

Lecture 4:

What Graphs Do and Making Bar Charts

February 12, 2018

Overview

Course Administration

Good, Bad and Ugly

Few, Chapters 6

Bar Charts in R

Course Administration

1. Will return proposal comments during programming
2. Rosa has graded problem sets – thank you
3. Grades posted?
4. Missing anything from me?

Next Week's Good Bad and Ugly

Monday by 9 am. Earlier is ok.

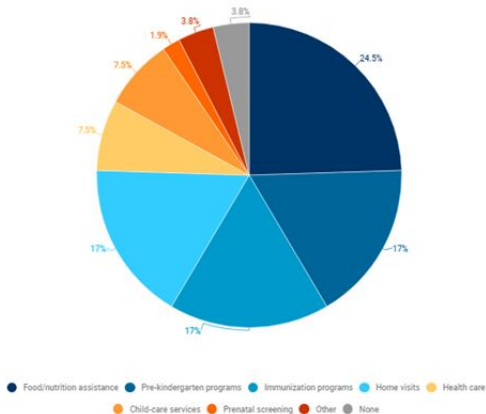
- Adam Brooks
- Gulfishan Khadim

This Week's Good Bad and Ugly

- Kelsey Wilson
- Nathan Rupp
- Haley Dunn

Kelsey's Example

Figure 1: Most Prevalent Types of ECD Programs Administered by Counties

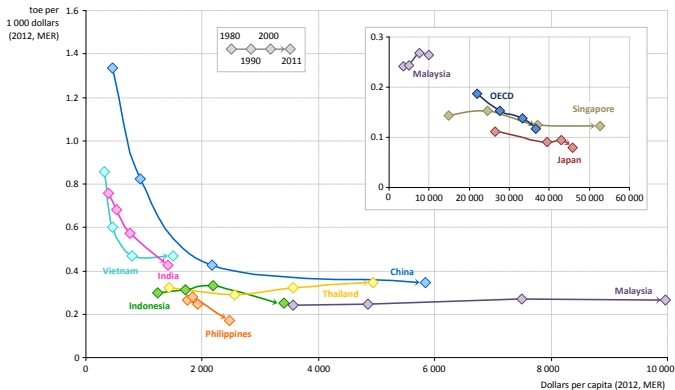


Note: Figure 1 represents the percent of respondents who indicated that counties in their state provide that type of ECD program.

Nathan's Example

© OECD/IEA, 2013

Figure 4.1 Trends in energy intensity and GDP per capita in selected countries, 1980 - 2011

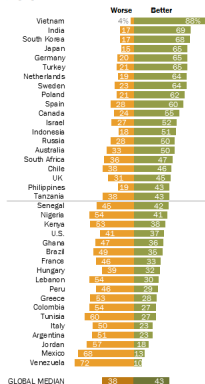


Note: GDP is measured at market exchange rates (MER) in year-2012 dollars.

Haley's Example

Globally, assessments vary on whether life is better or worse than 50 years ago

Life in our country today is ___ than it was 50 years ago
for people like me



Note: "About the same" responses not shown.
Source: Spring 2017 Global Attitudes Survey. Q3. U.S. survey conducted June 27-July 9, 2017.

PEW RESEARCH CENTER

Few:

Fundamental Variations of Graphs

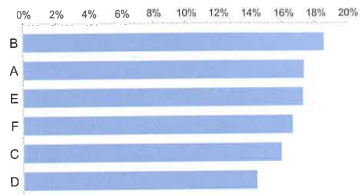
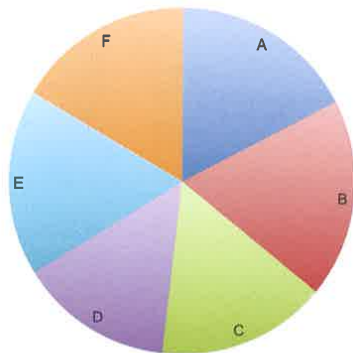
Today

1. Types of graphs
2. What you can communicate, by graph

1. Types of Graphs

- Points
- Lines
- Bars
- Boxes
- Shapes with varying 2-D areas
- Lines

Why to Avoid 2-D Sizes



Graph Design Solutions

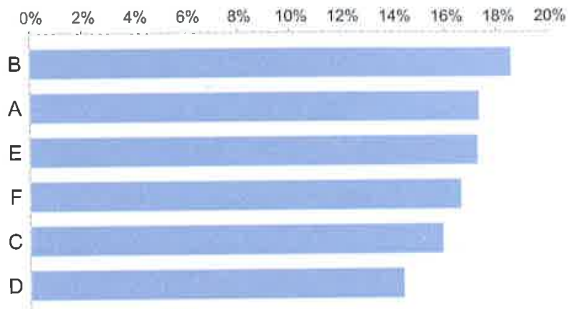
As we go through these, we'll discuss policy examples

1. Nominal Comparisons
2. Time Series Designs
3. Ranking Designs
4. Part-to-Whole Designs
5. Deviation Designs
6. Distribution Designs
7. Correlation Designs
8. Geospatial Designs

1. Nominal Comparisons

- Use bars: horizontal or vertical
- Or points to compare values
- Possible for not “too many” values

For Example



2. Time Series Designs

- Present data over time: months, days, hours, years, decades, ...
- Almost required to use horizontal axis left to right for time
- And usually a connected line, with or without dots
- If time intervals are not consistent, then maybe dots or bars
- Lines indicate connection between observations, so watch out if you're using them in another context

For Example

Market summary > KB Home

NYSE: KBH - Feb 9, 4:02 PM EST

29.42 USD 0.00 (0.00%)

1 day

5 day

1 month

3 month

1 year

5 year

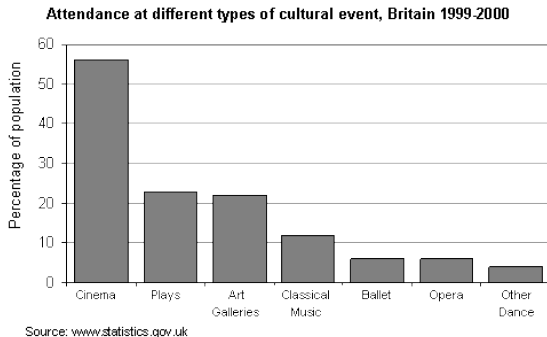
max



3. Ranking Designs

- Like nominal, but ranked
- So use a bar chart
- And sort by value
- Put the item you want to call attention to at top or left

For Example

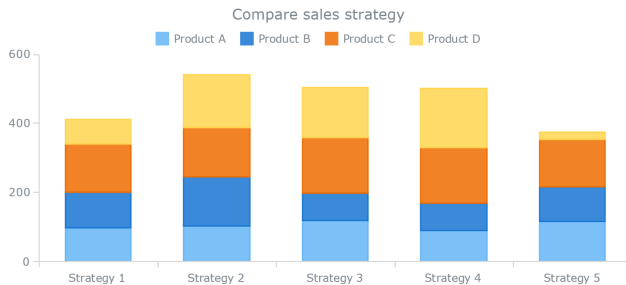


Courtesy of [this site](#)

4. Part-to-Whole Designs

- Comparison of shares
- Use simple bar
- Use stacked bar only when you want to compare across categories
- So use a bar chart
- And sort by value
- Put the item you want to call attention to at top or left

For Example

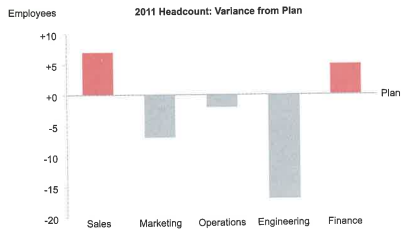
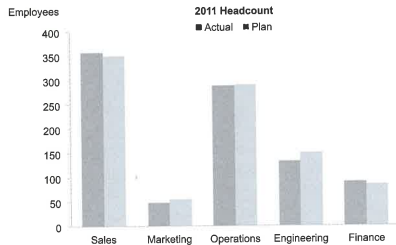


Courtesy of [this site](#)

5. Deviation Designs

- Highlight differences across types
- Paired bars
- Doesn't work too well with too many comparison categories
- Use stacked bar only when you want to compare across categories
- More sophisticated (not in Few): scatterplot and compare to 45 degree line

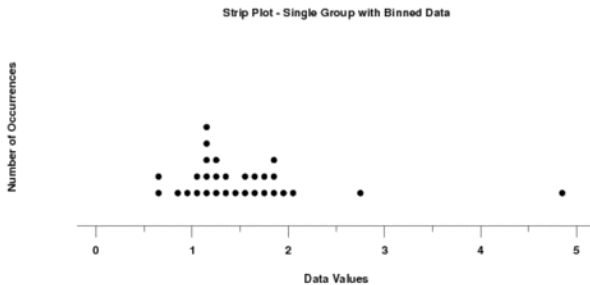
For Example



6. Distribution Designs

- Distributions can be continuous or by bin
- And you want to display one or many
- Use a bar chart
- Or a line chart
- Or a box plot – not too keen on these

For Example



Courtesy of [this site](#)

7. Correlation Designs

- Scatter chart
- Or scatter and trend line
- You can enhance scatter with color and weight variations
- Too many variations are not comprehensible
- No pic as you know this one

8. Geospatial Designs

- Map with
- Color fills
- Lines
- Much more on this later in the course

Bar Charts in R

Today's Goals

- A few non-graph commands
 - `ifelse`
 - `data.frame, c`
- For graphing, via `ggplot`
 - `geom_bar()`
 - `geom_text()`, `geom_label()`
 - `theme()`

A Basic Programming Command: `ifelse`

```
data$var <- ifelse(test_expression, [outcome if  
true], [outcome if false])
```

- `var`
 - Outcome is a variable in the dataframe `data`
 - Or something to do, instead of a variable
- `test_expression`
 - an expression that is evaluated, e.g. $x > y?$, $a = b?$
- After evaluation
 - if $x > y$, then you get the outcome if true – the second element
 - if $X < y$, then you get the outcome if false – the third element
 - you can nest another `ifelse` in the third one

Get Started and Make Your Own Dataframe: `data.frame`

```
dfname <- data.frame(col1 =, col2 =, ...)
```

- `data.frame` creates a dataframe called `dfname`
- write column as `name = c("e1", "e2", ... "en")`

Make a dataframe

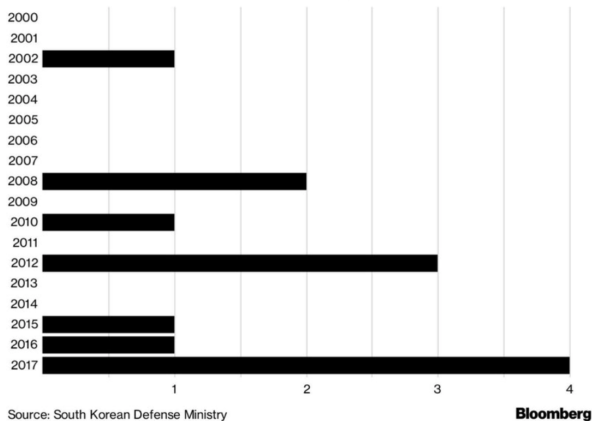
```
newframe <- data.frame(fruit = c("apples", "bananas",  
                                "pomegranates"),  
                       price.per.lb = c("2.49", "0.79",  
                                         "6"),  
                       junk = rep(1,  
                                length(c("apples",  
                                         "bananas",  
                                         "pomegranates")))  
newframe
```

```
##      fruit price.per.lb junk  
## 1    apples      2.49     1  
## 2   bananas      0.79     1  
## 3 pomegranates      6     1
```


Meryl's Example from Last Class

Runs Across the Border

Four North Korean soldiers have defected south this year -- the most since at least 2000



A Dataframe from Last Class's Bad Graph

```
# load north korean data
nkd <- data.frame(year = c("2000", "2001", "2002", "2003",
                           "2004", "2005", "2006", "2007",
                           "2008", "2009", "2010", "2011",
                           "2012", "2013", "2014", "2015",
                           "2016", "2017"),
                  defectors = c("0", "0", "1", "0", "0",
                                "0", "0", "0", "2", "0",
                                "1", "0", "3", "0", "0",
                                "1", "1", "4"))

nkd
```

##		year	defectors
## 1	2000		0
## 2	2001		0
## 3	2002		1
## 4	2003		0
## 5	2004		0

And on to ggplot

For today we're exploring

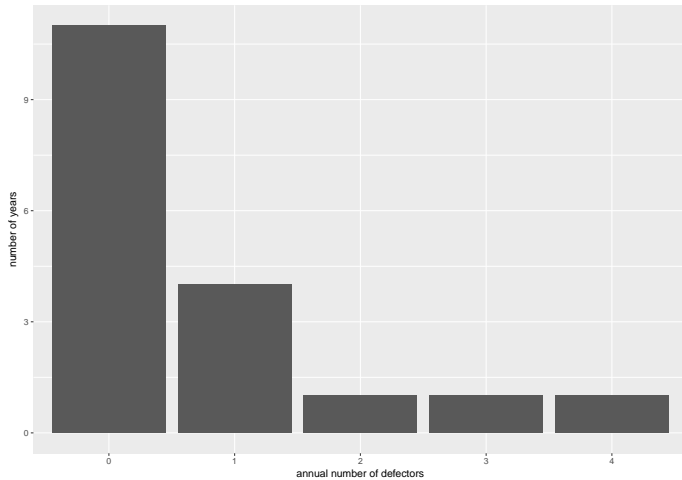
- `geom_bar`
- `geom_text`
- `theme`

Making a Bar Chart

```
# make a bar chart  
library(ggplot2)  
ggplot(nkd, aes(x=defectors)) + geom_bar() +  
  labs(x = "annual number of defectors",  
       y="number of years")
```

- ▶ call the ggplot library
- ▶ a similar first part to last lecture:
 - ▶ x axis is determined by quantity of defectors
- ▶ tell R we want a bar chart with geom_bar
 - ▶ default is to count the total number of observations by type (defectors)
- ▶ labs() makes the graph comprehensible

What It Looks Like



Other Options for `geom_bar()`

- ▶ if you want R to use the value in the dataframe, rather than counting observations, use `geom_bar(stat="identity")`
- ▶ you can control aesthetics within the bar via `geom_bar(aes(fill= [something]))`, useful for stacked graphs
- ▶ you can weight the totals
- ▶ zillions more are available

geom_text() to Put Things on Your Chart

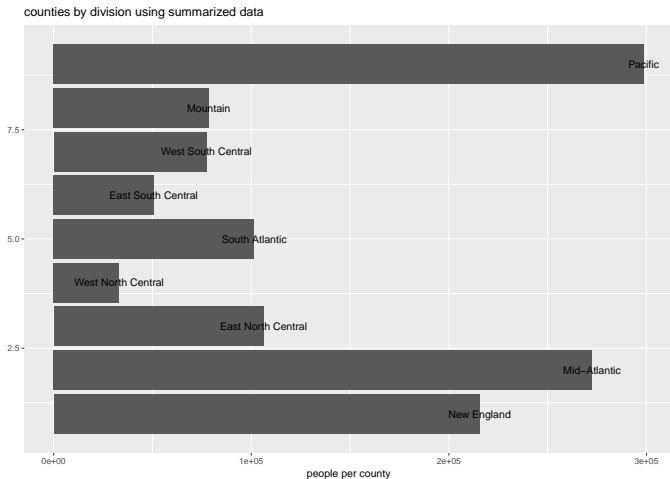
- ▶ puts variable value (maybe a fruit name) where you say based on the value of another variable
- ▶ very powerful: need to set up data the right way to use this power

geom_text() Example

Adding text to the chart with `geom_text`, telling R that the mapping for labels is `divisions$div.name`.

```
ggplot(data = divisions, aes(x = division, y=ppl.by.cnty))  
  geom_bar(stat = "identity") +  
  ggtitle("counties by division using summarized data") +  
  coord_flip() +  
  labs(x="", y="people per county") +  
  geom_text(mapping = aes(label=div.name))
```


What This Looks Like

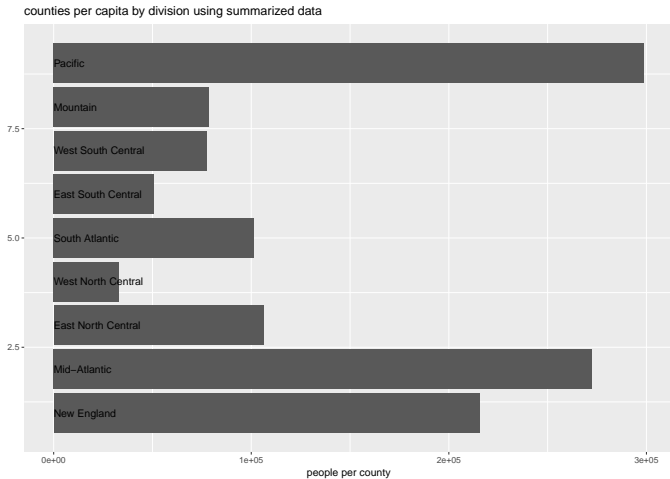


Fixing the Previous

```
# make labels legible
divisions$nada <- c(rep(0, length(divisions$div.name)))
ggplot(data = divisions, aes(x = division,
                             y=ppl.by.cnty)) +
  geom_bar(stat = "identity") +
  ggtitle("counties per capita by division
          using summarized data") +
  coord_flip() +
  labs(x="", y="people per county") +
  geom_text(mapping = aes(y=nada, label=div.name),
            hjust = 0)
```

- make a new variable that tells R where to put the name

How Does it Look?



ggplot's theme Commands

- ▶ a theme is a set of commands that standardize the look of the graph
- ▶ ggplot has a built-in default
- ▶ you can choose another default
- ▶ or modify the theme
- ▶ we'll focus on the latter

Modifying the Default Theme

- ▶ there are > 60 different parts of the default theme, including
 - ▶ `axis.ticks.x()`
 - ▶ `legend.title()`
 - ▶ `legend.box.margin()`
- ▶ see them all [here](#)
- ▶ in this class we mostly get rid of parts by adding the below to the `ggplot` command
 - ▶

```
theme(panel.grid.major = element_blank(),  
      panel.grid.minor = element_blank(),  
      axis.ticks.y=element_blank())
```

Try Today's Tutorial

- Pay attention to the output of each bit
- Go forth!

Next Lecture

- Turn in PS 4
- Read Few Chapter 6
- R Graphics Cookbook, Chapter 4
- Next policy brief deadline: April 2 for draft