Module B: Knowing Your Data

Slide Deck B2:

# Graphics for Numeric Variables

*The section in which we see some typical graphics used to understand/see numeric variables. Remember that the researcher must understand their data before trying to analyze it.*



## Today's Objectives

By the end of this slidedeck, you should

1. create the following univariate graphics
   - histogram
   - density plot
   - overlay plot
   - box-and-whiskers plot
2. create the following bivariate graphics
   - side-by-side boxplot
   - scatter plot
     - correlation
   - line plot
3. determine which graphic(s) are appropriate to tell the story of your data
4. interpret the presented graphics of others
5. modify graphics to make them presentation-worthy

## Code Preparation

To perform the code given in this slidedeck, please start R and run the following lines in the `Script` window in R:

```
source("http://rfs.kvasaheim.com/stat200.R")

dt = read.csv("http://rfs.kvasaheim.com/data/crime.csv")
attach(dt)
```





## The Theory of Graphics

**Recall**:
In R, there are three main graphics paradigms that can be used. We will be using the base graphics in this course. The metaphor it uses seems natural to me:

**The Painter's Canvas**

This is the framework in which you *plan* your painting. Then, you...

- define the parameters of the canvas (set parameters)
- start the canvas (initialize)
- add to the canvas (annotate)

## Some Basic Numeric Graphics

Basic **uni**variate numeric graphics consist of

- Histograms — numeric
- Density plots — numeric
- Box-and-Whiskers plots — numeric

Basic **bi**variate numeric graphics consist of

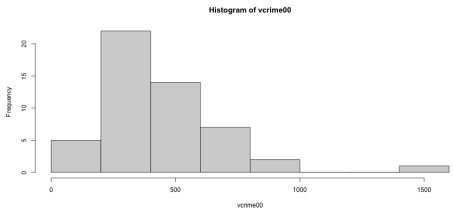
- Side-by-side Box-and-Whiskers plots — numeric × categorical
- Scatter plots — numeric × numeric





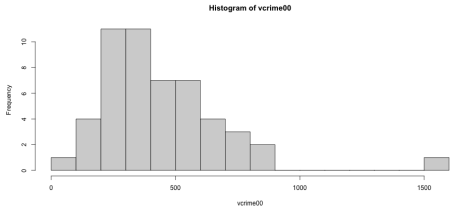
## Histogram

```
hist(vcrime00)
```



## Histogram

```
hist(vcrime00, breaks=11)
```

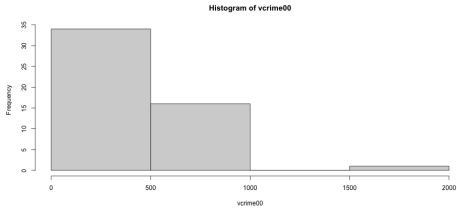






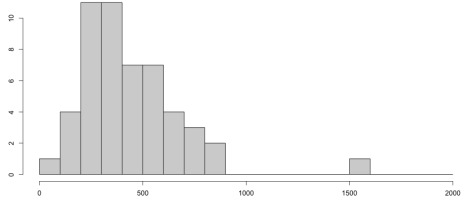

## Histogram

```
hist(vcrime00, breaks=5)
```



## Histogram

```
hist(vcrime00, breaks=seq(0,2000,100))
```

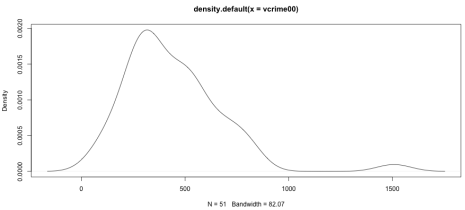






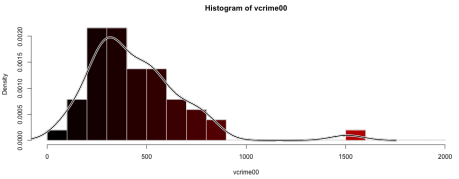

## Density Plot

```
plot(density(vcrime00))
```



## Density Plot

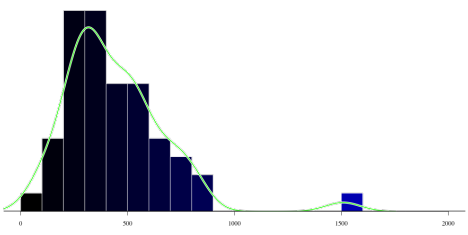
```
hist(vcrime00, freq=FALSE, breaks=seq(0,2000,100), col=rgb(0:20/20,0,0),
border="white")
lines(density(vcrime00))
```





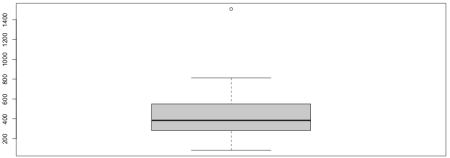



## Density Plot



## Box-and-Whiskers Plot

```
boxplot(vcrime00)
```

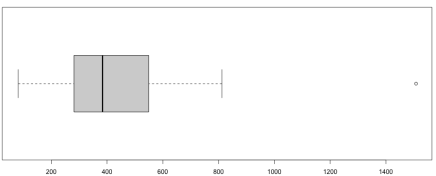






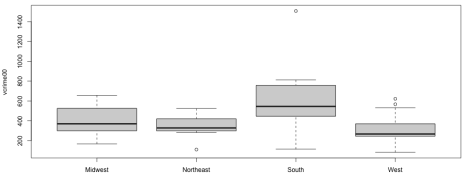

## Box-and-Whiskers Plot

```
boxplot(vcrime00, horizontal=TRUE)
```



## Side-by-Side Box-and-Whiskers Plot

```
boxplot(vcrime00 ~ census4)
```

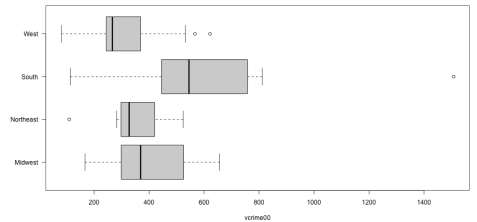




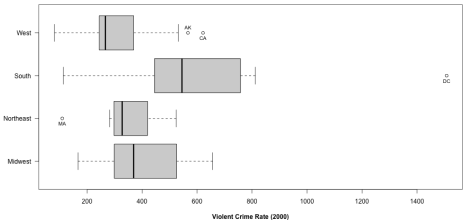



## Side-by-Side Box-and-Whiskers Plot

```
boxplot(vcrime00 ~ census4, horizontal=TRUE, las=1)
```



## Side-by-Side Box-and-Whiskers Plot

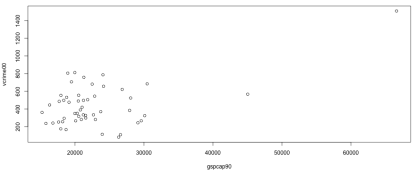







## Scatter Plot

```
plot(vcrime00 ~ gspcap90)
```




```
cor.test(vcrime00,gspcap90) # r=0.529153
```

## Modifying Graphics

Recall the metaphor for base graphics: **The Painter's Canvas**.

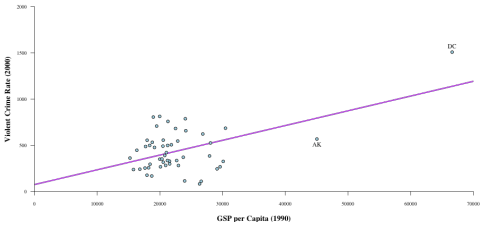
Plan your painting. Then, with that plan...

- define the parameters of the canvas
- start the canvas
- add to the canvas





## Scatter Plot Goal



## The Parameters

Here are some things to think about before starting your graphic:

- what margins will you use?
- what font? should it differ between labels and values?
- what sizes of the fonts? should it differ between labels and values?
- should the axes have additional space?
- what should the value orientation be?

All of these questions should be set in the first part of the graphic, the 'parameters' section.





## The Parameters

The following code:

- sets the margins to be 4, 4, 1, and 1 (lines at the bottom, left, top, and right);
- specifies the font family is serif, with labels bolded (**2**);
- specifies the labels are 20% larger and values are 20% smaller than default;
- requires the axes to meet at 0 (no axis padding); and
- forces all labels to be horizontal (for the sake of your readers).

```
par(mar=c(4,4,1,1))
par(family="serif", font.lab=2)
par(cex.lab=1.2, cex.axis=0.8)
par(xaxs="i", yaxs="i")
par(las=1)
```

## The Start

The following code:

- starts the graphic; and
- sets the viewing window to $(0, 70000) \times (0, 2000)$.

```
plot.new()
plot.window( xlim=c(0,70000), ylim=c(0,2000) )
```

Note that this will start a generic window of that size/dimension.

*Again, some graphing functions have their own "starting" functions. So, in those cases, the above two lines are not needed.*





## The Annotation

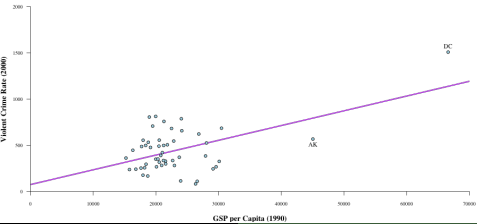
The following code:

- adds the line of best fit (spiffified);
- adds the points (light-blue-filled circles);
- adds the values and labels; and
- identifies the two outliers.

```
abline(lm(vcrime00~gspcap90), lwd=3, col="purple")
abline(lm(vcrime00~gspcap90), lwd=1, col="pink")

points(gspcap90,vcrime00, pch=21, bg="lightblue")

axis(1); axis(2)
title(xlab="GSP per Capita (1990)")
title(ylab="Violent Crime Rate (2000)")

text(gspcap90[24],vcrime00[24], label=scode[24], pos=3)
text(gspcap90[50],vcrime00[50], label=scode[50], pos=1)
```

## The Resulting Scatter Plot

These all come together to produce:

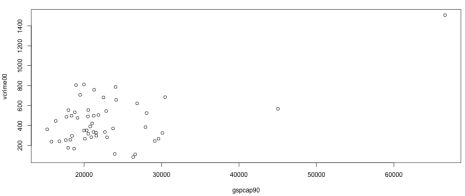







## The Original Scatter Plot

Recall the original (utilitarian) graphic:



## Maps in R

Note that maps are especially difficult to make, but well worth the effort for geographical data.

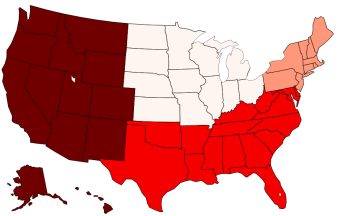
- Properly making maps is an art form.
- There are a lot of features of maps that need to be considered when plotting.
- Because of this, maps are especially difficult in any program, even in R.
- There are several packages that help in map creation.
  - `plotly`
  - `sp`
  - `sf`
- For the adventurous (or picky) amongst us, we can use R's strength.
  - One can modify any image as you want. Thus, starting with a map and modifying it is an option. It is rarely easy, but it is possible, and may be well worth it.
  - In my research, I tend to focus on creating maps from scratch, adding several features to them to increase the data density.





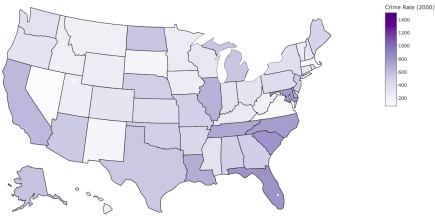
## Plotly Map of Categorical Data

Here is a map of the four census regions using `plotly`.



## Plotly Map of Numeric Data

Here is a map of the violent crime rate in 2000 using `plotly`.

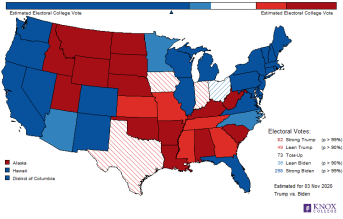







## A Map from Scratch

Here is a map I created to illustrate the likelihood a state will vote in each direction.



## Today's Objectives

Now that we have concluded this lecture, you should be able to

1. create the following univariate graphics
   - histogram
   - density plot
   - overlay plot
   - box-and-whiskers plot
2. create the following bivariate graphics
   - side-by-side boxplot
   - scatter plot
   - line plot
3. determine which graphic(s) are appropriate to tell the story of your data
4. interpret the presented graphics of others
5. modify graphics to make them presentation-worthy





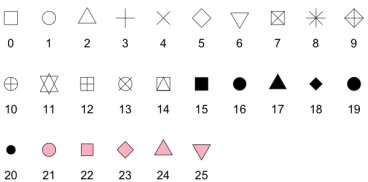
## Today's R Functions

In this slide deck, we covered the following R functions:

- `hist`
- `density`
- `boxplot`
- `plot`
- `par`
- `plot.new`
- `plot.window`
- `points`
- `lines`
- `abline`

## Plotting Characters

Several of the plotting characters (`pch`) available in R:








## Supplemental Activities

The following may be of interest to you in terms of today's topics:

- SCA 3a is for categorical graphics
- SCA 3b is for **numeric** graphics

Note that you can access all Statistical Computing Activities here:
`https://www.kvasaheim.com/courses/stat200/sca/`

## Supplemental Readings

The following are some readings that may be of interest to you in terms of graphing in R:

- R Graphics Cookbook, 2nd edition
  `https://r-graphics.org`
- R Coder
  `https://r-coder.com/r-graphs/`
- Statistical Methods and Data Analytics UCLA
  `https://stats.oarc.ucla.edu/r/codefragments/introduction/`
- Some colors and ideas:
  - `https://colorbrewer2.org/`
  - `https://r-charts.com/colors/`





## Supplemental Readings

The following may be of interest to you in terms of today's topics:

- Hawkes Learning: Section 2.3
- Intro to Modern Statistics: Chapters 5 and 6
- R for Starters: Nothing