Working with Messy Data

Christopher Teixeira

November 14, 2023

The author's affiliation with The MITRE Corporation is provided for identification pusposes only, and is not intended to convey or imply MITRE's concurrence with, or support for, the positions, opinions or viewpoints expressed by the author.

MITRE

A little about me…

Christopher Teixeira
Principal Data Scientist
The MITRE Corporation

Interests

- Data Analytics
- Applied Statistics
- Operations Research

Education

- MS in Operations Research, George Mason University
- BSc in Mathematics, Worcester Polytechnic Institute

Reading in data

R | Python

▼ Code

```r
# URL for the CMS data used in these slides
url <- "https://data.cms.gov/data-api/v1/dataset/8889d81e-2ee7-448f-8713-f071038289b

# Download and convert the JSON file to a data frame
df <- jsonlite::fromJSON(url)

# View the data frame in a friendly format for Quarto
knitr::kable(head(df), format="html")
```

Rndrng_NPI	Rndrng_Prvdr_Last_Org_Name	Rndrng_Prvdr_First_Name
1003000126	Enkeshafi	Ardalan
1003000134	Cibull	Thomas
1003000142	Khalil	Rashid
1003000423	Velotta	Jennifer

MITRE

Rndrng_NPI	Rndrng_Prvdr_Last_Org_Name	Rndrng_Prvdr_First_Name
1003000480	Rothchild	Kevin
1003000530	Semonche	Amanda

Changing data types

R | Python

▼ Code

```r
# Subset the data down to a select few to work with.
# Then convert "Bene_" and "Tot_" variables to numeric.
# The ID and zip code variables get converted to factors.
df.subset <- df |>
    select(Rndrng_NPI,
           Rndrng_Prvdr_Zip5,
           Tot_Benes,
           Tot_Srvcs,
           starts_with("Bene_")) |>
    mutate(across(starts_with(c("Bene_","Tot_")), as.numeric),
           across(starts_with("Rndrng_"), factor))

# View the data frame in a friendly format for Quarto
knitr::kable(head(df.subset), format="html")
```

Rndrng_NPI	Rndrng_Prvdr_Zip5	Tot_Benes	Tot_Srvcs	Bene_Avg_/
1003000126	20817	661	3749	
1003000134	60201	3216	7359	
1003000142	43623	239	1932	
1003000423	44106	69	738	
1003000480	80045	112	162	
1003000530	18951	404	1487	

MITRE

Exploratory data analysis

Exploratory data analysis (EDA) is used by data scientists to analyze and investigate data sets and summarize their main characteristics, often employing data visualization methods. It helps determine how best to manipulate data sources to get the answers you need, making it easier for data scientists to discover patterns, spot anomalies, test a hypothesis, or check assumptions. [1]

Four primary types of EDA:

1. **Univariate non-graphical**: Describe the data and find patterns that exist within a variable.
2. **Univariate graphical**: For a single variable, explore the values visaully using graphs like box plots or histograms.
3. **Multivariate nongraphical**: Describe the relationships between two or more variables in the data.
4. **Multivariate graphical**: Visualize the relationships between two or more variables through graphs like scatter plots or heat maps,

MITRE

1. IBM What is exploratory data analysis?

Applying EDA: Univariate

R | Python

▼ Code

```r
# Use the skimr package to examine the dataset.
# Produces a high level summary (# of rows/columns, column types)
# For each column type, it produces details about each variable.

library(skimr)
skim(df.subset)
```

Data summary

Name	df.subset
Number of rows	5000
Number of columns	36
_______________________	
Column type frequency:	
factor	2
numeric	34
________________________	
Group variables	None

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	to
Rndrng_NPI	0	1	FALSE	5000	1(1, 1(

MITRE

skim_variable	n_missing	complete_rate	ordered	n_unique	to
Rndrng_Prvdr_Zip5	0	1	FALSE	2809	77 55 02 10

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd
Tot_Benes	0	1.00	283.78	511.06
Tot_Srvcs	0	1.00	1718.13	10740.56
Bene_Avg_Age	0	1.00	71.43	5.54
Bene_Age_LT_65_Cnt	2011	0.60	53.89	74.11
Bene_Age_65_74_Cnt	792	0.84	142.20	237.40
Bene_Age_75_84_Cnt	1442	0.71	118.40	192.40
Bene_Age_GT_84_Cnt	2170	0.57	65.21	105.65
Bene_Feml_Cnt	613	0.88	180.66	307.23
Bene_Male_Cnt	613	0.88	138.98	237.98
Bene_Race_Wht_Cnt	1592	0.68	305.49	496.67
Bene_Race_Black_Cnt	3332	0.33	62.96	124.81
Bene_Race_API_Cnt	4094	0.18	23.43	52.69
Bene_Race_Hspnc_Cnt	3665	0.27	48.70	88.57

skim_variable	n_missing	complete_rate	mean	sd
Bene_Race_NatInd_Cnt	3348	0.33	1.98	13.68
Bene_Race_Othr_Cnt	4139	0.17	21.88	26.99
Bene_Dual_Cnt	1321	0.74	80.12	137.01
Bene_Ndual_Cnt	1321	0.74	282.12	482.44
Bene_CC_AF_Pct	1577	0.68	0.17	0.10
Bene_CC_Alzhmr_Pct	1623	0.68	0.21	0.16
Bene_CC_Asthma_Pct	2008	0.60	0.10	0.05
Bene_CC_Cncr_Pct	1623	0.68	0.15	0.11
Bene_CC_CHF_Pct	1208	0.76	0.28	0.17
Bene_CC_CKD_Pct	745	0.85	0.44	0.18
Bene_CC_COPD_Pct	1452	0.71	0.20	0.11
Bene_CC_Dprssn_Pct	751	0.85	0.33	0.15
Bene_CC_Dbts_Pct	753	0.85	0.37	0.13
Bene_CC_Hyplpdma_Pct	398	0.92	0.62	0.12
Bene_CC_Hyprtnsn_Pct	299	0.94	0.68	0.11
Bene_CC_IHD_Pct	774	0.85	0.42	0.16
Bene_CC_Opo_Pct	1810	0.64	0.11	0.06
Bene_CC_RAOA_Pct	528	0.89	0.49	0.13

Applying EDA: Multivariate

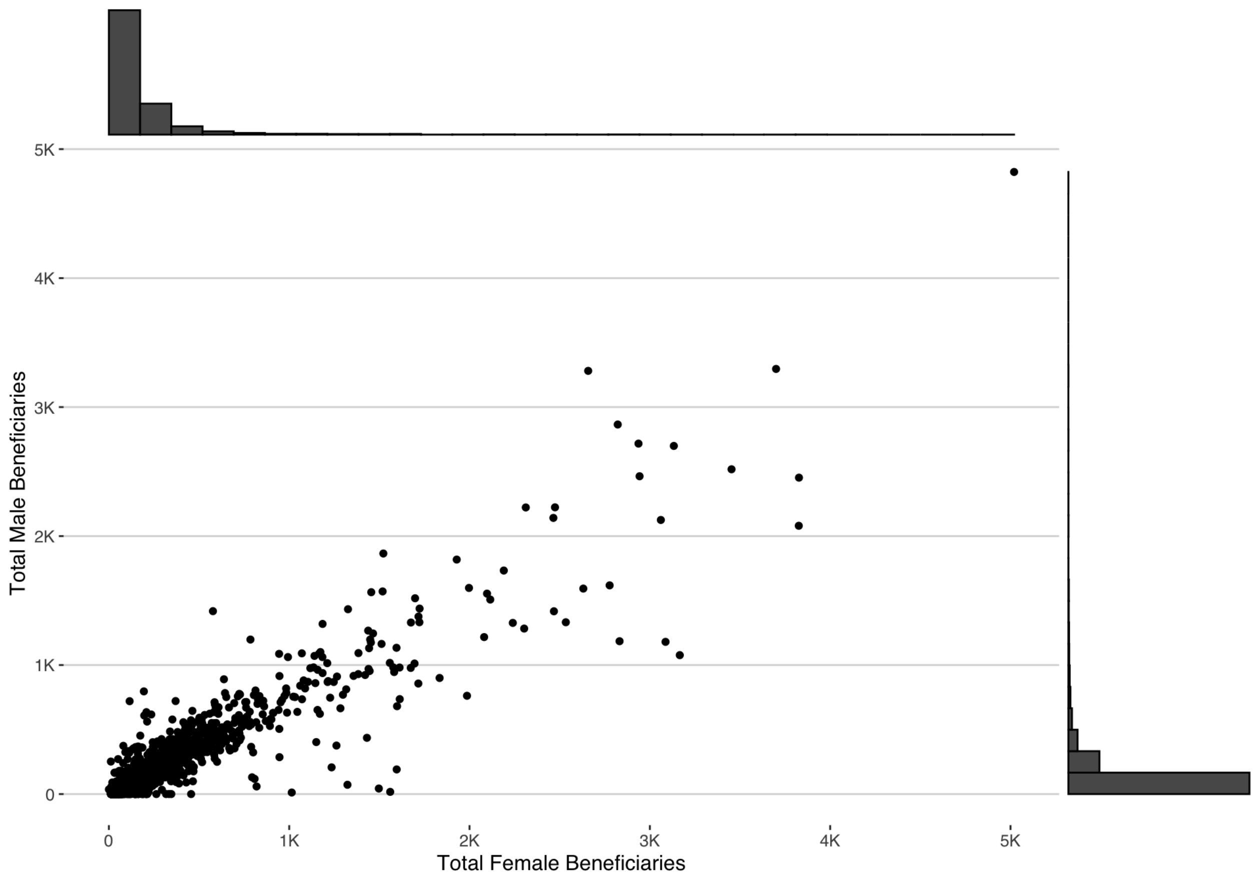
R | Python

▼ Code

```r
library(ggplot2)
library(ggExtra)
library(ggthemes)

# Create a scatter plot with the number of beneficiaries by gender.
g <- ggplot(df.subset,
            aes(x=Bene_Feml_Cnt,
                y=Bene_Male_Cnt)) +
    geom_point() +
    theme(legend.position="none") +
    labs(x="Total Female Beneficiaries",
         y="Total Male Beneficiaries") +
    scale_x_continuous(labels=label_number(scale_cut=cut_short_scale())) +
    scale_y_continuous(labels=label_number(scale_cut=cut_short_scale())) +
    theme_hc()

ggMarginal(g, type="histogram")
```



MITRE

Working with missing data

MITRE

Questions to ask when working with missing data

- Does "missing" mean something different from "0"?
  - If you have data on the amount of candy sold per day, does a missing value mean no candy was sold? or the amount of candy sold is unknown?
- Is "missing" captured in another way?
  - Sometimes negative values or "99" can imply a value is missing.
- Was there a change in how data was being captured?
  - For long standing data capture initiatives (e.g., surveys), the data collection methods can change without notice to the analysts.
    - Was the way data was being capture changed?
    - Did the range of values change?
    - Do the values represent something different?
- Does it make sense to replacing missing values?
  - If a variable is mostly missing, replacing it with any method could lead

MITRE

Imputing missing data

There are two general approaches:

- **Overly simple approach**: replace missing values with mean, median, or mode
- **Sophisticated approach**: replacing missing values by analyzing the full dataset and building a model per variable with missing data

Multivariate Imputation by Chained Equations (MICE)

R Python

▼ Code

```r
library(mice)

# Remove large dimensional variables
df.to.impute <- df.subset |> select(-Rndrng_NPI,-Rndrng_Pr

# Impute missing data using predictive mean matching
imputed <- df.to.impute |>
    mice(m=1, maxit=10, seed=42, method="pmm")

# Show the complete dataset including imputed values
complete(imputed)
```

Cautionary tales in working with data

MITRE

Data drift

Changes in the data can happen over time, resulting in “data drift” that can impact model performance or other decisions that can be overlooked if only near term changes are considered.

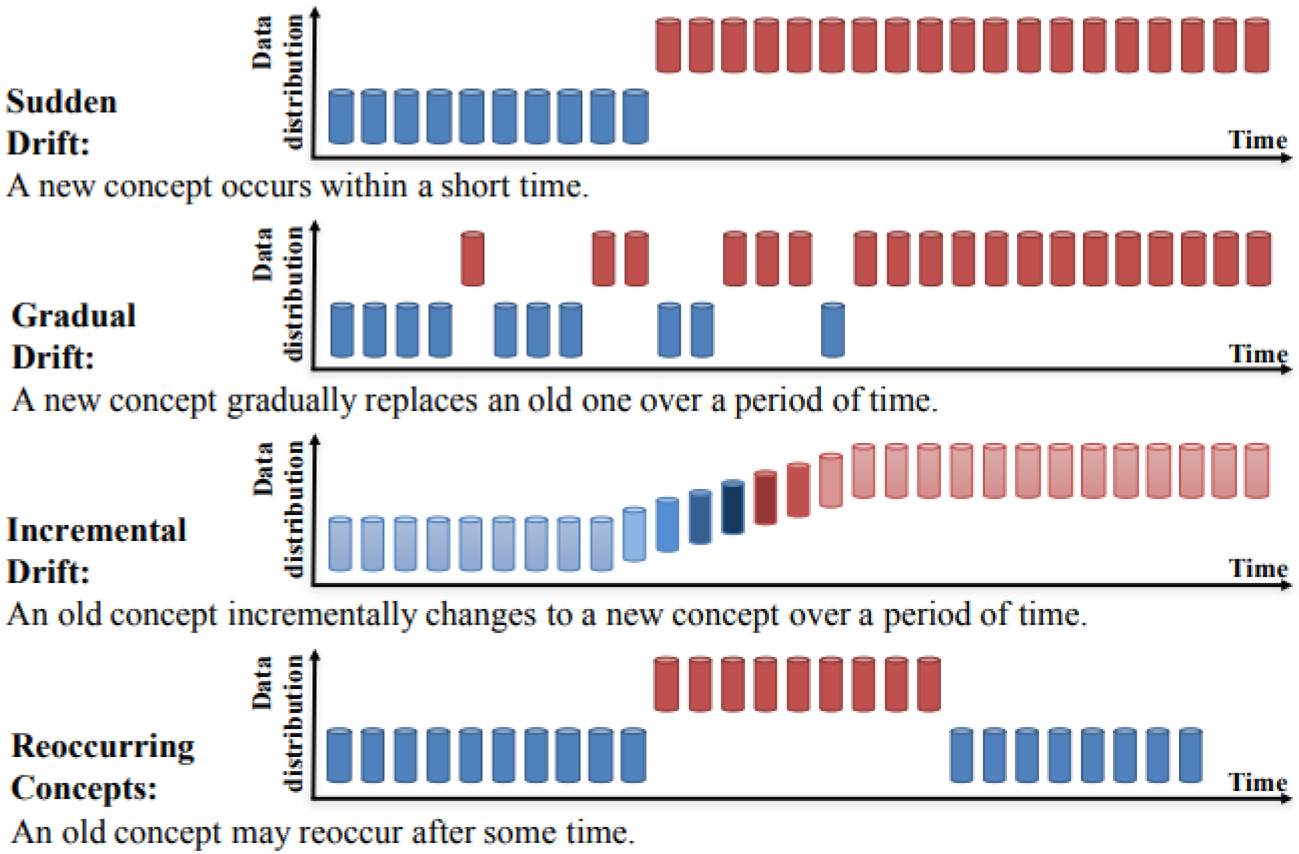


Image taken from DataCamp

Cognitive biases

Cognitive biases are systematic patterns of deviation from norm and/or rationality in judgment. They are often studied in psychology, sociology and behavioral economics.[1]

Some biases to be aware of:

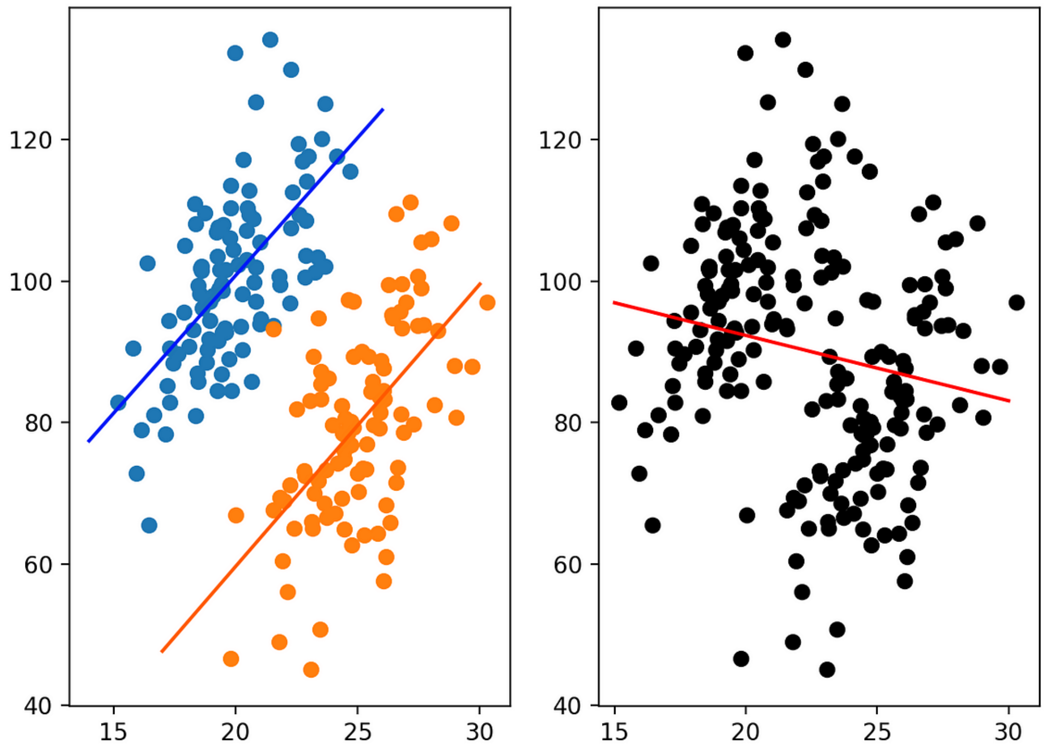
- **Survivorship bias**: Analyzing just the data that is available without analyzing the larger situation.
- **False causality**: Seeing correlation between two variables does not imply one causes the other to occur. [2]
- **Availability bias**: Drawing conclusions on limited data.
- **Confirmation bias**: Manipulating data to confirm your own hypothesis.

1. Haselton MG, Nettle D, Andrews PW (2005). "The evolution of cognitive bias" (PDF). In Buss DM (ed.). The Handbook of Evolutionary Psychology. Hoboken, NJ: John Wiley & Sons Inc. pp. 724–746.
2. Something fun to explore: Spurious correlations

Simpson’s paradox

Simpson’s paradox occurs when groups of data show one particular trend, but this trend is reversed when the groups are combined together. Understanding and identifying this paradox is important for correctly interpreting data. [1]

A baseball player can have higher batting average than another on each of two years, but lower than the other when the two are combined. In one case, David Justice had a higher batting average than Derek Jeter in 1995 and 1996, but across the two years, Jeter’s average was higher. [2]



1. Simpson, Edward H. (1951). “The Interpretation of Interaction in Contingency Tables”. Journal of the Royal Statistical Society, Series B. 13: 238–241.

2. Simpon’s Paradox by Brilliant

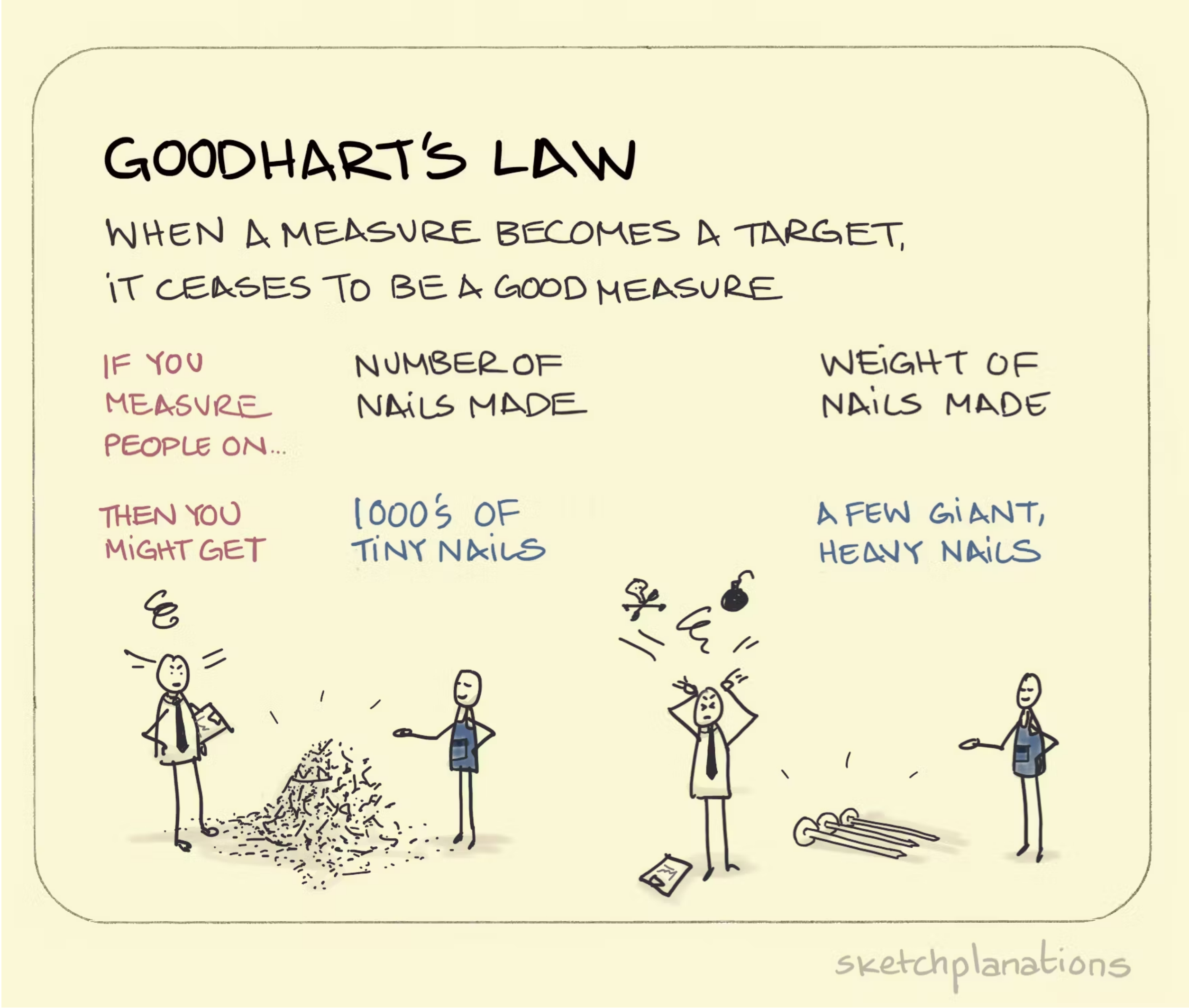
MITRE

Goodhart's law

Any observed statistical regularity will tend to collapse once pressure is placed upon it for control purposes. [1]

or a better way of putting it is:

When a measure becomes a target, it ceases to be a good measure. [2]



1. Goodhart, Charles (1975). "Problems of Monetary Management: The U.K. Experience". In Courakis, Anthony S. (ed.). Inflation, Depression, and Economic Policy in the West. Totowa, New Jersey: Barnes and Noble Books (published 1981). p. 116. ISBN 0-389-20144-8.
2. Strathern, Marilyn (1997). "'Improving ratings': audit in the British University system". European Review. John Wiley & Sons. 5 (3): 305–321

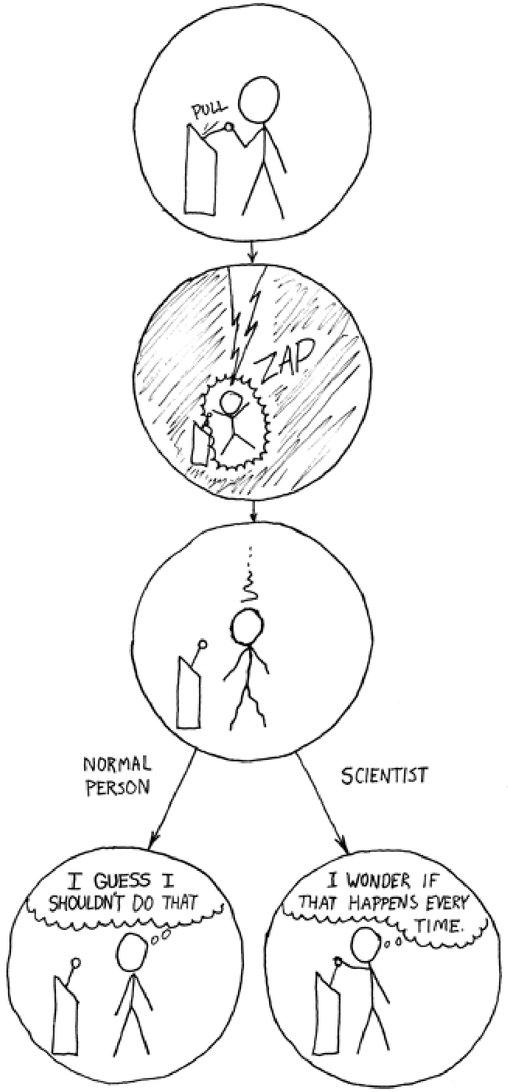
Sharing your data with others

Reproducibility and Replicability

Reproducibility is the ability of independent investigators to draw the same conclusions from an experiment by following the documentation shared by the original investigators. Hence, reproducibility requires that another, independent team of investigators have to conduct the same experiment. [1]

Easy tips for enabling others to reproduce your work:

- Use a static random seed
  - in R: `set.seed(42)`
  - in Python: `random.seed(42)`
- Document your environment
  - in R: `library(renv)`
  - in Python: `pip freeze > requirements.txt`
- Use version control (e.g., Github, Bitbucket)
- Use notebooks
  - in R: R Markdown
  - in Python: Jupyter
  - in either: Quarto



[2]

1. Gundersen Odd Erik. 2021. The fundamental principles of reproducibility. *Phil. Trans. R. Soc. A.* 379: 20200210. http://doi.org/10.1098/rsta.2020.0210

2. xkcd 242

Using pins

The pins package publishes data, models, and other R objects, making it easy to share them across projects and with your colleagues. You can pin objects to a variety of pin boards, including:

- folders (to share on a networked drive or with services like DropBox)
- Posit Connect
- Amazon S3
- Google Cloud Storage
- Azure storage
- Microsoft 365 (OneDrive and SharePoint).

Pins can be automatically versioned, making it straightforward to track changes, re-run analyses on historical data, and undo mistakes.[1] Pins is available in R and Python.

1. https://pins.rstudio.com/

Discussion / Contact Info

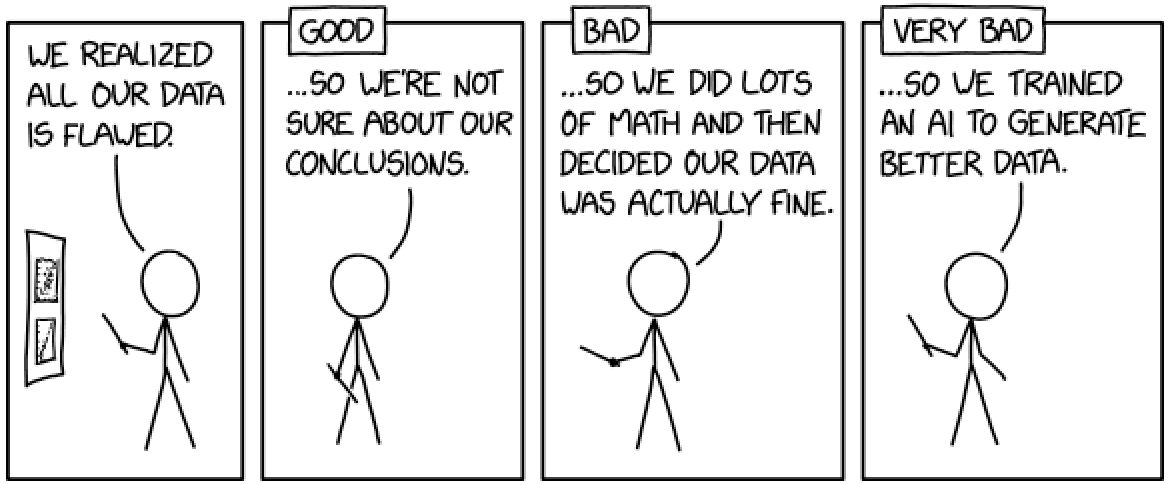
Christopher Teixeira

christopherteixeira.com

chris@christopherteixeira.com

in/christopherteixeira

ct-analytics



xkcd 2494

MITRE