Effective Communication

Acknowledgments:

This lecture is entirely based on:
STOR 390: Introduction to Data Science
Effective communication
04/11/17
https://idc9.github.io/stor390/slides/communication.pdf

Effective communication will make better at whatever you are doing

Final project grade

Communication is **context dependent**

Audience

Medium

Content

Time

Purpose

Differing types of audiences

Technical vs. non-technical

Familiarity with topic

Results vs. method

Native vs. non-native language

Mixed

Many mediums used in data science

Speaking

Text document

Static visualization

Dynamic visualization

Interactive application

Slide presentation

Web page

Literate programming

Communication is for more than just conveying results

Coding

Coordinating with collaborators

Asking for help

“Effective communication is optimization under constraints.”

–Trees, Maps, and Theorems

Lecture outline

Four general principles

Several strategies

Some examples

Four rules of communication

1. Adapt to your audience.

2. Maximize the signal to noise ratio.

3. Use effective redundancy.

4. Trade-offs.

1-3 are from Trees, Maps and Theorems

Adapt to your audience

Empathy

Understand your audience

Generosity

Effectiveness

“Much like being customer-minded in business or being user-friendly in software development, adapting to one’s audience is really a question of **effectiveness** **more than one of selflessness**.”

Trees, Maps and Theorems

Many types of audiences

Familiar or unfamiliar with the topic

Technical or non-technical

Expert in the topic

Native or non-native language speakers

Interested or uninterested

Mixed audience

Maximize signal to noise ratio

“Nothing is neutral in communication”

–Trees, Maps, and Theorems

Maximize signal to noise ratio

Audience sees/hears everything

Any detail either

- Helps convey message
- Hampers the message

1.**Support Vector Machine** is a very powerful and widely used **classification algrithm** used by many people who **machine learning** practitioners.

2. Support Vector Machine is an effective classification algorithm.

1.**Support Vector Machine** is a very powerful and widely used **classification algrithm** used by many people who **machine learning** practitioners.

Too wordy
Too much highlighting
Typos
Awkward grammar

2. Support Vector Machine is an effective classification algorithm.

Maximize signal to noise ratio

Audience sees/hears everything

Any detail either

- Helps convey message
- Hampers the message

Clear understanding of your message

Use effective redundancy

Communicate across multiple channels

Color
Text
Shape

Use effective redundancy

Communicate across multiple channels

Repetition

“Tell them what you are going to tell them. Tell them. Then tell them what you told them.”

– Aristotle (roughly)

Trade-offs

Time is usually the biggest cost

More vs. less detail

Targeting different audiences

“There ain’t no such thing as a free lunch.”

– (popularized by) Milton Friedman

Communication strategies

Revision

Message then details

Hierarchy

Easy to navigate structure

Communicate at different levels

Revise, revise, revise

Many rounds of revision

Outside feedback

“When revising go for the jugular.”

– Calum Carmichael

State the message first, then the details

“Too often, when we communicate with data, we don’t make our point clear. We leave our audience guessing. **Your audience should never have to guess what message you want them to know.** The onus is on the person communicating the information (you!) to make that clear.”

–Cole Knaflic

State the message first, then the details

Message > details

State message

- Explicitly
- At the beginning

No detective stories

Both macro and micro scale

Motivate the message

1. Motivation

2. Message

3. Details

Examples of message first

Executive summary

Upshot in title

- graphic
- Slide

Function names

str_extract vs. grep

Intuition then formal definition

State the message explicitly

State the message explicitly

You suck

vs.

You suck ;-)

“How can I know what I think until I see what I say.”

– Mr. Anderson

State the message first, then the details

Message > details

State message

- Explicitly
- At the beginning

No detective stories

Both macro and micro scale

Understand your thesis

Hierarchical is better than sequential

Humans process hierarchy better than sequence

Easier to remember

Depth proportional to document length

Examples of hierarchy

Sections, subsections

Kingdon, phylum, …

Helper functions

Grocery aisles

Sequential description

My research has both theoretical and applied components: dimensionality reduction for network valued random variables, temporally evolving preferential attachment models, support vector machine in high dimensional settings, DTI structural connectivity networks, text analysis of Supreme Court decisions.

Hierarchical description

My research has two components:

Theory

- Dimensionality reduction for network valued random variables.
- Temporally evolving preferential attachment models.
- Support vector machine in high dimensional settings.

Application

- DTI structural connectivity networks.
- Text analysis of Supreme Court decisions.

Make the structure easy to navigate

Structure visible at the beginning

Audience should know where they are

Floating TOC

Sections, subsections, page numbers

Transition slides

Communicate at different levels

Different types of audience members

One person can change types

Appendix

Message First

Executive summary

Data science examples

Static visualizations

Dynamic visualizations

Programming

R Markdown (literate programming)

Asking questions

Static visualizations

Exploration

Communication

Misleading plots

Exploratory plots: details over message and quantity over quality.

Many plots

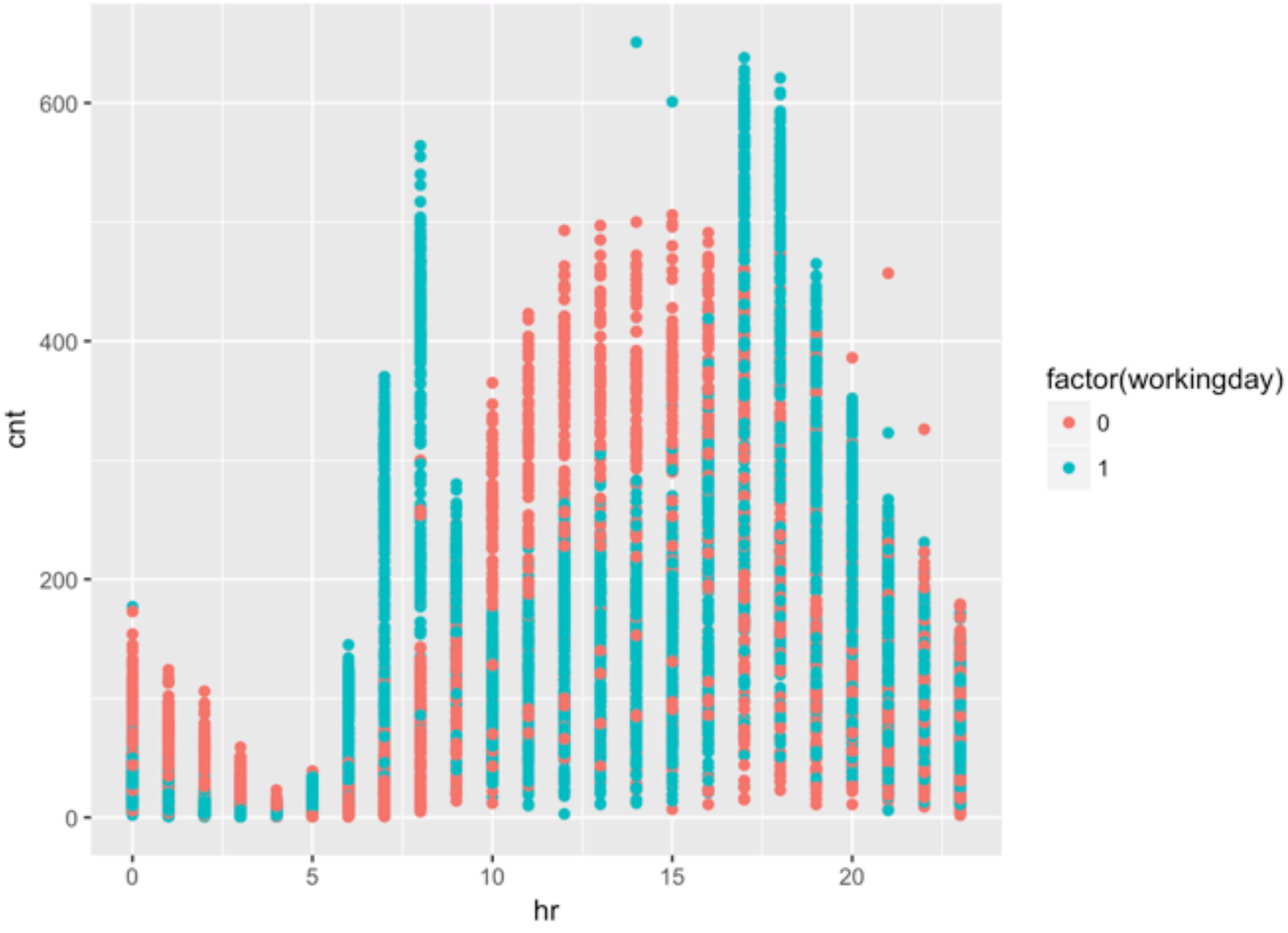
Rapid

Many details

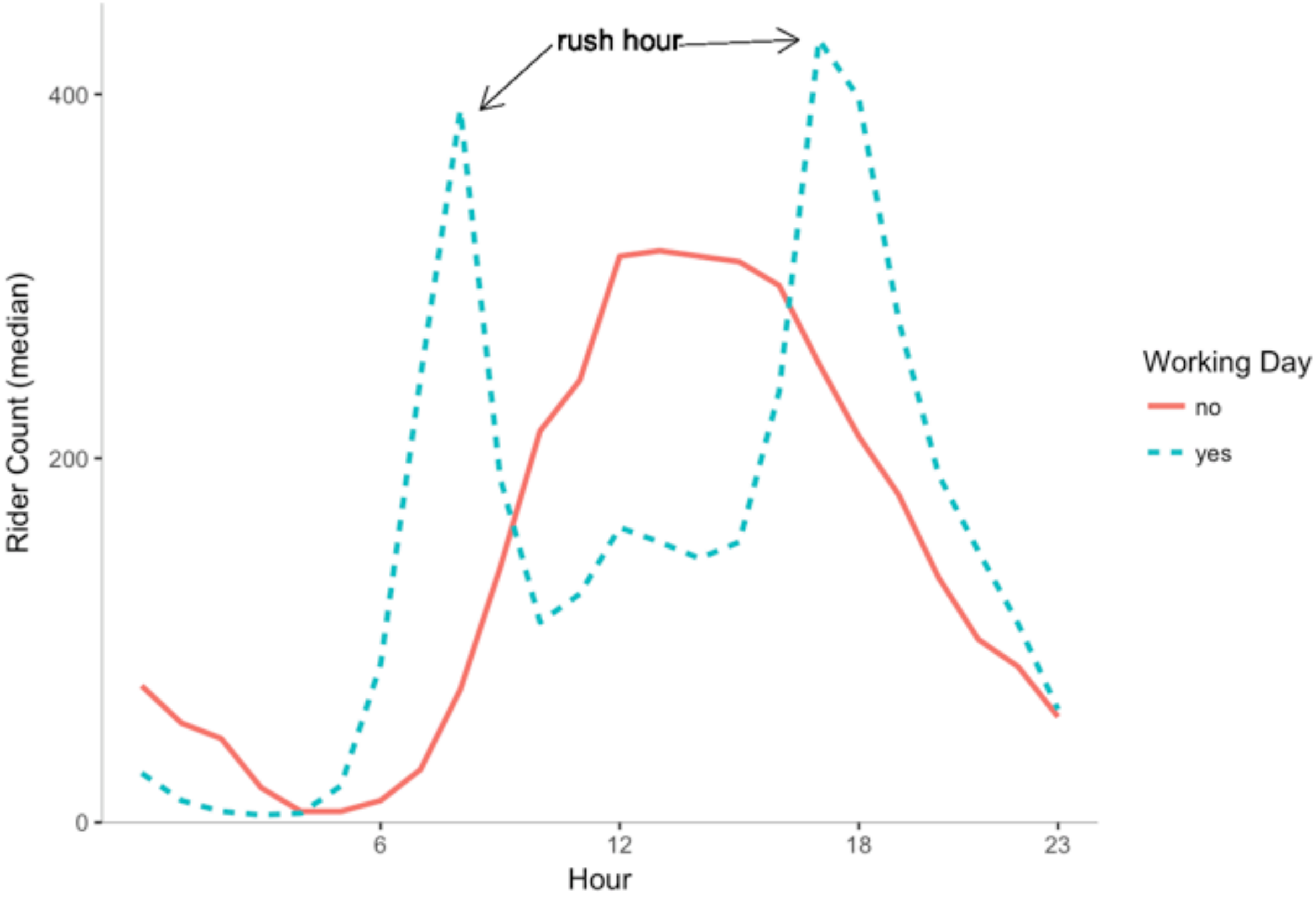
Communicatory plots: message over details, quality over quantity

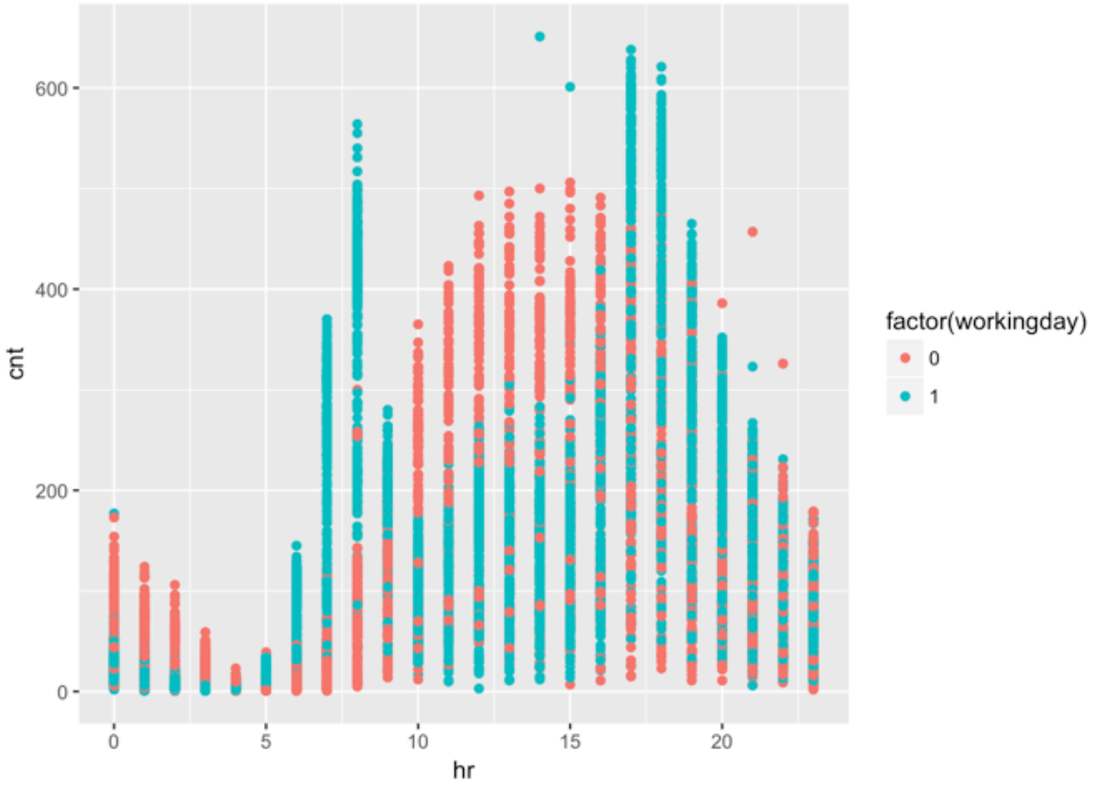
Declutter visualizations for communication

http://www.storytellingwithdata.com/blog/2017/3/29/declutter-this-graph

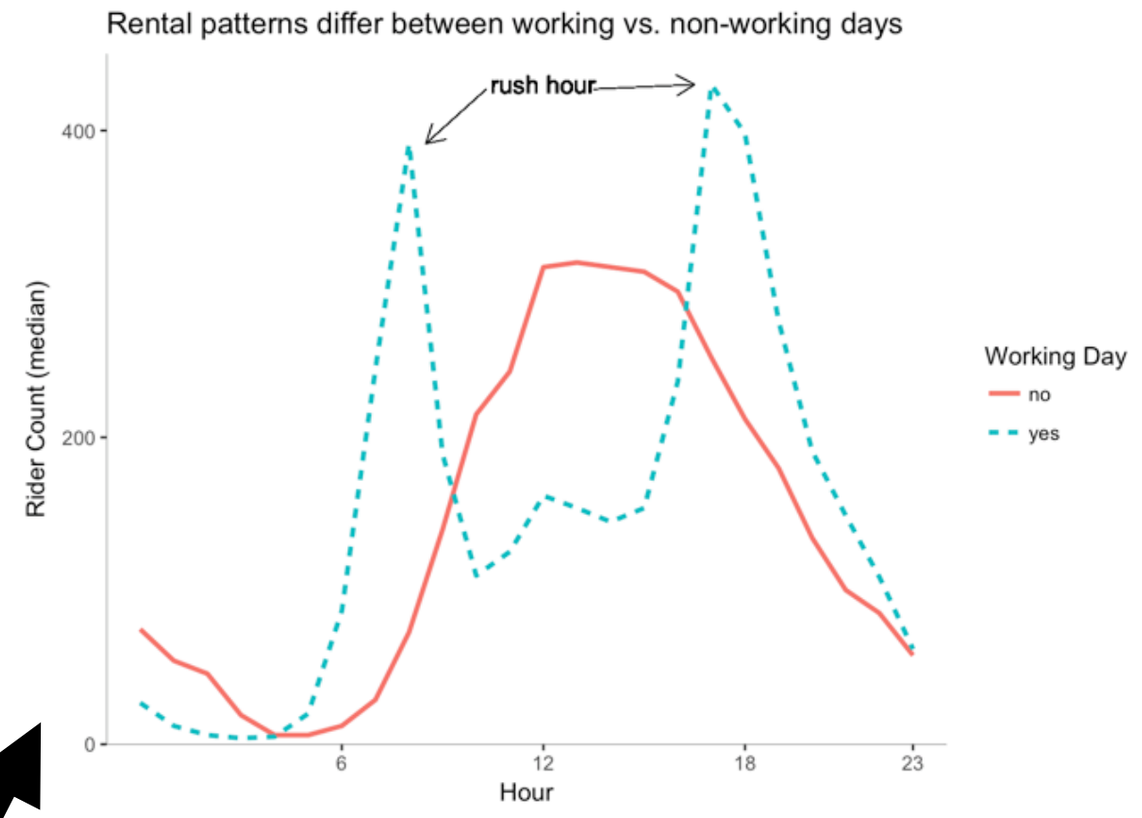

600
400
200
0
cnt
0
5
10
15
20
hr
factor(workingday)
0
1

Rental patterns differ between working vs. non-working days





2 lines of code



30 lines of code

Title states message
Median count vs. all points
Axes
Background grid
Annotation
Multiple codings for working day

Many ways to mislead with visualizations

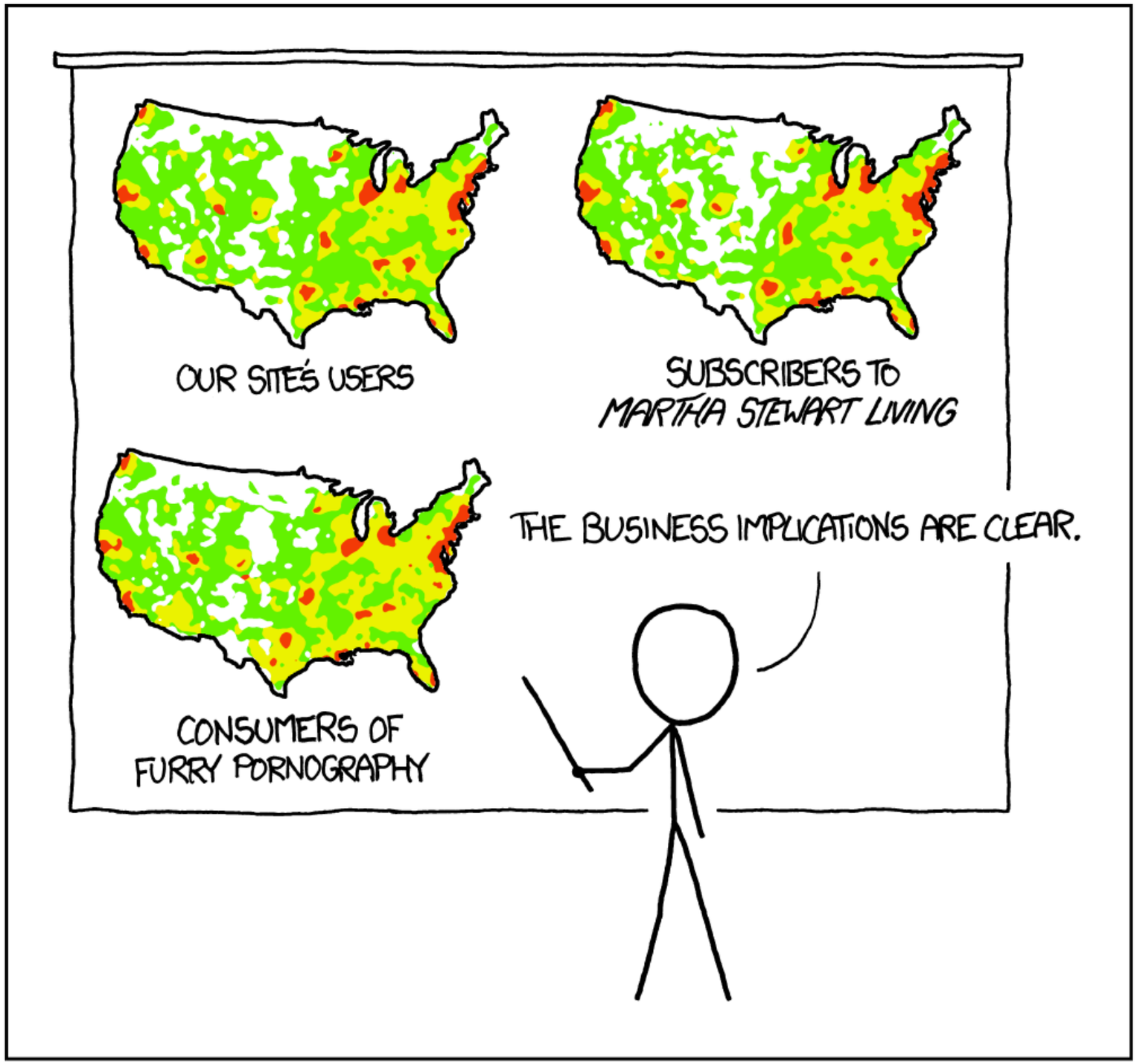
Axis scale

Axis range

Area scales quadratically

Color differences hard to perceive

Be skeptical of choropleths



PET PEEVE #208:
GEOGRAPHIC PROFILE MAPS WHICH ARE BASICALLY JUST POPULATION MAPS

https://xkcd.com/1138/

Dynamic visualizations have a time and a place

Time is an dimension

Interaction

Shiny

Skiing

Hip-hop vocabulary

P-hacking

Most concepts are best illustrated with a simple, static plot

Some cases when dynamic plots are effective

Several related points

Allows the audience to

- Look through the data
- Dig into individual data point

Dashboards

Programming is an act of communication

Two audiences

- Computer
- Future humans

Difficult to understand = bug

Write functions and readable code

Complex function -> many helper functions

Function, variable and file names

str_extract

mean_income

CamelCase or snake_case

Line breaks create hierarchy

Comments

Complex coding project should be organized into folders and sub-folders

https://github.com/juliasilge/tidytext

R Markdown

Text editor

Literate programming

R Markdown's capabilities

http://rmarkdown.rstudio.com/gallery.html

Text editor capabilities

Text formatting

bold, *italics*, ~~strikethrough~~

bold *italics* ~~strikethrough~~

Links

[text](www.diddukewin.com)

Sections and subsections

#, ##

Add block quotes

>

Lists, tables, images

R code

Customize html

Use formatting selectively

Too much emphasis is bad

Draw attention to important links

Consider the github repositories for the tidytext package (see here)

Floating TOC

Sections

RMD facilitates literate programming for data science

Code contains commentary about the code

RMD allows including code in the presentation of the results

Reproducibility

Code is the content of the analysis

How to ask questions effectively

Ask google before a human

Title that summarizes the problem

Spelling, grammar and punctuation

Words before code

Environment

- OS, R version, packages

Reproducible example

sessionInfo()

```
> sessionInfo()
R version 3.3.2 (2016-10-31)
Platform: x86_64-apple-darwin13.4.0 (64-bit)
Running under: macOS Sierra 10.12.3

locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

loaded via a namespace (and not attached):
 [1] backports_1.0.5 magrittr_1.5    rprojroot_1.1   htmltools_0.3.5 tools_3.3.2     yaml_2.1.14     Rcpp_0.12.9     stringi_1.1.2
 [9] rmarkdown_1.3   knitr_1.15.1    stringr_1.1.0   digest_0.6.12   evaluate_0.10
```

Include a **reproducible example**

Use built in R data sets if possible

Make code easy to understand

Environment

Minimal effort to run

Ideally copy/paste

dplyr::select function returning an error

When I load the dplyr and MASS packages in R the select() function from dplyr no longer works. If I run the following code

```
library(tidyverse)
library(MASS)

# attempt to select a column from a data frame
select(mtcars, mpg)
```

I get an error:

```
Error in select(mtcars, mpg) : unused argument (mpg)
```

My environment is listed below

```
> sessionInfo()
R version 3.3.2 (2016-10-31)
Platform: x86_64-apple-darwin13.4.0 (64-bit)
Running under: macOS Sierra 10.12.3

locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] MASS_7.3-45     dplyr_0.5.0     purrr_0.2.2     readr_1.0.0     tidyr_0.6.1     tibble_1.2      ggplot2_2.2.1   tidyverse_1.1.1

loaded via a namespace (and not attached):
 [1] Rcpp_0.12.9       plyr_1.8.4        forcats_0.2.0     tools_3.3.2       digest_0.6.12     jsonlite_1.3      lubridate_1.6.0
 [8] evaluate_0.10     nlme_3.1-130      gtable_0.2.0      lattice_0.20-34   psych_1.6.12      DBI_0.5-1         yaml_2.1.14
[15] parallel_3.3.2    haven_1.0.0       xml2_1.1.1        stringr_1.1.0     httr_1.2.1        knitr_1.15.1      hms_0.3
[22] rprojroot_1.1     grid_3.3.2        R6_2.2.0          readxl_0.1.1      foreign_0.8-67    rmarkdown_1.3     modelr_0.1.0
[29] reshape2_1.4.2    magrittr_1.5      backports_1.0.5   scales_0.4.1      htmltools_0.3.5   rvest_0.3.2       assertthat_0.1
[36] mnormt_1.5-5      colorspace_1.3-2  stringi_1.1.2     lazyeval_0.2.0    munsell_0.4.3     broom_0.4.1
```