SEOUL NATIONAL UNIVERSITY
Graduate School of Data Science

**M3239.003100: Data Analysis and Visualization**
Lecture 3

# Data Cleaning / Storing / Merging

Hyunwoo Park
Graduate School of Data Science
Seoul National University

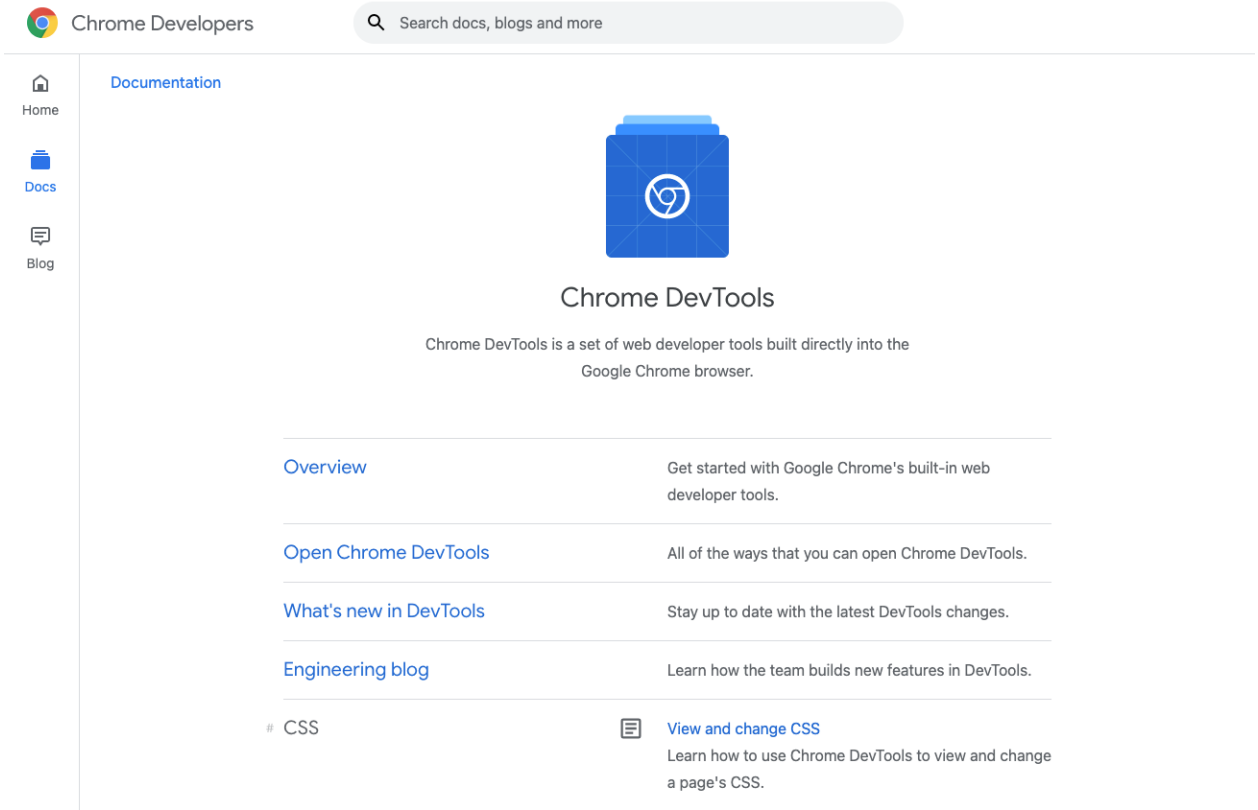
# Agenda

- Demo
  - Chrome Developer Tools
  - Naver Webtoon
  - Starbucks Locations in Korea
  - Code/Text Editor
- Data Cleaning
  - Types of data dirtiness
  - Importance of data cleaning
  - Data Wrangler / Open Refine
  - Regular Expression
  - OCR
- Data Storing
  - CSV (or other text file-based storage)
  - Sqlite
  - RDBS / NoSQL
- Data Merging
  - SQL
  - Python

# Demo

# Chrome Developer Tools

- https://developer.chrome.com/docs/devtools/

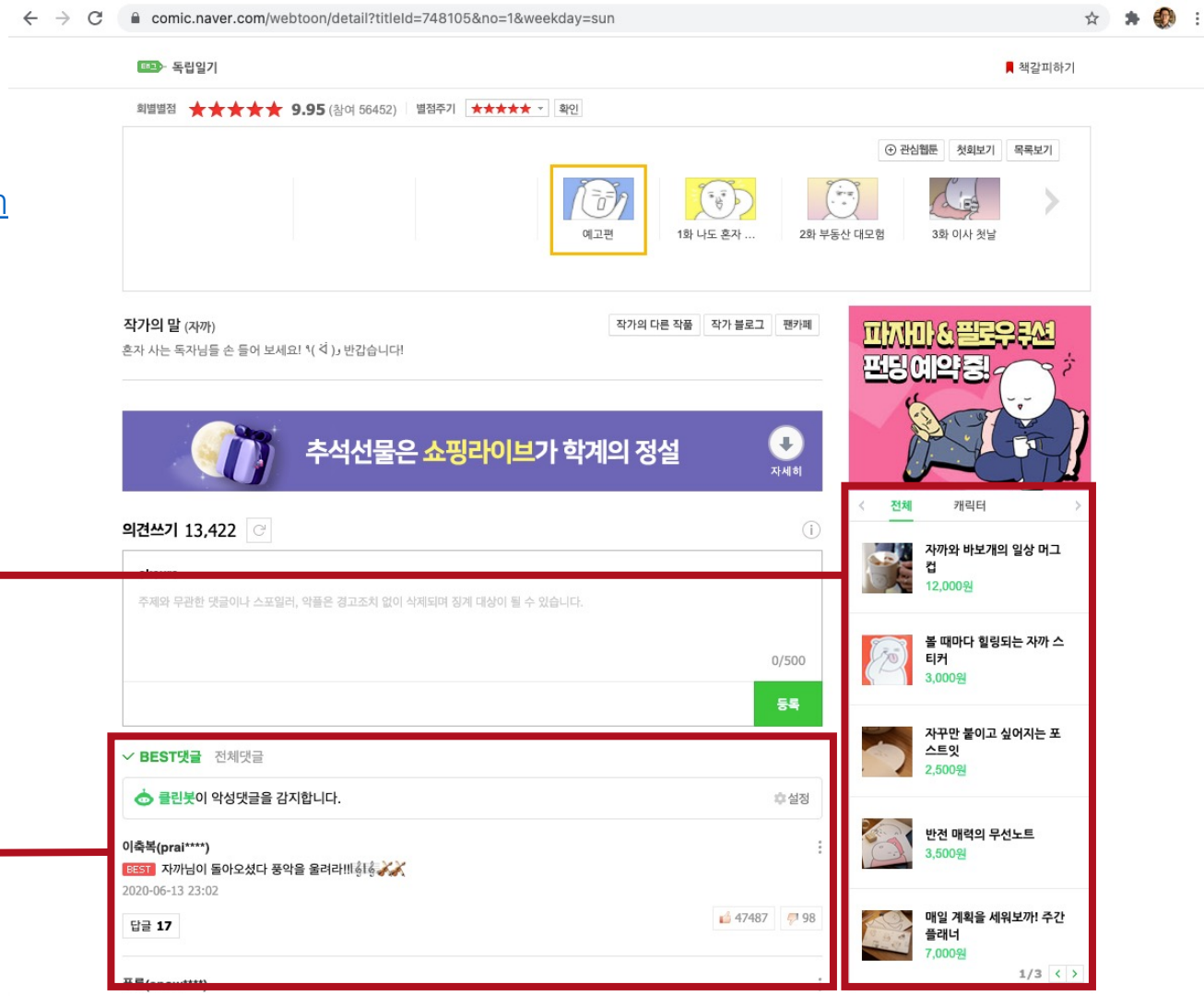


# Example: yestrade

- https://yestrade.go.kr/

# Example 1: Naver Webtoon

- https://comic.naver.com/webtoon/detail?titleId=748105&no=1&weekday=thu
- Related shop items are fetched as JSON asynchronously.
- Comments are in an IFRAME.



# Example 2: Starbucks Locations in Korea

- https://www.starbucks.co.kr/store/store_map.do

# Code/Text Editor (Sublime Text)

- Multiple Cursors
  - 1) Select next instance of the selected text
    - https://www.sublimetext.com/docs/multiple_selection_with_the_keyboard.html (See “Quick Add Next”)
  - 2) Select all instances of the selected text
    - https://stackoverflow.com/questions/12162047/how-to-select-all-instances-of-selected-region-in-sublime-text
  - 3) Column editing

On Mac OS you can use: CMD + CTRL + G

On Windows/Linux press Alt + F3.

- Move lines
- Duplicate lines

# Data Cleaning

# How dirty is real data?

# An example of dirty data

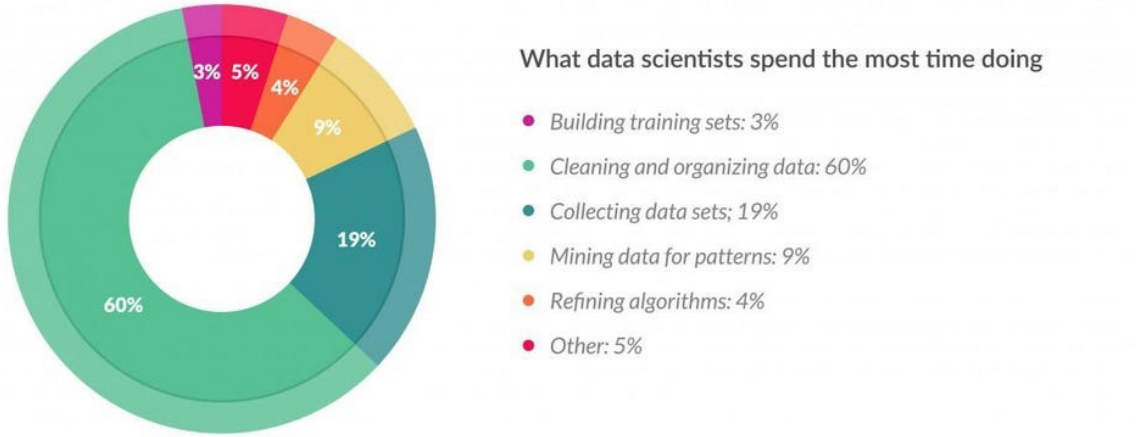
- Sep 9, 2021
- September 9, 21
- 9/9/21
- 2021-09-09
- 21/9/9
- 9th September 2021

# Types of data dirtiness

- Typos
- Missing data/fields
- Units (different)
- Data types
- Abbreviations
- Duplicates or variations of the same entity
- Encoding
- Dashes (en-dash, em-dash, hyphen), parentheses, delimiters
- White spaces

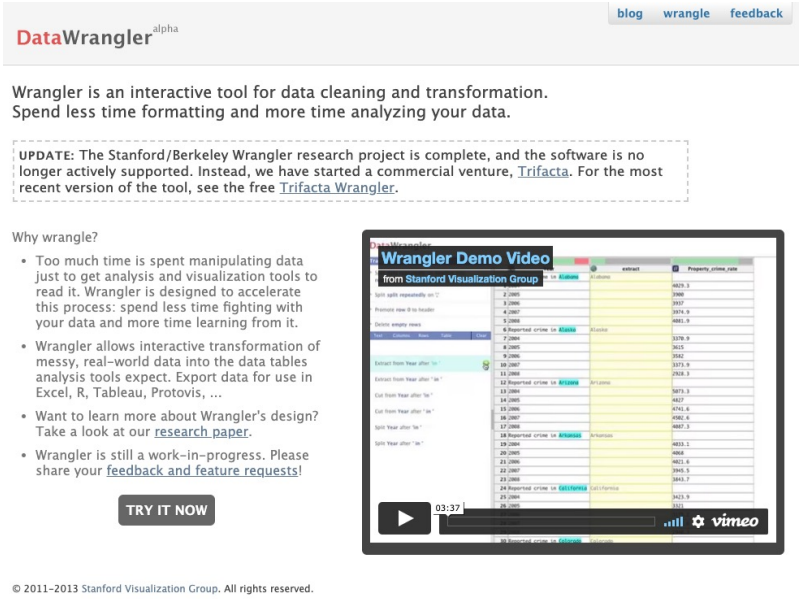
# Importance of data cleaning

- “80%” time spent on data preparation
- Cleaning Big Data: Most Time-Consuming,
  Least Enjoyable Data Science Task, Survey Says [Forbes]
  - https://www.forbes.com/sites/gilpress/2016/03/23/data-preparation-most-time-consuming-least-enjoyable-data-science-task-survey-says/
- https://en.wikipedia.org/wiki/Data_janitor

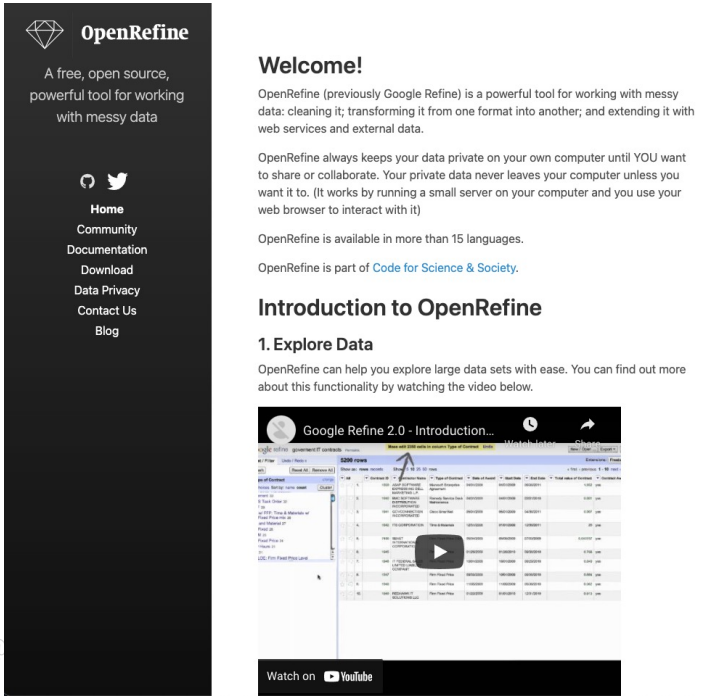


# Data cleaner tools

- Data Wrangler
  - Research at Stanford
  - http://vis.stanford.edu/wrangler/




- OpenRefine
  - Previously Google Refine
  - https://openrefine.org/



# What can Data Wrangler and OpenRefine do?

- [O] Clustering by similarity
- [O, W] Removing empty space
- [O, W] Reformatting
- [W, O] Comprehension/exporting of transformation (e./g., to excel, javascript)
- [W] Keyword extraction
- [O] Different unit (scaling, distribution); outliers [W] suggestions
- [W] Changing data types
- [O, W] undo/redo
- [O, W] Sorting
- [O] supporting scripting

# Regular expressions

- Python built-in package: re

Python » English 3.8.12 Documentation » The Python Standard Library » Text Processing Services »

Table of Contents

re — Regular expression operations
- Regular Expression Syntax
- Module Contents
- Regular Expression Objects
- Match Objects
- Regular Expression Examples
  - Checking for a Pair
  - Simulating scanf()
  - search() vs. match()
  - Making a Phonebook
  - Text Munging
  - Finding all Adverbs
  - Finding all Adverbs and their Positions
  - Raw String Notation
  - Writing a Tokenizer

Previous topic

string — Common string operations

Next topic

difflib — Helpers for computing deltas

This Page

Report a Bug
Show Source

## re — Regular expression operations

**Source code:** Lib/re.py

This module provides regular expression matching operations similar to those found in Perl.

Both patterns and strings to be searched can be Unicode strings (str) as well as 8-bit strings (bytes). However, Unicode strings and 8-bit strings cannot be mixed: that is, you cannot match a Unicode string with a byte pattern or vice-versa; similarly, when asking for a substitution, the replacement string must be of the same type as both the pattern and the search string.

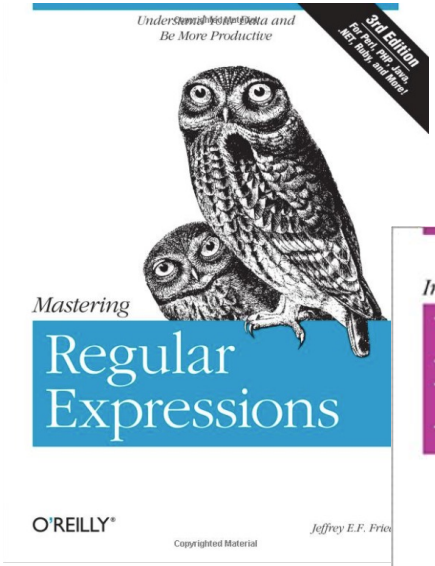
Regular expressions use the backslash character ('\') to indicate special forms or to allow special characters to be used without invoking their special meaning. This collides with Python's usage of the same character for the same purpose in string literals; for example, to match a literal backslash, one might have to write '\\\\' as the pattern string, because the regular expression must be \\, and each backslash must be expressed as \\ inside a regular Python string literal. Also, please note that any invalid escape sequences in Python's usage of the backslash in string literals now generate a DeprecationWarning and in the future this will become a SyntaxError. This behaviour will happen even if it is a valid escape sequence for a regular expression.

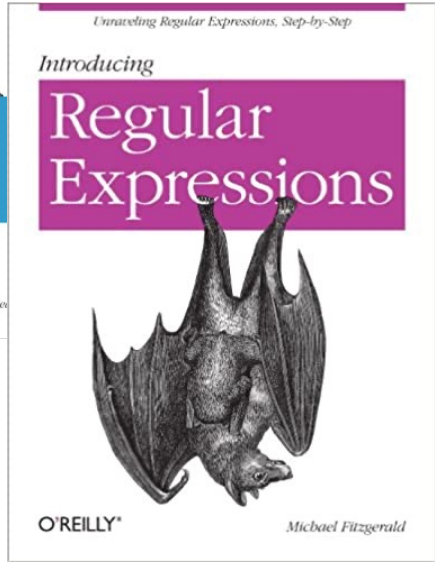
The solution is to use Python's raw string notation for regular expression patterns; backslashes are not handled in any special way in a string literal prefixed with 'r'. So r"\n" is a two-character string containing '\' and 'n', while "\n" is a one-character string containing a newline. Usually patterns will be expressed in Python code using this raw string notation.

It is important to note that most regular expression operations are available as module-level functions and methods on compiled regular expressions. The functions are shortcuts that don't require you to compile a regex object first, but miss some fine-tuning parameters.

**See also:** The third-party regex module, which has an API compatible with the standard library re module, but offers additional functionality and a more thorough Unicode support.

### Regular Expression Syntax



# OCR

- One powerful data collection method, yet producing dirty data
- https://cloud.google.com/vision
- Free alternative: tesseract

# Data Storing

# CSV or text-based files

- https://en.wikipedia.org/wiki/Comma-separated_values
- Easiest way to store data
- Some other text-based files
  - Fixed width file
  - Tab separated values (TSV)

- Adjacent fields must be separated by a single comma. However, comma is used as a decimal separator, semicolon, TAB, or other

```
1997,Ford,E350
```

- Any field *may* be *quoted* (that is, enclosed within double-quote ch

```
"1997","Ford","E350"
```

- Fields with embedded commas or double-quote characters must

```
1997,Ford,E350,"Super, luxurious truck"
```

- Each of the embedded double-quote characters must be represe

```
1997,Ford,E350,"Super, ""luxurious"" truck"
```

- Fields with embedded line breaks must be quoted (however, mar

```
1997,Ford,E350,"Go get one now
they are going fast"
```

Sqlite

SQLite

Home About Documentation Download License Support Purchase

## What Is SQLite?

SQLite is a C-language library that implements a small, fast, self-contained, high-reliability, full-featured, SQL database engine. SQLite is the most used database engine in the world. SQLite is built into all mobile phones and most computers and comes bundled inside countless other applications that people use every day. More Information...

The SQLite file format is stable, cross-platform, and backwards compatible and the developers pledge to keep it that way through the year 2050. SQLite database files are commonly used as containers to transfer rich content between systems [1] [2] [3] and as a long-term archival format for data [4]. There are over 1 trillion (1e12) SQLite databases in active use [5].

SQLite source code is in the public-domain and is free to everyone to use for any purpose.

## Latest Release

Version 3.36.0 (2021-06-18). Download Prior Releases

Comr

- Features
- When to
- Getting S
- Prior Rel
- SQL Syn
  - Prag
  - SQL
  - Dat
  - Agg
  - Win
  - Mat
  - JSO
- C/C++ I
  - Intr
  - List
- The TCL
- Quirks a
- Frequent
- Commit
- Bugs
- News

Hyun



Table of Contents



# sqlite3 — DB-API 2.0 interface for SQLite databases

**Source code:** Lib/sqlite3/

SQLite is a C library that provides a lightweight disk-based database that doesn't require a separate server process and allows accessing the database using a nonstandard variant of the SQL query language. Some applications can use SQLite for internal data storage. It's also possible to prototype an application using SQLite and then port the code to a larger database such as PostgreSQL or Oracle.

The sqlite3 module was written by Gerhard Häring. It provides a SQL interface compliant with the DB-API 2.0 specification described by PEP 249.

To use the module, you must first create a `Connection` object that represents the database. Here the data will be stored in the `example.db` file:

```
import sqlite3
con = sqlite3.connect('example.db')
```

You can also supply the special name `:memory:` to create a database in RAM.

Once you have a `Connection`, you can create a `Cursor` object and call its `execute()` method to perform SQL commands:

```
cur = con.cursor()

# Create table
cur.execute('''CREATE TABLE stocks
               (date text, trans text, symbol text, qty real, price rea

# Insert a row of data
cur.execute("INSERT INTO stocks VALUES ('2006-01-05','BUY','RHAT',100,3

# Save (commit) the changes
con.commit()

# We can also close the connection if we are done with it.
# Just be sure any changes have been committed or they will be lost.
con.close()
```

# RDBMS / NoSQL

- RDBMS
  - https://en.wikipedia.org/wiki/Relational_database#RDBMS
  - MySQL, PostgreSQL, SQLite
- NoSQL
  - https://en.wikipedia.org/wiki/NoSQL
  - MongoDB, HBase, Redis

# Data Wrangling with Pandas

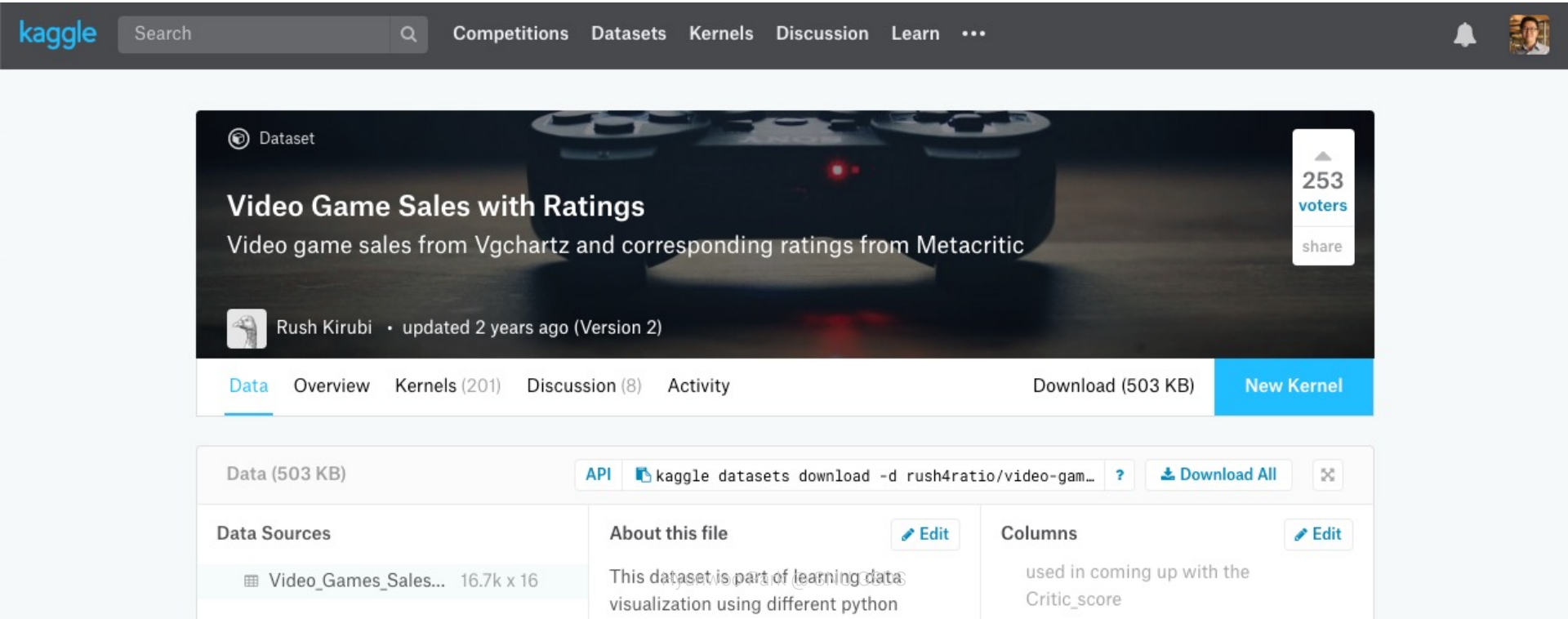
- https://www.tomasbeuzen.com/python-programming-for-data-science/chapters/chapter7-pandas.html
- https://www.tomasbeuzen.com/python-programming-for-data-science/chapters/chapter8-wrangling-basics.html
- https://www.tomasbeuzen.com/python-programming-for-data-science/chapters/chapter9-wrangling-advanced.html

- List of lists?

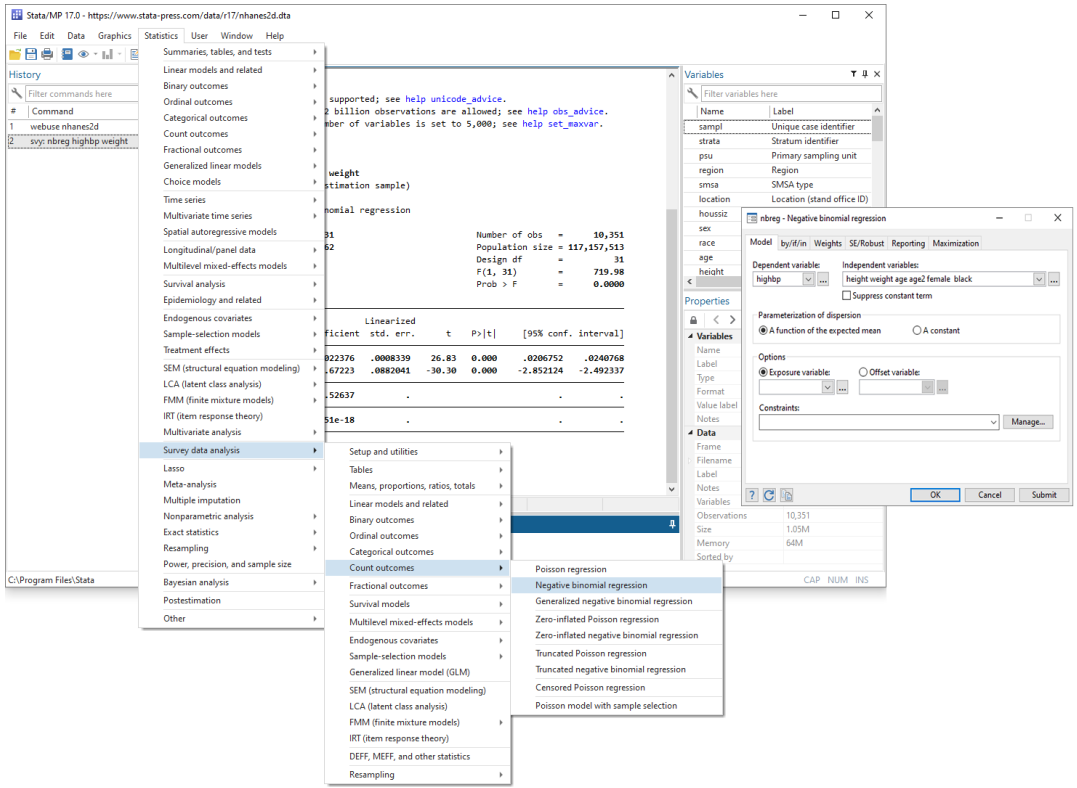
# Next Week

# Let's grab a dataset.

- Video Game Sales with Ratings
  https://www.kaggle.com/rush4ratio/video-game-sales-with-ratings



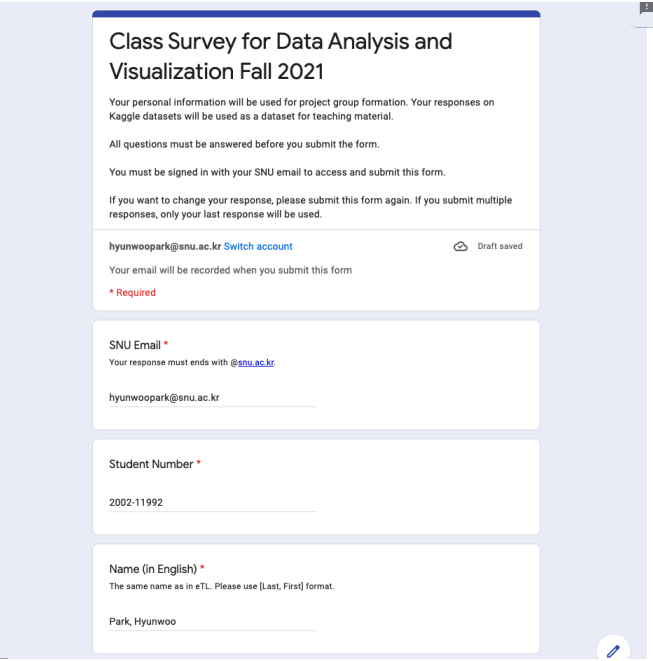
# Stata


Stata/MP 17.0 - https://www.stata-press.com/data/r17/nhanes2d.dta
File Edit Data Graphics Statistics User Window Help
History
Filter commands here
# Command
1 webuse nhanes2d
2 svy: nbreg highbp weight
Summaries, tables, and tests
Linear models and related
Binary outcomes
Ordinal outcomes
Categorical outcomes
Count outcomes
Fractional outcomes
Generalized linear models
Choice models
Time series
Multivariate time series
Spatial autoregressive models
Longitudinal/panel data
Multilevel mixed-effects models
Survival analysis
Epidemiology and related
Endogenous covariates
Sample-selection models
Treatment effects
SEM (structural equation modeling)
LCA (latent class analysis)
FMM (finite mixture models)
IRT (item response theory)
Multivariate analysis
Survey data analysis
Lasso
Meta-analysis
Multiple imputation
Nonparametric analysis
Exact statistics
Resampling
Power, precision, and sample size
Bayesian analysis
Postestimation
Other
supported; see help unicode_advice.
2 billion observations are allowed; see help obs_advice.
mber of variables is set to 5,000; see help set_maxvar.
weight
stimation sample)
nomial regression
Number of obs = 10,351
Population size = 117,157,513
Design df = 31
F(1, 31) = 719.98
Prob > F = 0.0000
Linearized
ficient std. err. t P>|t| [95% conf. interval]
022376 .0008339 26.83 0.000 .0206752 .0240768
.67223 .0882041 -30.30 0.000 -2.852124 -2.492337
.52637 . . . .
51e-18 . . . .
Setup and utilities
Tables
Means, proportions, ratios, totals
Linear models and related
Binary outcomes
Ordinal outcomes
Categorical outcomes
Count outcomes
Fractional outcomes
Survival models
Multilevel mixed-effects models
Endogenous covariates
Sample-selection models
Generalized linear model (GLM)
SEM (structural equation modeling)
LCA (latent class analysis)
FMM (finite mixture models)
IRT (item response theory)
DEFF, MEFF, and other statistics
Resampling
Poisson regression
Negative binomial regression
Generalized negative binomial regression
Zero-inflated Poisson regression
Zero-inflated negative binomial regression
Truncated Poisson regression
Truncated negative binomial regression
Censored Poisson regression
Poisson model with sample selection
Variables
Filter variables here
Name Label
sampl Unique case identifier
strata Stratum identifier
psu Primary sampling unit
region Region
smsa SMSA type
location Location (stand office ID)
houssiz
sex
race
age
height
Properties
Variables
Name
Label
Type
Format
Value label
Notes
Data
Frame
Filename
Label
Notes
Variables
Observations 10,351
Size 1.05M
Memory 64M
Sorted by
C:\Program Files\Stata
CAP NUM INS
nbreg - Negative binomial regression
Model by/if/in Weights SE/Robust Reporting Maximization
Dependent variable: highbp
Independent variables: height weight age age2 female black
Suppress constant term
Parameterization of dispersion
A function of the expected mean
A constant
Options
Exposure variable:
Offset variable:
Constraints:
Manage...
OK Cancel Submit

# Class Survey
# Due 9/13
# before class 12:30pm KST

# Example Response

https://forms.gle/SBsgZpFcAciLYRjP8



Class Survey for Data Analysis and Visualization Fall 2021

Your personal information will be used for project group formation. Your responses on Kaggle datasets will be used as a dataset for teaching material.

All questions must be answered before you submit the form.

You must be signed in with your SNU email to access and submit this form.

If you want to change your response, please submit this form again. If you submit multiple responses, only your last response will be used.

hyunwoopark@snu.ac.kr Switch account  Draft saved

Your email will be recorded when you submit this form

* Required

SNU Email *

Your response must ends with @snu.ac.kr.

hyunwoopark@snu.ac.kr

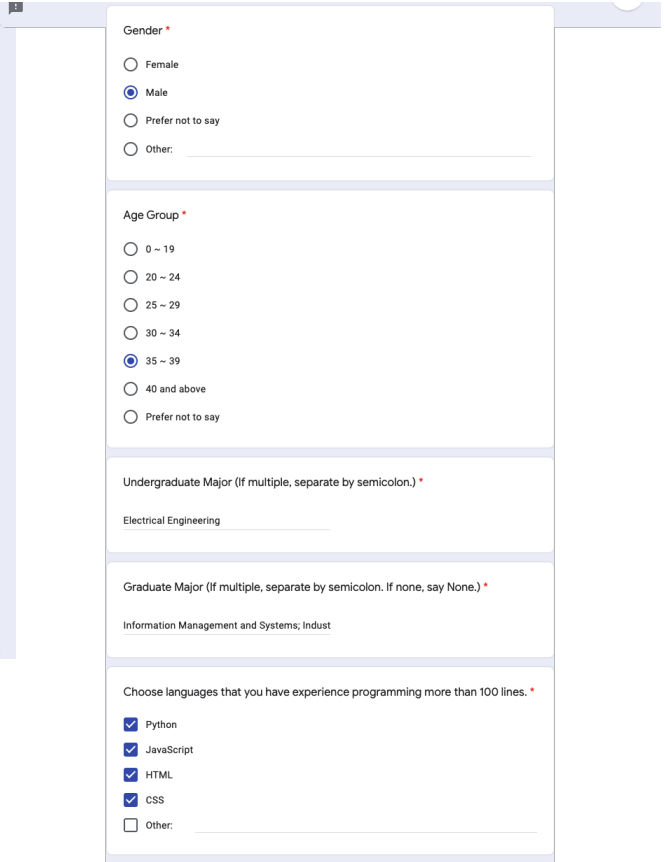
Student Number *

2002-11992

Name (in English) *

The same name as in eTL. Please use [Last, First] format.

Park, Hyunwoo



Gender *

- ○ Female
- ◉ Male
- ○ Prefer not to say
- ○ Other:

Age Group *

- ○ 0 ~ 19
- ○ 20 ~ 24
- ○ 25 ~ 29
- ○ 30 ~ 34
- ◉ 35 ~ 39
- ○ 40 and above
- ○ Prefer not to say

Undergraduate Major (If multiple, separate by semicolon.) *

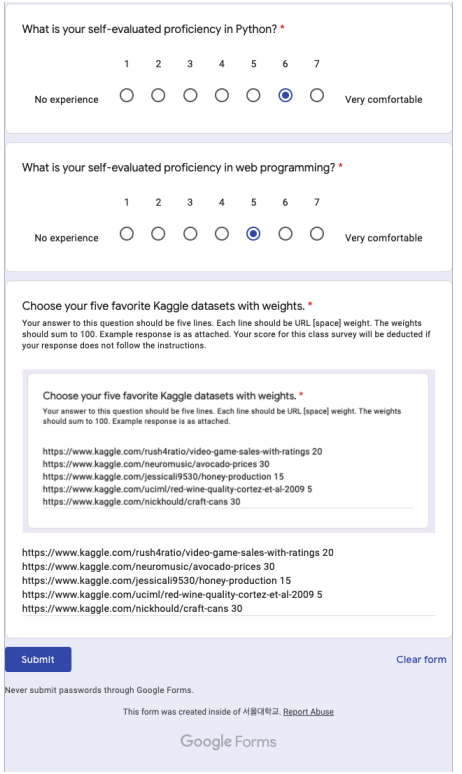
Electrical Engineering

Graduate Major (If multiple, separate by semicolon. If none, say None.) *

Information Management and Systems; Indust

Choose languages that you have experience programming more than 100 lines. *

- ☑ Python
- ☑ JavaScript
- ☑ HTML
- ☑ CSS
- ☐ Other:



What is your self-evaluated proficiency in Python? *

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | |
|---|---|---|---|---|---|---|---|---|
| No experience | ○ | ○ | ○ | ○ | ○ | ◉ | ○ | Very comfortable |

What is your self-evaluated proficiency in web programming? *

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | |
|---|---|---|---|---|---|---|---|---|
| No experience | ○ | ○ | ○ | ○ | ◉ | ○ | ○ | Very comfortable |

Choose your five favorite Kaggle datasets with weights. *

Your answer to this question should be five lines. Each line should be URL [space] weight. The weights should sum to 100. Example response is as attached. Your score for this class survey will be deducted if your response does not follow the instructions.

Choose your five favorite Kaggle datasets with weights. *

Your answer to this question should be five lines. Each line should be URL [space] weight. The weights should sum to 100. Example response is as attached.

https://www.kaggle.com/rush4ratio/video-game-sales-with-ratings 20
https://www.kaggle.com/neuromusic/avocado-prices 30
https://www.kaggle.com/jessicali9530/honey-production 15
https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009 5
https://www.kaggle.com/nickhould/craft-cans 30

https://www.kaggle.com/rush4ratio/video-game-sales-with-ratings 20
https://www.kaggle.com/neuromusic/avocado-prices 30
https://www.kaggle.com/jessicali9530/honey-production 15
https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009 5
https://www.kaggle.com/nickhould/craft-cans 30

Submit  Clear form

Never submit passwords through Google Forms.

This form was created inside of 서울대학교. Report Abuse

Google Forms

# Things to do / Reminders

- Class Survey (5%)
  - Due on 9/13 (Tue) before class 2pm KST.
- Review
  - Check out Chrome Developer Tools.
  - Watch Data Wranger and OpenRefine videos.
  - Try regular expression using re.
- eTL boards
  - Q&A Board for questions and answers
  - Discussion Board for resources or ideas to share
- See you Tuesday, 9/14.
  - Univariate Analysis and Visualization
  - Zoom only