Wrapping Up Descriptives

Matt Lawrence

October 7, 2019

Today we will continue using data from Chetty et al's 2014 paper "Where Is The Land Of Opportunity?". The `commuting_zones.csv` file on Canvas comes from the Opportunity Insights website which can be accessed `here`.

Load the data as a data frame called `cz` and load the tidyverse and pander packages.

```
library(tidyverse)
library(pander)

cz <- read.csv("https://raw.githubusercontent.com/mjclawrence/soci385/master/data/commuting_zones.csv")
```

We'll look at a small subset of the data to start so let's pull the following variables into a new data frame called `cz_subset`:

- `mobility` = measure of absolute upward mobility
- `gini` = Gini coefficient of income inequality; higher gini values indicate more inequality
- `urban` = binary variable for urban (1) or rural (0) commuting zone
- `hh_income` = median household family income in commuting zone
- `racial_seg` = measure of racial segregation

REPLACE THIS LINE WITH YOUR CODE

```
cz_subset <- cz %>%
     select(mobility, gini, urban, hh_income, racial_seg)
```

Correlation Matrix

One way to save some time when looking at multiple correlations is to create a matrix with all the possible correlations in your dataframe.

```
matrix <- round(cor(cz_subset, use = "complete.obs"),3)
```

Let's review the matrix!

```
matrix
```

```
##             mobility   gini  urban hh_income racial_seg
## mobility       1.000 -0.578 -0.361     0.051     -0.361
## gini          -0.578  1.000  0.313     0.069      0.281
## urban         -0.361  0.313  1.000     0.372      0.388
## hh_income      0.051  0.069  0.372     1.000      0.144
## racial_seg    -0.361  0.281  0.388     0.144      1.000
```

To only see some of the matrix, use indexing. We have indexed by value in the past, but we can also index by row and cell location in the dataframe. In this example, we want to pull the first three rows (1:3) and the first three columns (1:4):

```
matrix[1:3, 1:4]
```

```
##           mobility   gini  urban hh_income
## mobility     1.000 -0.578 -0.361     0.051
## gini        -0.578  1.000  0.313     0.069
## urban       -0.361  0.313  1.000     0.372
```

Introducing Multivariate Relationships - See Slides

Last week we identified individual points on a scatterplot to dig deeper into how two variables are associated. Another analytical tool is to examine whether the association differs at specific values of another variable.

Consider the correlation between racial segregation and income. Would you expect this correlation to be positive or negative? Strong or weak?

REPLACE THIS LINE WITH YOUR CODE

```
cor(cz_subset$hh_income, cz_subset$racial_seg,
    use = "complete")
```

```
## [1] 0.1445054
```

Would this association be the same in urban and rural commuting zones?

REPLACE THIS LINE WITH YOUR CODE

```
cor(cz_subset$racial_seg[cz_subset$urban==0],
    cz_subset$hh_income[cz_subset$urban==0],
    use = "complete")
```

```
## [1] -0.2621044
```

```
cor(cz_subset$racial_seg[cz_subset$urban==1],
    cz_subset$hh_income[cz_subset$urban==1],
    use = "complete")
```

```
## [1] 0.3231209
```

```
urban_rural_correlations <- cz_subset %>%
     group_by(urban) %>%
     summarize(racialseg_income_correlation =
                    cor(
          racial_seg, hh_income,
          use ="complete"))

urban_rural_correlations
```

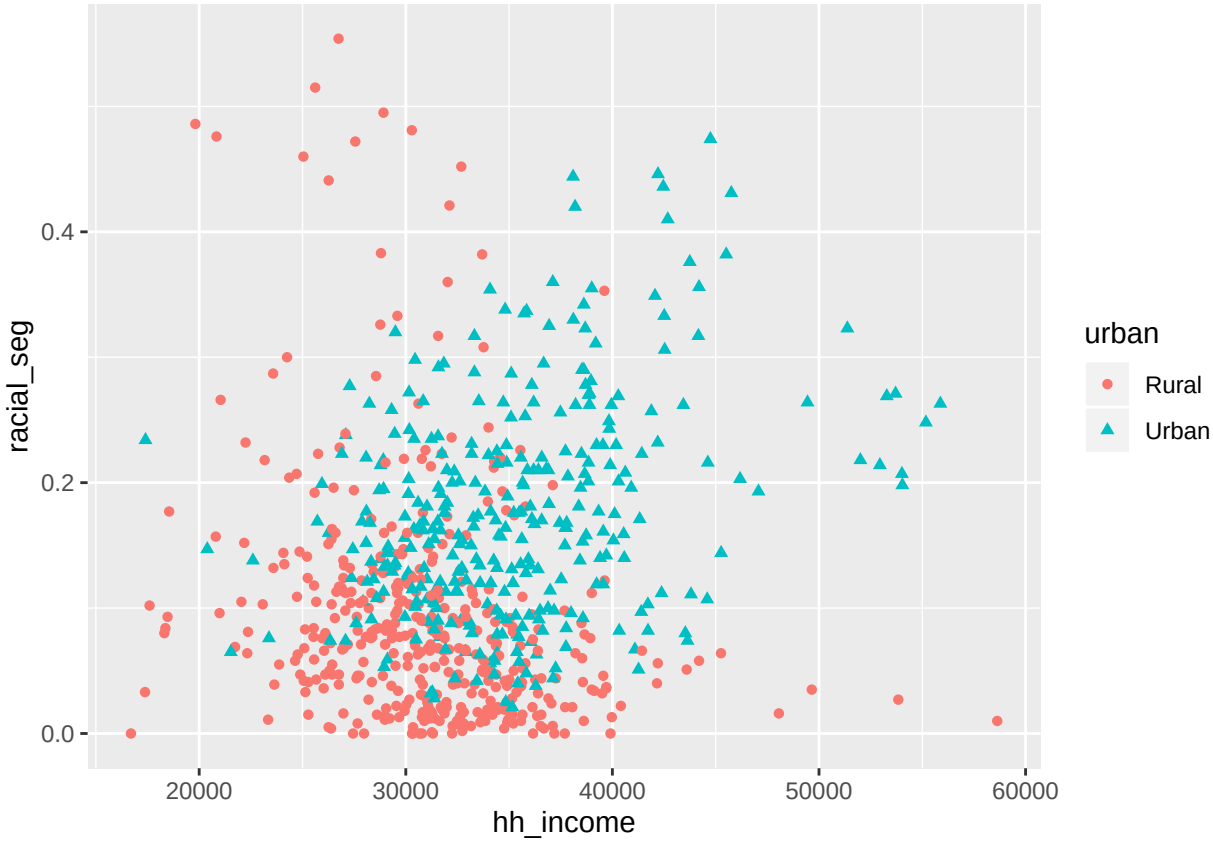
```
## # A tibble: 2 x 2
##   urban racialseg_income_correlation
##   <int>                        <dbl>
## 1     0                       -0.262
## 2     1                        0.323
```

We can visualize the different associations by geographic type if we use different colored points for urban and rural commuting zones. Before we get there, let's make sure the **urban** variable is recognized as a factor variable, and change the labels from "0" and "1" to "Rural" and "Urban".

```
cz_subset <- cz_subset %>%
     mutate(urban = factor(urban,
                           levels=c("0", "1"),
                           labels=c("Rural", "Urban")))
```

Now let's make the plot. We want urban and rural commuting zones to be in different colors and we want their points to have different shapes.

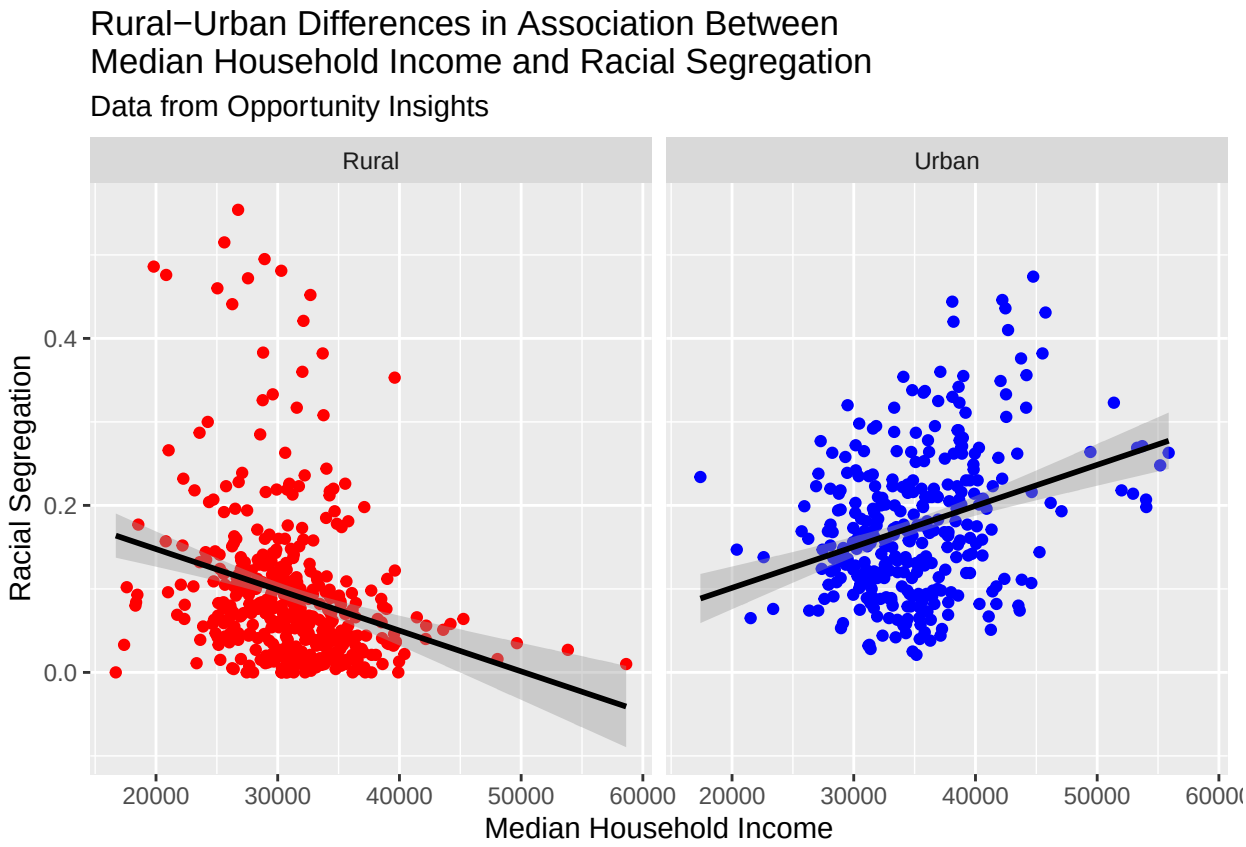
```
urban_rural_plot <- ggplot(cz_subset, aes(x = hh_income,
                                          y = racial_seg,
                                          color = urban,
                                          shape = urban))

urban_rural_plot + geom_point()
```



Extra Fancy: Remember we can use `facet_grid()` to display two plots in one figure.

```
urban_rural_facets <- ggplot(cz_subset, aes(x = hh_income, y = racial_seg))

urban_rural_facets + geom_point(aes(color = urban)) +
     geom_smooth(method = lm, color = "black") +
     facet_grid(.~urban) +
     scale_color_manual(values = c("Red", "Blue")) + guides(color = FALSE) +
     labs(x = "Median Household Income", y = "Racial Segregation",
          title = "Rural-Urban Differences in Association Between \nMedian Household Income and Racial S
```

```
        subtitle = "Data from Opportunity Insights")
```



Finally, let's create a three-way table showing how the mean of the racial segregation index varies at each quintile of median income between urban and rural commuting zones. Any thoughts on how to do this? What do we need?

REPLACE THIS LINE WITH YOUR CODE

```
cz_quintiles <- cz_subset %>%
     mutate(quintile = ntile(hh_income, 5)) %>%
     # Nice trick to get quintiles!
     group_by(quintile, urban) %>%
     # Grouping by multiple variables
     summarize(mean_racial_seg = round(mean(racial_seg),3))
# Round here

pander(cz_quintiles)
```

quintile	urban	mean_racial_seg
1	Rural	0.124
1	Urban	0.154
2	Rural	0.1
2	Urban	0.158
3	Rural	0.084
3	Urban	0.151

quintile	urban	mean_racial_seg
4	Rural	0.066
4	Urban	0.155
5	Rural	0.054
5	Urban	0.219

In the table above we have long data but might prefer wide data. This would be a good time to use `spread()`. In the `spread()` line in the next chunk, we are saying we want to create a column for each level of the `urban` variable, and we want the values of those variables to be the values of `mean_racial_seg` for each quintile-urban combination.

```
cz_spread <- cz_quintiles %>%
     spread(urban, mean_racial_seg)

cz_spread
```

```
## # A tibble: 5 x 3
## # Groups:   quintile [5]
##   quintile Rural Urban
##      <int> <dbl> <dbl>
## 1        1 0.124 0.154
## 2        2 0.1   0.158
## 3        3 0.084 0.151
## 4        4 0.066 0.155
## 5        5 0.054 0.219
```

Pander will make this table prettier...

```
pander(cz_spread)
```

quintile	Rural	Urban
1	0.124	0.154
2	0.1	0.158
3	0.084	0.151
4	0.066	0.155
5	0.054	0.219

The opposite of `spread()` is `gather()` which you use when you want to transform wide data into long data. In the example below, we want to collapse columns 2 and 3 into two new columns. The first new column will be called "urban" and its values will be the names of the existing columns. The second new column will be called "mean_racial_seg" and will take the values associated with the existing columns.

```
cz_quintiles <- cz_spread %>%
  gather(urban, mean_racial_seg, 2:3)
```

Take a look...

```
cz_quintiles
```

```
## # A tibble: 10 x 3
## # Groups:   quintile [5]
##    quintile urban mean_racial_seg
##       <int> <chr>           <dbl>
##  1        1 Rural           0.124
```

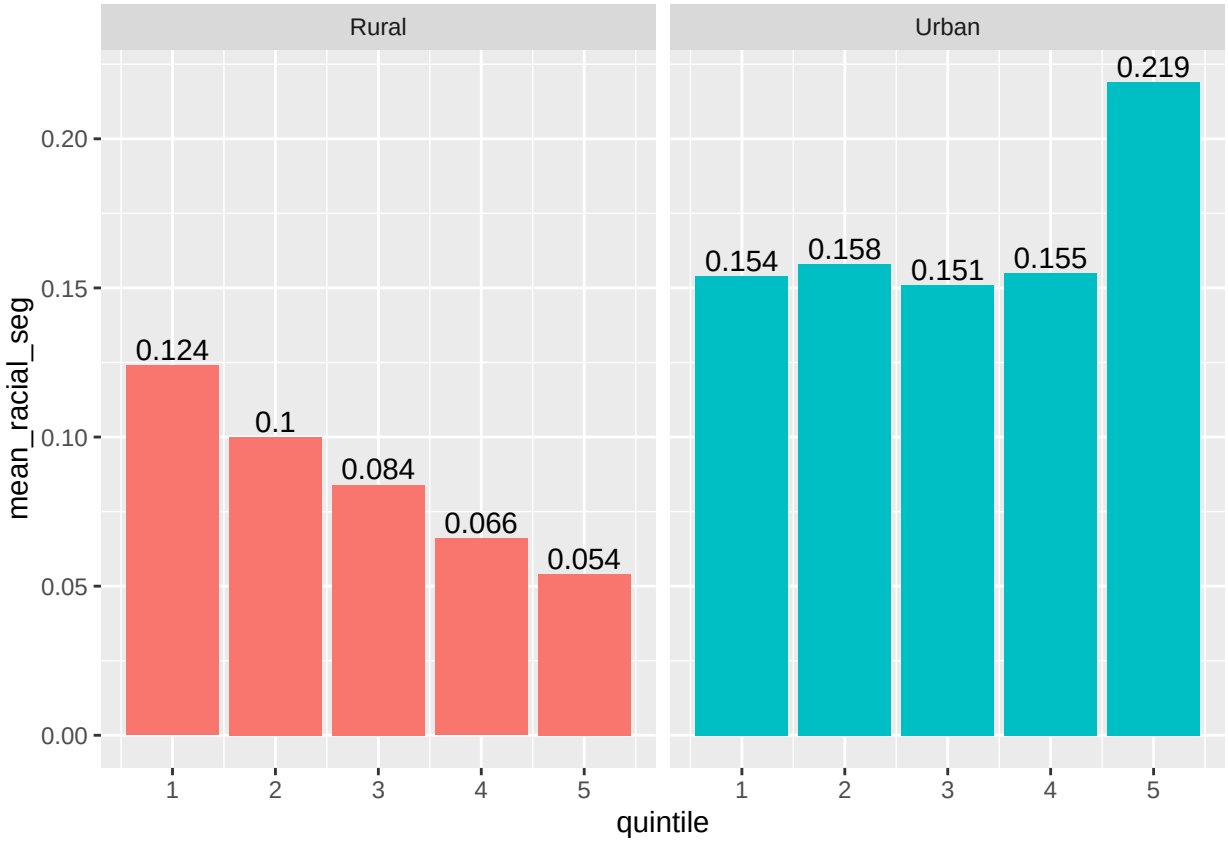
```
##  2        2 Rural           0.1
##  3        3 Rural           0.084
##  4        4 Rural           0.066
##  5        5 Rural           0.054
##  6        1 Urban           0.154
##  7        2 Urban           0.158
##  8        3 Urban           0.151
##  9        4 Urban           0.155
## 10        5 Urban           0.219
```

If there's time...

How could we visualize these three way relationships? Note that we'll want to use the long data for this rather than the wide data. Why?

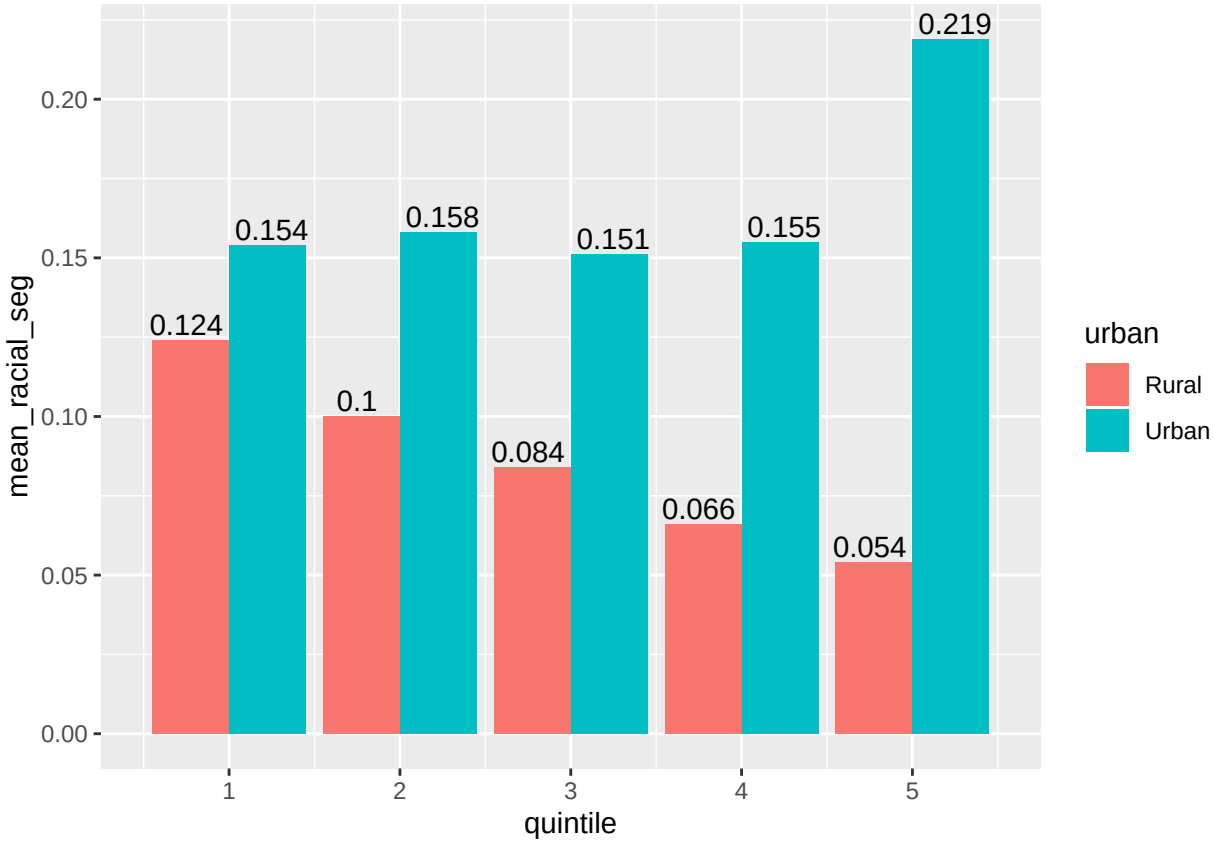
One option for a plot is to use side by side *plots*...

```
quintile_plot2 <- ggplot(cz_quintiles, aes(x = quintile, y = mean_racial_seg,
                                           fill = urban))

quintile_plot2 + geom_col() +
     geom_text(aes(label = mean_racial_seg, vjust = -.25)) +
     facet_grid(.~urban) + guides(fill = FALSE)
```



Another option is to change the "position" of the columns so they are next to each other on one plot...

```
quintile_plot1 <- ggplot(cz_quintiles, aes(x = quintile, y = mean_racial_seg,
                                           fill = urban,
                                           label = mean_racial_seg))

quintile_plot1 + geom_col(position = "dodge") + # For side by side columns
     geom_text(position = position_dodge(1), vjust = -.25)
```



A Word About Color

One popular package to use for adjusting color is the R Color Brewer package. You can install it and load it here:

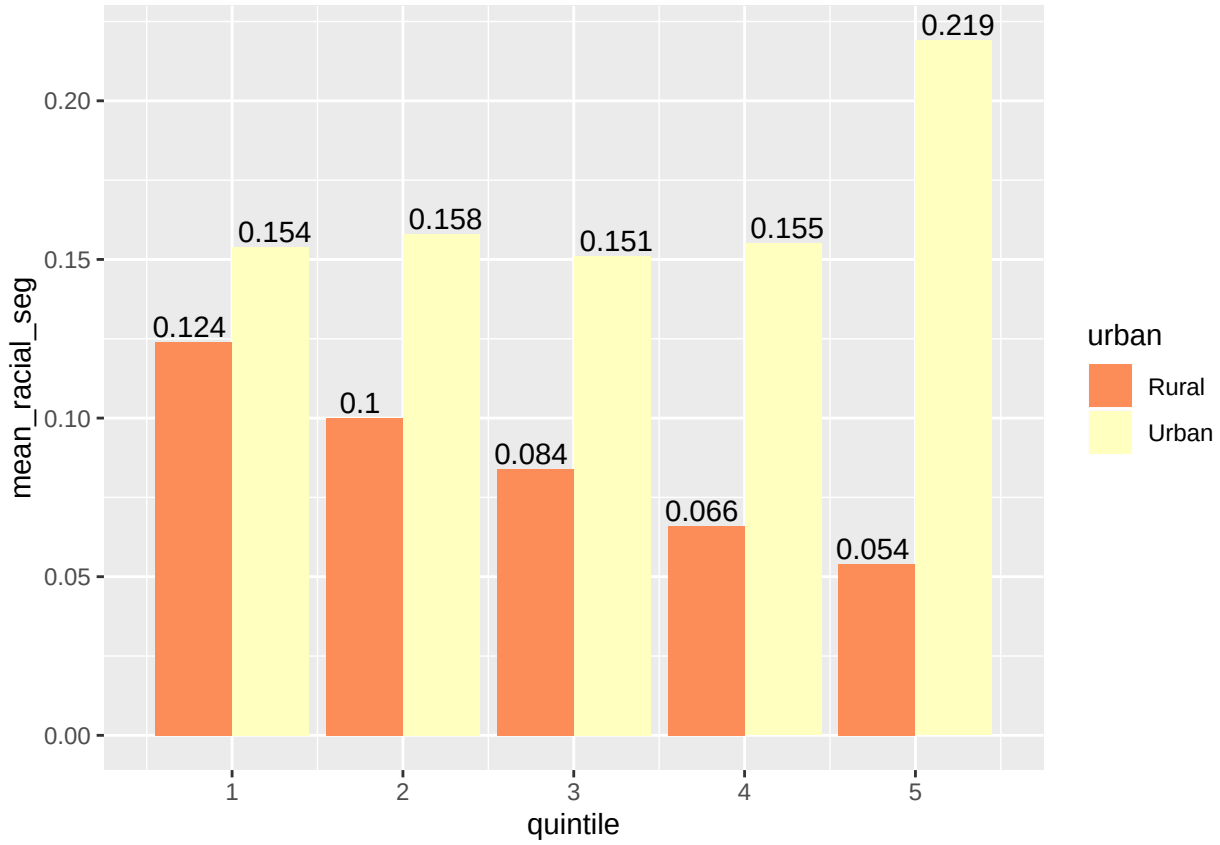
```
#install.packages("RColorBrewer")
library(RColorBrewer)
```

There's good information on Brewer's color options at this site. (Hold down the command button and click the link to open it!)

The site previews options for sequential (or continuous) variables and divergent (or factor/categorical/discrete) variables. Once you find a color scheme you like, input the palette name into the `scale_fill_brewer()` function.

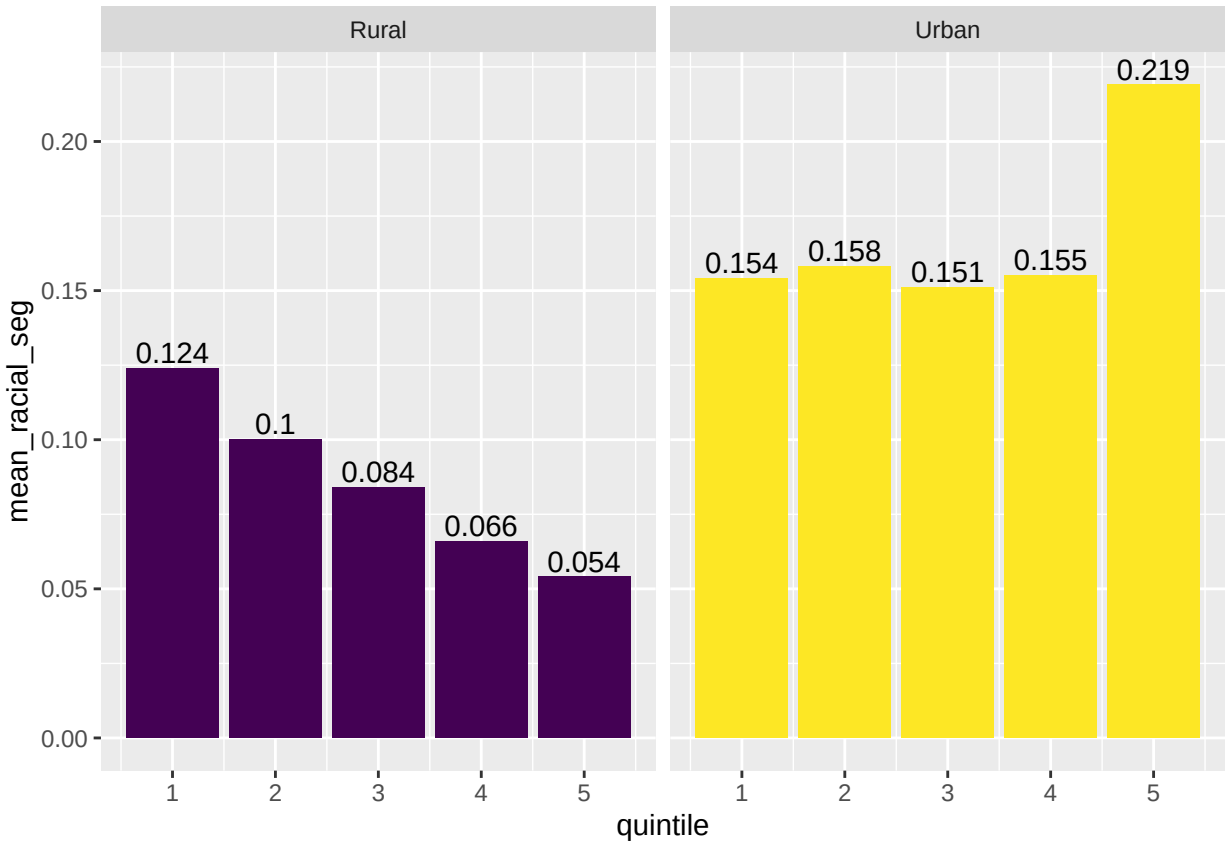
```
library(RColorBrewer)

quintile_plot1 + geom_col(position = "dodge") + # For side by side columns
     geom_text(position = position_dodge(1), vjust = -.25) +
```

```
  scale_fill_brewer(palette = "RdYlGn") # Palette Name
```



Want other colors? I like the `viridis color palettes`.

```
quintile_plot2 + geom_col() +
     geom_text(aes(label = mean_racial_seg, vjust = -.25)) +
     facet_grid(.~urban) + guides(fill = FALSE) +
     scale_fill_viridis_d() # The d is for a discrete variable
```


Rural
Urban
mean_racial_seg
0.20
0.15
0.10
0.05
0.00
0.124
0.1
0.084
0.066
0.054
0.154
0.158
0.151
0.155
0.219
1
2
3
4
5
quintile