County data - EDA

Tural Sadigov

8/10/22

Load libraries and R data

```
# our new universe
library(tidyverse)
```

```
-- Attaching packages --------------------------------------- tidyverse 1.3.2 --
v ggplot2 3.3.6     v purrr   0.3.4
v tibble  3.1.8     v dplyr   1.0.9
v tidyr   1.2.0     v stringr 1.4.0
v readr   2.1.2     v forcats 0.5.1
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
```

```
# load from R data file
load("county.rda")

# reading from CSV
county2 = read_csv('county.csv')
```

```
Rows: 3142 Columns: 15
-- Column specification --------------------------------------------------------
Delimiter: ","
chr  (5): name, state, metro, median_edu, smoking_ban
dbl (10): pop2000, pop2010, pop2017, pop_change, poverty, homeownership, mul...

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
  as_tibble(county2 == county)
```

```
# A tibble: 3,142 x 15
   name  state pop2000 pop2010 pop2017 pop_cha~1 poverty homeo~2 multi~3 unemp~4
   <lgl> <lgl> <lgl>   <lgl>   <lgl>   <lgl>     <lgl>   <lgl>   <lgl>   <lgl>
 1 TRUE  TRUE  TRUE    TRUE    TRUE    TRUE      TRUE    TRUE    TRUE    TRUE
 2 TRUE  TRUE  TRUE    TRUE    TRUE    TRUE      TRUE    TRUE    TRUE    TRUE
 3 TRUE  TRUE  TRUE    TRUE    TRUE    TRUE      TRUE    TRUE    TRUE    TRUE
 4 TRUE  TRUE  TRUE    TRUE    TRUE    TRUE      TRUE    TRUE    TRUE    TRUE
 5 TRUE  TRUE  TRUE    TRUE    TRUE    TRUE      TRUE    TRUE    TRUE    TRUE
 6 TRUE  TRUE  TRUE    TRUE    TRUE    TRUE      TRUE    TRUE    TRUE    TRUE
 7 TRUE  TRUE  TRUE    TRUE    TRUE    TRUE      TRUE    TRUE    TRUE    TRUE
 8 TRUE  TRUE  TRUE    TRUE    TRUE    TRUE      TRUE    TRUE    TRUE    TRUE
 9 TRUE  TRUE  TRUE    TRUE    TRUE    TRUE      TRUE    TRUE    TRUE    TRUE
10 TRUE  TRUE  TRUE    TRUE    TRUE    TRUE      TRUE    TRUE    TRUE    TRUE
# ... with 3,132 more rows, 5 more variables: metro <lgl>, median_edu <lgl>,
#   per_capita_income <lgl>, median_hh_income <lgl>, smoking_ban <lgl>, and
#   abbreviated variable names 1: pop_change, 2: homeownership, 3: multi_unit,
#   4: unemployment_rate
# i Use `print(n = ...)` to see more rows, and `colnames()` to see all variable names
```

Change the data format to tibble (advanced data frame) and view portion of it.

```
  county <- as_tibble(county)
  county
```

```
# A tibble: 3,142 x 15
   name    state pop2000 pop2010 pop2017 pop_c~1 poverty homeo~2 multi~3 unemp~4
   <fct>   <fct>   <dbl>   <dbl>   <int>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
 1 Autaug~ Alab~   43671   54571   55504    1.48    13.7    77.5     7.2    3.86
 2 Baldwi~ Alab~  140415  182265  212628    9.19    11.8    76.7    22.6    3.99
 3 Barbou~ Alab~   29038   27457   25270   -6.22    27.2    68      11.1    5.9
 4 Bibb C~ Alab~   20826   22915   22668    0.73    15.2    82.9     6.6    4.39
 5 Blount~ Alab~   51024   57322   58013    0.68    15.6    82       3.7    4.02
 6 Bulloc~ Alab~   11714   10914   10309   -2.28    28.5    76.9     9.9    4.93
 7 Butler~ Alab~   21399   20947   19825   -2.69    24.4    69      13.7    5.49
 8 Calhou~ Alab~  112249  118572  114728   -1.51    18.6    70.7    14.3    4.93
 9 Chambe~ Alab~   36583   34215   33713   -1.2     18.8    71.4     8.7    4.08
10 Cherok~ Alab~   23988   25989   25857   -0.6     16.1    77.5     4.3    4.05
```

```
# ... with 3,132 more rows, 5 more variables: metro <fct>, median_edu <fct>,
#   per_capita_income <dbl>, median_hh_income <int>, smoking_ban <fct>, and
#   abbreviated variable names 1: pop_change, 2: homeownership, 3: multi_unit,
#   4: unemployment_rate
# i Use `print(n = ...)` to see more rows, and `colnames()` to see all variable names
```

Select or pull out some variables to display them.

```
# select a feture/variable and displayfirst 10 elements
county %>%
  select(pop2000) %>%
  slice(1:10)
```

```
# A tibble: 10 x 1
   pop2000
     <dbl>
 1   43671
 2  140415
 3   29038
 4   20826
 5   51024
 6   11714
 7   21399
 8  112249
 9   36583
10   23988
```

```
# pull out the whole feature as vector/array
county %>%
  pull(pop2000) %>%
  head(10)
```

```
 [1]  43671 140415  29038  20826  51024  11714  21399 112249  36583  23988
```

We can count frequencies (numerical summary) of each level in categorical variables.

```
# how many counties has no smoking ban? some smoking ban?
county %>%
  count(smoking_ban)
```

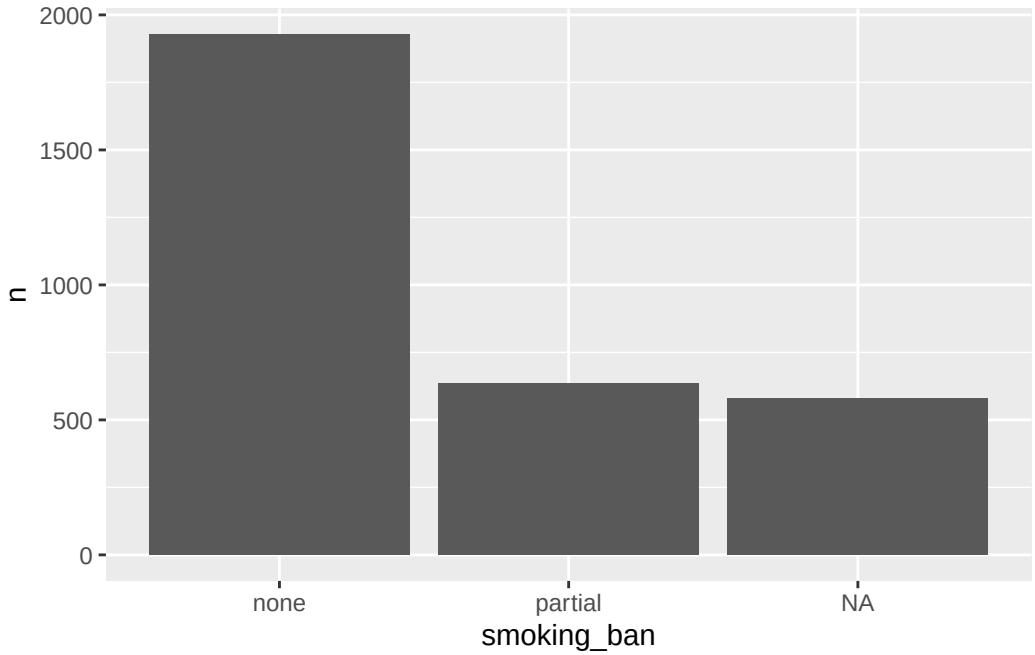
```
# A tibble: 3 x 2
  smoking_ban     n
  <fct>       <int>
1 none         1927
2 partial       635
3 <NA>          580
```

```
# how many counties has a metropolitan city in it?
county %>%
  count(metro)
```

```
# A tibble: 3 x 2
  metro     n
  <fct> <int>
1 no     1974
2 yes    1165
3 <NA>      3
```

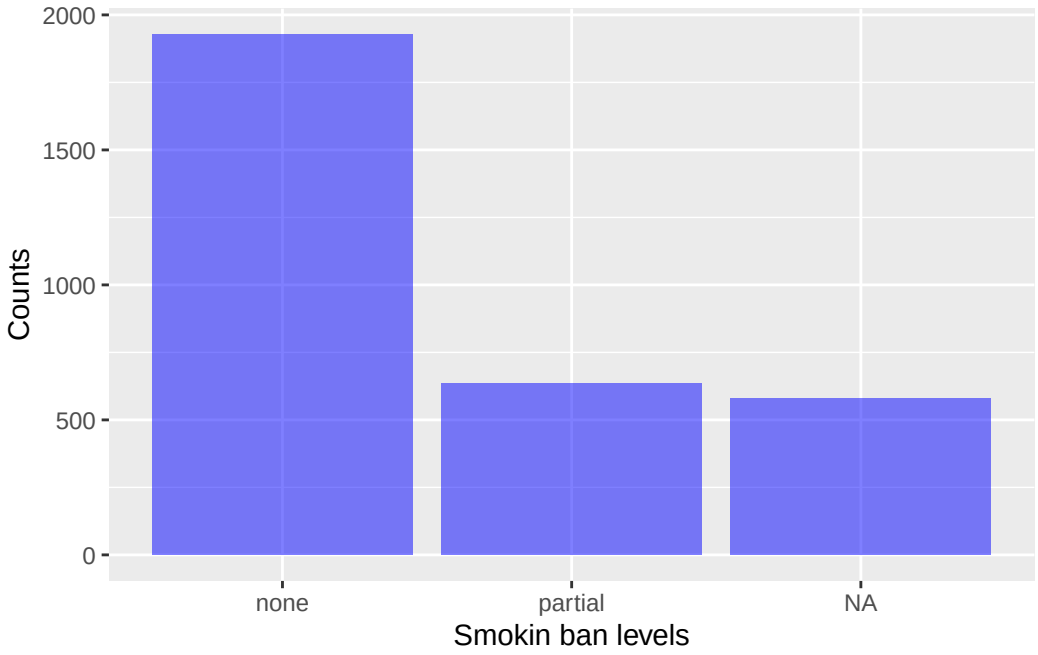
Display bar-plots for categorical variables above.

```
# how many counties has no smoking ban? some smoking ban?
county %>%
  count(smoking_ban) %>%
  ggplot(aes(x = smoking_ban, y = n)) +
  geom_col()
```



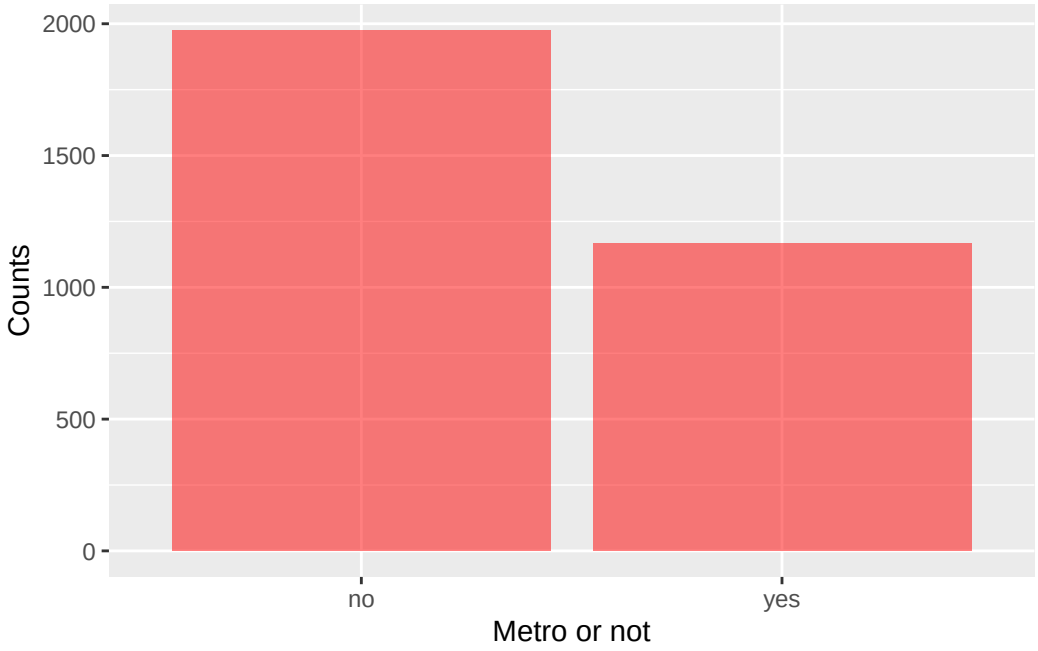
We can fill it with any color, change coordinate labels and even opaqueness.

```
# how many counties has no smoking ban? some smoking ban?
county %>%
  count(smoking_ban) %>%
  ggplot(aes(x = smoking_ban, y = n)) +
  geom_col(fill = 'blue', alpha = 0.5) +
  xlab('Smokin ban levels') +
  ylab('Counts')
```



Let's do this again for metro feature, but this time, lets drop rows with NAs.

```
county %>%
  count(metro) %>%
  drop_na() %>%
  ggplot(aes(x = metro, y = n)) +
  geom_col(fill = 'red', alpha = 0.5) +
  xlab('Metro or not') +
  ylab('Counts')
```



What if we would like to understand relationship between two categorical variables? Numerical summary.

```
county %>%
  count(metro, smoking_ban)
```

```
# A tibble: 8 x 3
  metro smoking_ban     n
  <fct> <fct>       <int>
1 no    none         1202
2 no    partial       413
3 no    <NA>          359
4 yes   none          723
5 yes   partial       222
6 yes   <NA>          220
7 <NA>  none            2
8 <NA>  <NA>            1
```

Drop NAs, and do it again.

```
county %>%
  drop_na() %>%
  count(metro, smoking_ban)
```

```
# A tibble: 4 x 3
  metro smoking_ban     n
  <fct> <fct>       <int>
1 no    none         1202
2 no    partial       413
3 yes   none          723
4 yes   partial       222
```

What about numerical variables? First, numerical summaries.

```
# quartiles and mean
county %>%
  pull(poverty) %>%
  summary()
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
   2.40   11.30   15.20   15.97   19.40   52.00       2
```

```
# spread: var
county %>%
  pull(poverty) %>%
  var(na.rm = T)
```

```
[1] 42.45412
```

```
# spread: sd
county %>%
  pull(poverty) %>%
  sd(na.rm = T)
```

```
[1] 6.515682
```

```
# spread: QR
county %>%
  pull(poverty) %>%
  IQR(na.rm = T)
```

```
[1] 8.1
```

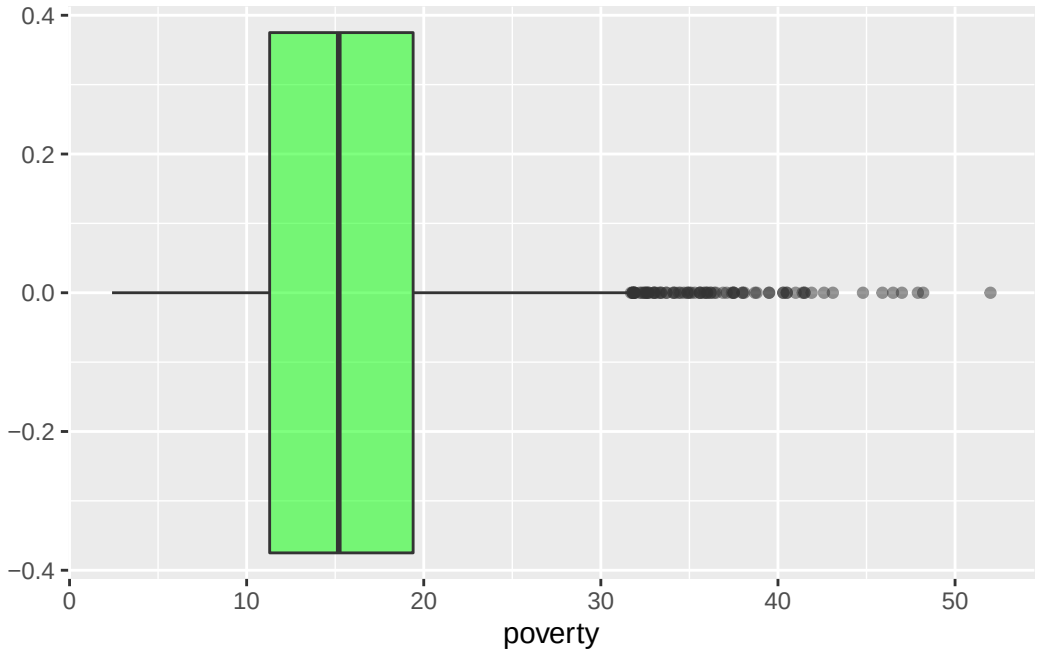
```
# spread: range
county %>%
  pull(poverty) %>%
  range(na.rm = T)
```

```
[1]  2.4 52.0
```

We can create distributions for numerical variables: dotplots, histograms, boxplots, smoothed histograms and more.

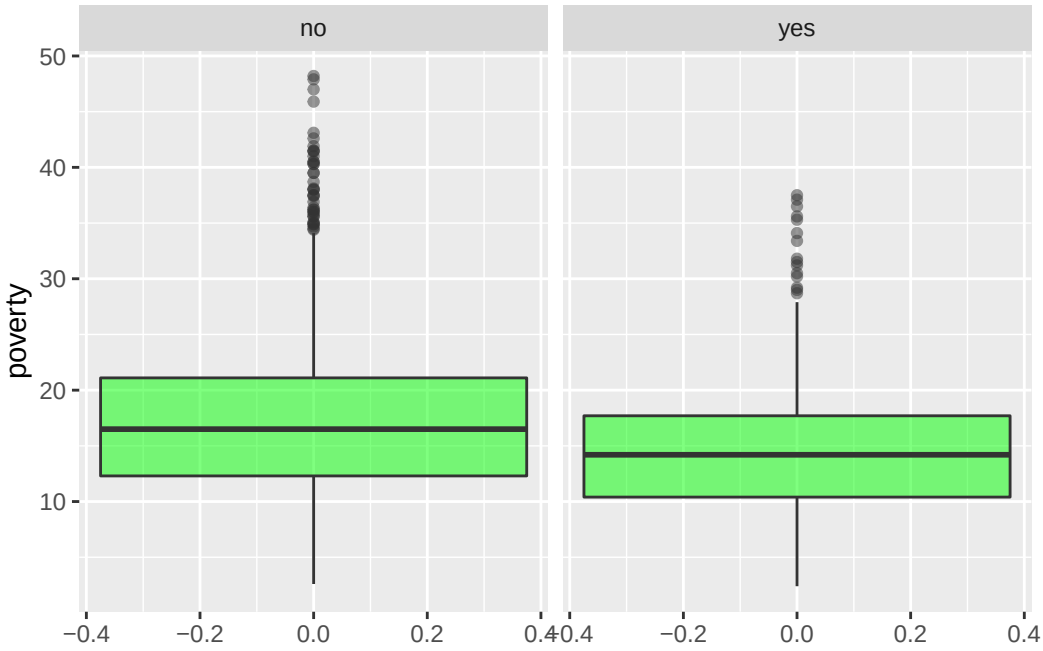
```
# boxplot
county %>%
  ggplot(aes(x = poverty)) +
  geom_boxplot(fill = 'green', alpha = 0.5)
```

```
Warning: Removed 2 rows containing non-finite values (stat_boxplot).
```



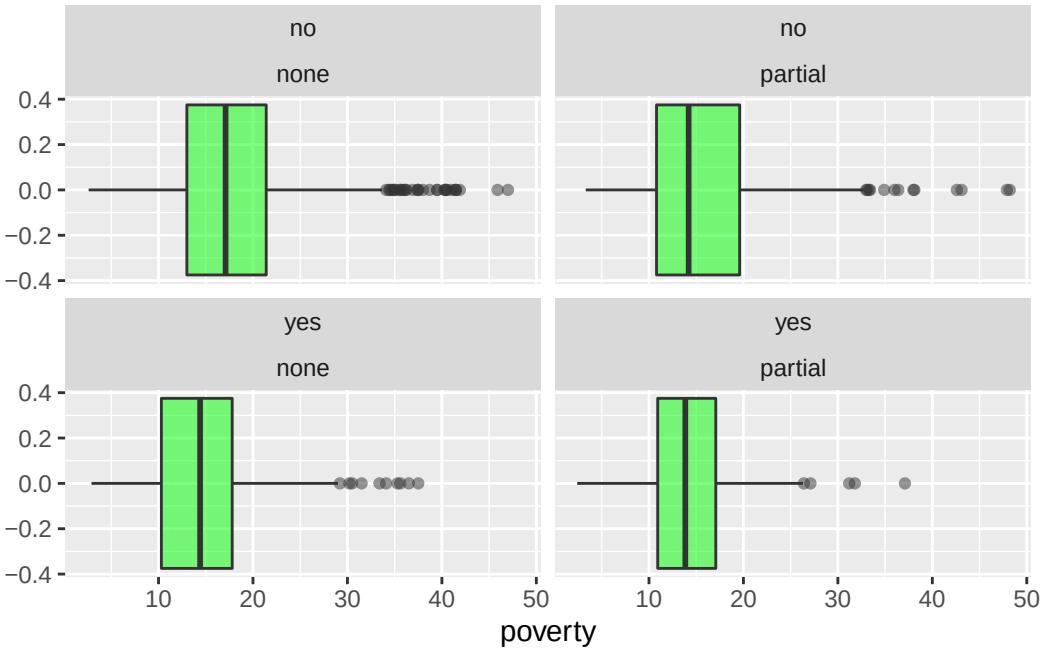
Boxplots for poverty with each level of metro (NAs dropped).

```
# boxplot - facet wrapped
county %>%
  drop_na() %>% # drop nas
  ggplot(aes(x = poverty)) +
  geom_boxplot(fill = 'green', alpha = 0.5) +
  facet_wrap(~metro) + # facet wrap with metro levels
  coord_flip() # make them vertical
```



We can facet wrap it with more than one categorical variable.

```
# boxplot - facet wrapped - twice!
county %>%
  drop_na() %>%
  ggplot(aes(x = poverty)) +
  geom_boxplot(fill = 'green', alpha = 0.5) +
  facet_wrap(~metro + smoking_ban) # facet wrap twice
```

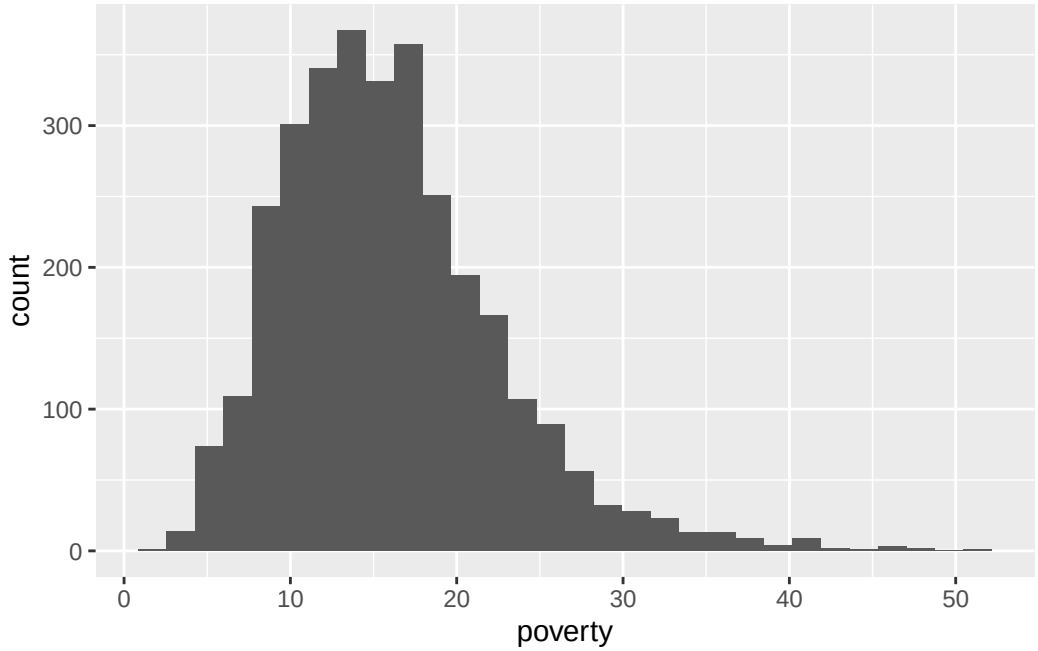


We could look at histogram for numerical variables.

```
# distribution - histogram
county %>%
  ggplot(aes(x = poverty)) +
  geom_histogram()
```

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

```
Warning: Removed 2 rows containing non-finite values (stat_bin).
```



Customize it.

```
# distribution - histogram
county %>%
  ggplot(aes(x = poverty)) +
  geom_histogram(aes(y = ..density..),
                 color = 'white',
                 fill = 'blue',
                 alpha = 0.5) +
  geom_density(lwd = 2, color = 'red')
```

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

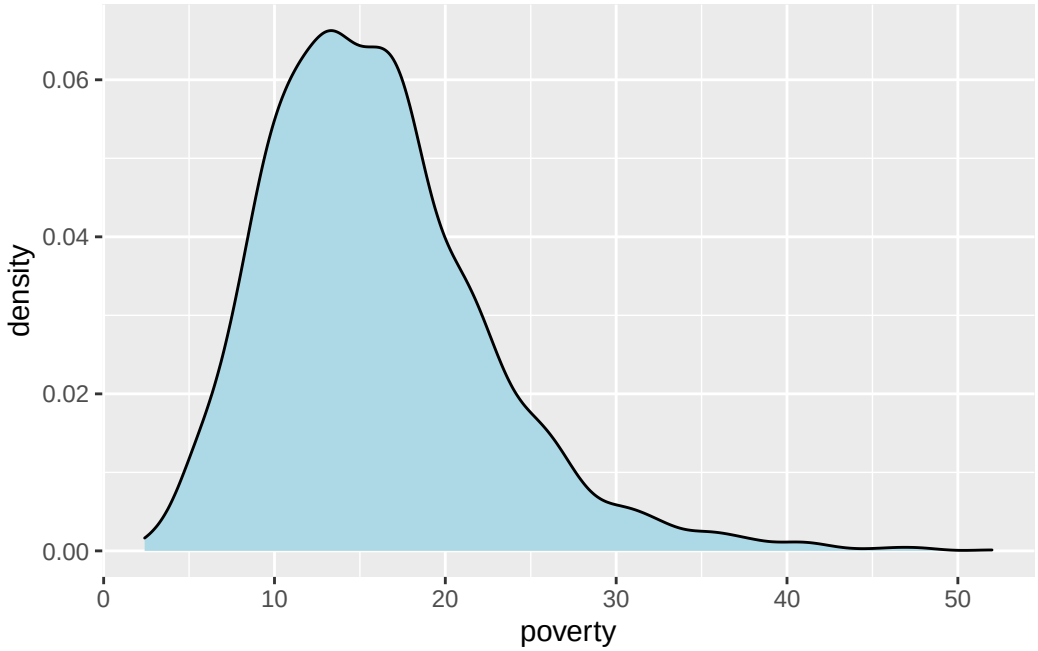
```
Warning: Removed 2 rows containing non-finite values (stat_bin).
```

```
Warning: Removed 2 rows containing non-finite values (stat_density).
```

Use only smoothed histogram (density) and fill it!

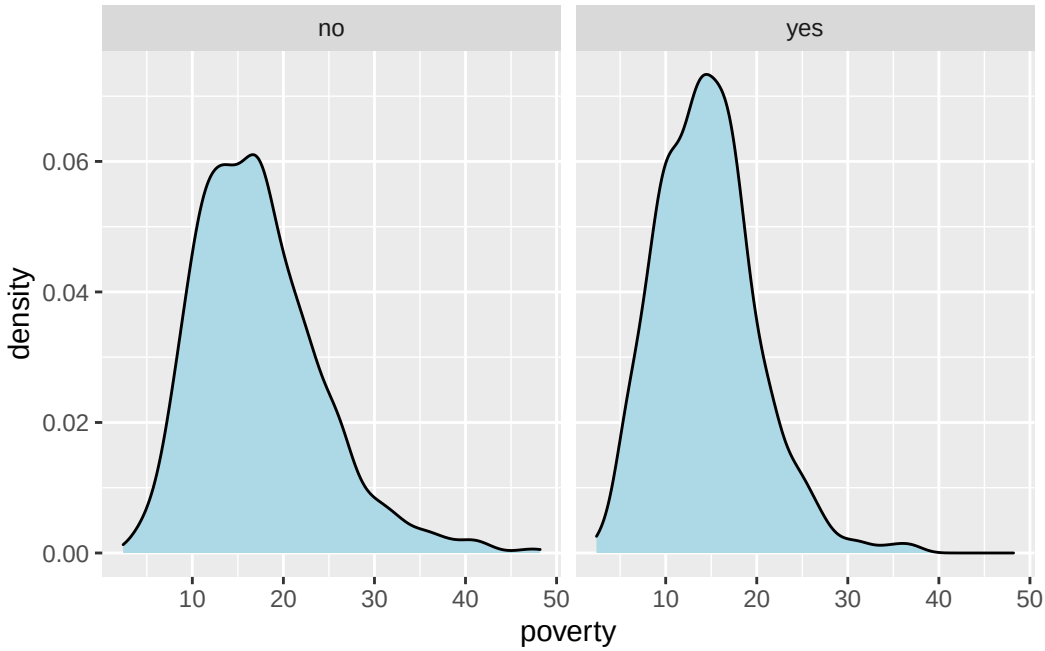
```
# distribution - histogram
county %>%
  ggplot(aes(x = poverty)) +
  geom_density(fill="lightblue")
```

```
Warning: Removed 2 rows containing non-finite values (stat_density).
```



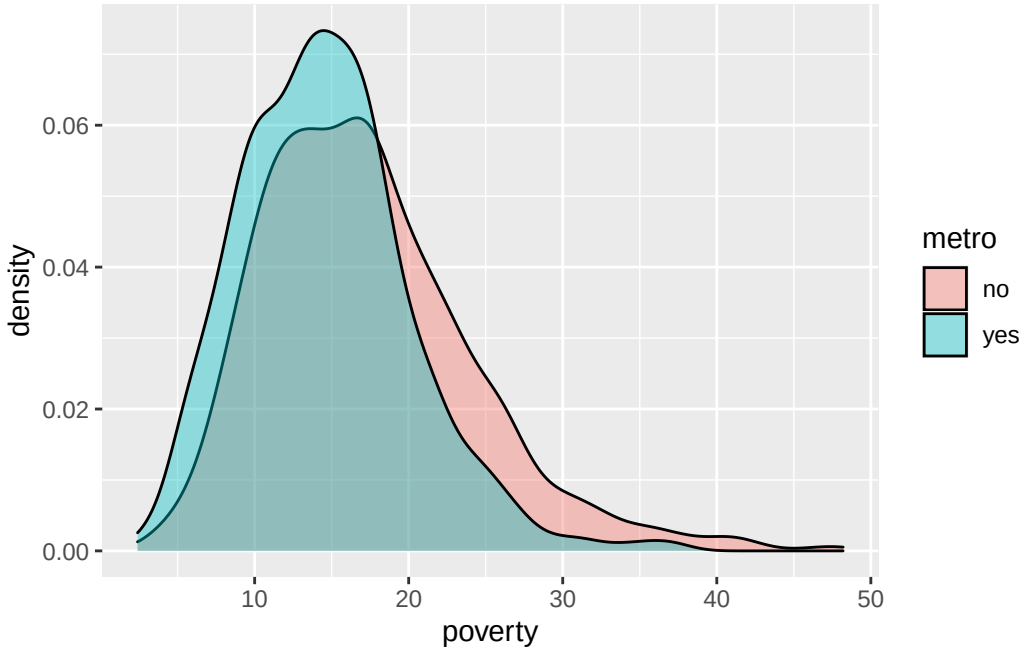
Can we facet-wrap it?

```
# distribution - histogram + facet wrap
county %>%
  drop_na() %>%
  ggplot(aes(x = poverty)) +
  geom_density(fill="lightblue") +
  facet_wrap(~metro)
```



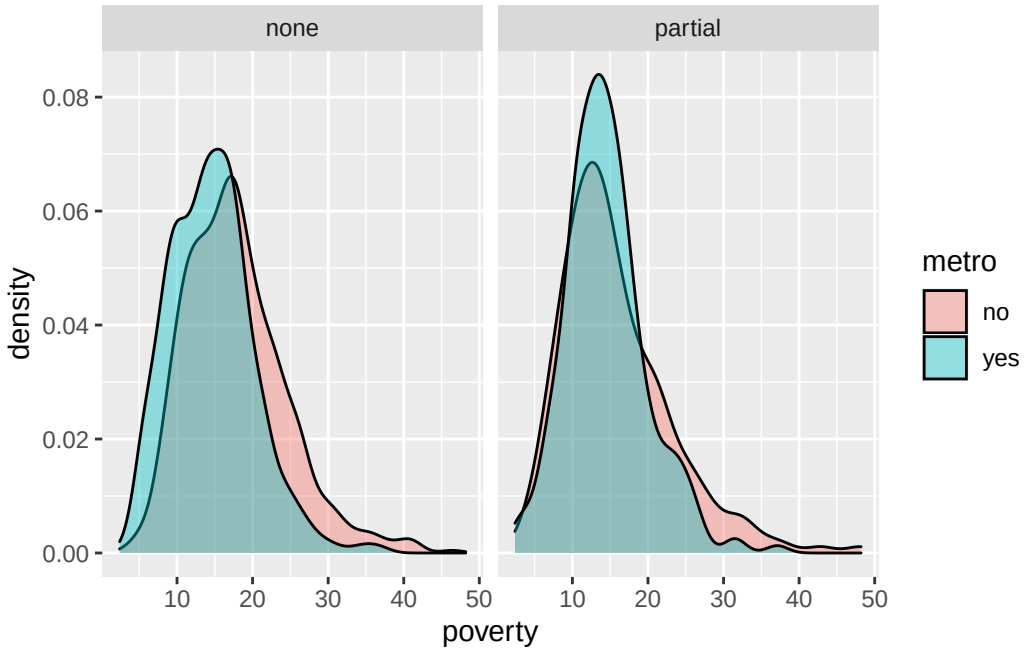
We can do better.

```
# distribution - histogram + facet wrap
county %>%
  drop_na() %>%
  ggplot(aes(x = poverty, fill= metro)) +
  geom_density(alpha = 0.4)
```



Now facet wrap it with smoking ban.

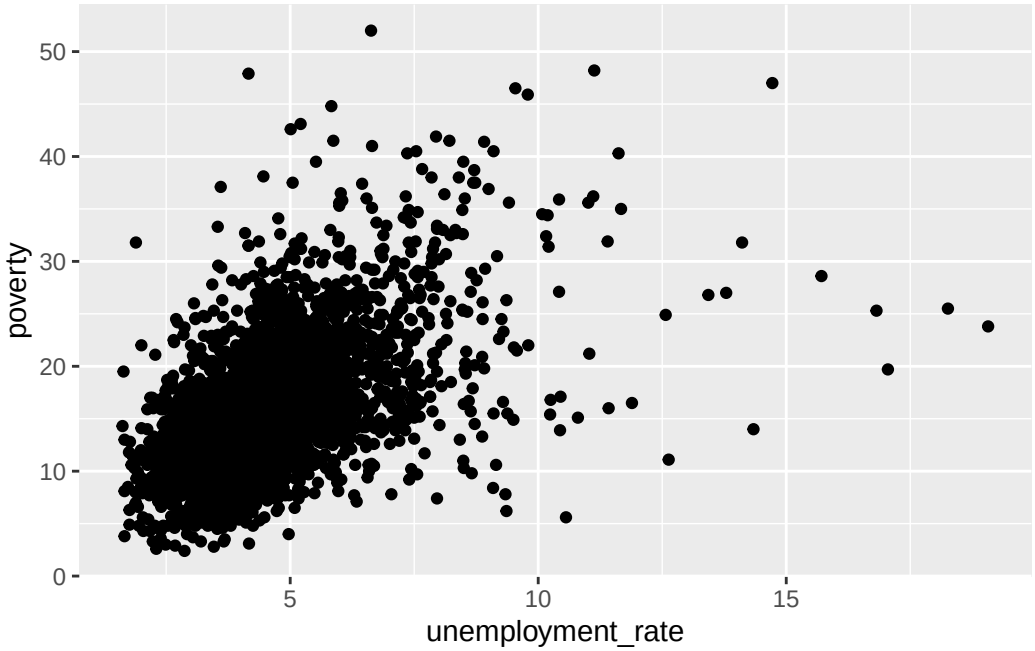
```
# distribution - histogram + facet wrap
county %>%
  drop_na() %>%
  ggplot(aes(x = poverty, fill= metro)) +
  geom_density(alpha = 0.4) +
  facet_wrap(~smoking_ban)
```



What about relationship between two numerical variables? Scatterplots.

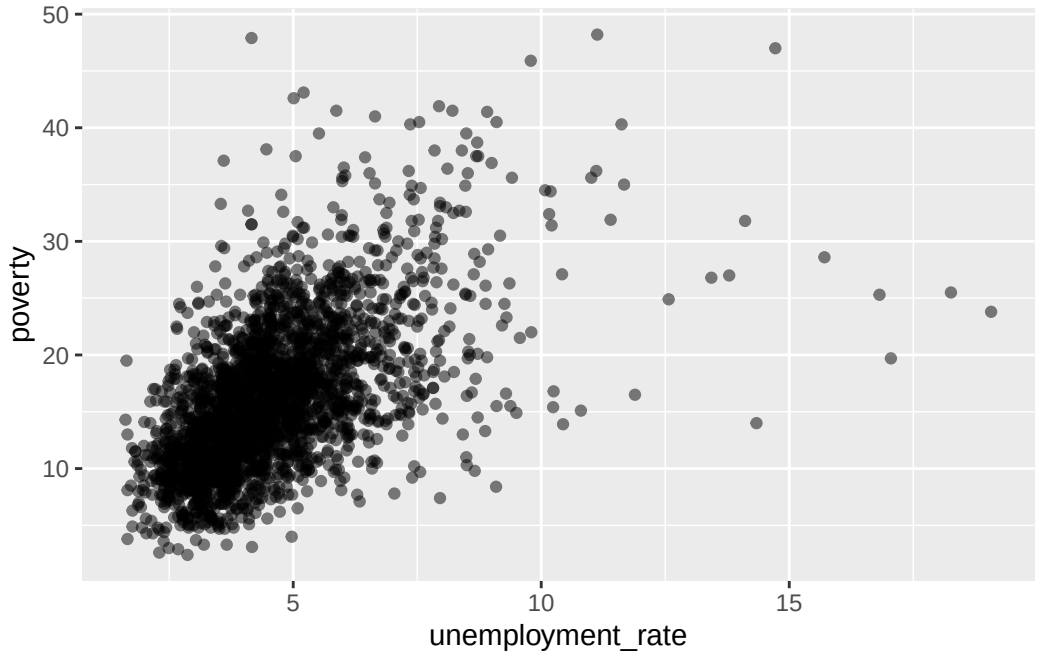
```
county %>%
  ggplot(aes(x = unemployment_rate, y = poverty)) +
  geom_point()
```

```
Warning: Removed 3 rows containing missing values (geom_point).
```



Add opaqueness.

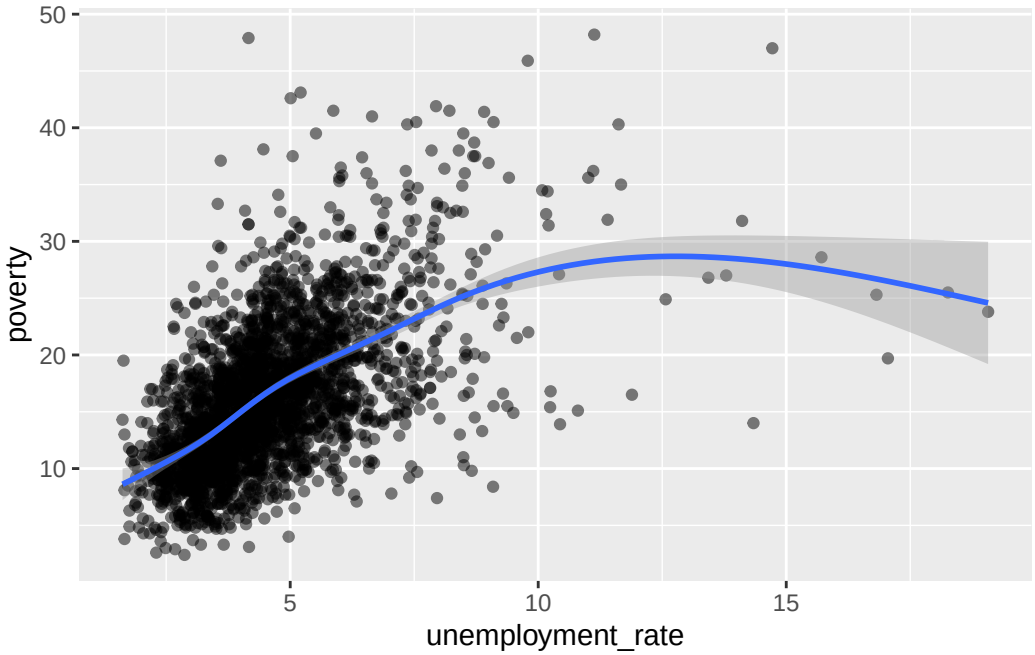
```
county %>%
  drop_na() %>%
  ggplot(aes(x = unemployment_rate, y = poverty)) +
  geom_point(alpha = 0.5)
```



Add trend.

```
county %>%
  drop_na() %>%
  ggplot(aes(x = unemployment_rate, y = poverty)) +
  geom_point(alpha = 0.5) +
  geom_smooth()
```

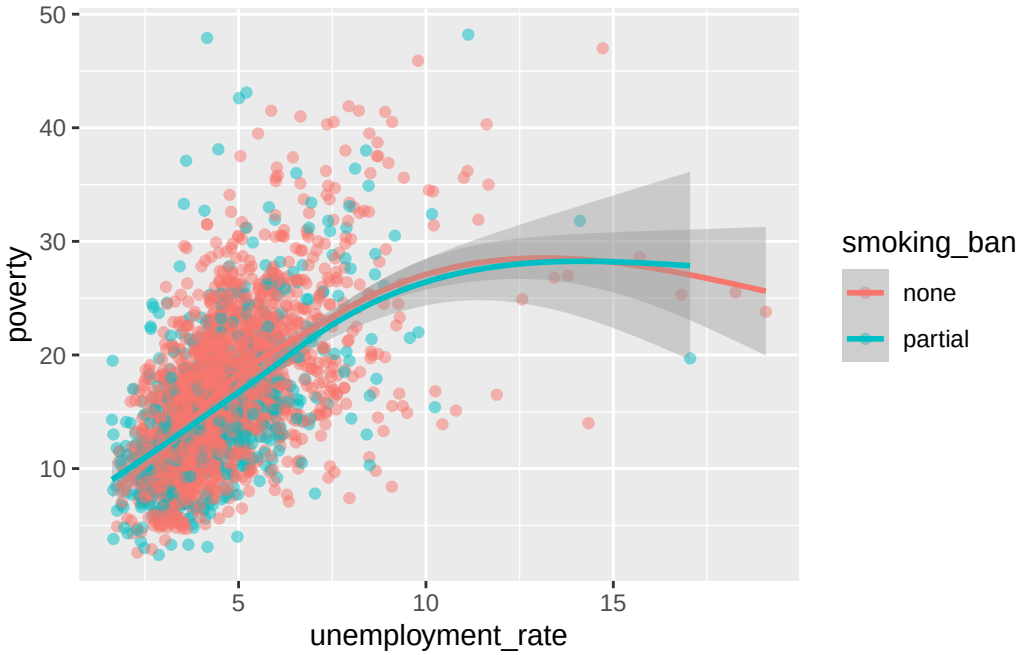
```
`geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



Color it according to metro.

```
county %>%
  drop_na() %>%
  ggplot(aes(x = unemployment_rate, y = poverty, color = smoking_ban)) +
  geom_point(alpha = 0.5) +
  geom_smooth()
```

```
`geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



Include smoking ban as color, and facet wrap it with metro.

```
county %>%
  drop_na() %>%
  ggplot(aes(x = unemployment_rate, y = poverty, color= smoking_ban)) +
  geom_point(alpha = 0.5) +
  geom_smooth(se = FALSE) +
  facet_wrap(~metro)
```

```
`geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```


no
yes
50
40
30
20
10
poverty
5
10
15
5
10
15
unemployment_rate
smoking_ban
none
partial