ISA 401/501: Business Intelligence & Data Visualization

08: Tidy Data in R

Fadel M. Megahed, PhD

Enders Associate Professor
Farmer School of Business
Miami University

@FadelMegahed
fmegahed
fmegahed@miamioh.edu
Automated Scheduler for Office Hours

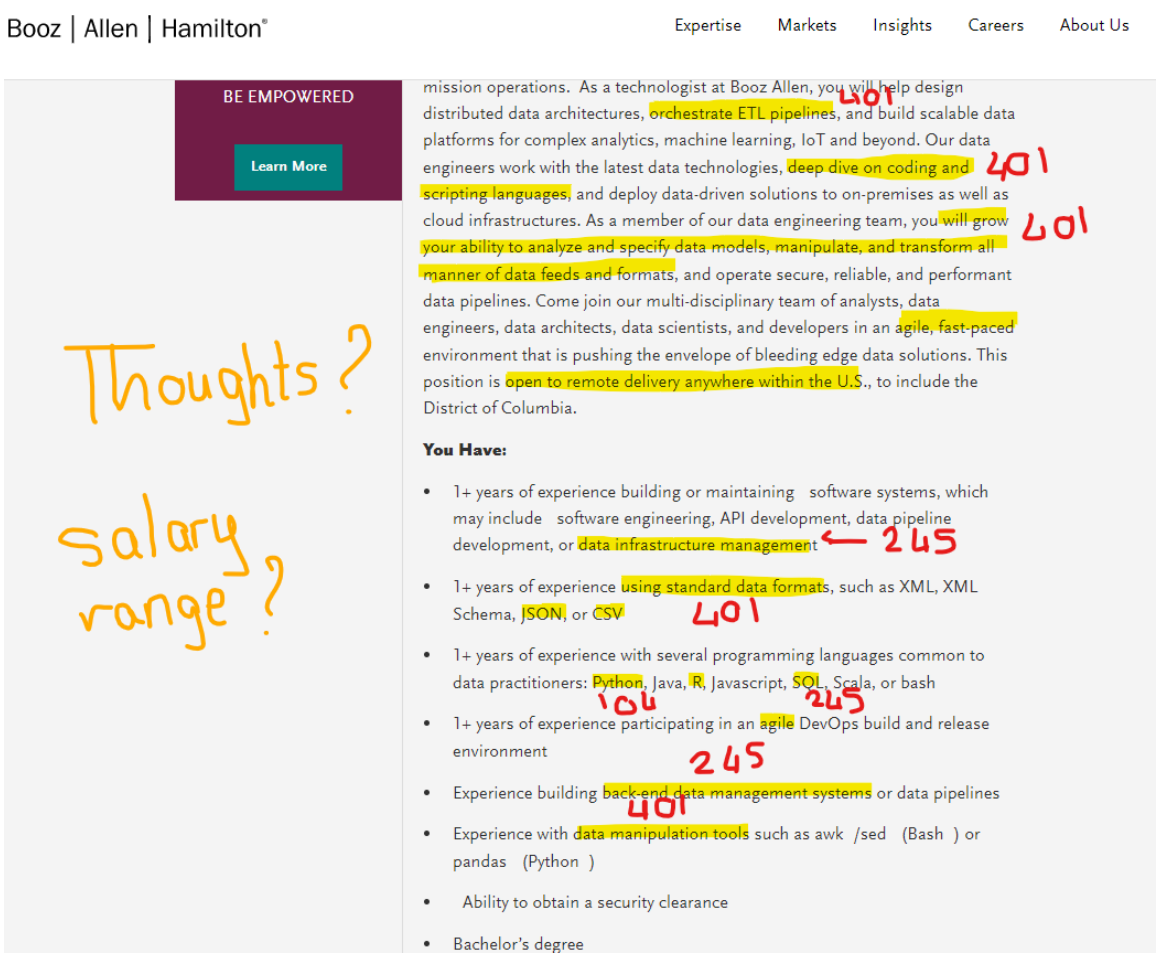
Fall 2022

Quick Refresher from Last Week

✅ Describe what is an API

✅ Download data using APIs

Open Jobs for your Existent Skillset



Booz | Allen | Hamilton®

Expertise Markets Insights Careers About Us

BE EMPOWERED

Learn More

mission operations. As a technologist at Booz Allen, you will help design distributed data architectures, orchestrate ETL pipelines, and build scalable data platforms for complex analytics, machine learning, IoT and beyond. Our data engineers work with the latest data technologies, deep dive on coding and scripting languages, and deploy data-driven solutions to on-premises as well as cloud infrastructures. As a member of our data engineering team, you will grow your ability to analyze and specify data models, manipulate, and transform all manner of data feeds and formats, and operate secure, reliable, and performant data pipelines. Come join our multi-disciplinary team of analysts, data engineers, data architects, data scientists, and developers in an agile, fast-paced environment that is pushing the envelope of bleeding edge data solutions. This position is open to remote delivery anywhere within the U.S., to include the District of Columbia.

Thoughts?

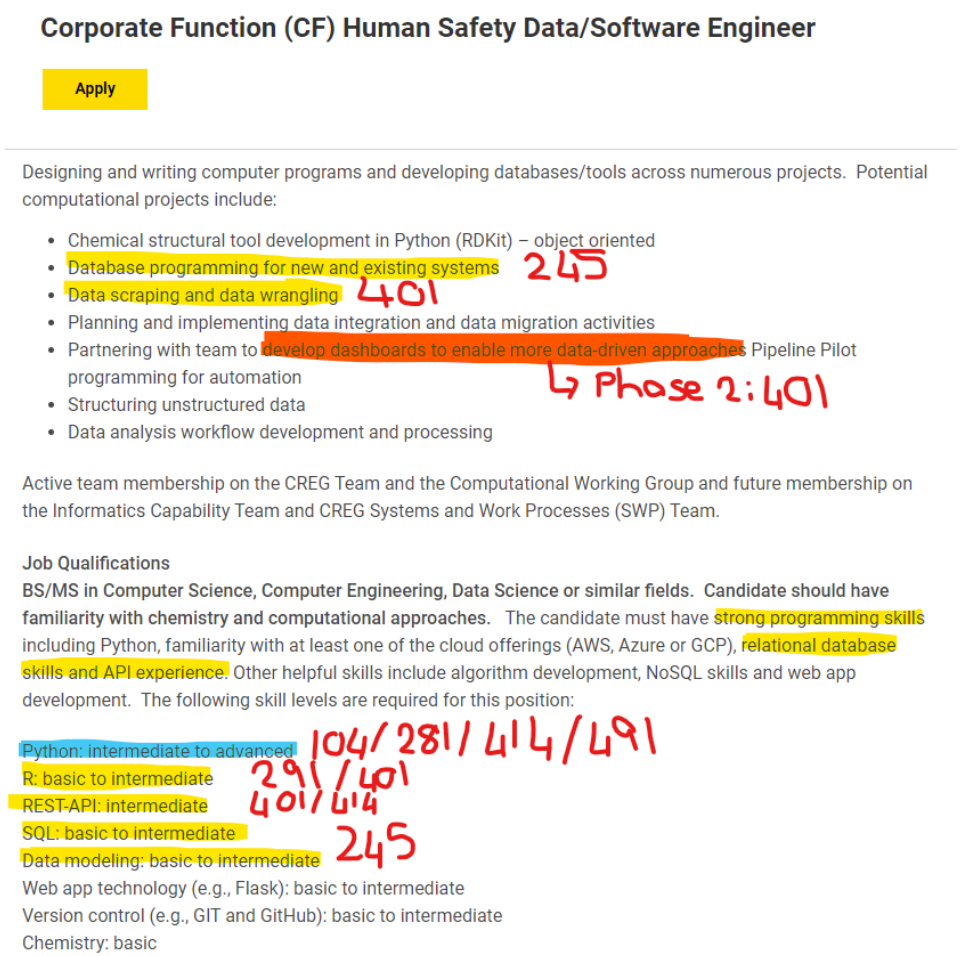
salary range?

You Have:

- 1+ years of experience building or maintaining software systems, which may include software engineering, API development, data pipeline development, or data infrastructure management
- 1+ years of experience using standard data formats, such as XML, XML Schema, JSON, or CSV
- 1+ years of experience with several programming languages common to data practitioners: Python, Java, R, Javascript, SQL, Scala, or bash
- 1+ years of experience participating in an agile DevOps build and release environment
- Experience building back-end data management systems or data pipelines
- Experience with data manipulation tools such as awk /sed (Bash) or pandas (Python)
- Ability to obtain a security clearance
- Bachelor's degree

Source: Data Engineer, Junior, Booz Allen

Open Jobs for your Existent Skillset



Corporate Function (CF) Human Safety Data/Software Engineer

Apply

Designing and writing computer programs and developing databases/tools across numerous projects. Potential computational projects include:

- Chemical structural tool development in Python (RDKit) – object oriented
- Database programming for new and existing systems
- Data scraping and data wrangling
- Planning and implementing data integration and data migration activities
- Partnering with team to develop dashboards to enable more data-driven approaches Pipeline Pilot programming for automation
- Structuring unstructured data
- Data analysis workflow development and processing

Active team membership on the CREG Team and the Computational Working Group and future membership on the Informatics Capability Team and CREG Systems and Work Processes (SWP) Team.

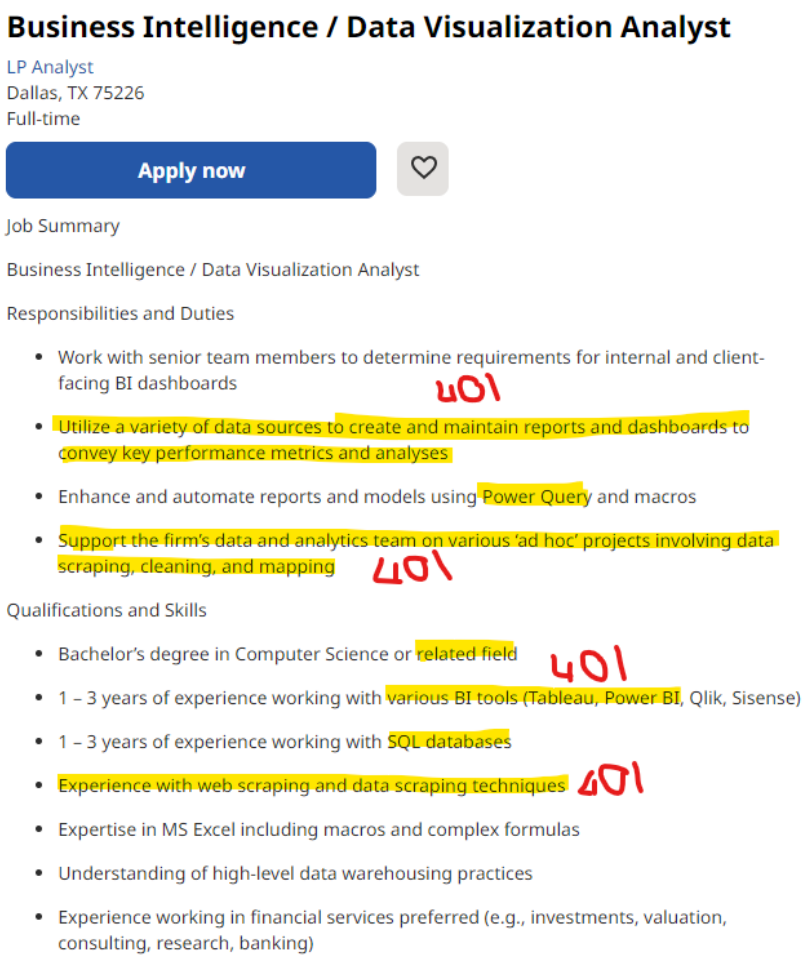
Job Qualifications

BS/MS in Computer Science, Computer Engineering, Data Science or similar fields. Candidate should have familiarity with chemistry and computational approaches. The candidate must have strong programming skills including Python, familiarity with at least one of the cloud offerings (AWS, Azure or GCP), relational database skills and API experience. Other helpful skills include algorithm development, NoSQL skills and web app development. The following skill levels are required for this position:

Python: intermediate to advanced
R: basic to intermediate
REST-API: intermediate
SQL: basic to intermediate
Data modeling: basic to intermediate
Web app technology (e.g., Flask): basic to intermediate
Version control (e.g., GIT and GitHub): basic to intermediate
Chemistry: basic

Source: Corporate Function (CF) Human Safety Data/Software Engineer, P&G

Open Jobs for your Existent Skillset



Business Intelligence / Data Visualization Analyst

LP Analyst
Dallas, TX 75226
Full-time

Apply now

Job Summary

Business Intelligence / Data Visualization Analyst

Responsibilities and Duties

- Work with senior team members to determine requirements for internal and client-facing BI dashboards
- Utilize a variety of data sources to create and maintain reports and dashboards to convey key performance metrics and analyses
- Enhance and automate reports and models using Power Query and macros
- Support the firm's data and analytics team on various 'ad hoc' projects involving data scraping, cleaning, and mapping

Qualifications and Skills

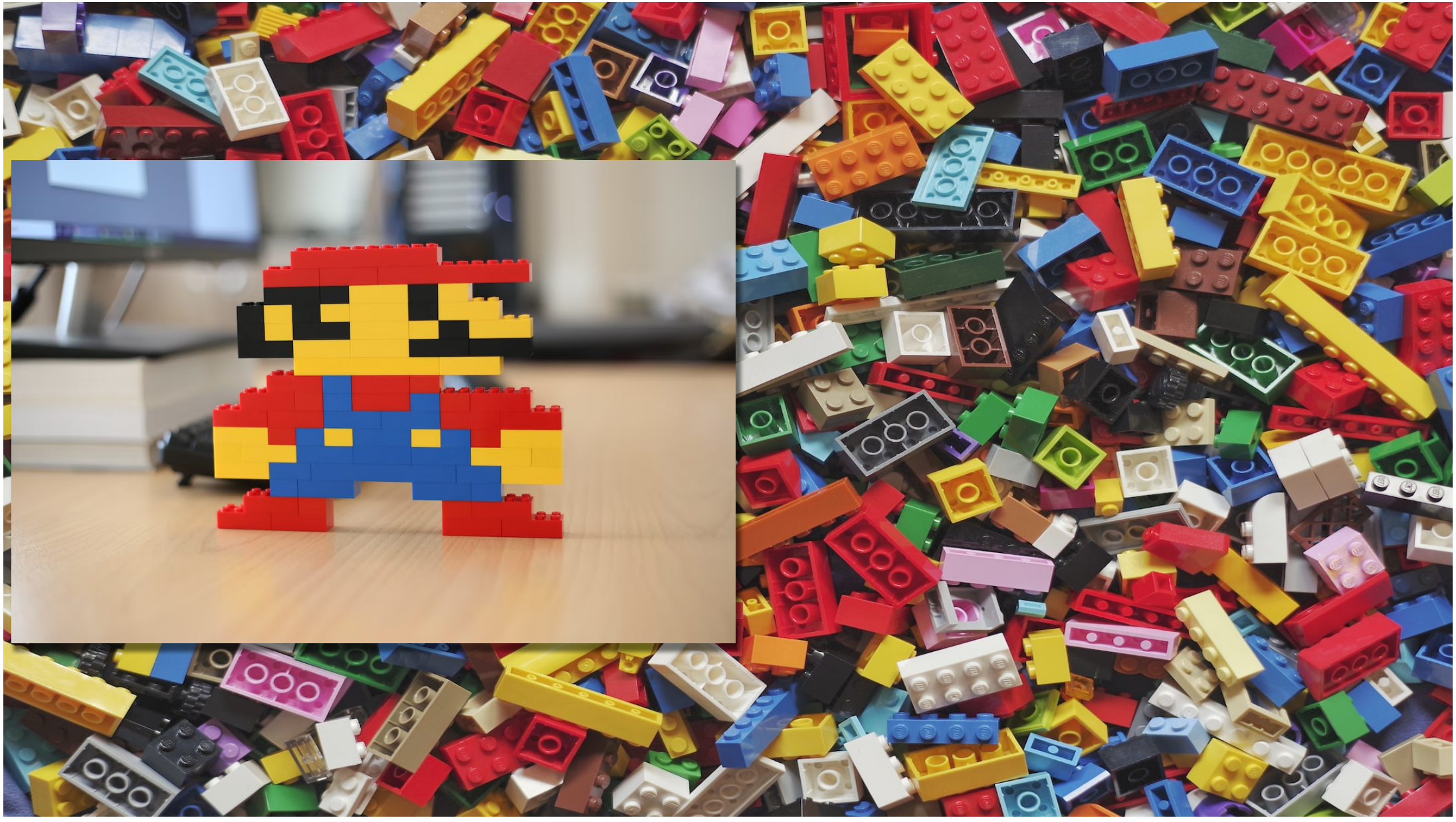
- Bachelor's degree in Computer Science or related field
- 1 – 3 years of experience working with various BI tools (Tableau, Power BI, Qlik, Sisense)
- 1 – 3 years of experience working with SQL databases
- Experience with web scraping and data scraping techniques
- Expertise in MS Excel including macros and complex formulas
- Understanding of high-level data warehousing practices
- Experience working in financial services preferred (e.g., investments, valuation, consulting, research, banking)

Source: Business Intelligence / Data Visualization Analyst, LP Analyst

Learning Objectives for Today's Class

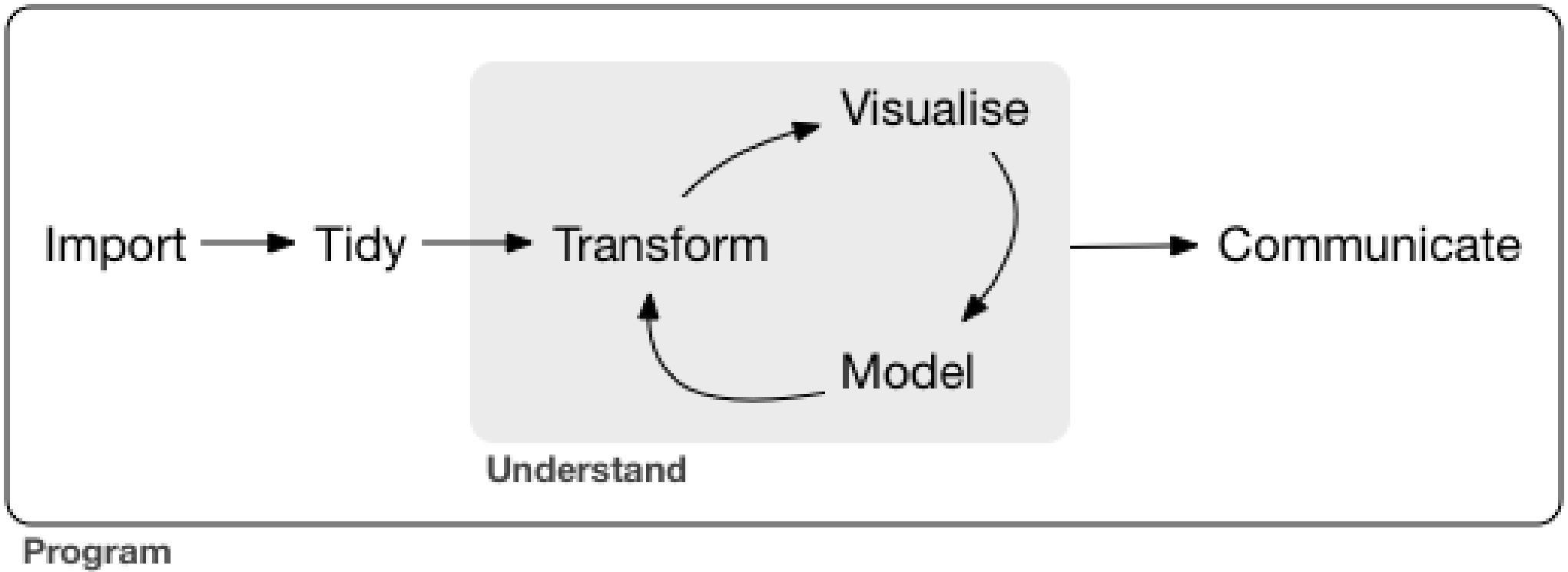
- Define tidy data
- Perform pivot and rectangling operations in 

Tidy Data 



Sources: All three images are obtained from Upsplash

The R for Data Science Workflow



Source: Image is from Wickham, H. Grolemnund, G. (2017). "R for Data Science", O'Reily. https://r4ds.had.co.nz/introduction.html

The Rationale for Tidy Data

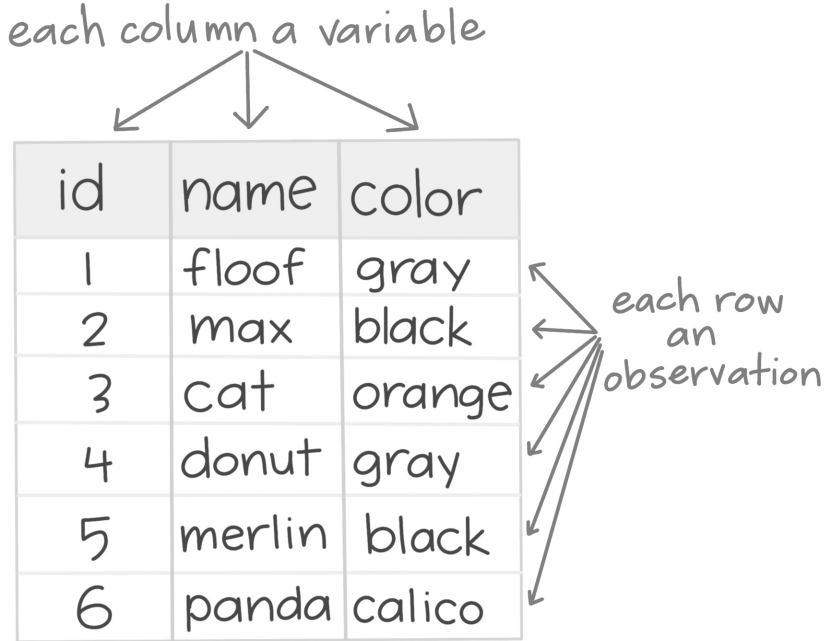
- The **tidy framework** provides a **consistent way to organize your data** in R.
- Getting your data into this format requires some **upfront work, but that work pays off in the long term.**
- Once you have tidy data and the tidy tools provided by packages in the `tidyverse`, you will spend **much less time munging data from one representation to another, allowing you to spend more time on the analytic questions at hand.**

Source: Slide is based on Wickham, H. Grolemnund, G. (2017). "R for Data Science", O'Reily. https://r4ds.had.co.nz/tidy-data.html.

> "TIDY DATA is a standard way of mapping the meaning of a dataset to its structure."
>
> —HADLEY WICKHAM

In tidy data:

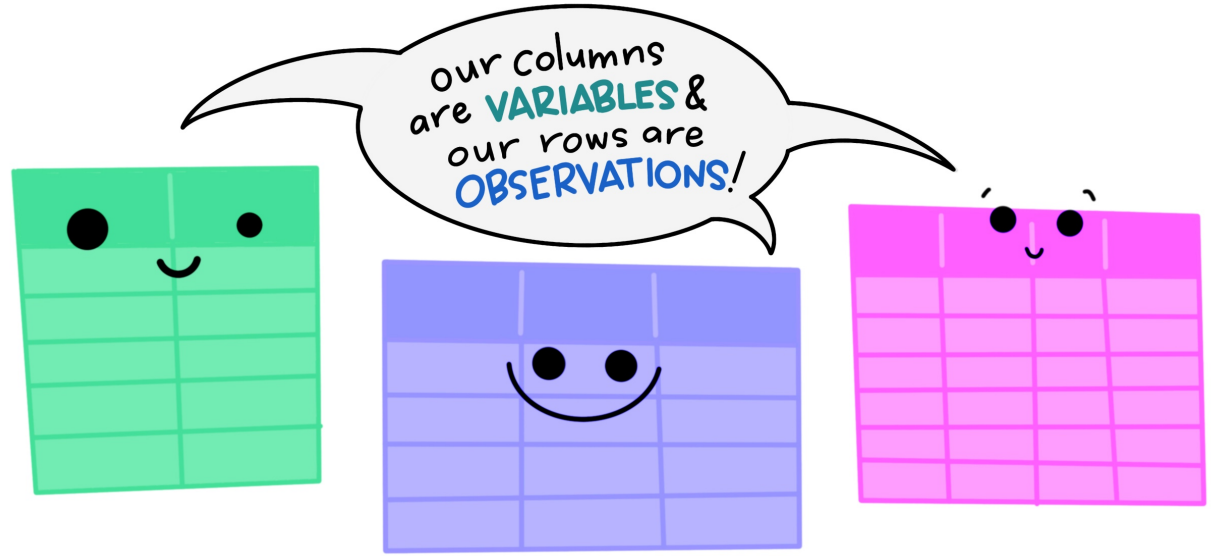
- each variable forms a column
- each observation forms a row
- each cell is a single measurement



Wickham, H. (2014). Tidy Data. Journal of Statistical Software *59* (10). DOI: 10.18637/jss.v059.i10

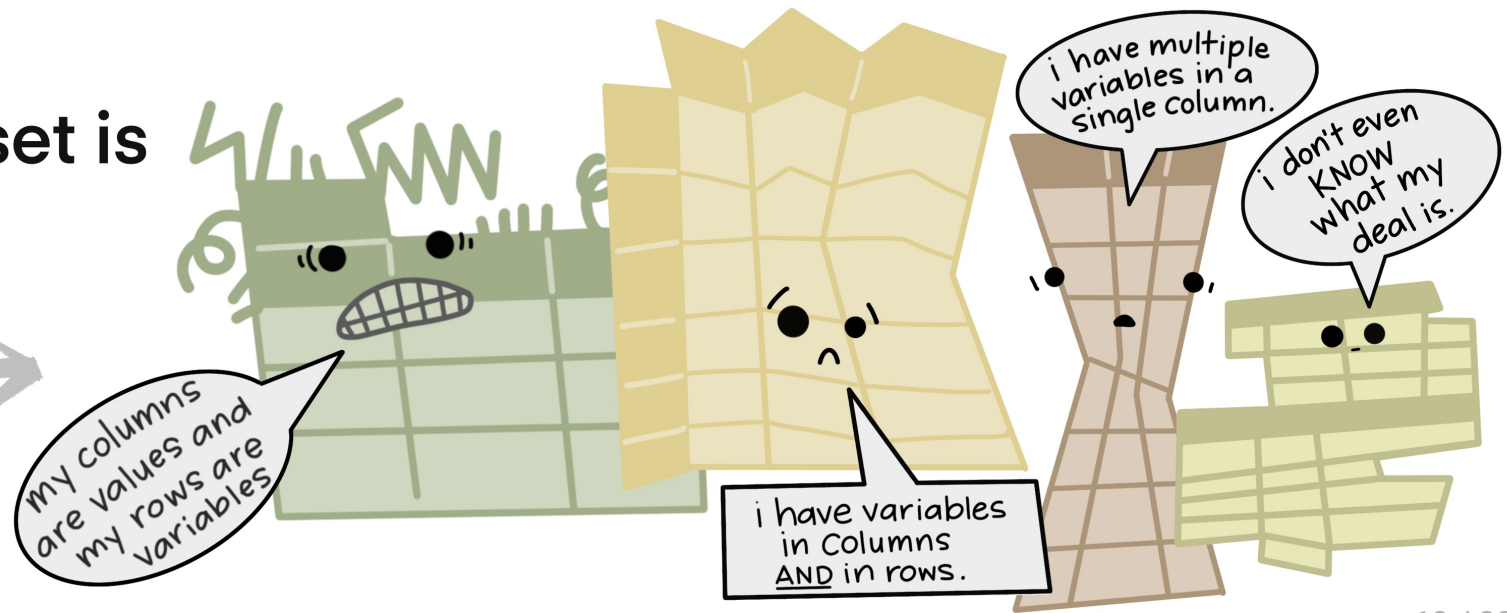
Source: Illustration is from the Openscapes blog Tidy Data for reproducibility, efficiency, and collaboration by Julia Lowndes and Allison Horst

The standard structure of tidy data means that "tidy datasets are all alike..."



"...but every messy dataset is messy in its own way."

–HADLEY WICKHAM



Source: Illustration is from the Openscapes blog Tidy Data for reproducibility, efficiency, and collaboration by Julia Lowndes and Allison Horst

tidy data ≠ clean data

The `movies` data is tidy but not clean.

```
movies <- as_tibble(jsonlite::read_json(
  "https://vega.github.io/vega-editor/app/data/movies.json",
  simplifyVector = TRUE))

movies %>%
  relocate(Release_Date, US_DVD_Sales) %>% # move cols to front
  slice(37:39, 268:269) %>% # filter specific row numbers
  print(width = 80) # print nicely
```

```
## # A tibble: 5 × 16
##   Release…¹ US_DV…² Title US_Gr…³ World…⁴ Produ…⁵ MPAA_…⁶ Runni…⁷ Distr…⁸ Source
##   <chr>       <int> <chr>   <int>   <dbl>   <int> <chr>     <int> <chr>   <chr>
## 1 9-Mar-94       NA Four…  5.27e7  2.43e8  4.5 e6 R            NA Gramer… Origi…
## 2 18-Oct-06      NA 51 B…  8.47e4  8.47e4  3.5 e5 Not Ra…      NA Truly … <NA>
## 3 1963-01-…      NA 55 D…  1   e7  1   e7  1.7 e7 <NA>         NA <NA>    Origi…
## 4 <NA>           NA Drei   0       0       7.2 e6 <NA>         NA <NA>    <NA>
## 5 16-Jan-98      NA The …  1.66e4  1.66e4  2.65e6 Not Ra…      NA Attitu… Origi…
## # … with 6 more variables: Major_Genre <chr>, Creative_Type <chr>,
## #   Director <chr>, Rotten_Tomatoes_Rating <int>, IMDB_Rating <dbl>,
## #   IMDB_Votes <int>, and abbreviated variable names ¹Release_Date,
```

Non-graded Activity: Tidy or Not?

05:00

Activity | table1 | table2 | table3 | table4a | table4b

- In the next five panels, there five tables all displaying the number of TB cases documented by the World Health Organization in Afghanistan, Brazil, and China between 1999 and 2000.
- The data contains values associated with four variables (country, year, cases, and population), but each table organizes the values in a different layout.
- **Based on the information in the previous slide, please document which of the table(s) is(are) tidy and if not, which rules are violated.**
- **Discuss your answer with your neighboring colleague.**

> *Note that you have a total of five minutes for this non-graded activity.*

Non-graded Activity: Tidy or Not?

05:00

Activity | table1 | table2 | table3 | table4a | table4b

```
## table1 from the tidyr package is printed below:
```

```
## # A tibble: 6 × 4
##   country      year  cases population
##   <chr>       <int>  <int>      <int>
## 1 Afghanistan  1999    745   19987071
## 2 Afghanistan  2000   2666   20595360
## 3 Brazil       1999  37737  172006362
## 4 Brazil       2000  80488  174504898
## 5 China        1999 212258 1272915272
## 6 China        2000 213766 1280428583
```

tidy/not-tidy: .....

data observations? data variables? .....

Non-graded Activity: Tidy or Not?

05:00

Activity | table1 | table2 | table3 | table4a | table4b

```
## table2 from the tidyr package is printed below:
```

```
## # A tibble: 12 × 4
##    country      year type            count
##    <chr>       <int> <chr>           <int>
##  1 Afghanistan  1999 cases             745
##  2 Afghanistan  1999 population   19987071
##  3 Afghanistan  2000 cases            2666
##  4 Afghanistan  2000 population   20595360
##  5 Brazil       1999 cases           37737
##  6 Brazil       1999 population  172006362
##  7 Brazil       2000 cases           80488
##  8 Brazil       2000 population  174504898
##  9 China        1999 cases          212258
## 10 China        1999 population 1272915272
## 11 China        2000 cases          213766
## 12 China        2000 population 1280428583
```

Non-graded Activity: Tidy or Not?

05:00

Activity | table1 | table2 | table3 | table4a | table4b

```
## table3 from the tidyr package is printed below:
```

```
## # A tibble: 6 × 3
##   country      year rate
## * <chr>       <int> <chr>
## 1 Afghanistan  1999 745/19987071
## 2 Afghanistan  2000 2666/20595360
## 3 Brazil       1999 37737/172006362
## 4 Brazil       2000 80488/174504898
## 5 China        1999 212258/1272915272
## 6 China        2000 213766/1280428583
```

tidy/not-tidy: .....

data observations? data variables? .....

Non-graded Activity: Tidy or Not?

05:00

Activity | table1 | table2 | table3 | table4a | table4b

```
## table4a from the tidyr package is printed below:
```

```
## # A tibble: 3 × 3
##   country     `1999` `2000`
## * <chr>        <int>  <int>
## 1 Afghanistan    745   2666
## 2 Brazil       37737  80488
## 3 China       212258 213766
```

tidy/not-tidy: .....

data observations? data variables? .....

rules broken (if any): .....

Non-graded Activity: Tidy or Not?

05:00

Activity | table1 | table2 | table3 | table4a | table4b

```
## table4b from the tidyr package is printed below:
```

```
## # A tibble: 3 × 3
##   country         `1999`     `2000`
## * <chr>            <int>      <int>
## 1 Afghanistan   19987071   20595360
## 2 Brazil       172006362  174504898
## 3 China       1272915272 1280428583
```

tidy/not-tidy: .....

data observations? data variables? .....

rules broken (if any): .....

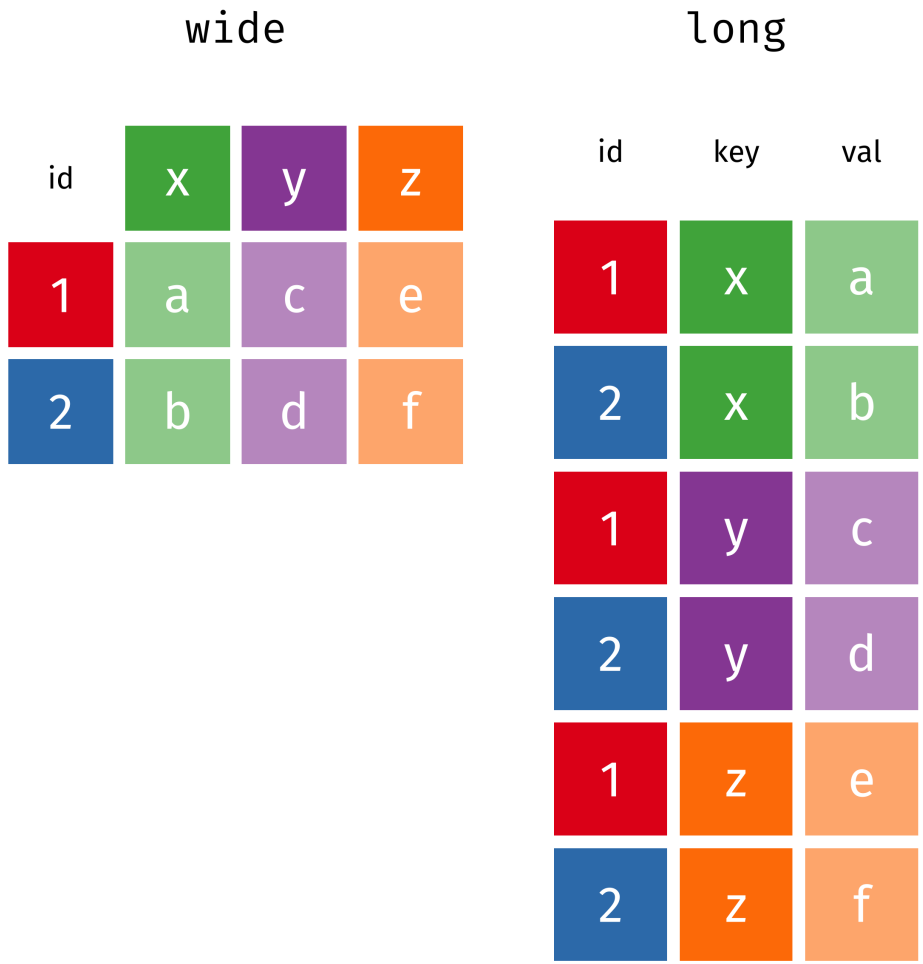
Getting Data into Tidy Format

Key Functions from the `tidyr`

type	`function()`	`function()`
pivoting	pivot_longer()	pivot_wider()
splitting/combining	seperate()	unite()
nesting/unnesting	nest()	unnest()
missing	complete()	fill()

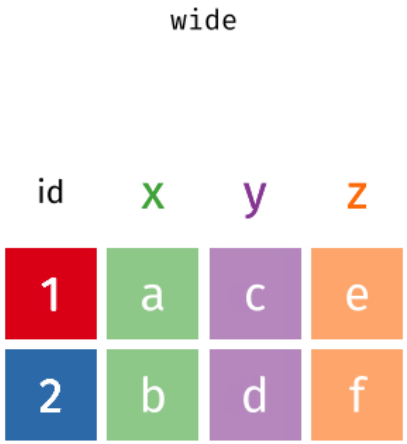
Source: Slide is based on Earo Wang's STAT 220 Slides

Wide Vs Long Data



Source: Image is from Garrick Aden-Buie's excellent tidyexplain GitHub Repository

`pviot_()` to Transform Wide from/to Long



Source: Image is from Garrick Aden-Buie's excellent tidyexplain GitHub Repository

The `pivot_longer()` Function

```
pivot_longer(wide,
       cols = x:z,
  names_to = "key",
 values_to = "val")
```

returns

Source: Image is from Garrick Aden-Buie's excellent tidyexplain GitHub Repository

`pivot_longer()` for table4a [1]

To tidy a dataset like this, we need to pivot the **offending columns into a new pair of variables**. To describe that operation we need **three parameters:**

- The set of columns whose names are values, not variables. In this example, those are the columns `1999` and `2000`.
- The name of the variable to move the column names to. Here it is `year`.
- The name of the variable to move the column values to. Here it's `cases`.

`pivot_longer()` for table4a [2]

country	year	cases
Afghanistan	1999	745
Afghanistan	2000	2666
Brazil	1999	37737
Brazil	2000	80488
China	1999	212258
China	2000	213766

country	1999	2000
Afghanistan	745	2666
Brazil	37737	80488
China	212258	213766

table4

Pivoting table4a into a longer, tidy form

`pivot_longer()` for table4a [3]

```
pivot_longer(table4a, c(`1999`, `2000`), names_to = "year", values_to = "cases")
```

```
## # A tibble: 6 × 3
##   country     year   cases
##   <chr>       <chr>  <int>
## 1 Afghanistan 1999     745
## 2 Afghanistan 2000    2666
## 3 Brazil      1999   37737
## 4 Brazil      2000   80488
## 5 China       1999  212258
## 6 China       2000  213766
```

The `pivot_wider()` Function

Source: Image is from Garrick Aden-Buie's excellent tidyexplain GitHub Repository

`pivot_wider()` for table2 [1]

- `pivot_wider()` is the opposite of `pivot_longer()`.
- You use it when an observation is scattered across multiple rows.
- For example, take table2: an observation is a country in a year, but each observation is spread across two rows.

```
head(table2, n = 3)
```

```
## # A tibble: 3 × 4
##   country      year type           count
##   <chr>       <int> <chr>          <int>
## 1 Afghanistan  1999 cases            745
## 2 Afghanistan  1999 population 19987071
## 3 Afghanistan  2000 cases           2666
```

`pivot_wider()` for table2 [2]

country	year	key	value
Afghanistan	1999	cases	745
Afghanistan	1999	population	19987071
Afghanistan	2000	cases	2666
Afghanistan	2000	population	20595360
Brazil	1999	cases	37737
Brazil	1999	population	172006362
Brazil	2000	cases	80488
Brazil	2000	population	174504898
China	1999	cases	212258
China	1999	population	1272915272
China	2000	cases	213766
China	2000	population	1280428583

table2

pivot_wider() for table2 [3]

```
pivot_wider(table2, names_from = type, values_from = count)
```

```
## # A tibble: 6 × 4
##   country      year  cases population
##   <chr>       <int>  <int>      <int>
## 1 Afghanistan  1999    745   19987071
## 2 Afghanistan  2000   2666   20595360
## 3 Brazil       1999  37737  172006362
## 4 Brazil       2000  80488  174504898
## 5 China        1999 212258 1272915272
## 6 China        2000 213766 1280428583
```

`seperate()` for table3 [1]

```
## # A tibble: 6 × 3
##   country      year rate
## * <chr>       <int> <chr>
## 1 Afghanistan  1999 745/19987071
## 2 Afghanistan  2000 2666/20595360
## 3 Brazil       1999 37737/172006362
## 4 Brazil       2000 80488/174504898
## 5 China        1999 212258/1272915272
## 6 China        2000 213766/1280428583
```

`table3` has a different problem:

- we have one column (`rate`) that contains two variables (`cases` and `population`).
- To fix this problem, we'll need the `separate()` function.

`seperate()` for table3 [2]

country	year	rate
Afghanistan	1999	**745** / 19987071
Afghanistan	2000	**2666** / 20595360
Brazil	1999	**37737** / 172006362
Brazil	2000	**80488** / 174504898
China	1999	**212258** / 1272915272
China	2000	**213766** / 1280428583

table3

country	year	cases	population
Afghanistan	1999	**745**	19987071
Afghanistan	2000	**2666**	20595360
Brazil	1999	**37737**	172006362
Brazil	2000	**80488**	174504898
China	1999	**212258**	1272915272
China	2000	**213766**	1280428583

Source: Slide is based on Wickham, H. Grolemnund, G. (2017). "R for Data Science", O'Reily. https://r4ds.had.co.nz/tidy-data.html.

`seperate()` for table3 [3]

```
separate(table3, rate, into = c("cases", "population"), convert = TRUE)
```

```
## # A tibble: 6 × 4
##   country      year  cases population
##   <chr>       <int>  <int>      <int>
## 1 Afghanistan  1999    745   19987071
## 2 Afghanistan  2000   2666   20595360
## 3 Brazil       1999  37737  172006362
## 4 Brazil       2000  80488  174504898
## 5 China        1999 212258 1272915272
## 6 China        2000 213766 1280428583
```

Non-graded Class Activity

05:00

Activity | Your Solution

> *In this five minute non-graded activity, please do the following*

- Go to https://usafacts.org/visualizations/coronavirus-covid-19-spread-map/
- Download the data for `Deaths` by clicking on the tab to the right of the page.
- **Tidy this data based on the information you have learned in today's class.**

Non-graded Class Activity

05:00

Activity | Your Solution

In your RStudio Session, please read the data, load the required packages and write the code needed to transform the `deaths` data into a tidy format.

Recap

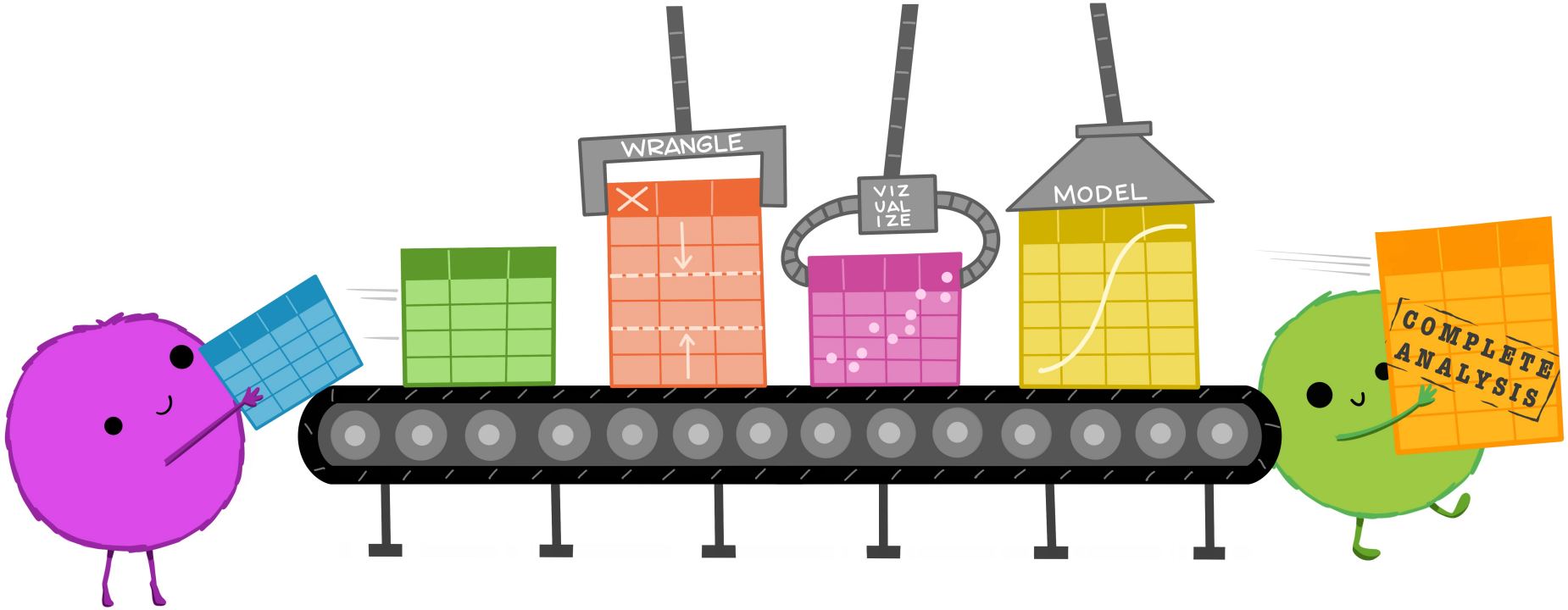
Summary of Main Points

By now, you should be able to do the following:

- Define tidy data
- Perform pivot and rectangling operations in 

Advantages of Tidy Data

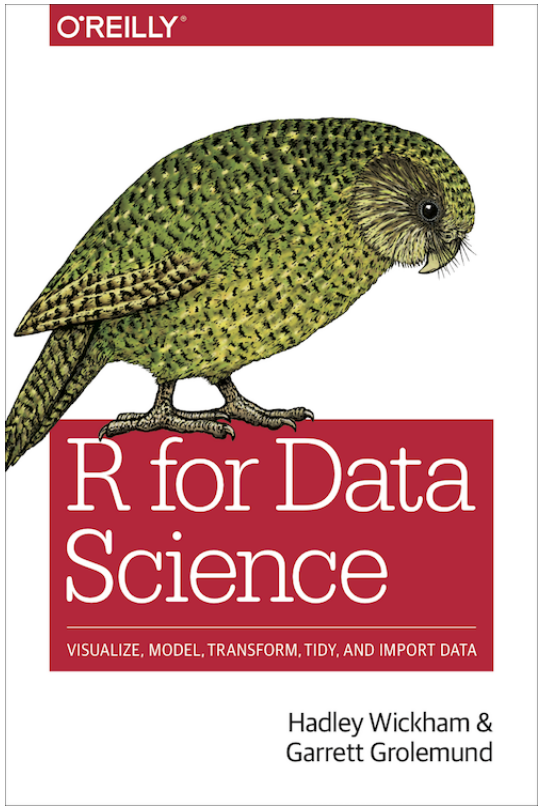
- one set of consistent tools for different datasets
- easier for automation and iteration



Source: Illustration is from the Openscapes blog Tidy Data for reproducibility, efficiency, and collaboration by Julia Lowndes and Allison Horst

Things to Do Prior to Next Class

Please go through the following two supplementary readings and complete assignment 08: tidy data.



- Tidy data
- {tidyr} cheatsheet