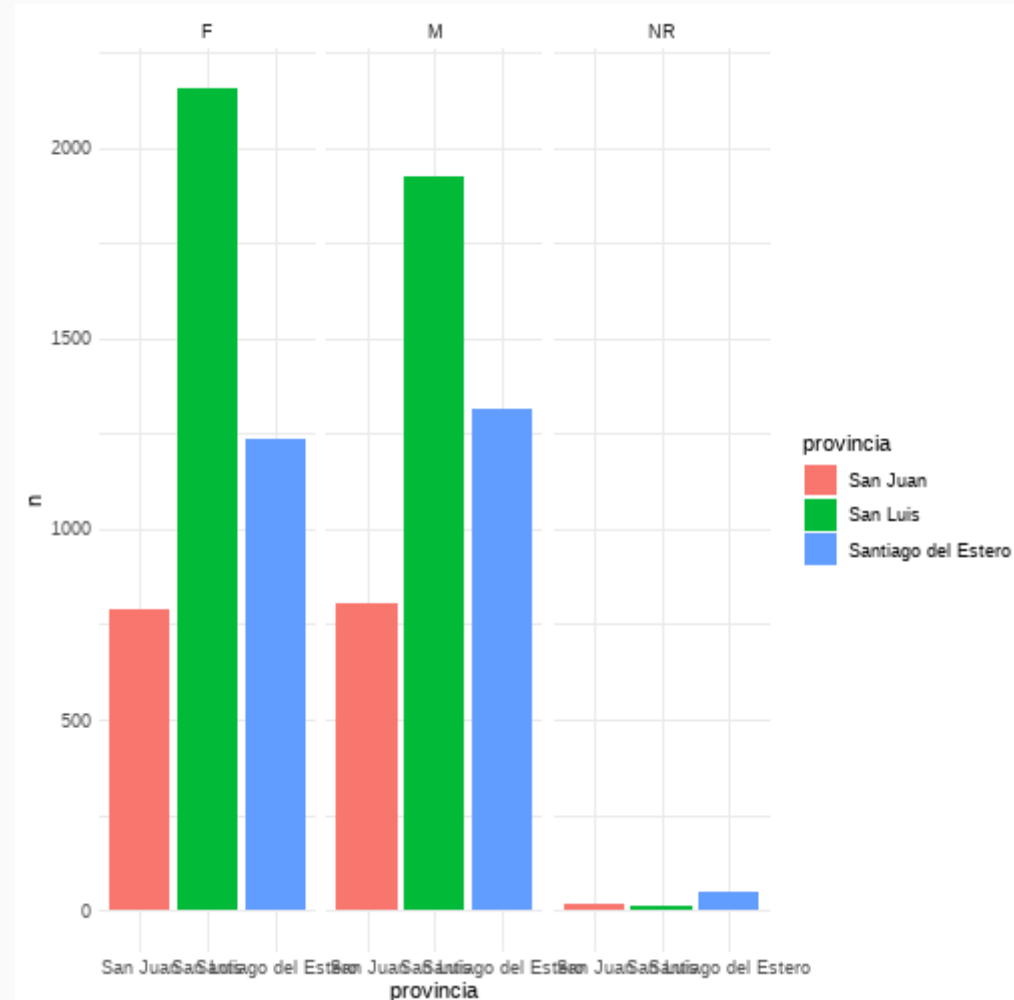
geom_col + faceteado + flípeo

```
base_covid %>%  
  rename(provincia = residencia_provincia_nombre) %  
  filter(provincia %in% c("San Luis", "San Juan", '  
  count(provincia, sexo) %>%  
  ggplot(mapping = aes(x = provincia,  
                        y = n)) +  
  geom_col(aes(fill = provincia)) +  
  theme_minimal()
```



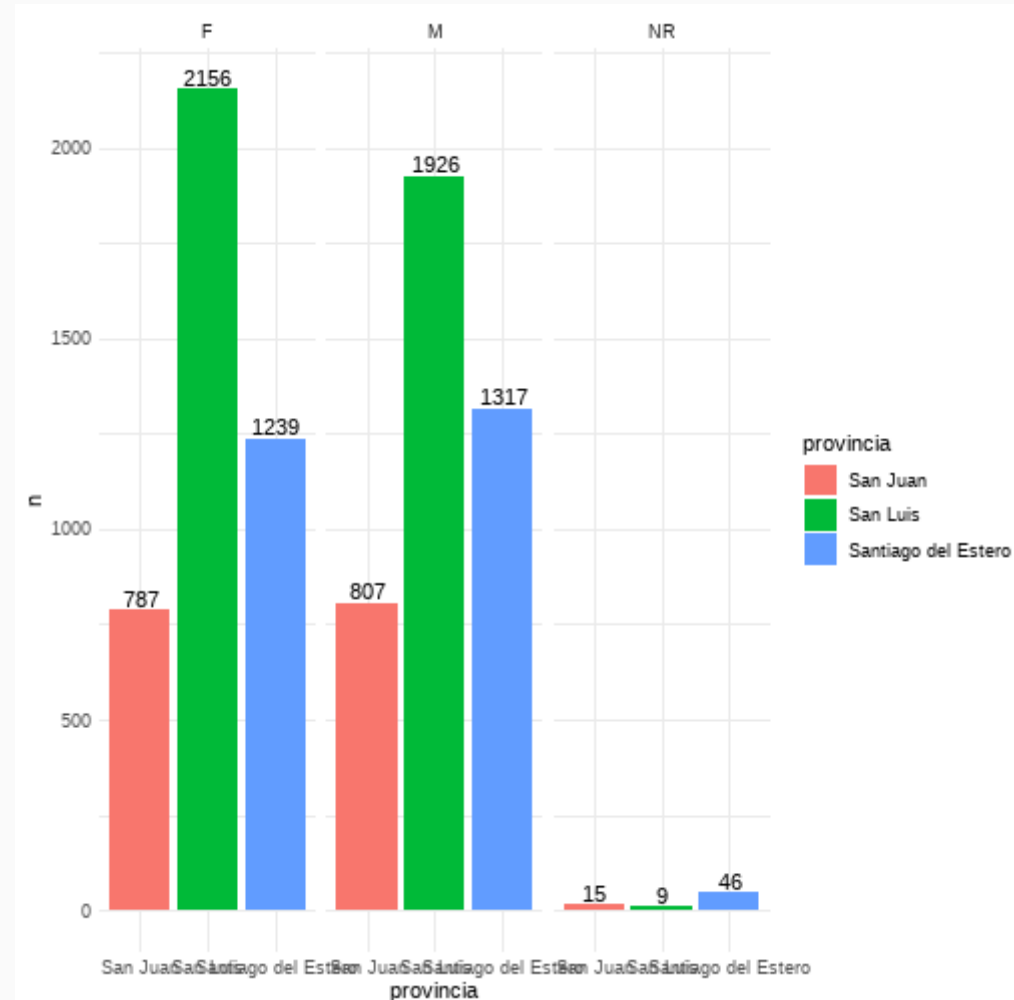
geom_col + faceteado + flipeo

```
base_covid %>%  
  rename(provincia = residencia_provincia_nombre) %  
  filter(provincia %in% c("San Luis", "San Juan", 'Santiago del Estero')) %>%  
  count(provincia, sexo) %>%  
  ggplot(mapping = aes(x = provincia,  
                        y = n)) +  
  geom_col(aes(fill = provincia)) +  
  theme_minimal() +  
  facet_wrap(facets = "sexo")
```



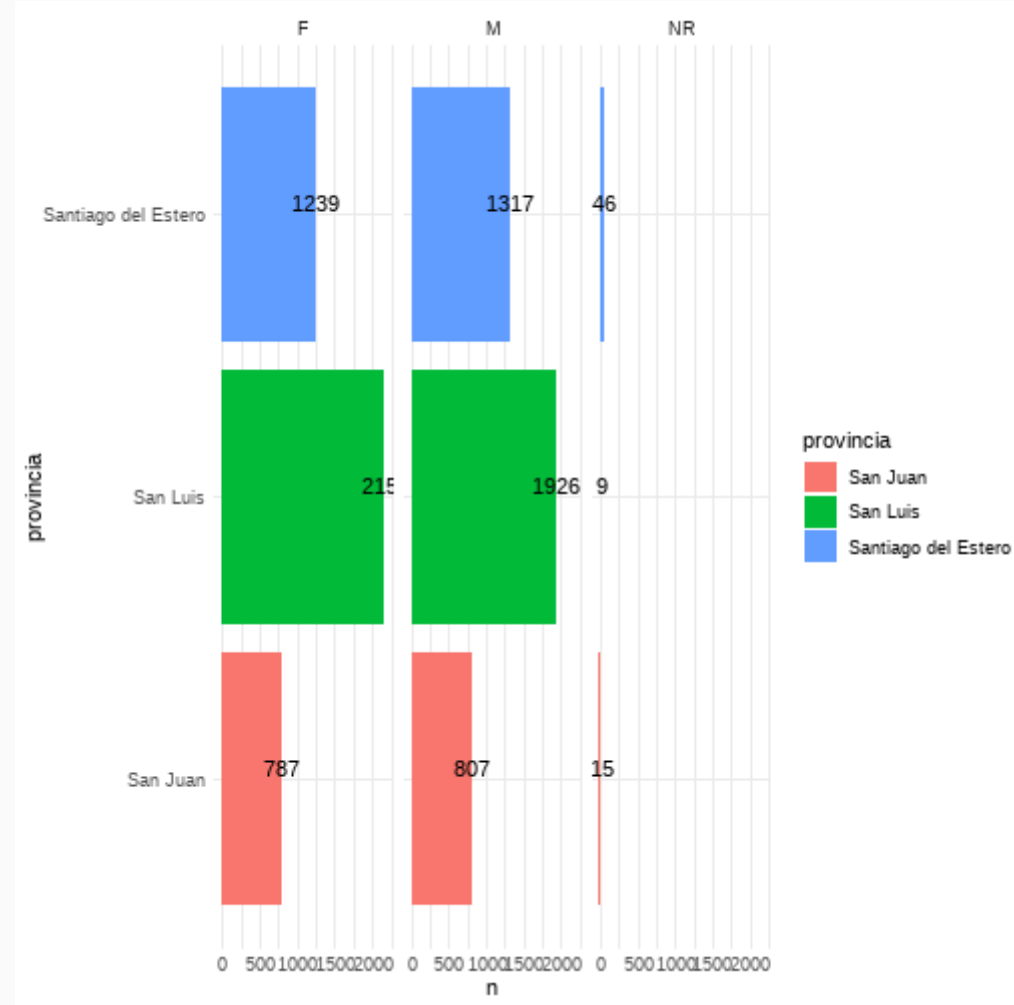
geom_col + faceteado + flipeo

```
base_covid %>%
  rename(provincia = residencia_provincia_nombre) %
  filter(provincia %in% c("San Luis", "San Juan", "Santiago del Estero")) %
  count(provincia, sexo) %>%
  ggplot(mapping = aes(x = provincia,
                        y = n)) +
  geom_col(aes(fill = provincia)) +
  theme_minimal() +
  facet_wrap(facets = "sexo") +
  geom_text(aes(label = n,
                vjust = -0.25))
```



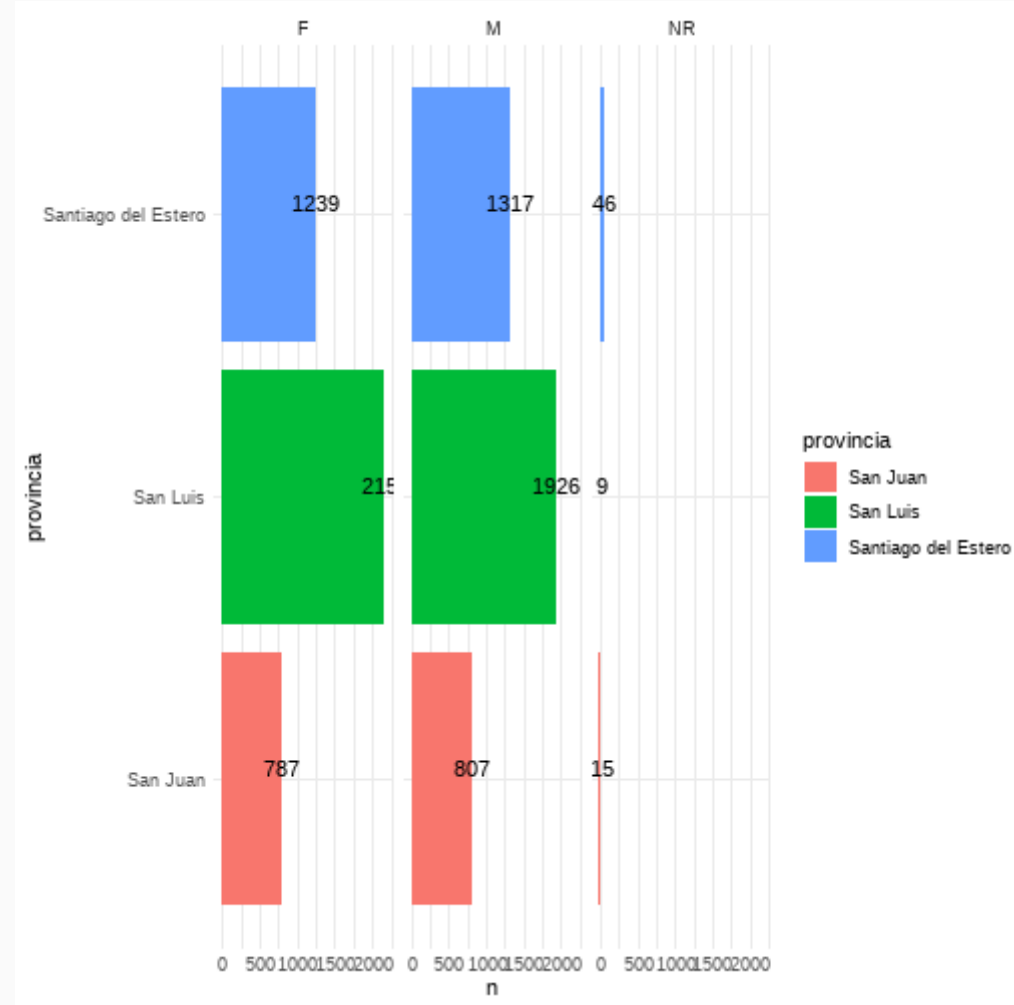
geom_col + faceteado + flipeo

```
base_covid %>%
  rename(provincia = residencia_provincia_nombre) %
  filter(provincia %in% c("San Luis", "San Juan", '
  count(provincia, sexo) %>%
  ggplot(mapping = aes(x = provincia,
                        y = n)) +
  geom_col(aes(fill = provincia)) +
  theme_minimal() +
  facet_wrap(facets = "sexo") +
  geom_text(aes(label = n),
            vjust = -0.25) +
  coord_flip()
```



geom_col + faceteado + flipeo

```
base_covid %>%
  rename(provincia = residencia_provincia_nombre) %
  filter(provincia %in% c("San Luis", "San Juan", '
  count(provincia, sexo) %>%
  ggplot(mapping = aes(x = provincia,
                        y = n)) +
  geom_col(aes(fill = provincia)) +
  theme_minimal() +
  facet_wrap(facets = "sexo") +
  geom_text(aes(label = n),
            vjust = -0.25) +
  coord_flip()
```



PRÁCTICA

Práctica

2). Identificar el error en la siguiente sentencia, corregirlo y correr el gráfico:

```
tabla_ejercicio <- base_covid %>%  
  rename(provincia = residencia_provincia_nombre) %>%  
  filter(provincia %in% c("Córdoba", "Salta", "La Pampa")) %>%  
  count(provincia, sexo)  
  
ggplot(data = provincia,  
       mapping = aes(x = tabla_ejercicio,  
                     y = n)) +  
  geom_col(aes(fill = provincia)) +  
  theme_minimal() +  
  facet_wrap(facets = "sexo") +  
  geom_text(aes(label = n),  
           vjust = -0.25)
```

Práctica

2). Identificar el error en la siguiente sentencia, corregirlo y correr el gráfico:

```
tabla_ejercicio_2 <- base_covid %>%
  rename(provincia = residencia_provincia_nombre) %>%
  filter(provincia %in% c("Córdoba", "Salta", "La Pampa")) %>%
  count(provincia, origen_financiamiento) %>%
  group_by(provincia) %>%
  mutate(peso = round(n / sum(n) * 100, digits = 2))

ggplot(data = tabla_ejercicio_2,
       mapping = aes(h = origen_financiamiento,
                     y = peso)) +
  geom_col(aes(relleno = origen_financiamiento)) +
  facet_wrap(facets = "provincia") +
  geom_text(aes(label = peso),
           vjust = -0.25) +
  theme_minimal()
```