Study Guide Week 6
DATA 110
Saidi

Exploring Relationships

Correlation and Scatterplots

Create a Scatterplot

In this example, look at US crime rates at the state level, in 2005, with rates per 100,000 population for crime types such as murder, robbery, and aggravated assault, as reported by the Census Bureau. There are 7 crime types in total. The dataset is clean to begin with.

```
library(tidyverse)
library(ggfortify)
library(htmltools)
library(plotly)
crime <- read_csv('http://datasets.flowingdata.com/crimeRatesByState2005.csv')
## Parsed with column specification:
## cols(
##   state = col_character(),
##   murder = col_double(),
##   forcible_rape = col_double(),
##   robbery = col_double(),
##   aggravated_assault = col_double(),
##   burglary = col_double(),
##   larceny_theft = col_double(),
##   motor_vehicle_theft = col_double(),
##   population = col_double()
```

```
## )
# source: U.S. Census Bureau and Nathan Yau
```

Check out the first few lines

state	murder	forcible_rape	robbery	aggravated_assault	burglary	larceny_theft
<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
United States	5.6	31.7	140.7	291.1	726.7	2286.3
Alabama	8.2	34.3	141.4	247.8	953.8	2650.0
Alaska	4.8	81.1	80.9	465.1	622.5	2599.1
Arizona	7.5	33.8	144.4	327.4	948.4	2965.2
Arkansas	6.7	42.9	91.1	386.8	1084.6	2711.2
California	6.9	26.0	176.1	317.3	693.3	1916.5

6 rows | 1-7 of 9 columns

2023-02-25

Part I - Exploring the data through scatterplots

Create a Scatterplot

In this example, look at US crime rates at the state level, in 2005, with rates per 100,000 population for crime types such as murder, robbery, and aggravated assault, as reported by the Census Bureau. There are 7 crime types in total. The dataset is clean to begin with.

```
library(tidyverse)
library(ggfortify)
library(plotly)

crime <- read_csv('http://datasets.flowingdata.com/crimeRatesByState2005.csv')

Rows: 52 Columns: 9
── Column specification ────────────────────────────────────────────────────────
Delimiter: ","
chr (1): state
dbl (8): murder, forcible_rape, robbery, aggravated_assault, burglary, larce...
```

```
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.

# source: U.S. Census Bureau and Nathan Yau
```

Check out the first few lines

```
# A tibble: 6 × 9
  state   murder forcible_rape robbery aggravated_assault burglary larceny_theft
  <chr>    <dbl>         <dbl>   <dbl>              <dbl>    <dbl>         <dbl>
1 United…    5.6          31.7   141.                291.     727.         2286.
2 Alabama    8.2          34.3   141.                248.     954.         2650 
3 Alaska     4.8          81.1    80.9               465.     622.         2599.
4 Arizona    7.5          33.8   144.                327.     948.         2965.
5 Arkans…    6.7          42.9    91.1               387.    1085.         2711.
6 Califo…    6.9          26     176.                317.     693.         1916.
# ℹ 2 more variables: motor_vehicle_theft <dbl>, population <dbl>
```

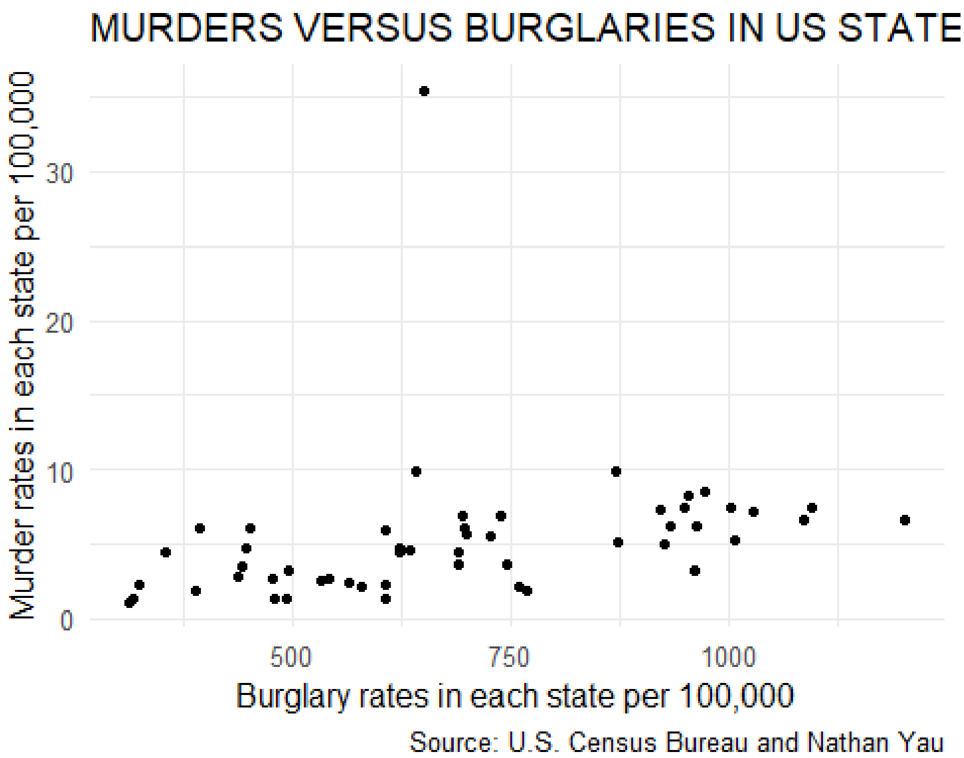
Notice

The data has a column for the state and then the rest are rates for various crimes. Now make a quick scatterplot.

Map variables in the data onto the X and Y axes and change the axes labels and theme

The default gray theme of ggplot2 has a rather academic look. See here and here for how to use the theme option to customize individual elements of a chart. Use one of the ggplot2 built-in themes, and then customize the fonts.

```
p1 <- ggplot(crime, aes(x = burglary, y = murder)) +
  labs(title = "MURDERS VERSUS BURGLARIES IN US STATES PER 100,000",
  caption = "Source: U.S. Census Bureau and Nathan Yau",
  x = "Burglary rates in each state per 100,000",
  y = "Murder rates in each state per 100,000") +
  theme_minimal(base_size = 12)
p1 + geom_point() # add the points
```

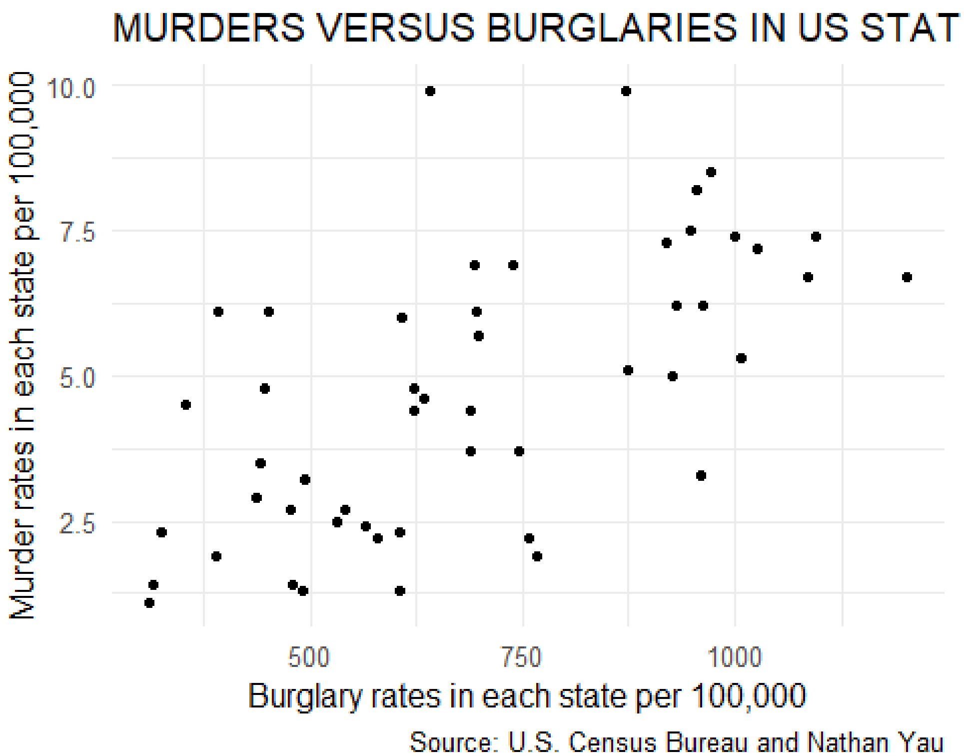


What is going on with the outlier?

The one point far higher than the rest represents Washington, D.C., which had a much higher murder rate of 35.4. The states with the next highest murder rate at that time were Louisiana and Maryland at 9.9 per 100,000.

Remove D.C. and US averages and replot:

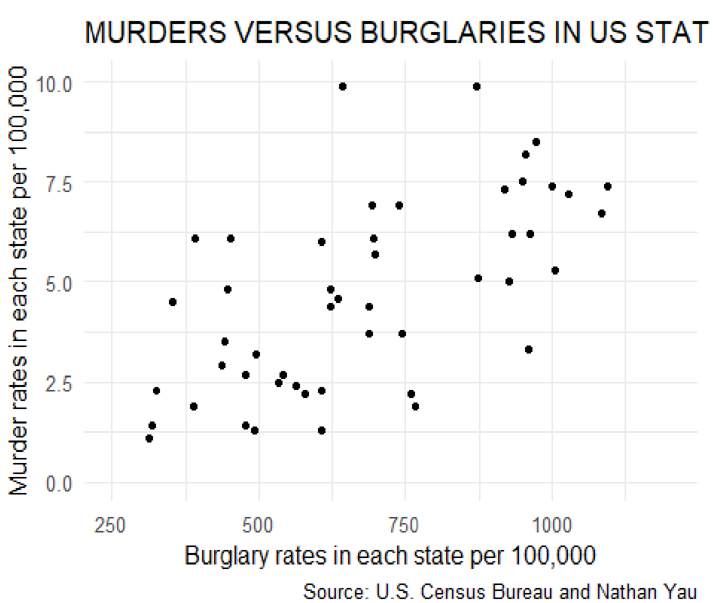
```
crime2 <- crime[crime$state != "District of Columbia",]
crime2 <- crime2[crime2$state != "United States",]

p2 <- ggplot(crime2, aes(x = burglary, y = murder)) +
  labs(title = "MURDERS VERSUS BURGLARIES IN US STATES PER 100,000",
  caption = "Source: U.S. Census Bureau and Nathan Yau",
  x = "Burglary rates in each state per 100,000",
  y = "Murder rates in each state per 100,000") +
  theme_minimal(base_size = 12)
p2 + geom_point()
```



Now the scatterplot appears to show a correlation

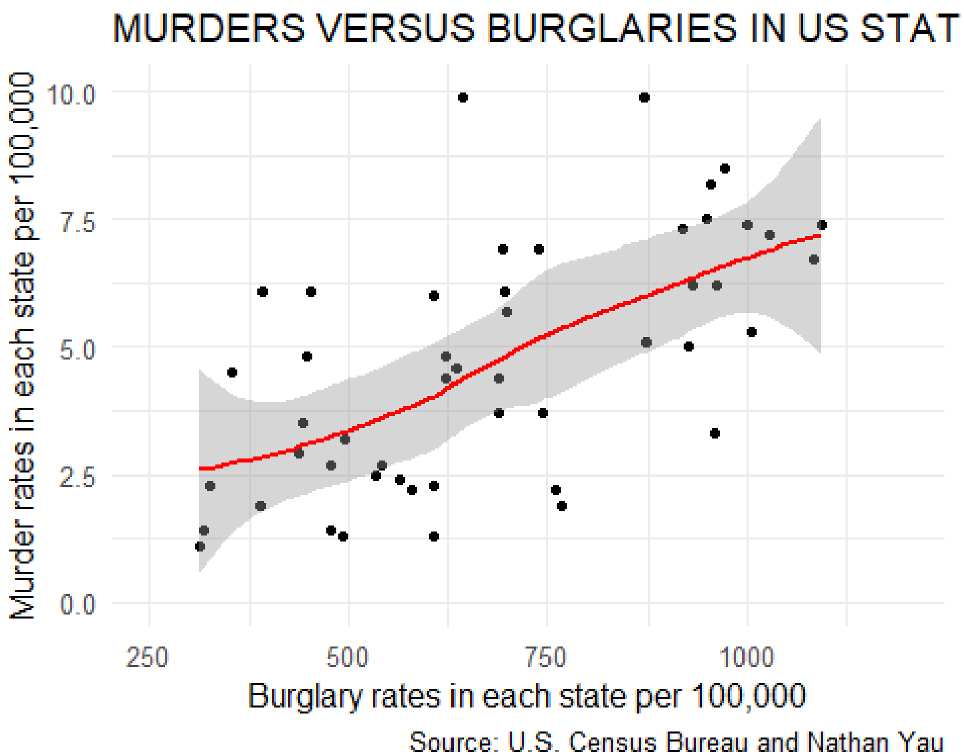
Fix the axes to start at 0.

```
p3 <- p2 + geom_point() + xlim(250,1200)+ ylim(0,10)
p3

Warning: Removed 1 row containing missing values or values outside the scale rang
e
(`geom_point()`).
```



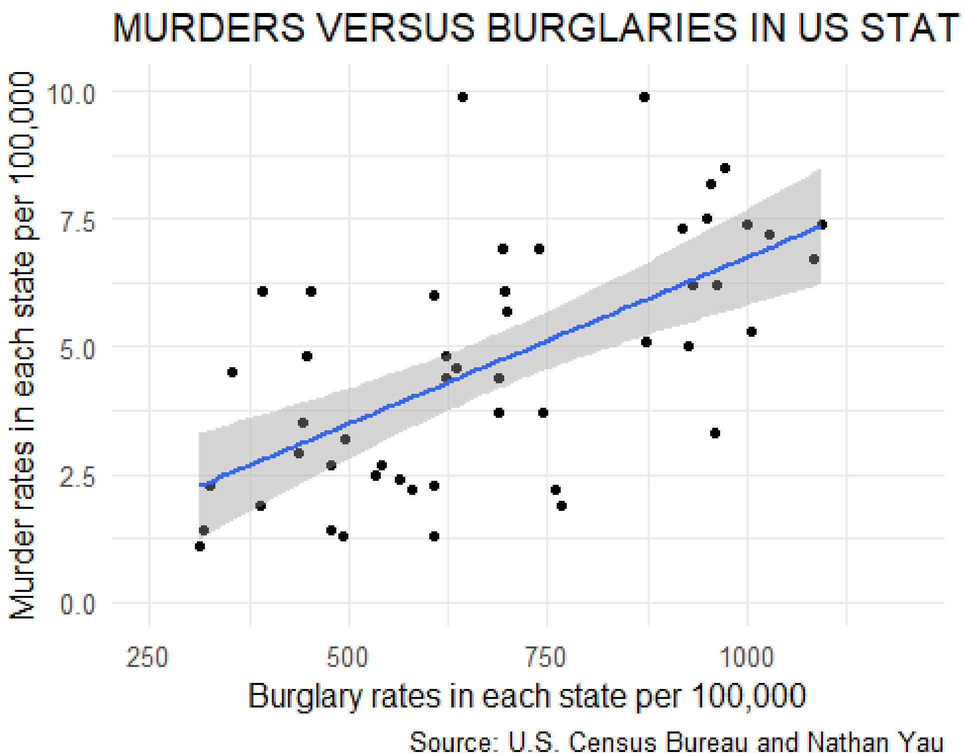
Add a smoother in red with a confidence interval

```
p4 <- p3 + geom_smooth(color = "red")
p4

`geom_smooth()` using method = 'loess' and formula = 'y ~ x'

Warning: Removed 1 row containing non-finite outside the scale range
(`stat_smooth()`).

Warning: Removed 1 row containing missing values or values outside the scale rang
e
(`geom_point()`).
```



Add a linear regression with confidence interval

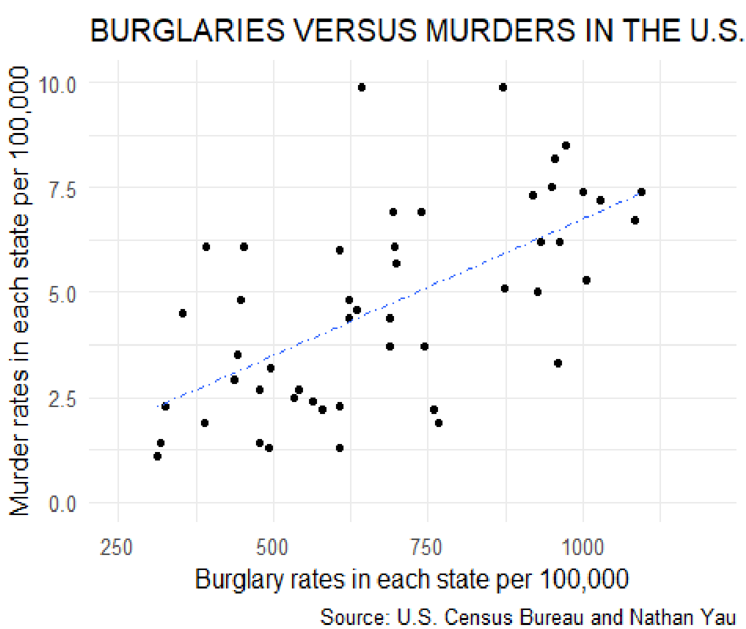
```
p5 <- p3 + geom_smooth(method='lm',formula=y~x)
p5

Warning: Removed 1 row containing non-finite outside the scale range
(`stat_smooth()`).

Warning: Removed 1 row containing missing values or values outside the scale rang
e
(`geom_point()`).
```



Add a title, make the line dashed, and remove the confidence interval band

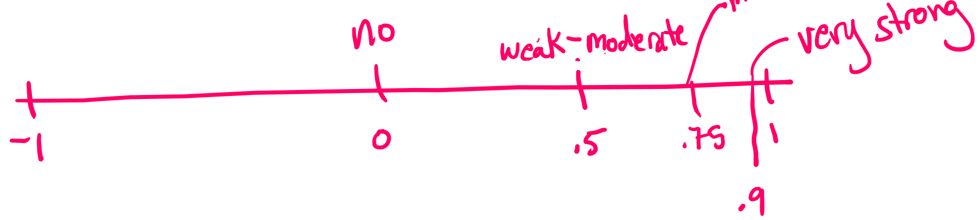
The command se = FALSE takes away the CI band

```
p6 <- p3 + geom_smooth(method='lm',formula=y~x, se = FALSE, linetype= "dotdash",
size = 0.3) +
  ggtitle("BURGLARIES VERSUS MURDERS IN THE U.S.")
```



Cor = code correlation (data$x, data$y)

Correlation Scale



Part II - Regression and Modeling

What is the linear equation of that linear regression model?

In the form, y=mx + b, we use the command, lm(y~x), meaning, fit the predictor variable x into the model to predict y. Look at the values of (Intercept) and murder. The column, Estimate gives the value you need in your linear model. The column for Pr(>|t|) p-value and is the describes whether the predictor is useful to the model. The more asterisks, the more the variable contributes to the model.

```
cor(crime2$burglary, crime2$murder)

[1] 0.6231757
```

moderate

```
fit1 <- lm(murder ~ burglary, data = crime2)
summary(fit1)


Call:
lm(formula = murder ~ burglary, data = crime2)

Residuals:
    Min      1Q  Median      3Q     Max
-3.2924 -1.2156 -0.2142  1.1749  5.4978

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.395519   0.825748   0.479    0.634
burglary    0.006247   0.001132   5.521 1.34e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.87 on 48 degrees of freedom
Multiple R-squared:  0.3883,    Adjusted R-squared:  0.3756
F-statistic: 30.48 on 1 and 48 DF,  p-value: 1.342e-06
```

y ~ x

y = mx + b

slope

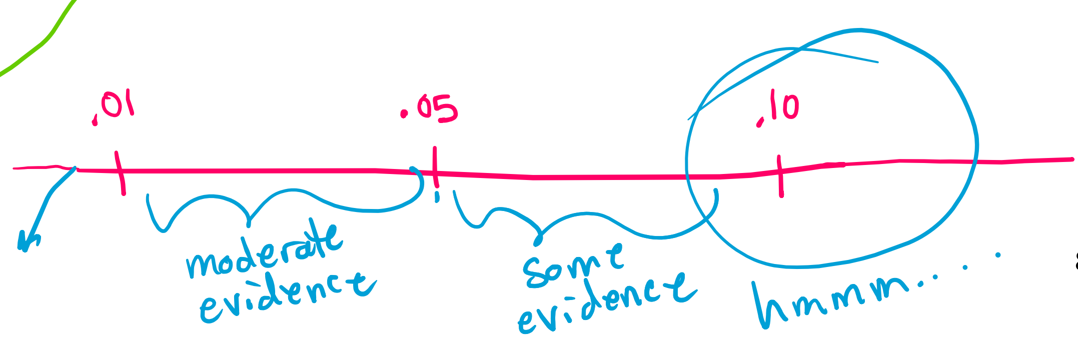
y-intercept (0, y)

p-value

.00000134

p-value scale

(significance scale)



What does the output mean?

Cor stands for "correlation". This is a value between (inclusively) -1 and 1. The correlation coefficient tells how strong or weak the correlation is. Values closer to +/- 1 are strong correlation (the sign is determined by the linear slope), values close to +/- 0.5 are weak correlation, and values close to zero have no correlation.

The model has the equation: murder = 0.0062(burglary) + 0.396

Y = slope(x) + y-int.

The slope may be interpreted in the following: For each additional burglary per 100,000, there is a predicted increase of 0.006 murders. per 100,000.

The p-value on the right of burglary has 3 asterisks which suggests it is a meaningful variable to explain the linear increase in murders. But we also need to look at the Adjusted R-Squared value. It states that about 38% of the variation in the observations may be explained by the model. In other words, 62% of the variation in the data is likely not explained by this model.

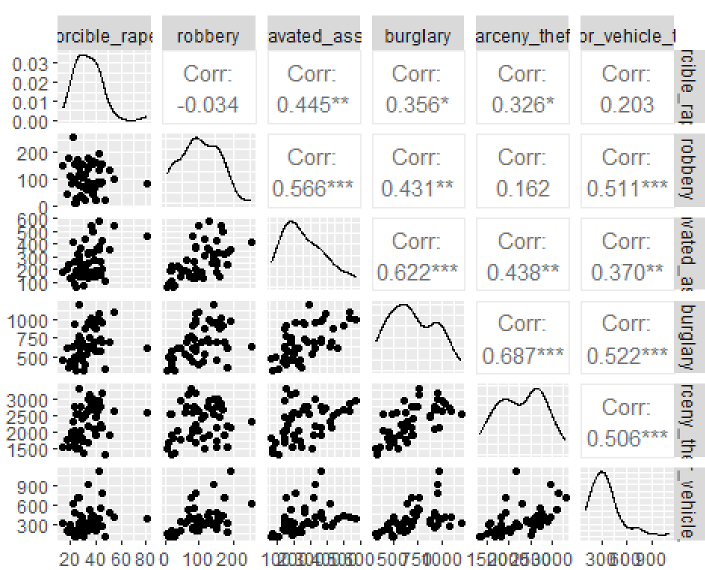
What about more variables?

Can a model with more predictors also be used? What would we be trying to predict?

Is there an easier way to compare multiple variables using a scatterplot matrix?

Check out the pairwise comparisons with density curves and correlation output

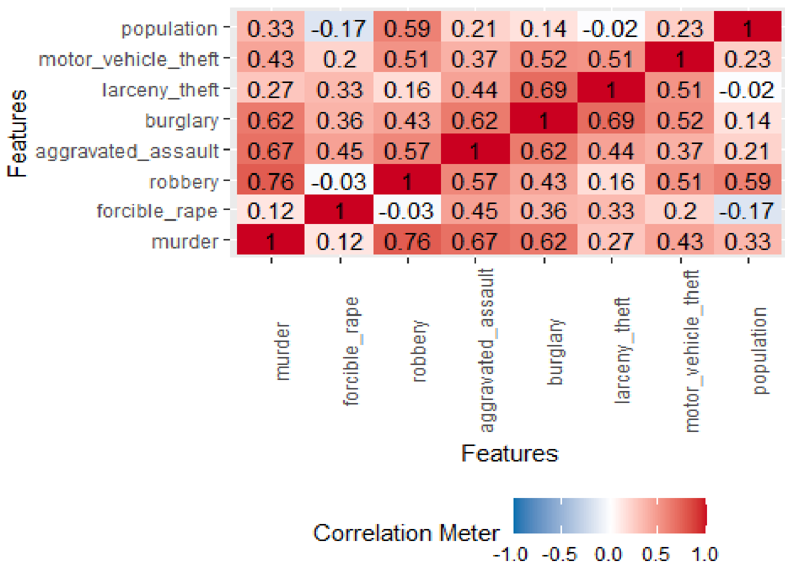
```
library(GGally)
ggpairs(crime2, columns = 3:8)  # only include predictor variables in the matrix
```



Another method: Use a correlation plot to explore the correlation among all variables

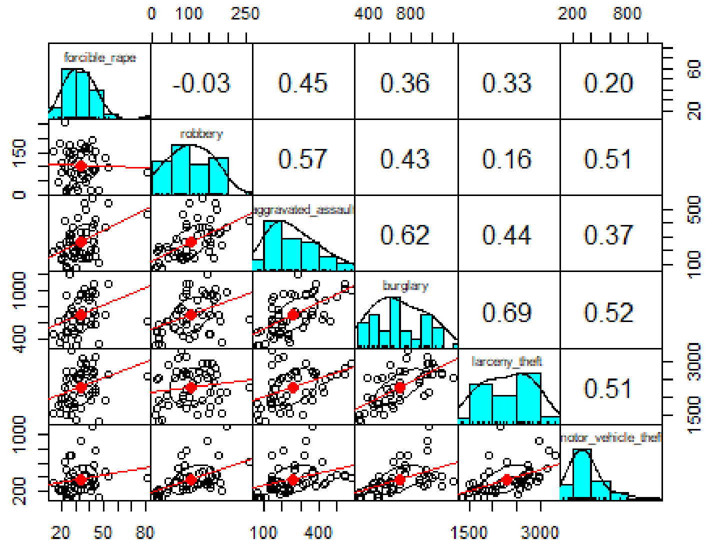
This correlation plot shows similar pairwise results as above, but in a heatmap of correlation values.

```
#install.packages("DataExplorer")
library(DataExplorer)
plot_correlation(crime2)

Warning in dummify(data, maxcat = maxcat): Ignored all discrete features since
`maxcat` set to 20 categories!
```



A third option to explore correlations using library(psych)

```
library(psych)
pairs.panels(crime2[3:8],   # plot distributions and correlations for all the dat
a
             gap = 0,
             pch = 21,
             lm = TRUE)
```



Collinearity

The key goal of multiple regression analysis is to isolate the relationship between EACH INDEPENDENT VARIABLE and the DEPENDENT VARIABLE.

COLLINEARITY means explanatory variables are correlated and thus NOT INDEPENDENT. The more correlated the variables, the more difficult it is to change one variable without changing the other. This is important to keep in mind. The two different matrices gave slightly different correlation information. We are concerned with dependence of 2 or more variables.

The two variables with the highest correlation of 0.68 or 0.69 are burglary and larceny_theft.

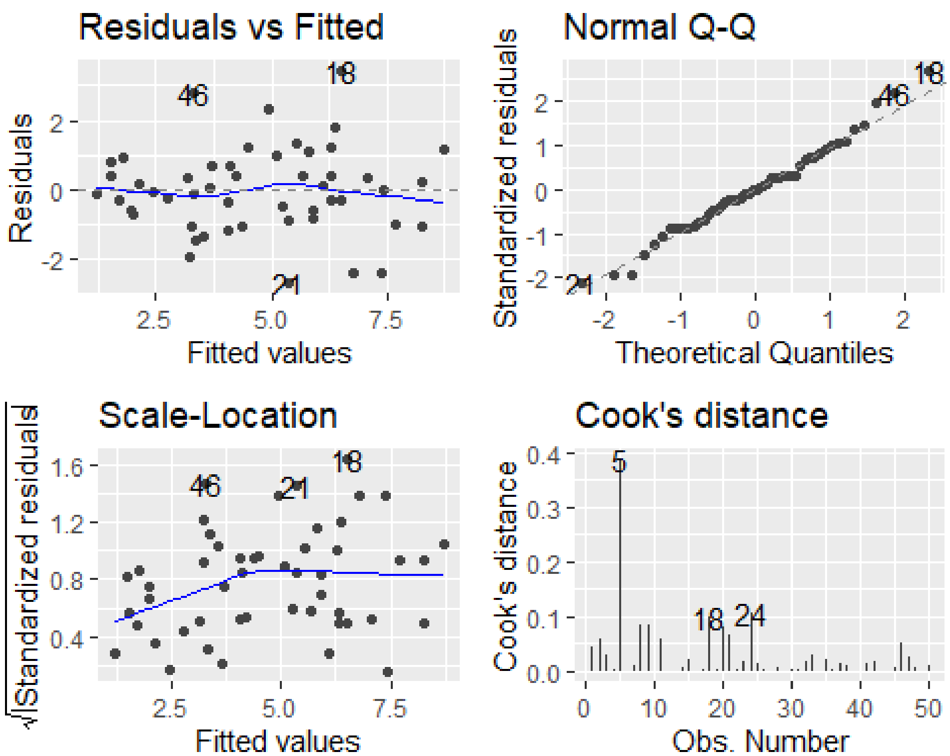
Now try to make a multiple regression model.

With multiple regression, there are several strategies for comparing variable inputs into a model. I will show you backward elimination. In backward elimination, start with all possible predictor variables with your response variable. In this case, we will use: burglary forcible_rape aggravated_assault larceny_theft motor_vehicle_theft Perform a model fit with all predictors.

1. Look at the p-value for each variable - if it is relatively small (< 0.10), then it is likely contributing to the model.
2. Check out the residual plots. A good model will have a relatively straight horizontal red line across the scatterplot between residuals plotted with fitted values (see below for a good residuals plot). You can also look at the other plots (Normal QQ, Scale-Location, and Residuals vs Leverage), but for now we will focus on the residual vs. fitted plot. The more curved the red line, the more likely that a better model exists.
3. Look at the output for the Adjusted R-Squared value at the bottom of the output. The interpretation is:

__% (from the adjusted r-squared value) of the variation in the observations may be explained by this model. The higher the adjusted R-squared value, the better the model. We use the adjusted R-squared value because it compensates for more predictors mathematically increasing the normal R-squared value.

```r
fit2 <- lm(murder ~ robbery + burglary + forcible_rape + aggravated_assault + lar
ceny_theft + motor_vehicle_theft + population, data = crime2)
summary(fit2)


Call:
lm(formula = murder ~ robbery + burglary + forcible_rape + aggravated_assault +
    larceny_theft + motor_vehicle_theft + population, data = crime2)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.6687 -0.7794 -0.0333  0.6965  3.4105 

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)    
(Intercept)          1.073e+00  1.086e+00   0.988 0.328984    
robbery              2.239e-02  5.990e-03   3.738 0.000555 ***
burglary             4.106e-03  1.334e-03   3.078 0.003665 ** 
forcible_rape       -1.426e-02  2.109e-02  -0.676 0.502798    
aggravated_assault   4.319e-03  2.351e-03   1.837 0.073303 .  
larceny_theft       -7.895e-04  5.621e-04  -1.404 0.167537    
motor_vehicle_theft -2.964e-04  1.276e-03  -0.232 0.817454    
population          -3.744e-08  3.723e-08  -1.006 0.320309    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.338 on 42 degrees of freedom
Multiple R-squared:  0.7259,	Adjusted R-squared:  0.6802 
F-statistic: 15.89 on 7 and 42 DF,  p-value: 5.48e-10

autoplot(fit2, 1:4, nrow=2, ncol=2)
```



What does these diagnostic plots mean?

4. Residual plot essentiall indicates whether a linear model is appropriate - you can see this by the blue line showing relatively horizontal. If it is not relatively horizontal, a linear plot may not be appropriate.

5. QQPlot indicates whether the distribution is relatively normal. Observations that might be outliers are indicated by their row number.

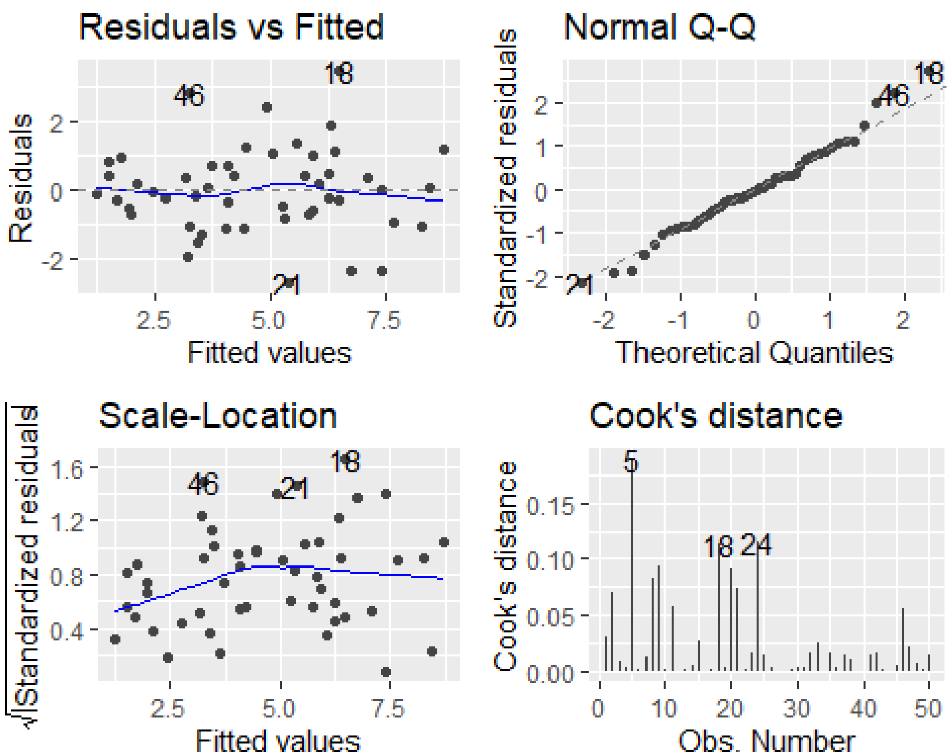
6. Scale-Location indicates homogeneous variance (homeoscedacity). Influential observations that are skewing the variance distribution are indicated.

7. Cook's Distance indicates which outliers have high leverage, meaning that some outliers may not cause the model to violate basic assumptions required for the regression analysis (see #1-3). If outliers have high leverage, then they may be causing problems for your model. You can try to remove those observations, especially if they appear in any of the other 3 plots above.

What are we really trying to predict?

If we are trying to predict murder rates, then we can see if any of the predictor variables contribute to this model. Note the adjusted R-squared value is 68.01% The only variable that does not appear to be as significant as the others is motor_vehicle_theft. So drop that and re-run the model.

```r
fit3 <- lm(murder ~ robbery + burglary + forcible_rape + aggravated_assault + lar
ceny_theft + population, data = crime2)
summary(fit3)


Call:
lm(formula = murder ~ robbery + burglary + forcible_rape + aggravated_assault +
    larceny_theft + population, data = crime2)

Residuals:
    Min      1Q  Median      3Q     Max
-2.6913 -0.7289 -0.0276  0.6978  3.4248

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)         1.154e+00  1.017e+00   1.134 0.263070
robbery             2.177e-02  5.305e-03   4.104 0.000178 ***
burglary            4.077e-03  1.314e-03   3.104 0.003372 **
forcible_rape      -1.515e-02  2.051e-02  -0.739 0.464187
aggravated_assault  4.442e-03  2.266e-03   1.960 0.056449 .
larceny_theft      -8.368e-04  5.182e-04  -1.615 0.113655
population         -3.708e-08  3.678e-08  -1.008 0.319085
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.323 on 43 degrees of freedom
Multiple R-squared:  0.7255,    Adjusted R-squared:  0.6872
F-statistic: 18.94 on 6 and 43 DF,  p-value: 1.221e-10

autoplot(fit3, 1:4, nrow=2, ncol=2)
```



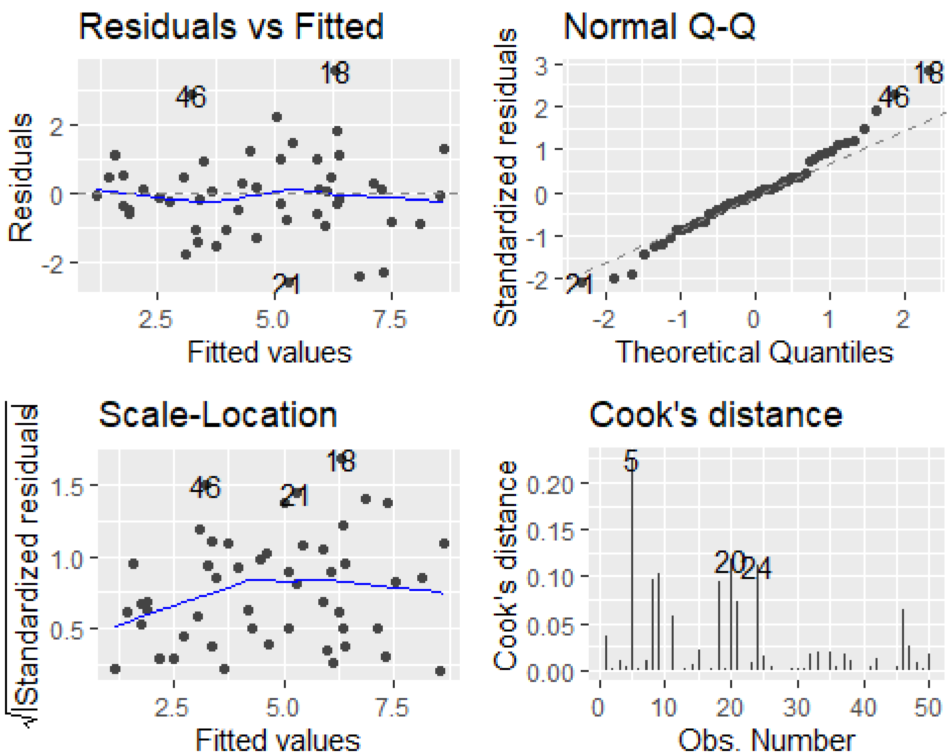
Drop motor_vehicle_theft - the adjusted R-squared value improved slightly to 68.7%.

Maybe try removing forcible rape since it had a large p-value of 0.51. Don't forget to check the diagnostic plots.

```
fit4 <- lm(murder ~ robbery + burglary + aggravated_assault + larceny_theft + pop
ulation, data = crime2)
summary(fit4)


Call:
lm(formula = murder ~ robbery + burglary + aggravated_assault +
    larceny_theft + population, data = crime2)

Residuals:
    Min      1Q  Median      3Q     Max
-2.5994 -0.7290 -0.0557  0.5274  3.5978

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)         8.058e-01  8.971e-01   0.898  0.37394
robbery             2.290e-02  5.054e-03   4.531 4.46e-05 ***
burglary            3.962e-03  1.297e-03   3.053  0.00383 **
aggravated_assault  3.710e-03  2.027e-03   1.830  0.07399 .
larceny_theft      -8.467e-04  5.153e-04  -1.643  0.10750
population         -3.482e-08  3.647e-08  -0.955  0.34485
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 1.317 on 44 degrees of freedom
Multiple R-squared:  0.722, Adjusted R-squared:  0.6904
F-statistic: 22.86 on 5 and 44 DF,  p-value: 3.111e-11
```

```r
autoplot(fit4, 1:4, nrow=2, ncol=2)
```



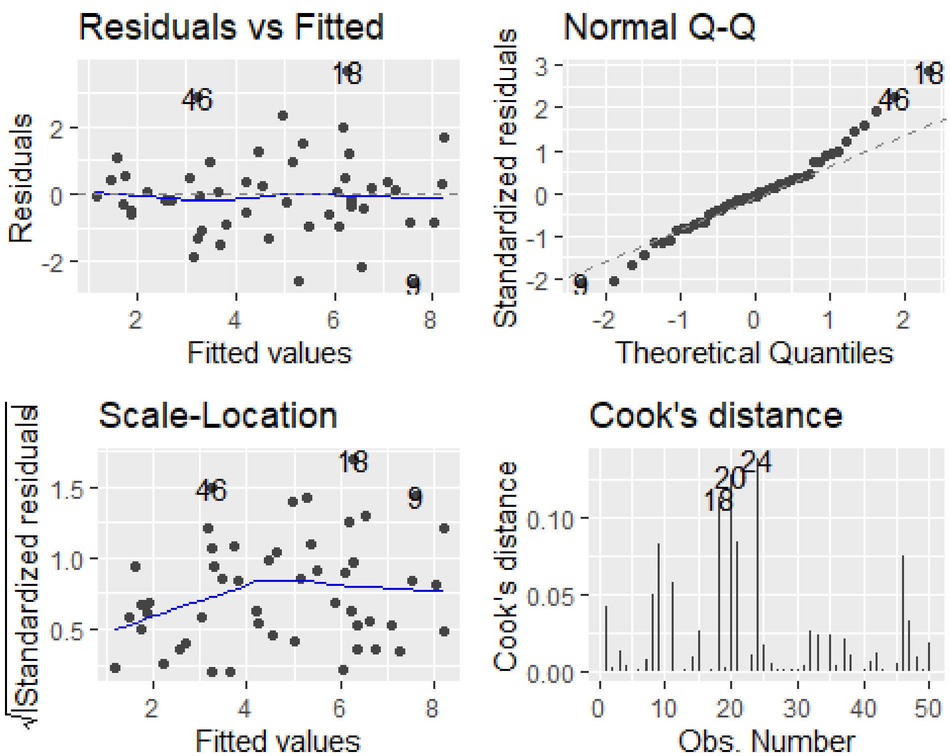
The adjusted R-squared went up to 69%. The residuals plot looks about the same.

One final model - the simplest (parsimonious) by removing population.

```r
fit5 <- lm(murder ~ robbery + burglary + aggravated_assault + larceny_theft, data
= crime2)
summary(fit5)


Call:
lm(formula = murder ~ robbery + burglary + aggravated_assault + 
    larceny_theft, data = crime2)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.6290 -0.7670 -0.0601  0.4779  3.6348 

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)    
(Intercept)        0.7555163  0.8946439   0.844  0.40286    
robbery            0.0201084  0.0041195   4.881 1.36e-05 ***
burglary           0.0040134  0.0012950   3.099  0.00334 ** 
aggravated_assault 0.0039521  0.0020089   1.967  0.05533 .  
```

```
larceny_theft      -0.0008325  0.0005146  -1.618  0.11268
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.315 on 45 degrees of freedom
Multiple R-squared:  0.7163,    Adjusted R-squared:  0.691
F-statistic:  28.4 on 4 and 45 DF,  p-value: 8.396e-12
```

```
autoplot(fit5, 1:4, nrow=2, ncol=2)
```



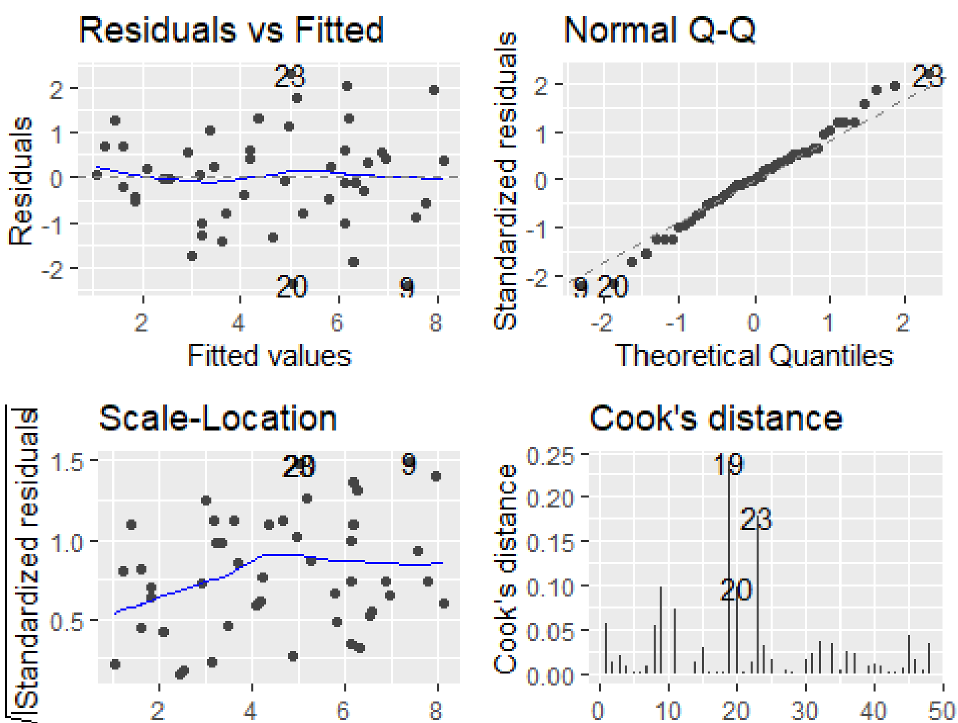
The residuals plot shows observations 46 and 18 have an effect on the residuals plot as well having high scale-location values.

Louisiana is 18 and 46 is Virginia

Try the last model, but remove those 2 observations:

```
options(scipen = 0)
crime3 <- crime2[-c(18,46),]
fit6 <- lm(murder ~ robbery + burglary + aggravated_assault + larceny_theft, data
= crime3)
summary(fit6)


Call:
lm(formula = murder ~ robbery + burglary + aggravated_assault +
    larceny_theft, data = crime3)
```

```
Residuals:
     Min       1Q   Median       3Q      Max 
-2.38619 -0.62797  0.01056  0.58757  2.28866 

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)    
(Intercept)         0.6210438  0.7722820   0.804 0.425724    
robbery             0.0202223  0.0035864   5.639 1.22e-06 ***
burglary            0.0044251  0.0011349   3.899 0.000334 ***
aggravated_assault  0.0031196  0.0017625   1.770 0.083825 .  
larceny_theft      -0.0008685  0.0004461  -1.947 0.058109 .  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.133 on 43 degrees of freedom
Multiple R-squared:  0.7743,    Adjusted R-squared:  0.7533 
F-statistic: 36.88 on 4 and 43 DF,  p-value: 2.227e-13

autoplot(fit6, 1:4, nrow=2, ncol=2)
```



ⓘ Interesting! the Adj R^2 increased quite a bit. What do we do - removing the outliers improves the model, but these may be important states to study what is going on

The adjusted R^2 went up to about 75%, which is an improvement. Was it worth removing those two state observations to improve our model? Or do those two states' information tell us something more to investigate?

One last attempt - we can compare the last models to see if removing population is an improvement on the model using ANOVA

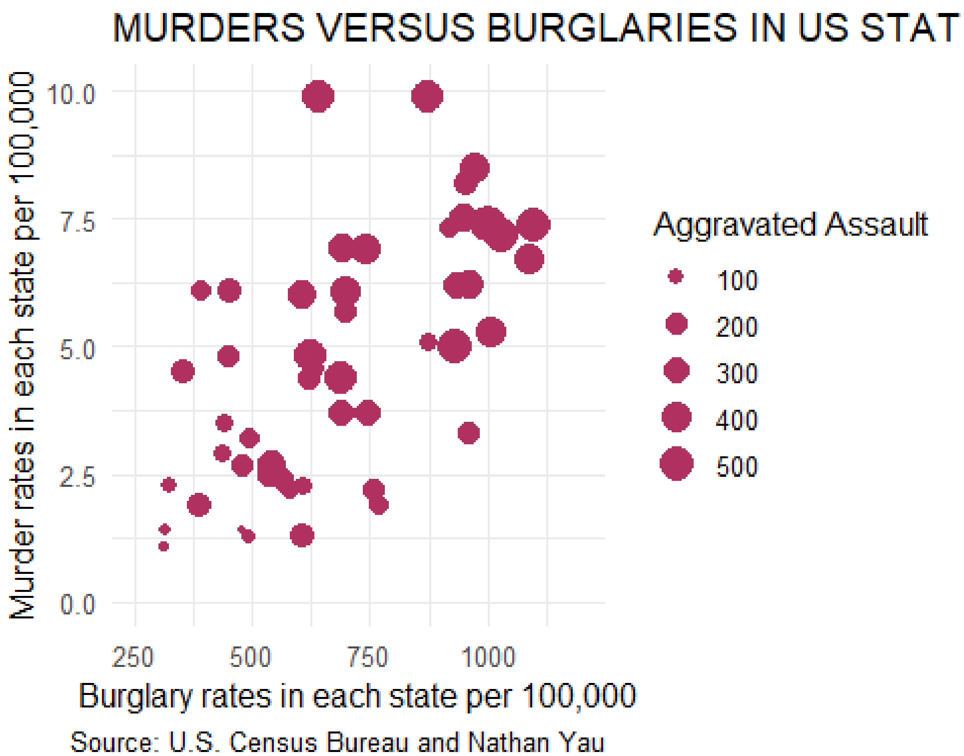
ANOVA (analysis of variance) compares 2 models, one simpler than the other. If the result is a small p-value, then the larger model is better than the smaller model

```
anova(fit5, fit4)

Analysis of Variance Table

Model 1: murder ~ robbery + burglary + aggravated_assault + larceny_theft
Model 2: murder ~ robbery + burglary + aggravated_assault + larceny_theft +
    population
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1     45 77.852
2     44 76.272  1    1.5806 0.9118 0.3448
```

We can see that the p-value is large, so we choose the simpler model. There is no compelling evidence that population contributes significantly to the model.

Back to simply murders and burglaries - bring in the next most important variable to the relationship - aggravated assault (by small p-value) as a size of the circle

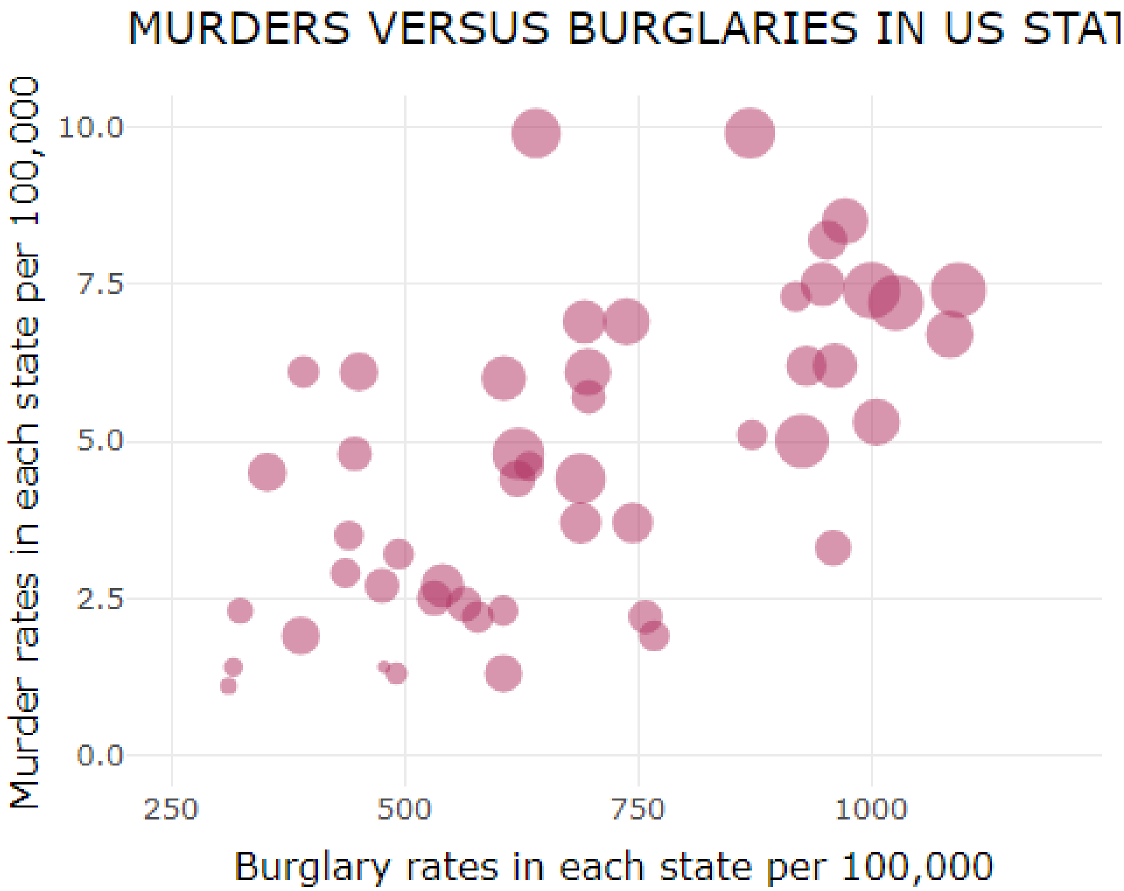
```
options(scipen = 999)
p2 +
  geom_point(aes(size = aggravated_assault), color = "maroon") +
  xlim(250,1200) +
  ylim(0,10) +
  labs(title = "MURDERS VERSUS BURGLARIES IN US STATES PER 100,000",
  caption = "Source: U.S. Census Bureau and Nathan Yau",
  x= "Burglary rates in each state per 100,000",
  y= "Murder rates in each state per 100,000",
  size = "Aggravated Assault") +
  theme_minimal(base_size = 12)

Warning: Removed 1 row containing missing values or values outside the scale rang
e
(`geom_point()`).
```



Finally, add some interactivity to the plot with plotly

⚠ Plotly is "dirty", meaning it adds interactive mouse-over tootip capabilities, but it causes other elements of the plot to stop working. In this case, we lose the legend.

```
p <- ggplot(crime2,
            aes(x = burglary,
                y = murder,
                size = aggravated_assault,
                text = paste("State:", state, "Population:", population))) +
  geom_point(alpha = 0.5, color = "maroon") +
  xlim(250,1200) +
  ylim(0,10) +
  labs(title = "MURDERS VERSUS BURGLARIES IN US STATES PER 100,000",
  caption = "Source: U.S. Census Bureau and Nathan Yau",
  x= "Burglary rates in each state per 100,000",
  y= "Murder rates in each state per 100,000",
  size = "Aggravated Assault") +
  theme_minimal(base_size = 12)
p <- ggplotly(p)
p
```



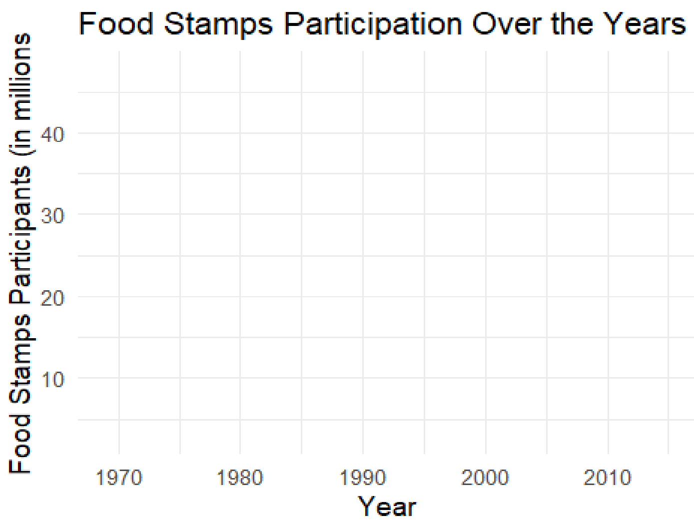
Make a series of charts from food stamps data

Now we will explore a series of other geom functions using the food stamps data.

Load the data, map variables onto the X and Y axes, and save chart template

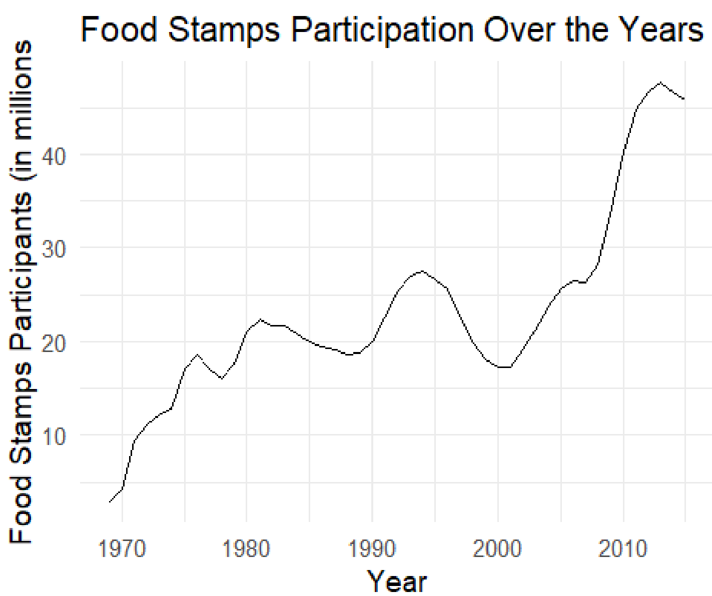
```
# load data
setwd("C:/Users/rsaidi/Dropbox/Rachel/MontColl/Datasets/Datasets")
food_stamps <- read_csv("food_stamps.csv")

Rows: 47 Columns: 3
── Column specification ────────────────────────────────────────────────
Delimiter: ","
dbl (3): year, participants, costs


# save basic chart template
food_stamps_chart <- ggplot(food_stamps, aes(x = year, y = participants)) +
  labs(title = "Food Stamps Participation Over the Years",
       x = "Year",
       y = "Food Stamps Participants (in millions") +
```

```
  theme_minimal(base_size = 14)
food_stamps_chart
```



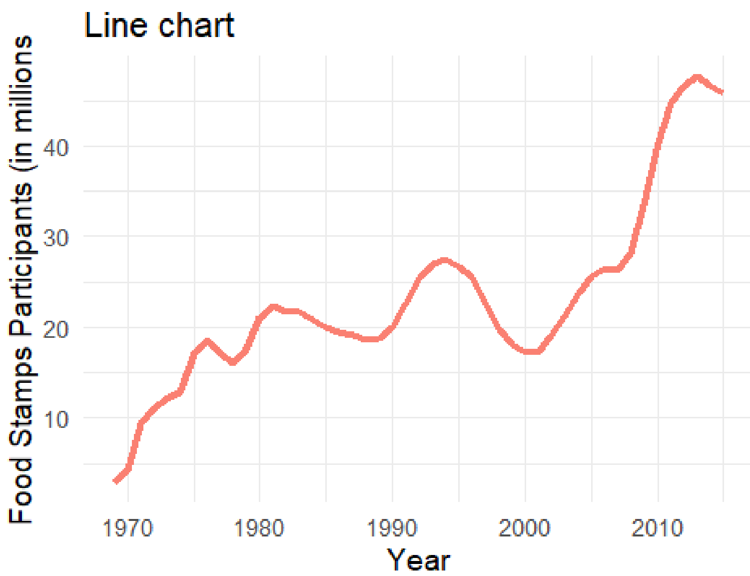
Make a line chart

```
food_stamps_chart +
  geom_line()
```



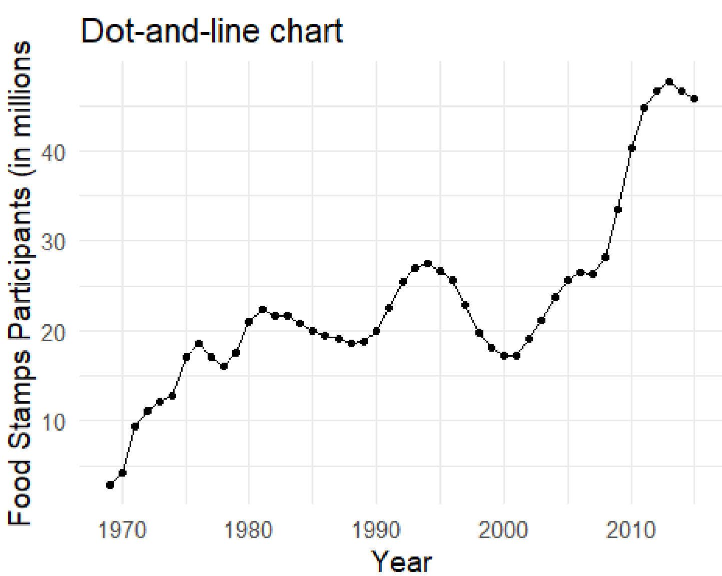
Customize the line, and add a title

```
food_stamps_chart +
  geom_line(size = 1.5, color = "salmon") +
  ggtitle("Line chart")
```



Add a second layer to make a dot-and-line chart

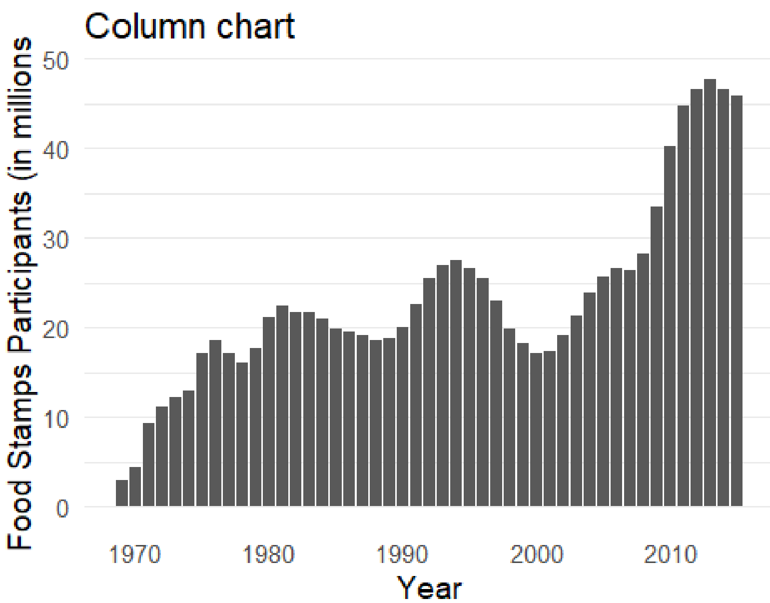
```
food_stamps_chart +
  geom_line() +
  geom_point() +
  ggtitle("Dot-and-line chart")
```



Make a column chart, then flip its coordinates to make a bar chart

```
# Make a column chart
food_stamps_chart +
  geom_bar(stat = "identity") +
  ggtitle("Column chart") +
```

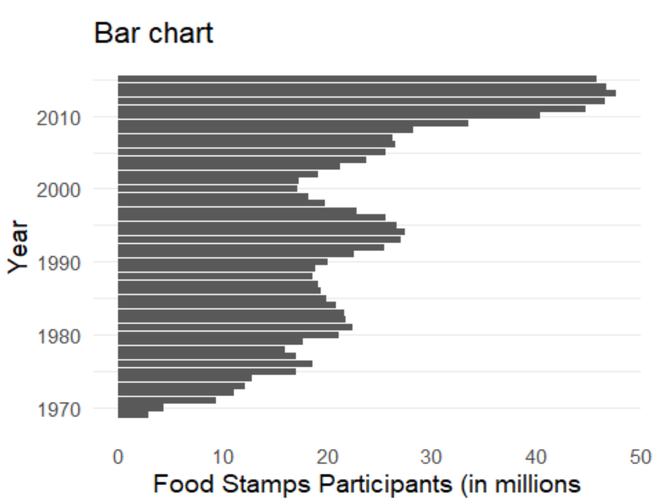
```
  theme(panel.grid.major.x = element_blank(),
        panel.grid.minor.x = element_blank())
```



geom_bar works a little differently to the geoms we have considered previously. If you have not mapped data values to the Y axis with aes, its default behavior is to set the heights of the bars by counting the number of records for values along the X axis. If you have mapped a variable to the Y axis, and want the heights of the bars to represent values in the data, use you must use stat="identity".

coord_flip switches the X and Y axes.

```
# Make a bar chart
food_stamps_chart +
  geom_bar(stat = "identity") +
  ggtitle("Bar chart") +
  theme(panel.grid.major.x = element_blank(),
        panel.grid.minor.x = element_blank()) +
  coord_flip()
```



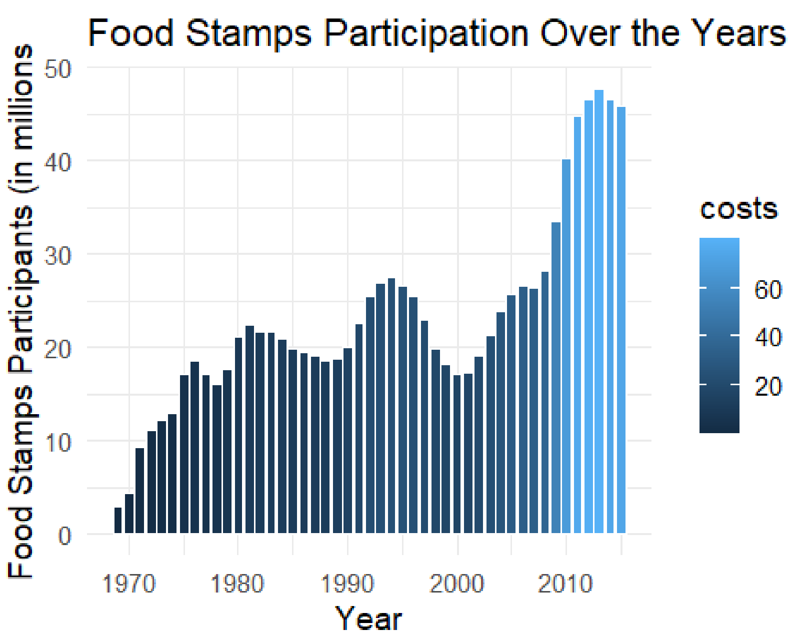
The difference between color and fill

For some geoms, notably geom_bar, you can set color for their outline as well as the interior of the shape.

When setting colors, color refers to the outline, fill to the interior of the shape.

Map fill color to the values of a continuous variable

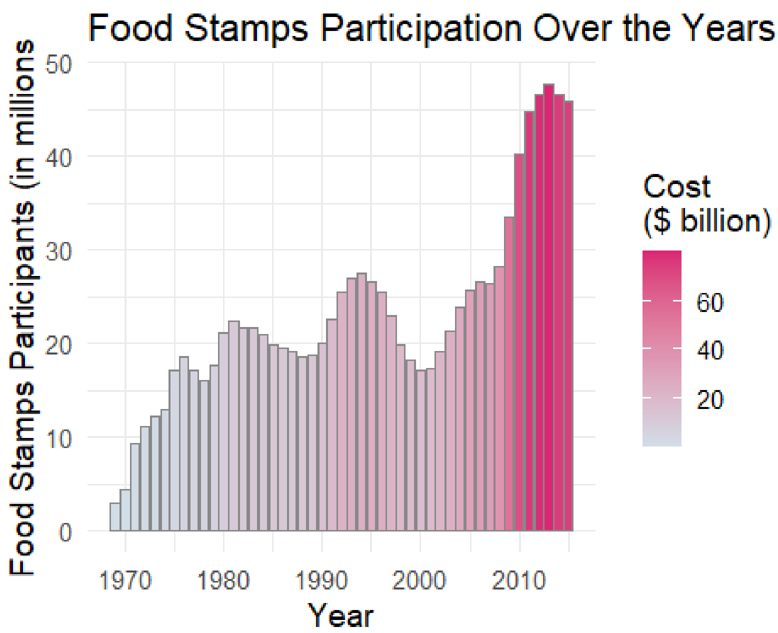
```
# fill the bars according to values for the cost of the program
food_stamps_chart +
  geom_bar(stat = "identity", color= "white", aes(fill = costs, direction = -1))

Warning in geom_bar(stat = "identity", color = "white", aes(fill = costs, :
Ignoring unknown aesthetics: direction
```



This code uses an aes mapping to color the bars according values for the costs of the program, in billions of dollars. ggplot2 recognizes that costs is a continuous variable, but its default sequential scheme applies more intense blues to lower values, which is counterintuitive.

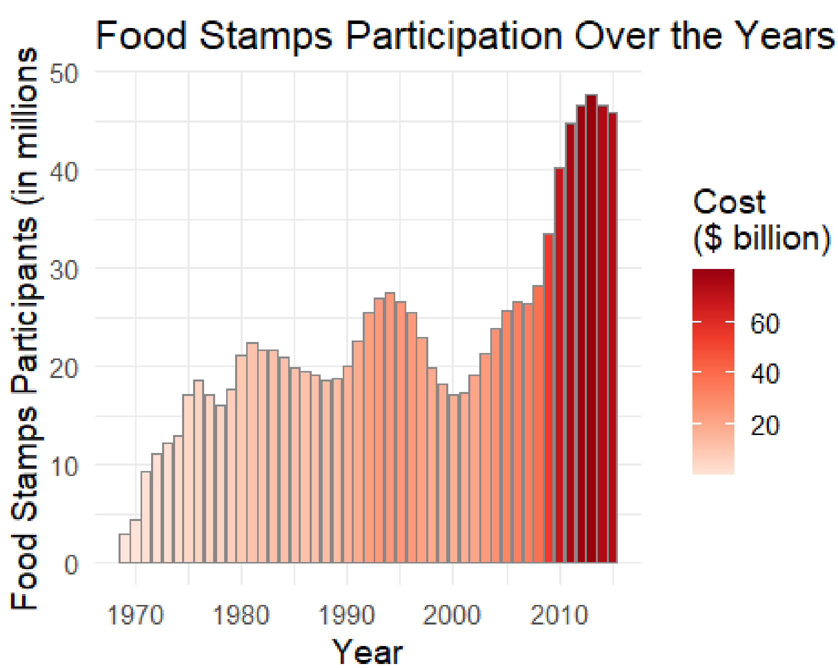
Use a ColorBrewer sequential color palette

```
# use a colorbrewer gradient levels for intensity
food_stamps_chart +
  geom_bar(stat = "identity", color = "#888888", aes(fill = costs)) +
  scale_fill_gradient(name = "Cost\n($ billion)", low = "#d1dee8", high = "#d9277
4")
```



scale_fill_distiller (and scale_color_distiller) work like scale_color_brewer, but set color gradients for ColorBrewer's sequential and diverging color palettes; direction = 1 ensures that larger numbers are mapped to more intense colors (direction = -1 reverses the color mapping). Try changing the code I have: scale_fill_gradient() to scale_fill_distiller with different directions (1 or -1).

```
food_stamps_chart +
  geom_bar(stat = "identity", color = "#888888", aes(fill = costs)) +
  scale_fill_distiller(name = "Cost\n($ billion)", palette = "Reds", direction =
1)
```



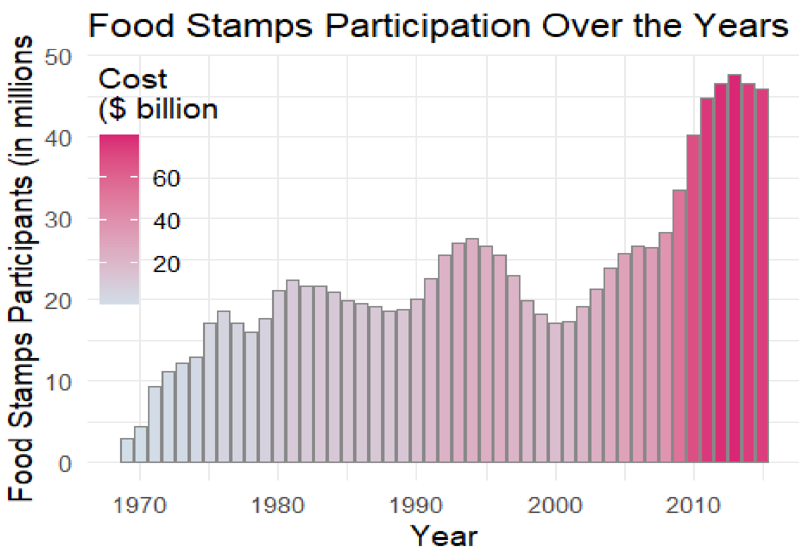
Notice also the in the title for the legend. This introduces a new line.

Control the position of the legend

This code uses the theme function to moves the legend from its default position to the right of the chart to use some empty space on the chart itself.

```
food_stamps_chart +
  geom_bar(stat="identity", color = "#888888", aes(fill=costs)) +
  scale_fill_gradient(name = "Cost\n($ billion", low = "#d1dee8", high = "#d92774
") +
  theme(legend.position=c(0.1,0.7))
```

```
Warning: A numeric `legend.position` argument in `theme()` was deprecated in ggpl
ot2
3.5.0.
ℹ Please use the `legend.position.inside` argument of `theme()` instead.
```

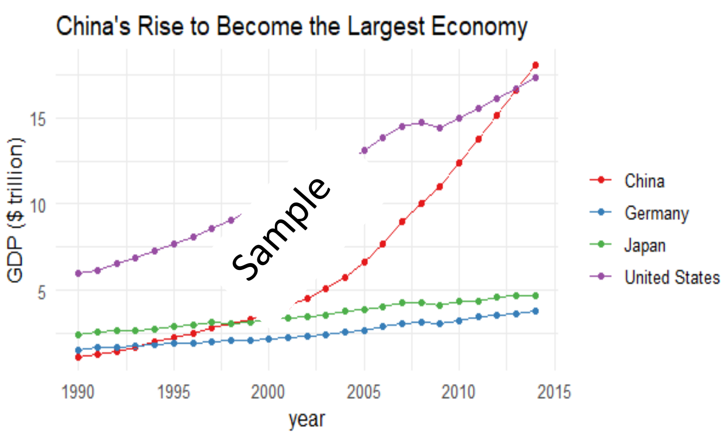


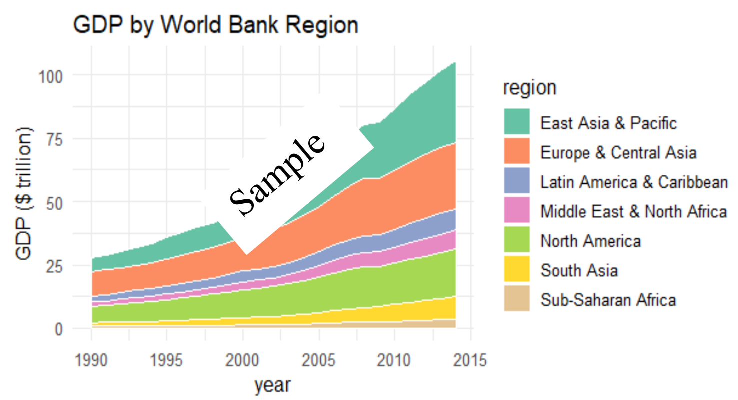
The coordinates for the legend are given as a list: The first number sets the horizontal position, from left to right, on a scale from 0 to 1; the second number sets the vertical position, from bottom to top, again on a scale from 0 to 1.

Week 6 Homework Assignment

1. **(Ungraded)** Complete copying notes on scatterplotting, correlation, and regression analysis.

2. **(Worth up to 10 points for each chart)** Use **`dplyr`** and **`ggplot2`** (from **`tidyverse)`** to process data and draw these two charts (shown below) from the **`nations`** dataset. You do NOT need to incorporate interactivity, but you can, if you want to challenge yourself. Both charts should be created on one single Quarto document and rendered to rpubs.





Details for Nations Dataset Charts Assignment

- For both charts, you will first need to create a new variable in the data, using `mutate` from **dplyr**, giving the **gdp** of each country in trillions of dollars, by multiplying `gdp_percap` by `population` and dividing by a trillion. 10^12

- Draw both charts with **ggplot2**.

- For the first chart, you will need to `filter` the data with **dplyr** for the four desired countries. When making the chart with **ggplot2** you will need to add both `geom_point` and `geom_line` layers, and use the `Set1 ColorBrewer` palette using: `scale_color_brewer(palette = "Set1").`

- For the second chart, using **dplyr** you will need to `group_by region` and `year`, and then `summarize` on your mutated value for gdp using `summarise(sum_GDP = sum(gdp, na.rm = TRUE)).` (There will be null values, or NAs, in this data, so you will need to use `na.rm = TRUE).`

- Each region's area will be generated by the command `geom_area` ()

- When drawing the chart with **ggplot2**, you will need to use the `Set2` ColorBrewer palette using `scale_fill_brewer(palette = "Set2")`

- Think about the difference between `fill` and `color` when making the chart, and where the above `fill command needs to go in order for the regions to fill with the different colors` when making the chart, and put a very thin white line around each area.

Render your code for both charts and save your work in rpubs. Submit the link on the assignment dropbox **by 11:59 pm on Sunday, March 9th .**