Computational Data Analytics for Economists

Lecture 2

Web Scraping and Tools for Scientific Programming

Helge Liebert   Anthony Strittmatter

Outline

1. Introduction
2. Basic tools
3. Web APIs
4. HTML and the DOM
5. A simple web scraper
6. Regular Expressions
7. Assignment

What is this lecture about?

- Most research in economics now involves scientific programming.
- Introduce tools and ideas that may make daily research tasks easier.
- Many of these have been developed by other scientists or IT professionals.
- Some are almost as old as computers, others are novel.
- Focus is going to be on data processing.

Economics and computer science

- Data management is not taught in introductory econometrics.
- Computer science often involves processing data.
- Problems you are likely to encounter have been solved.
- CS offers tools and concepts that economists can profit from.
    - Tools: Remote servers, databases, version control, acessing APIs for data, text processing and analysis, geospatial analysis, OCR, automation, . . .
    - Concepts: Time complexity of algorithms, computational cost, databases.

Topics

- General points about scientific programming and working with computers.
- Use processing of web data to introduce various ideas along the way.

- Unix tools and the command line.
- Accessing data on the web: APIs.
- Gathering (unstructured) web data and transforming it into structured data ("*web scraping*").
- Regular expressions.

Goal: Understanding the tools available

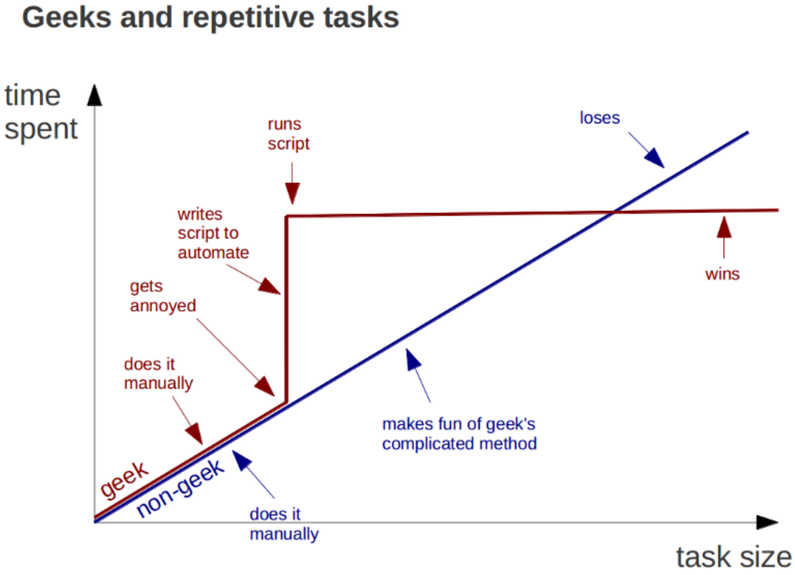
- Navigate the jungle: Isolating a particular tool is often harder than understanding how it works.
- Point you towards the resources and explain their general concepts.
- Methods may sometimes offer the only feasible solution to a problem.
- They may also help you solve tasks efficiently.
- Knowing that a problem can be solved and how is worth a lot.
- Leave you marginally more computer literate.

Goal: Automation

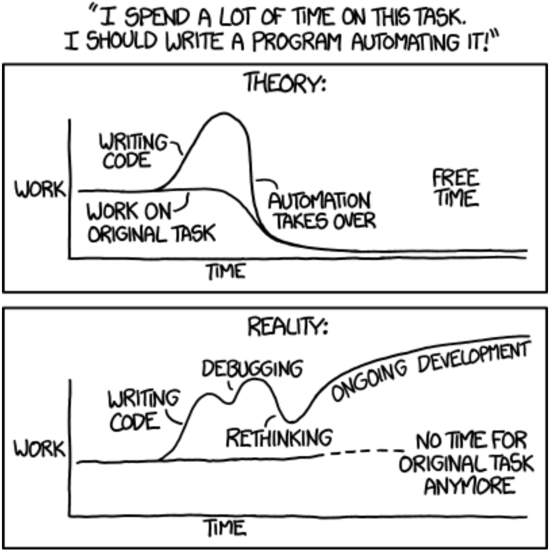
- Digitalization offers exciting data for research. But: *Data is messy*.
- Gathering or processing data often involves repetitive manual tasks.
- Disadvantages:
  - Manual tasks are often not well documented or reproducible ex post.
  - Manual work is frustrating and a huge time-sink.
  - Manual work may not be feasible with large data.

➥ Automation helps!
  - Frees you to engage in other work.
  - You learn new things.
  - Should you encounter the same class of problem in the future, you already have a solution at hand.

Automation


Geeks and repetitive tasks
time spent
task size
runs script
writes script to automate
gets annoyed
does it manually
geek
non-geek
does it manually
makes fun of geek's complicated method
loses
wins

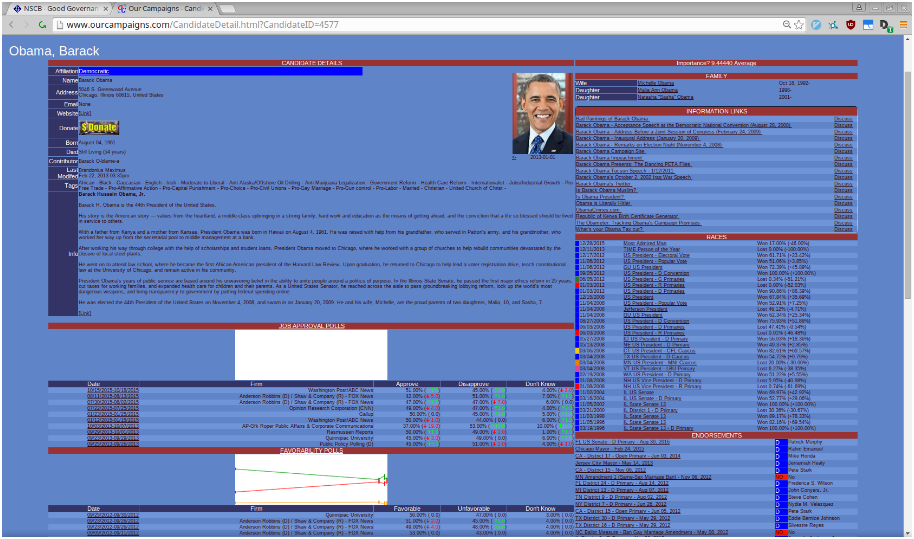
Automation


"I SPEND A LOT OF TIME ON THIS TASK.
I SHOULD WRITE A PROGRAM AUTOMATING IT!"
THEORY:
WORK
TIME
WRITING CODE
WORK ON ORIGINAL TASK
AUTOMATION TAKES OVER
FREE TIME
REALITY:
WORK
TIME
WRITING CODE
DEBUGGING
RETHINKING
ONGOING DEVELOPMENT
NO TIME FOR ORIGINAL TASK ANYMORE

Guiding problem

- How to turn unstructured into structured data?
- Consider a situation where
  - You want to get data from the internet.
  - The data is in unstructured/semi-structured form.
  - You want to transform it into a differently structured format for further use.
  - You need to filter the available information.

From this . . .



Obama, Barack

CANDIDATE DETAILS

Affiliation	Democratic
Name	Barack Obama
Address	5046 S. Greenwood Avenue Chicago, Illinois 60615, United States
Email	None
Website	[Link]
Donate	$ Donate
Born	August 04, 1961
Died	Still Living (54 years)
Contributor	Barack O-blame-a
Last Modifed	Brandonius Maximus Feb 22, 2013 03:35pm
Tags	African - Black - Caucasian - English - Irish - Moderate-to-Liberal - Anti Alaska/Offshore Oil Drilling - Anti Marijuana Legalization - Government Reform - Health Care Reform - Internationalist - Jobs/Industrial Growth - Pro Free Trade - Pro-Affirmative Action - Pro-Capital Punishment - Pro-Choice - Pro-Civil Unions - Pro-Gay Marriage - Pro-Gun control - Pro-Labor - Married - Christian - United Church of Christ -

<- 2013-01-01

Info

Barack Hussein Obama, Jr.

Barack H. Obama is the 44th President of the United States.

His story is the American story — values from the heartland, a middle-class upbringing in a strong family, hard work and education as the means of getting ahead, and the conviction that a life so blessed should be lived in service to others.

With a father from Kenya and a mother from Kansas, President Obama was born in Hawaii on August 4, 1961. He was raised with help from his grandfather, who served in Patton's army, and his grandmother, who worked her way up from the secretarial pool to middle management at a bank.

After working his way through college with the help of scholarships and student loans, President Obama moved to Chicago, where he worked with a group of churches to help rebuild communities devastated by the closure of local steel plants.

He went on to attend law school, where he became the first African-American president of the Harvard Law Review. Upon graduation, he returned to Chicago to help lead a voter registration drive, teach constitutional law at the University of Chicago, and remain active in his community.

President Obama's years of public service are based around his unwavering belief in the ability to unite people around a politics of purpose. In the Illinois State Senate, he passed the first major ethics reform in 25 years, cut taxes for working families, and expanded health care for children and their parents. As a United States Senator, he reached across the aisle to pass groundbreaking lobbying reform, lock up the world's most dangerous weapons, and bring transparency to government by putting federal spending online.

He was elected the 44th President of the United States on November 4, 2008, and sworn in on January 20, 2009. He and his wife, Michelle, are the proud parents of two daughters, Malia, 10, and Sasha, 7.

[Link]

JOB APPROVAL POLLS

Date	Firm	Approve	Disapprove	Don't Know
10/15/2015-10/18/2015	Washington Post/ABC News	51.00% (↑1.0)	45.00% (↑1.0)	4.00% (↓ 2.0)
08/11/2015-08/13/2015	Anderson Robbins (D) / Shaw & Company (R) - FOX News	42.00% (↓ 5.0)	51.00% (↑4.0)	7.00% (↑1.0)
07/30/2015-08/02/2015	Anderson Robbins (D) / Shaw & Company (R) - FOX News	47.00% (↑7.0)	47.00% (↓ 7.0)	6.00% (0.0)
07/22/2015-07/25/2015	Opinion Research Corporation (CNN)	49.00% (↓ 4.0)	47.00% (↑2.0)	4.00% (↑2.0)
01/23/2015-01/25/2015	Gallup	50.00% (0.0)	45.00% (↑1.0)	5.00% (↑5.0)
01/12/2015-01/15/2015	Washington Post/ABC News	50.00% (↓ 1.0)	44.00% (0.0)	6.00% (↑1.0)
10/03/2013-10/07/2013	AP-Gfk Roper Public Affairs & Corporate Communications	37.00% (↓ 19.0)	53.00% (↑13.0)	10.00% (↑10.0)
09/29/2013-10/01/2013	Rasmussen Reports	50.00% (↑1.0)	49.00% (↓ 1.0)	1.00% (↑1.0)
09/23/2013-09/29/2013	Quinnipiac University	45.00% (↓ 3.0)	49.00% (0.0)	6.00% (↑3.0)
09/25/2013-09/26/2013	Public Policy Polling (D)	45.00% (↑3.0)	51.00% (↓ 2.0)	4.00% (↓ 1.0)

FAVORABILITY POLLS

Date	Firm	Favorable	Unfavorable	Don't Know
09/25/2012-09/30/2012	Quinnipiac University	50.00% (0.0)	47.00% (0.0)	3.00% (0.0)
09/23/2012-09/26/2012	Anderson Robbins (D) / Shaw & Company (R) - FOX News	51.00% (↓ 2.0)	45.00% (↑2.0)	4.00% (0.0)
09/23/2012-09/26/2012	Anderson Robbins (D) / Shaw & Company (R) - FOX News	49.00% (↓ 4.0)	48.00% (↑5.0)	4.00% (0.0)
09/09/2012-09/11/2012	Anderson Robbins (D) / Shaw & Company (R) - FOX News	53.00% (0.0)	43.00% (0.0)	4.00% (0.0)

Importance? 9.44440 Average

FAMILY

Wife	Michelle Obama	Oct 18, 1992-
Daughter	Malia Ann Obama	1998-
Daughter	Natasha "Sasha" Obama	2001-

INFORMATION LINKS

Bad Paintings of Barack Obama	Discuss
Barack Obama - Acceptance Speech at the Democratic National Convention (August 28, 2008)	Discuss
Barack Obama - Address Before a Joint Session of Congress (February 24, 2009)	Discuss
Barack Obama - Inaugural Address (January 20, 2009)	Discuss
Barack Obama - Remarks on Election Night (November 4, 2008)	Discuss
Barack Obama Campaign Site	Discuss
Barack Obama Impeachment	Discuss
Barack Obama Presents: The Dancing PETA Flies	Discuss
Barack Obama Tucson Speech - 1/12/2011	Discuss
Barack Obama's October 2, 2002 Iraq War Speech	Discuss
Barack Obama's Twitter	Discuss
Is Barack Obama Muslim?	Discuss
Is Obama President?	Discuss
Obama is Literally Hitler	Discuss
ObamaCrimes.com	Discuss
Republic of Kenya Birth Certificate Generator	Discuss
The Obameter: Tracking Obama's Campaign Promises	Discuss
What's your Obama Tax cut?	Discuss

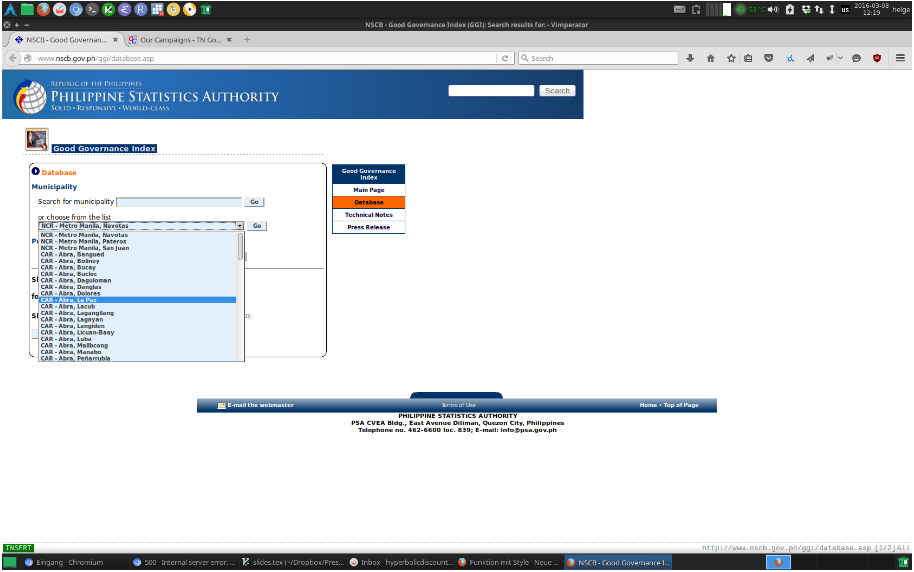
RACES

🟦	12/28/2015	Most Admired Man	Won 17.00% (-46.00%)
🟦	12/11/2013	TIME Person of the Year	Lost 0.00% (-100.00%)
🟦	12/17/2012	US President - Electoral Vote	Won 61.71% (+23.42%)
🟦	11/06/2012	US President - Popular Vote	Won 51.06% (+3.85%)
🟦	11/06/2012	GU US President	Won 72.39% (+45.89%)
🟦	09/05/2012	US President - D Convention	Won 100.00% (+100.00%)
🟩	06/05/2012	US President - G Primaries	Lost 0.34% (-51.21%)
🟥	01/03/2012	US President - R Primaries	Lost 0.00% (-52.03%)
🟦	01/03/2012	US President - D Primaries	Won 90.86% (+88.39%)
🟦	12/15/2008	US President	Won 67.84% (+35.69%)
🟦	11/04/2008	US President - Popular Vote	Won 52.91% (+7.25%)
🟦	11/04/2008	Jefferson President	Lost 46.12% (-4.71%)
🟦	11/04/2008	GU US President	Won 62.34% (+25.34%)
🟦	08/27/2008	US President - D Convention	Won 75.93% (+51.86%)
🟦	06/03/2008	US President - D Primaries	Lost 47.41% (-0.54%)
🟥	06/03/2008	US President - R Primaries	Lost 0.01% (-46.48%)
🟦	05/27/2008	ID US President - D Primary	Won 56.03% (+18.36%)
🟦	05/13/2008	NE US President - D Primary	Won 49.37% (+2.85%)
🟨	03/06/2008	CT US President - CFL Caucus	Won 82.61% (+69.57%)
🟦	03/04/2008	TX US President - D Caucus	Won 54.72% (+9.79%)
🟧	03/04/2008	MN US President - MNI Caucus	Lost 20.00% (-30.00%)
🟪	03/04/2008	VT US President - LBU Primary	Lost 6.27% (-38.35%)
🟦	02/19/2008	WA US President - D Primary	Won 51.22% (+5.55%)
🟦	01/08/2008	NH US Vice President - D Primary	Lost 5.95% (-40.98%)
🟥	01/08/2008	NH US Vice President - R Primary	Lost 0.74% (-61.69%)
🟦	11/02/2004	IL US Senate	Won 69.97% (+42.92%)
🟦	03/16/2004	IL US Senate - D Primary	Won 52.77% (+29.06%)
🟦	11/05/2002	IL State Senate 13	Won 100.00% (+100.00%)
🟦	03/21/2000	IL District 1 - D Primary	Lost 30.36% (-30.67%)
🟦	11/03/1998	IL State Senate 13	Won 89.17% (+78.33%)
🟦	11/05/1996	IL State Senate 13	Won 82.16% (+69.54%)
🟦	03/19/1996	IL State Senate 13 - D Primary	Won 100.00% (+100.00%)

ENDORSEMENTS

FL US Senate - D Primary - Aug 30, 2016	D	Patrick Murphy
Chicago Mayor - Feb 24, 2015	D	Rahm Emanuel
CA - District 17 - Open Primary - Jun 03, 2014	D	Mike Honda
Jersey City Mayor - May 14, 2013	D	Jerramiah Healy
CA - District 15 - Nov 06, 2012	D	Pete Stark
MN Amendment 1 (Same-Sex Marriage Ban) - Nov 06, 2012	NO	No
FL District 24 - D Primary - Aug 14, 2012	D	Frederica S. Wilson
MI District 13 - D Primary - Aug 07, 2012	D	John Conyers, Jr.
TN District 9 - D Primary - Aug 02, 2012	D	Steve Cohen
NY District 7 - D Primary - Jun 26, 2012	D	Nydia M. Velazquez
CA - District 15 - Open Primary - Jun 05, 2012	D	Pete Stark
TX District 30 - D Primary - May 29, 2012	D	Eddie Bernice Johnson
TX District 16 - D Primary - May 29, 2012	D	Silvestre Reyes
NC Ballot Measure - Ban Gay Marriage Amendment - May 08, 2012	NO	No

. . . or this



NSCB - Good Governance Index (GGI): Search results for: - Vimperator

www.nscb.gov.ph/ggi/database.asp

REPUBLIC OF THE PHILIPPINES
PHILIPPINE STATISTICS AUTHORITY
SOLID • RESPONSIVE • WORLD-CLASS

Good Governance Index

Database

Municipality

Search for municipality [ ] Go

or choose from the list

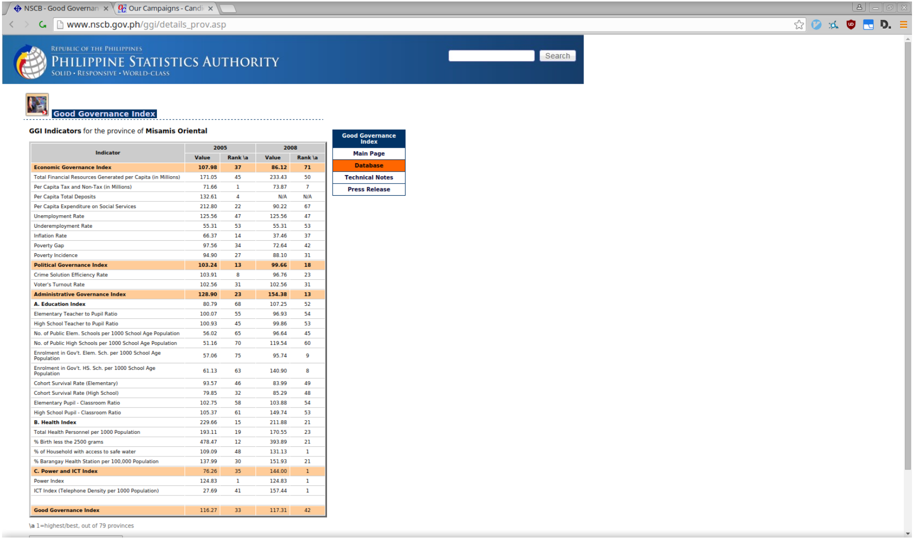
NCR - Metro Manila, Navotas

NCR - Metro Manila, Navotas
NCR - Metro Manila, Pateros
NCR - Metro Manila, San Juan
CAR - Abra, Bangued
CAR - Abra, Boliney
CAR - Abra, Bucay
CAR - Abra, Bucloc
CAR - Abra, Daguioman
CAR - Abra, Danglas
CAR - Abra, Dolores
CAR - Abra, La Paz
CAR - Abra, Lacub
CAR - Abra, Lagangilang
CAR - Abra, Lagayan
CAR - Abra, Langiden
CAR - Abra, Licuan-Baay
CAR - Abra, Luba
CAR - Abra, Malibcong
CAR - Abra, Manabo
CAR - Abra, Peñarrubia

PHILIPPINE STATISTICS AUTHORITY
PSA CVEA Bldg., East Avenue Diliman, Quezon City, Philippines
Telephone no. 462-6600 loc. 839; E-mail: info@psa.gov.ph

INSERT
http://www.nscb.gov.ph/ggi/database.asp [1/2] All

. . . and this


NSCB - Good Governan
Our Campaigns - Candi
www.nscb.gov.ph/ggi/details_prov.asp
Republic of the Philippines
Philippine Statistics Authority
Solid • Responsive • World-class
Search
Good Governance Index
Main Page
Database
Technical Notes
Press Release

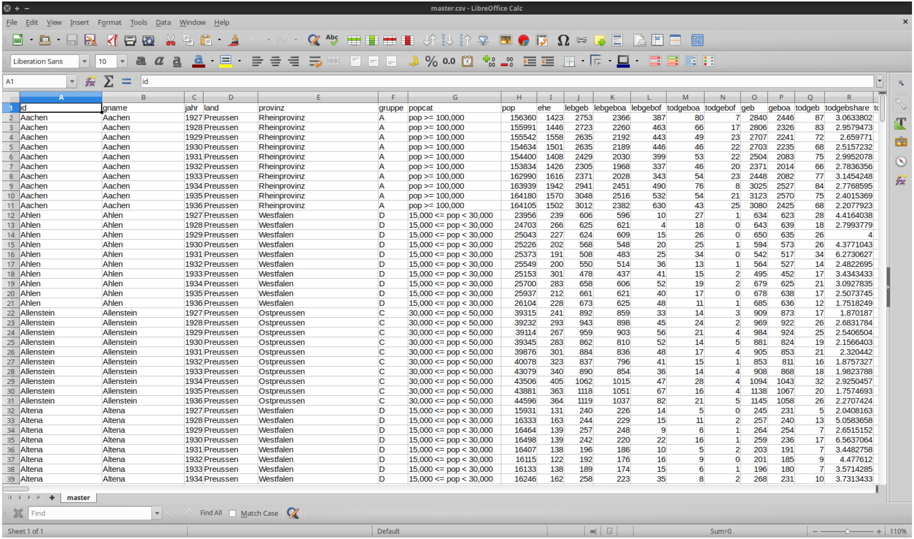
Good Governance Index

GGI Indicators for the province of **Misamis Oriental**

Indicator	2005		2008	
	Value	Rank \a	Value	Rank \a
Economic Governance Index	**107.98**	**37**	**86.12**	**71**
Total Financial Resources Generated per Capita (in Millions)	171.05	45	233.43	50
Per Capita Tax and Non-Tax (in Millions)	71.66	1	73.87	7
Per Capita Total Deposits	132.61	4	N/A	N/A
Per Capita Expenditure on Social Services	212.80	22	90.22	67
Unemployment Rate	125.56	47	125.56	47
Underemployment Rate	55.31	53	55.31	53
Inflation Rate	66.37	14	37.46	37
Poverty Gap	97.56	34	72.64	42
Poverty Incidence	94.90	27	88.10	31
Political Governance Index	**103.24**	**13**	**99.66**	**18**
Crime Solution Efficiency Rate	103.91	8	96.76	23
Voter's Turnout Rate	102.56	31	102.56	31
Administrative Governance Index	**128.90**	**23**	**154.38**	**13**
A. Education Index	80.79	68	107.25	52
Elementary Teacher to Pupil Ratio	100.07	55	96.93	54
High School Teacher to Pupil Ratio	100.93	45	99.86	53
No. of Public Elem. Schools per 1000 School Age Population	56.02	65	96.64	45
No. of Public High Schools per 1000 School Age Population	51.16	70	119.54	60
Enrolment in Gov't. Elem. Sch. per 1000 School Age Population	57.06	75	95.74	9
Enrolment in Gov't. HS. Sch. per 1000 School Age Population	61.13	63	140.90	8
Cohort Survival Rate (Elementary)	93.57	46	83.99	49
Cohort Survival Rate (High School)	79.85	32	85.29	48
Elementary Pupil - Classroom Ratio	102.75	58	103.88	54
High School Pupil - Classroom Ratio	105.37	61	149.74	53
B. Health Index	229.66	15	211.88	21
Total Health Personnel per 1000 Population	193.11	19	170.55	23
% Birth less the 2500 grams	478.47	12	393.89	21
% of Household with access to safe water	109.09	48	131.13	1
% Barangay Health Station per 100,000 Population	137.99	30	151.93	21
C. Power and ICT Index	76.26	35	144.00	1
Power Index	124.83	1	124.83	1
ICT Index (Telephone Density per 1000 Population)	27.69	41	157.44	1
Good Governance Index	116.27	33	117.31	42

\a 1=highest/best, out of 79 provinces

. . . to this.


master.csv - LibreOffice Calc
File Edit View Insert Format Tools Data Window Help
Liberation Sans
10
A1
id
master
Find
Find All
Match Case
Sheet 1 of 1
Default
Sum=0
110%

id	gname	jahr	land	provinz	gruppe	popcat	pop	ehe	lebgeb	lebgeboa	lebgebof	todgeboa	todgebof	geb	geboa	todgeb	todgebshare	tc
Aachen	Aachen	1927	Preussen	Rheinprovinz	A	pop >= 100,000	156360	1423	2753	2366	387	80	7	2840	2446	87	3.0633802	
Aachen	Aachen	1928	Preussen	Rheinprovinz	A	pop >= 100,000	155991	1446	2723	2260	463	66	17	2806	2326	83	2.9579473	
Aachen	Aachen	1929	Preussen	Rheinprovinz	A	pop >= 100,000	155542	1558	2635	2192	443	49	23	2707	2241	72	2.659771	
Aachen	Aachen	1930	Preussen	Rheinprovinz	A	pop >= 100,000	154634	1501	2635	2189	446	46	22	2703	2235	68	2.5157232	
Aachen	Aachen	1931	Preussen	Rheinprovinz	A	pop >= 100,000	154400	1408	2429	2030	399	53	22	2504	2083	75	2.9952078	
Aachen	Aachen	1932	Preussen	Rheinprovinz	A	pop >= 100,000	153834	1426	2305	1968	337	46	20	2371	2014	66	2.7836356	
Aachen	Aachen	1933	Preussen	Rheinprovinz	A	pop >= 100,000	162990	1616	2371	2028	343	54	23	2448	2082	77	3.1454248	
Aachen	Aachen	1934	Preussen	Rheinprovinz	A	pop >= 100,000	163939	1942	2941	2451	490	76	8	3025	2527	84	2.7768595	
Aachen	Aachen	1935	Preussen	Rheinprovinz	A	pop >= 100,000	164180	1570	3048	2516	532	54	21	3123	2570	75	2.4015369	
Aachen	Aachen	1936	Preussen	Rheinprovinz	A	pop >= 100,000	164105	1502	3012	2382	630	43	25	3080	2425	68	2.2077923	
Ahlen	Ahlen	1927	Preussen	Westfalen	D	15,000 <= pop < 30,000	23956	239	606	596	10	27	1	634	623	28	4.4164038	
Ahlen	Ahlen	1928	Preussen	Westfalen	D	15,000 <= pop < 30,000	24703	266	625	621	4	18	0	643	639	18	2.7993779	
Ahlen	Ahlen	1929	Preussen	Westfalen	D	15,000 <= pop < 30,000	25043	227	624	609	15	26	0	650	635	26	4	
Ahlen	Ahlen	1930	Preussen	Westfalen	D	15,000 <= pop < 30,000	25226	202	568	548	20	25	1	594	573	26	4.3771043	
Ahlen	Ahlen	1931	Preussen	Westfalen	D	15,000 <= pop < 30,000	25373	191	508	483	25	34	0	542	517	34	6.2730627	
Ahlen	Ahlen	1932	Preussen	Westfalen	D	15,000 <= pop < 30,000	25549	200	550	514	36	13	1	564	527	14	2.4822695	
Ahlen	Ahlen	1933	Preussen	Westfalen	D	15,000 <= pop < 30,000	25153	301	478	437	41	15	2	495	452	17	3.4343433	
Ahlen	Ahlen	1934	Preussen	Westfalen	D	15,000 <= pop < 30,000	25700	283	658	606	52	19	2	679	625	21	3.0927835	
Ahlen	Ahlen	1935	Preussen	Westfalen	D	15,000 <= pop < 30,000	25937	212	661	621	40	17	0	678	638	17	2.5073745	
Ahlen	Ahlen	1936	Preussen	Westfalen	D	15,000 <= pop < 30,000	26104	228	673	625	48	11	1	685	636	12	1.7518249	
Allenstein	Allenstein	1927	Preussen	Ostpreussen	C	30,000 <= pop < 50,000	39315	241	892	859	33	14	3	909	873	17	1.870187	
Allenstein	Allenstein	1928	Preussen	Ostpreussen	C	30,000 <= pop < 50,000	39232	293	943	898	45	24	2	969	922	26	2.6831784	
Allenstein	Allenstein	1929	Preussen	Ostpreussen	C	30,000 <= pop < 50,000	39114	267	959	903	56	21	4	984	924	25	2.5406504	
Allenstein	Allenstein	1930	Preussen	Ostpreussen	C	30,000 <= pop < 50,000	39345	283	862	810	52	14	5	881	824	19	2.1566403	
Allenstein	Allenstein	1931	Preussen	Ostpreussen	C	30,000 <= pop < 50,000	39876	301	884	836	48	17	4	905	853	21	2.320442	
Allenstein	Allenstein	1932	Preussen	Ostpreussen	C	30,000 <= pop < 50,000	40078	323	837	796	41	15	1	853	811	16	1.8757327	
Allenstein	Allenstein	1933	Preussen	Ostpreussen	C	30,000 <= pop < 50,000	43079	340	890	854	36	14	4	908	868	18	1.9823788	
Allenstein	Allenstein	1934	Preussen	Ostpreussen	C	30,000 <= pop < 50,000	43506	405	1062	1015	47	28	4	1094	1043	32	2.9250457	
Allenstein	Allenstein	1935	Preussen	Ostpreussen	C	30,000 <= pop < 50,000	43881	363	1118	1051	67	16	4	1138	1067	20	1.7574693	
Allenstein	Allenstein	1936	Preussen	Ostpreussen	C	30,000 <= pop < 50,000	44596	364	1119	1037	82	21	5	1145	1058	26	2.2707424	
Altena	Altena	1927	Preussen	Westfalen	D	15,000 <= pop < 30,000	15931	131	240	226	14	5	0	245	231	5	2.0408163	
Altena	Altena	1928	Preussen	Westfalen	D	15,000 <= pop < 30,000	16333	163	244	229	15	11	2	257	240	13	5.0583658	
Altena	Altena	1929	Preussen	Westfalen	D	15,000 <= pop < 30,000	16464	139	257	248	9	6	1	264	254	7	2.6515152	
Altena	Altena	1930	Preussen	Westfalen	D	15,000 <= pop < 30,000	16498	139	242	220	22	16	1	259	236	17	6.5637064	
Altena	Altena	1931	Preussen	Westfalen	D	15,000 <= pop < 30,000	16407	138	196	186	10	5	2	203	191	7	3.4482758	
Altena	Altena	1932	Preussen	Westfalen	D	15,000 <= pop < 30,000	16115	122	192	176	16	9	0	201	185	9	4.477612	
Altena	Altena	1933	Preussen	Westfalen	D	15,000 <= pop < 30,000	16133	138	189	174	15	6	1	196	180	7	3.5714285	
Altena	Altena	1934	Preussen	Westfalen	D	15,000 <= pop < 30,000	16246	162	258	223	35	8	2	268	231	10	3.7313433	

Which language to choose?

- *Anything* can be done in *any* language. Convenience varies.
- Concepts and toolkits transfer easily most of the time.
- Trade-off: Prior knowledge vs. task suitability.
- Never re-invent the wheel.

- Choose a high-level, dynamic language. Ideally free and open source.
- Specialized languages: R, MATLAB, Octave, Gauss, Julia, …
- General-purpose languages: Python, Perl, Ruby, …
- Choice is use case- and taste-specific. Popular is typically better.

R

- R is the major statistical programming language.
- It is free, used in many sciences and in industry. Good documentation.
- New models are typically first published and implemented in R.
- Having data processing and analysis in the same language is nice.
- Good library support for common tools (e.g. databases, regular expressions).
- Specific tasks for which high-level wrapper functions are not available may be very cumbersome.
- In recent years, R development has been very active and libaries exist for almost anything.

Python

- General-purpose programming language, supports object-oriented programming.
- Reads like english. Explicit and clear. Whitespace matters, no braces. (*“There should be one obvious way to do it”*.)
- Used extensively in industry and sciences. Good documentation.
- Libraries for almost anything.
- Many science-related libraries exist for other languages, but rarely are they as mature.
- Less support for statistical modeling (but growing).
- Less suited for interactive data work.

Recommendation

- Research ex ante which libraries are most mature and best for solving your specific problem.
- Focus on getting things done. Utilize prior knowledge.
- Rule-of-thumb:
  - Simple data processing:
    Stick with R. Augment with other tools where required.
  - More involved projects:
    Go with Python. You can still analyze data in R.
- I am proposing a mix of R, Python and Bash (Unix-Shell),
  . . . and whatever program your co-authors are using.
- R, Python and SQL are highly valued on the job market, knowing your way around a terminal is useful.
- This course uses R, but I will provide some equivalent python code.

Why not Stata, Matlab, Gauss or similar?

- Advantage: Many domain-specific models supported.
- Less support for almost anything else.
- Much less flexible for anything not to do with data analysis or numerics.
- Difficult to deploy on a server. Often tied to a GUI.
- Less popular, smaller userbase. Proprietary and expensive.
- You can still rely on them for estimation after your data is clean.

Why not Perl or Ruby?

Perl

- *“There’s more than one way to do it.”*
- Lots of special cases, reliance on hidden magic, bad readability.
- You may want to work together with somebody else.
- You may want to understand your own code in a few months time.
- Less popular in sciences.

Ruby

- Everything is an object. Intuitive.
- Different focus.
- Even less popular in sciences.
- Less support.

A few things to get started

- What you need for this course:
  - R.
  - A text editor or an IDE (like RStudio).

- You want to use Python:
  - A Python distribution (use Anaconda) (and possibly a shell).
  - A text editor or an IDE (like Spyder).
  - Which version, 2.7 or 3?
    - Python 3 if you plan to use Python regularly in the future.

- You want to use the command line interface (CLI) and have access to shell tools:
  - Terminal and Bash (Linux, MacOS), package manager (default on Linux, use iterm2 and homebrew on MacOS).
  - Cygwin, Windows Subsystem for Linux, Linux in a virtual machine, dual boot (Windows).

A note on operating systems

- MacOS or Linux offer built-in access to a Unix shell (Bash).
- Further software is managed via a package management system and distributed via software repositories.
- On Linux, use your package manager to install anything you require.
- On MacOS, familiarize yourself with Homebrew. Install iterm2 if you want a fancier terminal.

- For Windows, many tools are not available or cumbersome to use. Dependency resolution can be a nightmare.
- Windows does not provide proper access to a Unix shell.
- Even reliably installing Python was a chore until recently (now use Anaconda.)

Command line interpreters and shells

- A shell is an interface that lets you interact with your computer.
- A CLI using a programming language that allows you to execute programs and scripts.
- Unix-based operating systems (Linux, MacOS) have Bash pre-installed.
- Windows has cmd (or PowerShell). These are not a viable replacement. Cygwin or WSL may be. Git Bash is incomplete.

- Some examples:

```
cd somedir/subdir # navigate to a folder
cd .. # navigate to parent directory
cd # return to your home folder
ls # list directory contents
R # start the R console
```

- *CTRL + c* aborts a process, *CTRL + d* quits.

Examples

- Some examples:

```
vim myscript.r # edit your R script with vim
R -f myscript.r # execute your R script
python myscript.py # execute your python script
git add myscript.py # stage file for version control
git commit myscript.py # stage file for version control
man ssh # display manual pages for the ssh program
ssh myusername@13.438.14.673 # secure shell login to your remote serve
shutdown -h +20 # shut off computer in 20 minutes
```

- Sounds tedious? It can be. But it can also be extremely powerful.

- Convert all your pdf files in a folder to text and search them.

```
for file in *.pdf; do pdftotext "$file"; done
grep -icr "keyword" *.txt
```

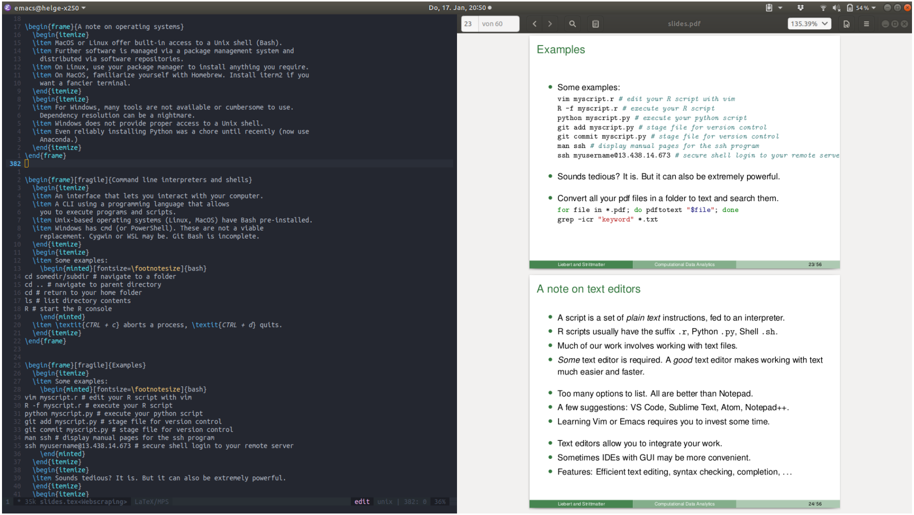
A note on text editors

- A script is a set of *plain text* instructions, fed to an interpreter.
- R scripts usually have the suffix `.r`, Python `.py`, Shell `.sh`.
- Much of our work involves working with text files.
- *Some* text editor is required. A *good* text editor makes working with text much easier and faster.

- Too many options to list. All are better than Notepad.
- A few options: VS Code, Sublime Text, Atom, Notepad++.
- Learning Vim or Emacs requires you to invest some time.

- Text editors allow you to integrate your work.
- Sometimes IDEs with GUI may be more convenient.
- Features: Efficient text editing, syntax checking, completion, ...

A possible setup



```
\begin{frame}{A note on operating systems}
  \begin{itemize}
  \item MacOS or Linux offer built-in access to a Unix shell (Bash).
  \item Further software is managed via a package management system and
    distributed via software repositories.
  \item On Linux, use your package manager to install anything you require.
  \item On MacOS, familiarize yourself with Homebrew. Install iterm2 if you
    want a fancier terminal.
  \end{itemize}
  \begin{itemize}
  \item For Windows, many tools are not available or cumbersome to use.
    Dependency resolution can be a nightmare.
  \item Windows does not provide proper access to a Unix shell.
  \item Even reliably installing Python was a chore until recently (now use
    Anaconda.)
  \end{itemize}
\end{frame}


\begin{frame}[fragile]{Command line interpreters and shells}
  \begin{itemize}
  \item An interface that lets you interact with your computer.
  \item A CLI using a programming language that allows
    you to execute programs and scripts.
  \item Unix-based operating systems (Linux, MacOS) have Bash pre-installed.
  \item Windows has cmd (or PowerShell). These are not a viable
    replacement. Cygwin or WSL may be. Git Bash is incomplete.
  \end{itemize}
  \begin{itemize}
  \item Some examples:
    \begin{minted}[fontsize=\footnotesize]{bash}
cd somedir/subdir # navigate to a folder
cd .. # navigate to parent directory
cd # return to your home folder
ls # list directory contents
R # start the R console
    \end{minted}
  \item \textit{CTRL + c} aborts a process, \textit{CTRL + d} quits.
  \end{itemize}
\end{frame}


\begin{frame}[fragile]{Examples}
  \begin{itemize}
  \item Some examples:
    \begin{minted}[fontsize=\footnotesize]{bash}
vim myscript.r # edit your R script with vim
R -f myscript.r # execute your R script
python myscript.py # execute your python script
git add myscript.py # stage file for version control
git commit myscript.py # stage file for version control
man ssh # display manual pages for the ssh program
ssh myusername@13.438.14.673 # secure shell login to your remote server
    \end{minted}
  \end{itemize}
  \begin{itemize}
  \item Sounds tedious? It is. But it can also be extremely powerful.
  \end{itemize}
  \begin{itemize}
```

```
1  * 35k slides.tex<Webscraping>  LaTeX/MPS                    edit  unix | 382: 0  36%
```

Examples

- Some examples:

```
vim myscript.r # edit your R script with vim
R -f myscript.r # execute your R script
python myscript.py # execute your python script
git add myscript.py # stage file for version control
git commit myscript.py # stage file for version control
man ssh # display manual pages for the ssh program
ssh myusername@13.438.14.673 # secure shell login to your remote serve
```

- Sounds tedious? It is. But it can also be extremely powerful.

- Convert all your pdf files in a folder to text and search them.

```
for file in *.pdf; do pdftotext "$file"; done
grep -icr "keyword" *.txt
```

Liebert and Strittmatter | Computational Data Analytics | 23/ 56

A note on text editors

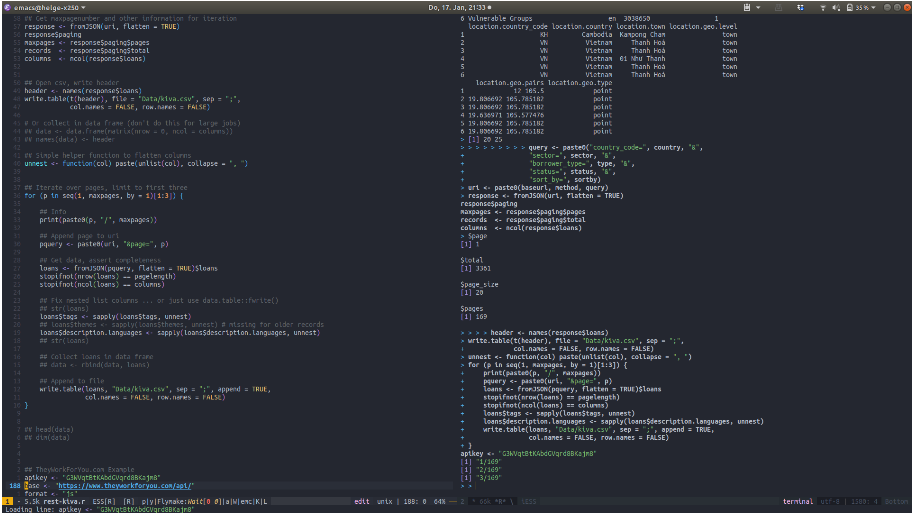
- A script is a set of *plain text* instructions, fed to an interpreter.
- R scripts usually have the suffix `.r`, Python `.py`, Shell `.sh`.
- Much of our work involves working with text files.
- *Some* text editor is required. A *good* text editor makes working with text much easier and faster.

- Too many options to list. All are better than Notepad.
- A few suggestions: VS Code, Sublime Text, Atom, Notepad++.
- Learning Vim or Emacs requires you to invest some time.

- Text editors allow you to integrate your work.
- Sometimes IDEs with GUI may be more convenient.
- Features: Efficient text editing, syntax checking, completion, . . .

Liebert and Strittmatter | Computational Data Analytics | 24/ 56

. . . that is universal

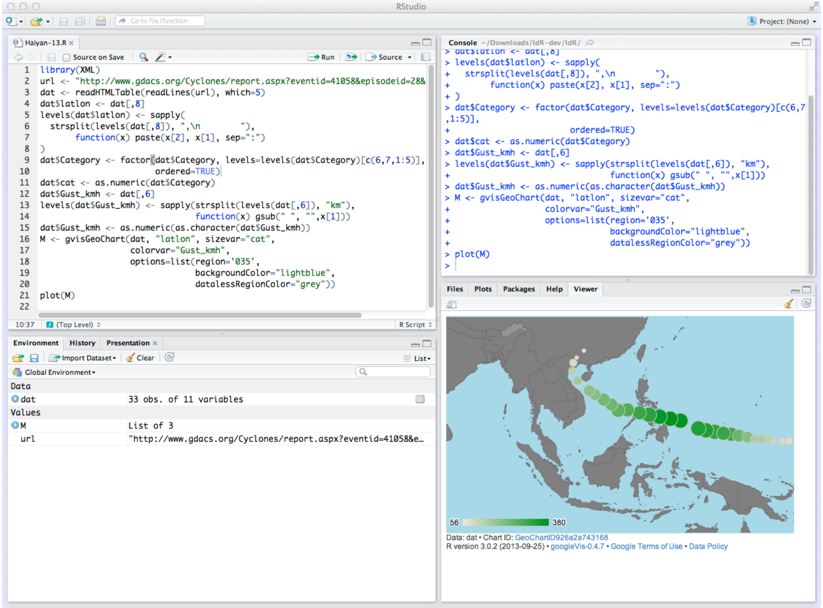

emacs@helge-x250
Do, 17. Jan, 21:33
35 %
1 - 5.5k rest-kiva.r ESS[R] [R] p|y|Flymake:Wait[0 0]|a|W|emc|K|L edit unix | 188: 0 64%
2 * 66k *R* \ iESS terminal utf-8 | 1580: 4 Bottom

```r
## Get maxpagenumber and other information for iteration
response <- fromJSON(uri, flatten = TRUE)
response$paging
maxpages <- response$paging$pages
records  <- response$paging$total
columns  <- ncol(response$loans)


## Open csv, write header
header <- names(response$loans)
write.table(t(header), file = "Data/kiva.csv", sep = ";",
            col.names = FALSE, row.names = FALSE)

# Or collect in data frame (don't do this for large jobs)
## data <- data.frame(matrix(nrow = 0, ncol = columns))
## names(data) <- header

## Simple helper function to flatten columns
unnest <- function(col) paste(unlist(col), collapse = ", ")


## Iterate over pages, limit to first three
for (p in seq(1, maxpages, by = 1)[1:3]) {

    ## Info
    print(paste0(p, "/", maxpages))

    ## Append page to uri
    pquery <- paste0(uri, "&page=", p)

    ## Get data, assert completeness
    loans <- fromJSON(pquery, flatten = TRUE)$loans
    stopifnot(nrow(loans) == pagelength)
    stopifnot(ncol(loans) == columns)

    ## Fix nested list columns ... or just use data.table::fwrite()
    ## str(loans)
    loans$tags <- sapply(loans$tags, unnest)
    ## loans$themes <- sapply(loans$themes, unnest) # missing for older records
    loans$description.languages <- sapply(loans$description.languages, unnest)
    ## str(loans)

    ## Collect loans in data frame
    ## data <- rbind(data, loans)

    ## Append to file
    write.table(loans, "Data/kiva.csv", sep = ";", append = TRUE,
                col.names = FALSE, row.names = FALSE)
}


## head(data)
## dim(data)



## TheyWorkForYou.com Example
apikey <- "G3WVqtBtKAbdGVqrd8BKajm8"
base <- "https://www.theyworkforyou.com/api/"
format <- "js"
```

```
Loading line: apikey <- "G3WVqtBtKAbdGVqrd8BKajm8"
```

```
6 Vulnerable Groups                    en  3038650                1
  location.country_code location.country location.town location.geo.level
1                    KH         Cambodia  Kampong Cham               town
2                    VN          Vietnam     Thanh Hoá               town
3                    VN          Vietnam     Thanh Hoá               town
4                    VN          Vietnam  01 Như Thanh               town
5                    VN          Vietnam     Thanh Hoá               town
6                    VN          Vietnam     Thanh Hoá               town
   location.geo.pairs location.geo.type
1            12 105.5             point
2 19.806692 105.785182             point
3 19.806692 105.785182             point
4 19.636971 105.577476             point
5 19.806692 105.785182             point
6 19.806692 105.785182             point
> [1] 20 25
> > > > > > > > > query <- paste0("country_code=", country, "&",
+                "sector=", sector, "&",
+                "borrower_type=", type, "&",
+                "status=", status, "&",
+                "sort_by=", sortby)
> uri <- paste0(baseurl, method, query)
> response <- fromJSON(uri, flatten = TRUE)
response$paging
maxpages <- response$paging$pages
records  <- response$paging$total
columns  <- ncol(response$loans)
> $page
[1] 1

$total
[1] 3361

$page_size
[1] 20

$pages
[1] 169

> > > > header <- names(response$loans)
> write.table(t(header), file = "Data/kiva.csv", sep = ";",
+             col.names = FALSE, row.names = FALSE)
> unnest <- function(col) paste(unlist(col), collapse = ", ")
> for (p in seq(1, maxpages, by = 1)[1:3]) {
+     print(paste0(p, "/", maxpages))
+     pquery <- paste0(uri, "&page=", p)
+     loans <- fromJSON(pquery, flatten = TRUE)$loans
+     stopifnot(nrow(loans) == pagelength)
+     stopifnot(ncol(loans) == columns)
+     loans$tags <- sapply(loans$tags, unnest)
+     loans$description.languages <- sapply(loans$description.languages, unnest)
+     write.table(loans, "Data/kiva.csv", sep = ";", append = TRUE,
+                 col.names = FALSE, row.names = FALSE)
+ }
apikey <- "G3WVqtBtKAbdGVqrd8BKajm8"
[1] "1/169"
[1] "2/169"
[1] "3/169"
> > 
```

Rstudio



```r
library(XML)
url <- "http://www.gdacs.org/Cyclones/report.aspx?eventid=41058&episodeid=28&
dat <- readHTMLTable(readLines(url), which=5)
dat$latlon <- dat[,8]
levels(dat$latlon) <- sapply(
  strsplit(levels(dat[,8]), ",\n         "),
      function(x) paste(x[2], x[1], sep=":")
)
dat$Category <- factor(dat$Category, levels=levels(dat$Category)[c(6,7,1:5)],
                      ordered=TRUE)
dat$cat <- as.numeric(dat$Category)
dat$Gust_kmh <- dat[,6]
levels(dat$Gust_kmh) <- sapply(strsplit(levels(dat[,6]), "km"),
                              function(x) gsub(" ", "",x[1]))
dat$Gust_kmh <- as.numeric(as.character(dat$Gust_kmh))
M <- gvisGeoChart(dat, "latlon", sizevar="cat",
                  colorvar="Gust_kmh",
                  options=list(region='035',
                              backgroundColor="lightblue",
                              datalessRegionColor="grey"))
plot(M)
```

```
> dat$latlon <- dat[,8]
> levels(dat$latlon) <- sapply(
+   strsplit(levels(dat[,8]), ",\n         "),
+       function(x) paste(x[2], x[1], sep=":")
+ )
> dat$Category <- factor(dat$Category, levels=levels(dat$Category)[c(6,7
,1:5)],
+                       ordered=TRUE)
> dat$cat <- as.numeric(dat$Category)
> dat$Gust_kmh <- dat[,6]
> levels(dat$Gust_kmh) <- sapply(strsplit(levels(dat[,6]), "km"),
+                               function(x) gsub(" ", "",x[1]))
> dat$Gust_kmh <- as.numeric(as.character(dat$Gust_kmh))
> M <- gvisGeoChart(dat, "latlon", sizevar="cat",
+                   colorvar="Gust_kmh",
+                   options=list(region='035',
+                               backgroundColor="lightblue",
+                               datalessRegionColor="grey"))
> plot(M)
>
```

Data

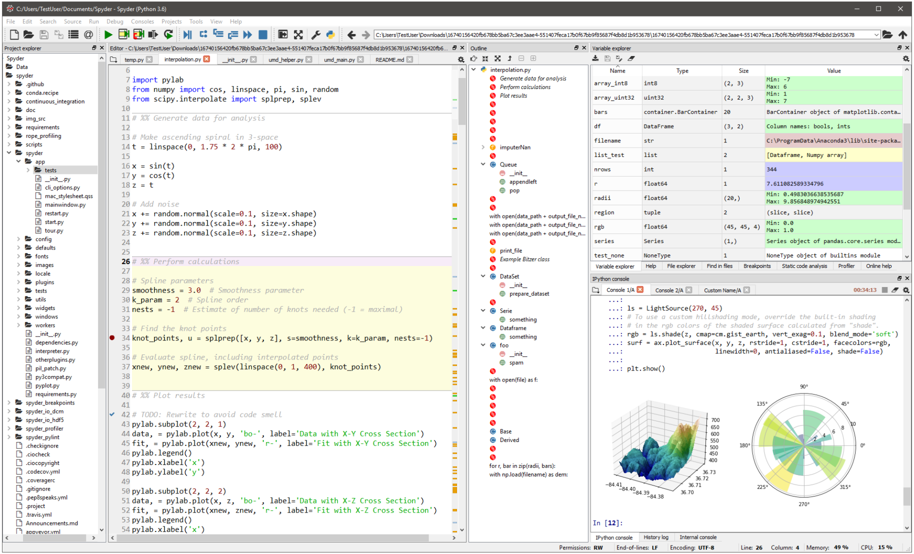
dat	33 obs. of 11 variables

Values

M	List of 3
url	"http://www.gdacs.org/Cyclones/report.aspx?eventid=41058&e…

Data: dat • Chart ID: GeoChartID926a2a743168
R version 3.0.2 (2013-09-25) • googleVis-0.4.7 • Google Terms of Use • Data Policy

Spyder



```python

import pylab
from numpy import cos, linspace, pi, sin, random
from scipy.interpolate import splprep, splev

# %% Generate data for analysis

# Make ascending spiral in 3-space
t = linspace(0, 1.75 * 2 * pi, 100)

x = sin(t)
y = cos(t)
z = t

# Add noise
x += random.normal(scale=0.1, size=x.shape)
y += random.normal(scale=0.1, size=y.shape)
z += random.normal(scale=0.1, size=z.shape)


# %% Perform calculations

# Spline parameters
smoothness = 3.0  # Smoothness parameter
k_param = 2  # Spline order
nests = -1  # Estimate of number of knots needed (-1 = maximal)

# Find the knot points
knot_points, u = splprep([x, y, z], s=smoothness, k=k_param, nests=-1)

# Evaluate spline, including interpolated points
xnew, ynew, znew = splev(linspace(0, 1, 400), knot_points)


# %% Plot results

# TODO: Rewrite to avoid code smell
pylab.subplot(2, 2, 1)
data, = pylab.plot(x, y, 'bo-', label='Data with X-Y Cross Section')
fit, = pylab.plot(xnew, ynew, 'r-', label='Fit with X-Y Cross Section')
pylab.legend()
pylab.xlabel('x')
pylab.ylabel('y')

pylab.subplot(2, 2, 2)
data, = pylab.plot(x, z, 'bo-', label='Data with X-Z Cross Section')
fit, = pylab.plot(xnew, znew, 'r-', label='Fit with X-Z Cross Section')
pylab.legend()
pylab.xlabel('x')
```

Name	Type	Size	Value
array_int8	int8	(2, 3)	Min: -7 Max: 6
array_uint32	uint32	(2, 2, 3)	Min: 1 Max: 7
bars	container.BarContainer	20	BarContainer object of matplotlib.conta…
df	DataFrame	(3, 2)	Column names: bools, ints
filename	str	1	C:\ProgramData\Anaconda3\lib\site-packa…
list_test	list	2	[Dataframe, Numpy array]
nrows	int	1	344
r	float64	1	7.611082589334796
radii	float64	(20,)	Min: 0.4983036638535687 Max: 9.856848974942551
region	tuple	2	(slice, slice)
rgb	float64	(45, 45, 4)	Min: 0.0 Max: 1.0
series	Series	(1,)	Series object of pandas.core.series mod…
test_none	NoneType	1	NoneType object of builtins module

```
   ...:
   ...: ls = LightSource(270, 45)
   ...: # To use a custom hillshading mode, override the built-in shading
   ...: # in the rgb colors of the shaded surface calculated from "shade".
   ...: rgb = ls.shade(z, cmap=cm.gist_earth, vert_exag=0.1, blend_mode='soft')
   ...: surf = ax.plot_surface(x, y, z, rstride=1, cstride=1, facecolors=rgb,
   ...:                        linewidth=0, antialiased=False, shade=False)
   ...:
   ...: plt.show()

In [12]:
```

Permissions: RW   End-of-lines: LF   Encoding: UTF-8   Line: 26   Column: 4   Memory: 49 %   CPU: 15 %

Version control

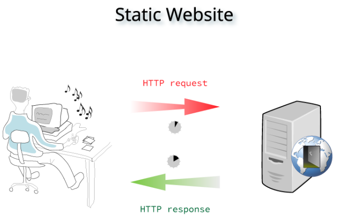
- Ubiquitous in IT, extremely useful in many purposes.
- Somewhat less useful for statistical data management and analysis due to different workflow. Do not version control data.
- Still good to know about and understand the basics.
- Git is the dominant version control software today.
- *ProGit* is a good and free resource. Skim the first few chapters.
- Sufficient to grasp the concept and know the basic commands.
- Lots of programs are hosted on public git repositories.

Getting started—things to consider before you begin

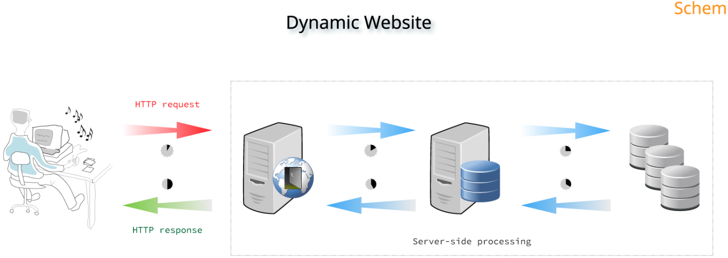
- Pick up the phone and try to get the data directly.
- Search if somebody has already faced the same or a similar problem.
- Does the site or service provide an API that you can access directly?
- Is there a wrapper for it?
- Is the website only online for a limited time? Do you want an original snapshot as a backup? Is it more convenient to filter your data offline?

Static vs. dynamic websites

Scheme A



Scheme B



Save an offline copy

- Use the shell utilities `wget` or `curl` to download the complete site.
- Also useful if you just want a set of files (e.g. pdf documents) from the same site directory.
- Convenient for static sites of limited size.
- Infeasible for large sites or sites that create content dynamically.

Examples

- Simple http GET request.

```
wget http://www.google.com
```

- Recursively download a website.

```
wget -r http://www.some-site.com/some-subdir/
```

- Download all pdfs from a site.

```
wget -r -A.pdf http://url-to-webpage-with-pdfs/
```

- Mirror a site offline and convert links for local browsing.

```
wget --mirror -p --convert-links -p ./local-dir
      http://target-website.com
```

Web APIs

- Data providers often offer Web APIs (*Application Programming Interface*) to access data.
- Allow programmable access to data via a defined set of HTTP messages. Similar to visiting a website: you specify a URL and information is sent to your machine.
- With a website, you receive code interpreted by your browser (HTML, CSS, JavaScript). With an API, you receive data.
- Usually in JSON (*JavaScript Object Notation*) or XML (*Extensible Markup Language*) format.

Web APIs

- Often just two steps:
  1. Construct the URL query that serves as the API request.
  2. Process the response message the API sends back.
- Examples:
  - `https://api.kivaws.org/v1/loans/newest.html`
  - `https://api.kivaws.org/v1/loans/newest.json`
  - `https://api.kivaws.org/v1/loans/search.json?sector=Agriculture&country=VN`
  - `https://www.theyworkforyou.com/api/getMPs?&key=someapikeyhere&output=js`
- Libraries may offer wrappers for APIs: `WDI`, `wbstats`, `twfy`, `pvsR`, Google Maps, OpenStreetMap/OSRM, . . .
- Sometimes it is possible to reverse engineer a site's internal API rather than scraping the HTML.

HTML and the Document Object Model

- Extracting information from the web requires a basic understanding of HTML and the associated Document Object Model (DOM).
- HTML elements provide the structure and content of web pages.
- Typically consist of `<start>` and `</end>` tags, with content in between.

  ```
  <tagname>Content here</tagname>
  ```

- A page consists of nested elements.
- The `html` element is the outer-most element, nesting the `head` and `body` elements, which in turn have nested elements.
- Nesting structure of elements can be represented by a tree (DOM).

Document Object Model

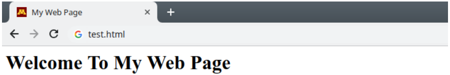
- The DOM is a programming interface for HTML and XML documents.
- Provides a structured representation of the document.
- A document as a group of nodes, each node representing a part of the document.
- Allows programmatic access to the tree to change the structure, style and content of the document.
- Connects web pages to scripts or programming languages.

A simple HTML page

A simple HTML page:

```
<html>
  <head>
    <title>My Web Page</title>
  </head>
  <body>
    <h1>Welcome To My Web Page</h1>
  </body>
</html>
```

How a browser renders this page:



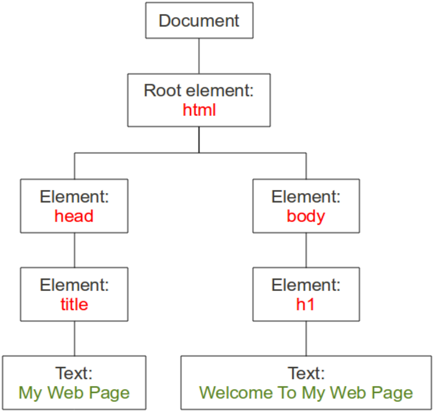
Welcome To My Web Page

HTML and the DOM

A simple HTML page:

```
<html>
  <head>
    <title>My Web Page
    </title>
  </head>
  <body>
    <h1>Welcome To My Web Page
    </h1>
  </body>
</html>
```

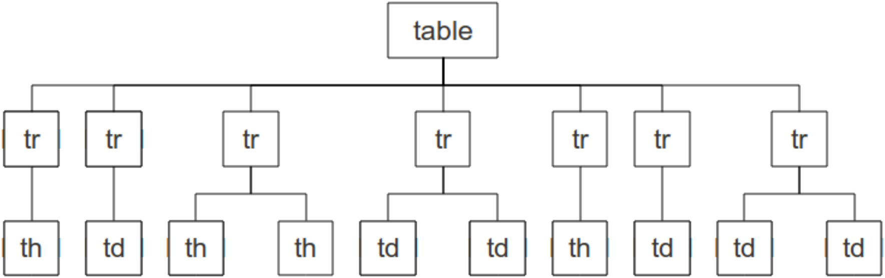
Corresponding node tree:



DOM node trees

- HTML DOM views a document as a tree structure called node tree.
- Everything in an HTML document is a node.
    - The entire document is a document node
    - Every HTML element is an element node
    - Every HTML attribute is an attribute node
    - Text content in the HTML elements is a text node
- Nodes can be accessed through the tree.
- Nodes may be assigned unique id attributes.

Example: An HTML table element



- Tables are represented by a top-level `table` element.
- The `table` element nests `tr` (*table row*) elements.
- These nest `th` (*table header*) and `td` (*table data*) element cells.

HTML and the DOM

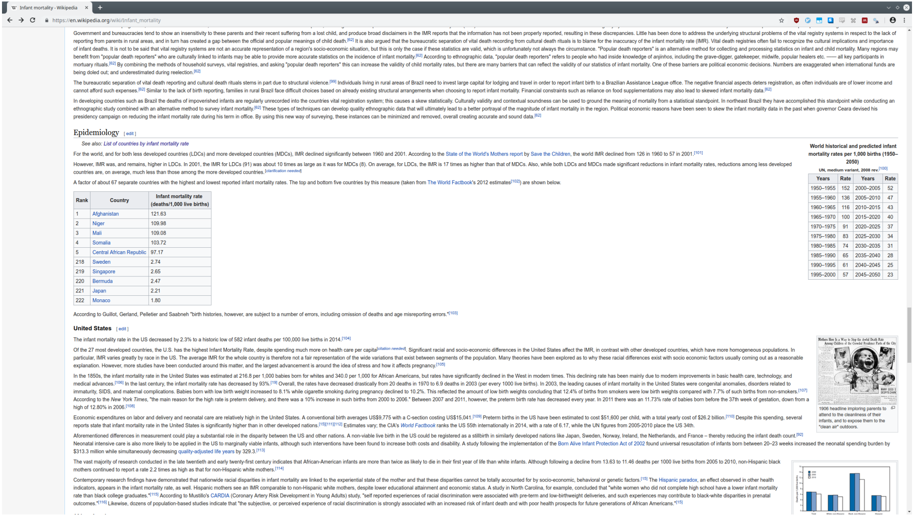
- HTML tags can have attributes and text content.

```
<tag attribute="value" attribute2="value">Text content.</tag>
```

- Example page:

```
<html>
  <head>
    <title>My Web Page</title>
  </head>
  <body>
    <h1>Welcome To My Web Page</h1>
    <img src="picture.png" alt="example photo" width="100">
    <a href="pagelink.html" id="pageid">Check this other page.</a>
  </body>
</html>
```

Data from the web



Infant mortality - Wikipedia
https://en.wikipedia.org/wiki/Infant_mortality
Mothers Here Is a Way to Stop the Awful Death Rate Among Children of the Crowded Residence Parts of the City
MANY GOOD S DESPITE CHANC
2000
2005
2010
Deaths per 1,000 live births
Total
White non-Hispanic
Black non-Hispanic
Hispanic

Government and bureaucracies tend to show an insensitivity to these parents and their recent suffering from a lost child, and produce broad disclaimers in the IMR reports that the information has not been properly reported, resulting in these discrepancies. Little has been done to address the underlying structural problems of the vital registry systems in respect to the lack of reporting from parents in rural areas, and in turn has created a gap between the official and popular meanings of child death.[62] It is also argued that the bureaucratic separation of vital death recording from cultural death rituals is to blame for the inaccuracy of the infant mortality rate (IMR). Vital death registries often fail to recognize the cultural implications and importance of infant deaths. It is not to be said that vital registry systems are not an accurate representation of a region's socio-economic situation, but this is only the case if these statistics are valid, which is unfortunately not always the circumstance. "Popular death reporters" is an alternative method for collecting and processing statistics on infant and child mortality. Many regions may benefit from "popular death reporters" who are culturally linked to infants may be able to provide more accurate statistics on the incidence of infant mortality.[62] According to ethnographic data, "popular death reporters" refers to people who had inside knowledge of *anjinhos*, including the grave-digger, gatekeeper, midwife, popular healers etc. —— all key participants in mortuary rituals.[62] By combining the methods of household surveys, vital registries, and asking "popular death reporters" this can increase the validity of child mortality rates, but there are many barriers that can reflect the validity of our statistics of infant mortality. One of these barriers are political economic decisions. Numbers are exaggerated when international funds are being doled out; and underestimated during reelection.[62]

The bureaucratic separation of vital death reporting and cultural death rituals stems in part due to structural violence.[99] Individuals living in rural areas of Brazil need to invest large capital for lodging and travel in order to report infant birth to a Brazilian Assistance League office. The negative financial aspects deters registration, as often individuals are of lower income and cannot afford such expenses.[62] Similar to the lack of birth reporting, families in rural Brazil face difficult choices based on already existing structural arrangements when choosing to report infant mortality. Financial constraints such as reliance on food supplementations may also lead to skewed infant mortality data.[62]

In developing countries such as Brazil the deaths of impoverished infants are regularly unrecorded into the countries vital registration system; this causes a skew statistically. Culturally validity and contextual soundness can be used to ground the meaning of mortality from a statistical standpoint. In northeast Brazil they have accomplished this standpoint while conducting an ethnographic study combined with an alternative method to survey infant mortality.[62] These types of techniques can develop quality ethnographic data that will ultimately lead to a better portrayal of the magnitude of infant mortality in the region. Political economic reasons have been seen to skew the infant mortality data in the past when governor Ceara devised his presidency campaign on reducing the infant mortality rate during his term in office. By using this new way of surveying, these instances can be minimized and removed, overall creating accurate and sound data.[62]

Epidemiology [edit]

See also: List of countries by infant mortality rate

For the world, and for both less developed countries (LDCs) and more developed countries (MDCs), IMR declined significantly between 1960 and 2001. According to the State of the World's Mothers report by Save the Children, the world IMR declined from 126 in 1960 to 57 in 2001.[101]

However, IMR was, and remains, higher in LDCs. In 2001, the IMR for LDCs (91) was about 10 times as large as it was for MDCs (8). On average, for LDCs, the IMR is 17 times as higher than that of MDCs. Also, while both LDCs and MDCs made significant reductions in infant mortality rates, reductions among less developed countries are, on average, much less than those among the more developed countries.[clarification needed]

A factor of about 67 separate countries with the highest and lowest reported infant mortality rates. The top and bottom five countries by this measure (taken from The World Factbook's 2012 estimates[102]) are shown below.

Rank	Country	Infant mortality rate (deaths/1,000 live births)
1	Afghanistan	121.63
2	Niger	109.98
3	Mali	109.08
4	Somalia	103.72
5	Central African Republic	97.17
218	Sweden	2.74
219	Singapore	2.65
220	Bermuda	2.47
221	Japan	2.21
222	Monaco	1.80

World historical and predicted infant mortality rates per 1,000 births (1950–2050)

UN, medium variant, 2008 rev.[100]

Years	Rate	Years	Rate
1950–1955	152	2000–2005	52
1955–1960	136	2005–2010	47
1960–1965	116	2010–2015	43
1965–1970	100	2015–2020	40
1970–1975	91	2020–2025	37
1975–1980	83	2025–2030	34
1980–1985	74	2030–2035	31
1985–1990	65	2035–2040	28
1990–1995	61	2040–2045	25
1995–2000	57	2045–2050	23

According to Guillot, Gerland, Pelletier and Saabneh "birth histories, however, are subject to a number of errors, including omission of deaths and age misreporting errors."[103]

United States [edit]

The infant mortality rate in the US decreased by 2.3% to a historic low of 582 infant deaths per 100,000 live births in 2014.[104]

Of the 27 most developed countries, the U.S. has the highest Infant Mortality Rate, despite spending much more on health care per capita[citation needed]. Significant racial and socio-economic differences in the United States affect the IMR, in contrast with other developed countries, which have more homogeneous populations. In particular, IMR varies greatly by race in the US. The average IMR for the whole country is therefore not a fair representation of the wide variations that exist between segments of the population. Many theories have been explored as to why these racial differences exist with socio economic factors usually coming out as a reasonable explanation. However, more studies have been conducted around this matter, and the largest advancement is around the idea of stress and how it affects pregnancy.[105]

1906 headline imploring parents to attend to the cleanliness of their infants, and to expose them to the "clean air" outdoors.

In the 1850s, the infant mortality rate in the United States was estimated at 216.8 per 1,000 babies born for whites and 340.0 per 1,000 for African Americans, but rates have significantly declined in the West in modern times. This declining rate has been mainly due to modern improvements in basic health care, technology, and medical advances.[106] In the last century, the infant mortality rate has decreased by 93%.[19] Overall, the rates have decreased drastically from 20 deaths in 1970 to 6.9 deaths in 2003 (per every 1000 live births). In 2003, the leading causes of infant mortality in the United States were congenital anomalies, disorders related to immaturity, SIDS, and maternal complications. Babies born with low birth weight increased to 8.1% while cigarette smoking during pregnancy declined to 10.2%. This reflected the amount of low birth weights concluding that 12.4% of births from smokers were low birth weights compared with 7.7% of such births from non-smokers.[107] According to the *New York Times*, "the main reason for the high rate is preterm delivery, and there was a 10% increase in such births from 2000 to 2006." Between 2007 and 2011, however, the preterm birth rate has decreased every year. In 2011 there was an 11.73% rate of babies born before the 37th week of gestation, down from a high of 12.80% in 2006.[108]

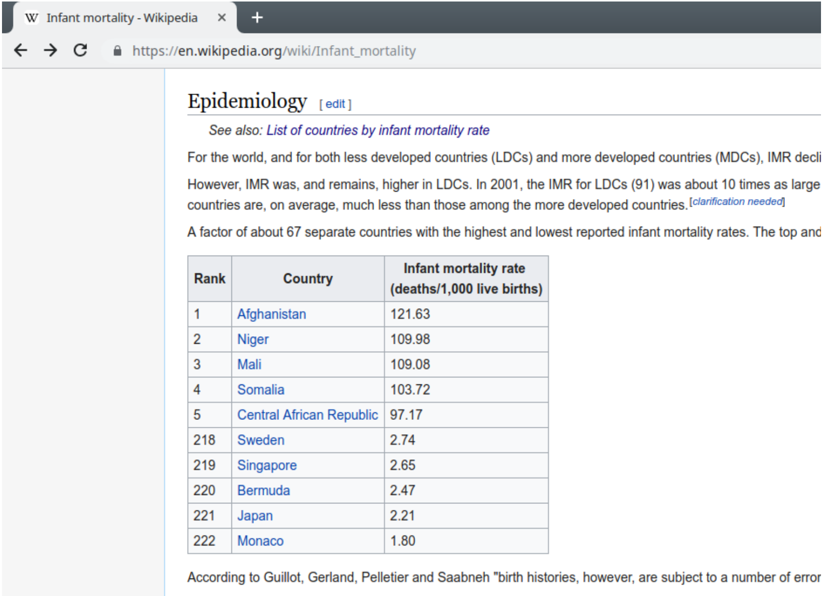
Economic expenditures on labor and delivery and neonatal care are relatively high in the United States. A conventional birth averages US$9,775 with a C-section costing US$15,041.[109] Preterm births in the US have been estimated to cost $51,600 per child, with a total yearly cost of $26.2 billion.[110] Despite this spending, several reports state that infant mortality rate in the United States is significantly higher than in other developed nations.[15][111][112] Estimates vary; the CIA's *World Factbook* ranks the US 55th internationally in 2014, with a rate of 6.17, while the UN figures from 2005-2010 place the US 34th.

Aforementioned differences in measurement could play a substantial role in the disparity between the US and other nations. A non-viable live birth in the US could be registered as a stillbirth in similarly developed nations like Japan, Sweden, Norway, Ireland, the Netherlands, and France – thereby reducing the infant death count.[92] Neonatal intensive care is also more likely to be applied in the US to marginally viable infants, although such interventions have been found to increase both costs and disability. A study following the implementation of the Born Alive Infant Protection Act of 2002 found universal resuscitation of infants born between 20–23 weeks increased the neonatal spending burden by $313.3 million while simultaneously decreasing quality-adjusted life years by 329.3.[113]

The vast majority of research conducted in the late twentieth and early twenty-first century indicates that African-American infants are more than twice as likely to die in their first year of life than white infants. Although following a decline from 13.63 to 11.46 deaths per 1000 live births from 2005 to 2010, non-Hispanic black mothers continued to report a rate 2.2 times as high as that for non-Hispanic white mothers.[114]

Contemporary research findings have demonstrated that nationwide racial disparities in infant mortality are linked to the experiential state of the mother and that these disparities cannot be totally accounted for by socio-economic, behavioral or genetic factors.[15] The Hispanic paradox, an effect observed in other health indicators, appears in the infant mortality rate, as well. Hispanic mothers see an IMR comparable to non-Hispanic white mothers, despite lower educational attainment and economic status. A study in North Carolina, for example, concluded that "white women who did not complete high school have a lower infant mortality rate than black college graduates."[115] According to Mustillo's CARDIA (Coronary Artery Risk Development in Young Adults) study, "self reported experiences of racial discrimination were associated with pre-term and low-birthweight deliveries, and such experiences may contribute to black-white disparities in prenatal outcomes."[116] Likewise, dozens of population-based studies indicate that "the subjective, or perceived experience of racial discrimination is strongly associated with an increased risk of infant death and with poor health prospects for future generations of African Americans."[15]

Wikipedia on infant mortality



W Infant mortality - Wikipedia ×
+
https://en.wikipedia.org/wiki/Infant_mortality

Epidemiology [edit]

See also: List of countries by infant mortality rate

For the world, and for both less developed countries (LDCs) and more developed countries (MDCs), IMR decli

However, IMR was, and remains, higher in LDCs. In 2001, the IMR for LDCs (91) was about 10 times as large countries are, on average, much less than those among the more developed countries.[*clarification needed*]

A factor of about 67 separate countries with the highest and lowest reported infant mortality rates. The top and

Rank	Country	Infant mortality rate (deaths/1,000 live births)
1	Afghanistan	121.63
2	Niger	109.98
3	Mali	109.08
4	Somalia	103.72
5	Central African Republic	97.17
218	Sweden	2.74
219	Singapore	2.65
220	Bermuda	2.47
221	Japan	2.21
222	Monaco	1.80

According to Guillot, Gerland, Pelletier and Saabneh "birth histories, however, are subject to a number of error

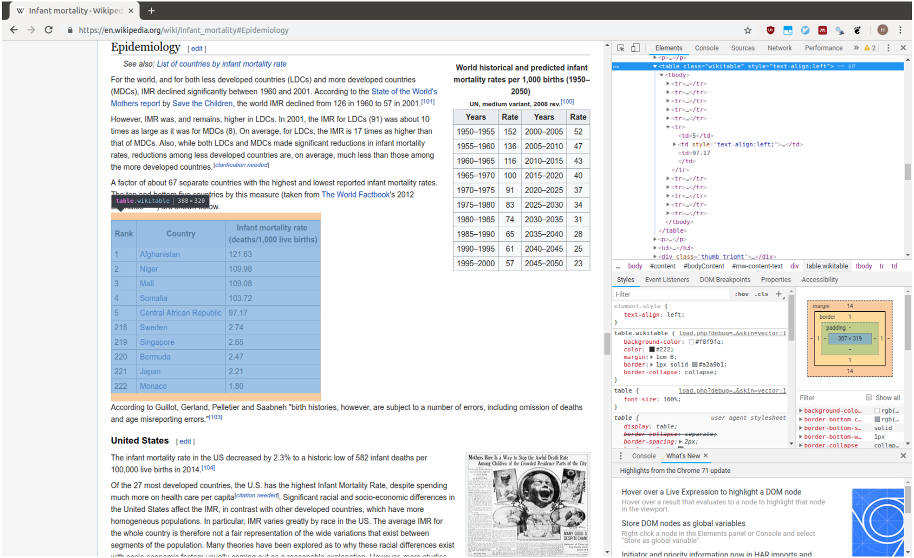
Fetching a table from Wikipedia

```r
library(rvest)

# 1) fetch and parse the website
page <- read_html("https://en.wikipedia.org/wiki/Infant_mortality")
# 2) extract the html node containing the table
table <- html_node(page,
                   xpath = "//*[@id='mw-content-text']/div/table[2]")
# 3) extract the table as a data frame
mrates <- html_table(table)
```

Inspecting the HTML source

- Convenient with modern browsers: Use the developer tools.
- Right-click *Inspect* (for Chrome there is also SelectorGadget).
- Look at the HTML source to grasp the structure.
- Find out how to navigate the site.
- Find the element(s) you want to extract.
- Get the Xpath expression or CSS selector to extract elements.

HTML elements visualized



Epidemiology [edit]

See also: List of countries by infant mortality rate

For the world, and for both less developed countries (LDCs) and more developed countries (MDCs), IMR declined significantly between 1960 and 2001. According to the State of the World's Mothers report by Save the Children, the world IMR declined from 126 in 1960 to 57 in 2001.[101]

However, IMR was, and remains, higher in LDCs. In 2001, the IMR for LDCs (91) was about 10 times as large as it was for MDCs (8). On average, for LDCs, the IMR is 17 times as higher than that of MDCs. Also, while both LDCs and MDCs made significant reductions in infant mortality rates, reductions among less developed countries are, on average, much less than those among the more developed countries.[clarification needed]

A factor of about 67 separate countries with the highest and lowest reported infant mortality rates. The top and bottom five countries by this measure (taken from The World Factbook's 2012 estimates) are shown below.

Rank	Country	Infant mortality rate (deaths/1,000 live births)
1	Afghanistan	121.63
2	Niger	109.98
3	Mali	109.08
4	Somalia	103.72
5	Central African Republic	97.17
218	Sweden	2.74
219	Singapore	2.65
220	Bermuda	2.47
221	Japan	2.21
222	Monaco	1.80

World historical and predicted infant mortality rates per 1,000 births (1950–2050)

UN, medium variant, 2008 rev.[100]

Years	Rate	Years	Rate
1950–1955	152	2000–2005	52
1955–1960	136	2005–2010	47
1960–1965	116	2010–2015	43
1965–1970	100	2015–2020	40
1970–1975	91	2020–2025	37
1975–1980	83	2025–2030	34
1980–1985	74	2030–2035	31
1985–1990	65	2035–2040	28
1990–1995	61	2040–2045	25
1995–2000	57	2045–2050	23

According to Guillot, Gerland, Pelletier and Saabneh "birth histories, however, are subject to a number of errors, including omission of deaths and age misreporting errors."[103]

United States [edit]

The infant mortality rate in the US decreased by 2.3% to a historic low of 582 infant deaths per 100,000 live births in 2014.[104]

Of the 27 most developed countries, the U.S. has the highest Infant Mortality Rate, despite spending much more on health care per capita[citation needed]. Significant racial and socio-economic differences in the United States affect the IMR, in contrast with other developed countries, which have more homogeneous populations. In particular, IMR varies greatly by race in the US. The average IMR for the whole country is therefore not a fair representation of the wide variations that exist between segments of the population. Many theories have been explored as to why these racial differences exist

Mothers Here Is a Way to Stop the Awful Death Rate Among Children of the Crowded Residence Parts of the City

```
▸<p>…</p>
▾<table class="wikitable" style="text-align:left"> == $0
  ▾<tbody>
    ▸<tr>…</tr>
    ▸<tr>…</tr>
    ▸<tr>…</tr>
    ▸<tr>…</tr>
    ▸<tr>…</tr>
    ▾<tr>
        <td>5</td>
      ▸<td style="text-align:left;">…</td>
        <td>97.17
        </td>
      </tr>
    ▸<tr>…</tr>
    ▸<tr>…</tr>
    ▸<tr>…</tr>
    ▸<tr>…</tr>
    ▸<tr>…</tr>
    </tbody>
  </table>
▸<p>…</p>
▸<h3>…</h3>
▸<div class="thumb tright">…</div>
```

… body #content #bodyContent #mw-content-text div table.wikitable tbody tr td

```
element.style {
  text-align: left;
}
table.wikitable {                load.php?debug=…&skin=vector:1
  background-color: □#f8f9fa;
  color: ■#222;
  margin:▸ 1em 0;
  border:▸ 1px solid ■#a2a9b1;
  border-collapse: collapse;
}
table {                          load.php?debug=…&skin=vector:1
  font-size: 100%;
}
table {                          user agent stylesheet
  display: table;
  ~~border-collapse: separate;~~
  border-spacing:▸ 2px;
```

margin 14, border 1, padding -, 387 × 319

```
▸ background-colo…   □rgb(…
▸ border-bottom-c…   ■rgb(…
▸ border-bottom-s…   solid
▸ border-bottom-w…   1px
▸ border-collapse    collap…
```

Highlights from the Chrome 71 update

Hover over a Live Expression to highlight a DOM node

Hover over a result that evaluates to a node to highlight that node in the viewport.

Store DOM nodes as global variables

Right-click a node in the Elements panel or Console and select "Store as global variable".

Initiator and priority information now in HAR imports and

Infant mortality rates from Wikipedia

```
<table class="wikitable" style="text-align:left">
  <tbody>
    <tr>
      <th>Rank</th>
      <th>Country</th>
      <th>Infant mortality rate <br> (deaths/1,000 live births)</th>
    </tr>
    <tr>
      <td>1</td>
      <td style="text-align:left;"><a href="/wiki/Afghanistan" title="Afghanistan">Afghanistan</a></td>
      <td>121.63</td>
    </tr>
    <tr>
      <td>2</td>
      <td style="text-align:left;"><a href="/wiki/Niger" title="Niger">Niger</a></td>
      <td>109.98</td>
    </tr>
    <tr>
      <td>3</td>
      <td style="text-align:left;"><a href="/wiki/Mali" title="Mali">Mali</a></td>
      <td>109.08</td>
    </tr>
    <tr>
      <td>4</td>
      <td style="text-align:left;"><a href="/wiki/Somalia" title="Somalia">Somalia</a></td>
      <td>103.72</td>
    </tr>
    ...
  </tbody>
</table>
```

CSS selectors and XPath expressions

```
# fetch and parse the website
page <- read_html("https://en.wikipedia.org/wiki/Infant_mortality")
# list the table nodes
html_nodes(page, "table")
# using xpath expressions or css selectors is equivalent
table <- html_node(page,
                   xpath = "//*[@id='mw-content-text']/div/table[2]")
table <- html_node(page,
                   css = "#mw-content-text > div > table:nth-child(121)")
```

The general structure

- There is no universal recipe. But most programs follow a certain structure.
  1. Open a website mimicking a browser and navigate it (optional).
  2. Get the page source HTML and feed it to a parser.
  3. Extract the elements you need.
  4. Filter and arrange them as needed and save them.
  5. Repeat 1.–4. until you have everything you want.
  6. Output your data.

Navigating to another page

```r
# Open infant mortality page
session <- html_session("https://en.wikipedia.org/wiki/Infant_mortality")
# Goto page on Somalia
session <- follow_link(session, "Somalia")
# Read the source
page <- read_html(session)
# Extract html
table <- html_node(page,
                   xpath = "//*[@id='mw-content-text']/div/table[4]")
regions <- html_table(table)
```

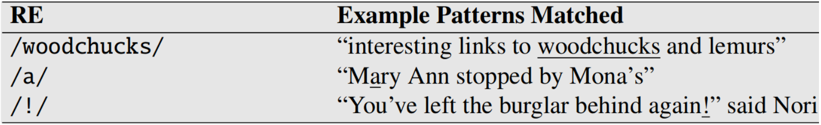
Filtering links

```
# read wiki page
page <- read_html("https://en.wikipedia.org/wiki/Infant_mortality")
# get the links
wikilinks <- html_attr(html_nodes(page, "a"), "href")
# use regex to filter internal links:
#   select only articles, no files or category pages,
#   matching with mortality or somalia
links <- grep("^(?!.*:)(/wiki/.*Mortality)|(/wiki/.*Somalia)", wikilinks,
              ignore.case = TRUE, value = TRUE, perl = TRUE)
links <- unique(links)
# go to first selected article page and process it
session <- jump_to(session, links[1])
page <- read_html(session)
html_nodes(page, "title")
```

Regular expressions

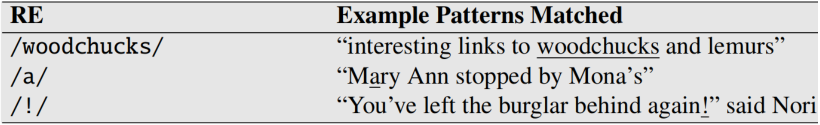
- Regular expressions are character sequences defining a search pattern.
- Usually used for find/replace operations on strings, or for validation.
- Regexes are an *extremely* helpful tool.
- Easy to grasp, complex to master.
- Pin a cheatsheet to your office wall.
- But: Regular expressions are not parsers. Always use a dedicated HTML parser to extract elements.

Regular expressions

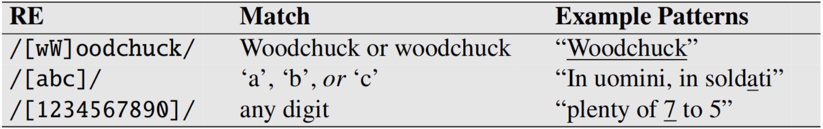


RE	Example Patterns Matched
`/woodchucks/`	“interesting links to woodchucks and lemurs”
`/a/`	“Mary Ann stopped by Mona’s”
`/!/`	“You’ve left the burglar behind again!” said Nori

Regular expressions

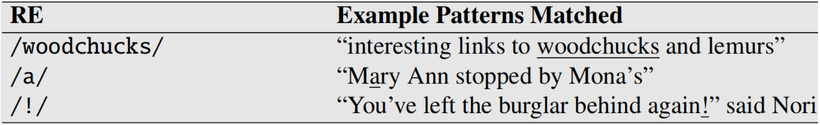


RE	Example Patterns Matched
`/woodchucks/`	“interesting links to woodchucks and lemurs”
`/a/`	“Mary Ann stopped by Mona’s”
`/!/`	“You’ve left the burglar behind again!” said Nori

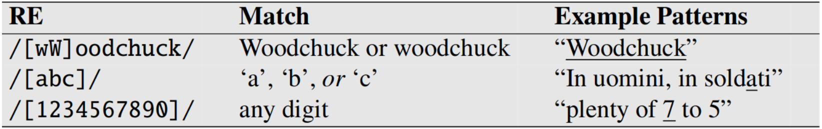


RE	Match	Example Patterns
`/[wW]oodchuck/`	Woodchuck or woodchuck	“Woodchuck”
`/[abc]/`	‘a’, ‘b’, *or* ‘c’	“In uomini, in soldati”
`/[1234567890]/`	any digit	“plenty of 7 to 5”

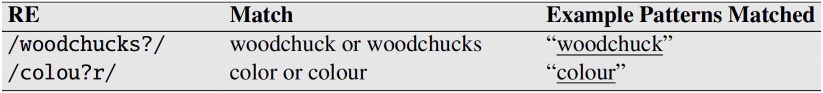
Regular expressions



RE	Example Patterns Matched
`/woodchucks/`	"interesting links to woodchucks and lemurs"
`/a/`	"Mary Ann stopped by Mona's"
`/!/`	"You've left the burglar behind again!" said Nori

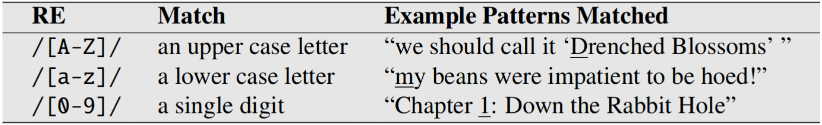


RE	Match	Example Patterns
`/[wW]oodchuck/`	Woodchuck or woodchuck	"Woodchuck"
`/[abc]/`	'a', 'b', *or* 'c'	"In uomini, in soldati"
`/[1234567890]/`	any digit	"plenty of 7 to 5"



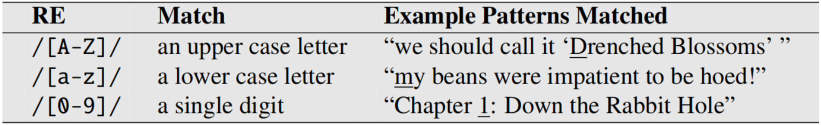
RE	Match	Example Patterns Matched
`/woodchucks?/`	woodchuck or woodchucks	"woodchuck"
`/colou?r/`	color or colour	"colour"

Regular expressions

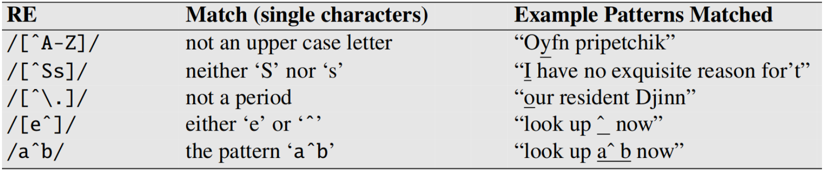


RE	Match	Example Patterns Matched
`/[A-Z]/`	an upper case letter	“we should call it ‘Drenched Blossoms’ ”
`/[a-z]/`	a lower case letter	“my beans were impatient to be hoed!”
`/[0-9]/`	a single digit	“Chapter 1: Down the Rabbit Hole”

Regular expressions



RE	Match	Example Patterns Matched
/[A-Z]/	an upper case letter	“we should call it ‘Drenched Blossoms’ ”
/[a-z]/	a lower case letter	“my beans were impatient to be hoed!”
/[0-9]/	a single digit	“Chapter 1: Down the Rabbit Hole”



RE	Match (single characters)	Example Patterns Matched
/[^A-Z]/	not an upper case letter	“Oyfn pripetchik”
/[^Ss]/	neither ‘S’ nor ‘s’	“I have no exquisite reason for’t”
/[^\.]/	not a period	“our resident Djinn”
/[e^]/	either ‘e’ or ‘^’	“look up ^ now”
/a^b/	the pattern ‘a^b’	“look up a^ b now”

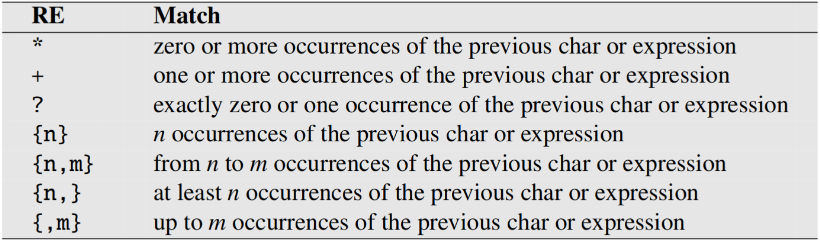
Regular expressions

RE	Match	Example Patterns Matched
/[A-Z]/	an upper case letter	“we should call it ‘Drenched Blossoms’ ”
/[a-z]/	a lower case letter	“my beans were impatient to be hoed!”
/[0-9]/	a single digit	“Chapter 1: Down the Rabbit Hole”

RE	Match (single characters)	Example Patterns Matched
/[^A-Z]/	not an upper case letter	“Oyfn pripetchik”
/[^Ss]/	neither ‘S’ nor ‘s’	“I have no exquisite reason for’t”
/[^\.]/	not a period	“our resident Djinn”
/[e^]/	either ‘e’ or ‘^’	“look up ^ now”
/a^b/	the pattern ‘a^b’	“look up a^ b now”

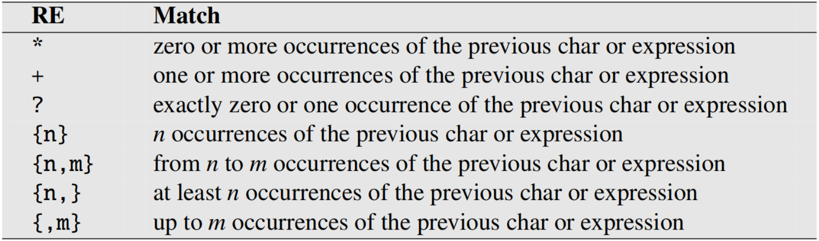
RE	Expansion	Match	First Matches
\d	[0-9]	any digit	Party␣of␣5
\D	[^0-9]	any non-digit	Blue␣moon
\w	[a-zA-Z0-9_]	any alphanumeric/underscore	Daiyu
\W	[^\w]	a non-alphanumeric	!!!!
\s	[␣\r\t\n\f]	whitespace (space, tab)	
\S	[^\s]	Non-whitespace	in␣Concord

Regular expressions

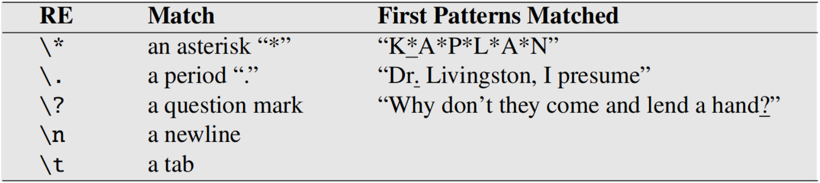


RE	Match
`*`	zero or more occurrences of the previous char or expression
`+`	one or more occurrences of the previous char or expression
`?`	exactly zero or one occurrence of the previous char or expression
`{n}`	*n* occurrences of the previous char or expression
`{n,m}`	from *n* to *m* occurrences of the previous char or expression
`{n,}`	at least *n* occurrences of the previous char or expression
`{,m}`	up to *m* occurrences of the previous char or expression

Regular expressions



RE	Match
*	zero or more occurrences of the previous char or expression
+	one or more occurrences of the previous char or expression
?	exactly zero or one occurrence of the previous char or expression
{n}	*n* occurrences of the previous char or expression
{n,m}	from *n* to *m* occurrences of the previous char or expression
{n,}	at least *n* occurrences of the previous char or expression
{,m}	up to *m* occurrences of the previous char or expression



RE	Match	First Patterns Matched
*	an asterisk "*"	"K*A*P*L*A*N"
\.	a period "."	"Dr. Livingston, I presume"
\?	a question mark	"Why don't they come and lend a hand?"
\n	a newline	
\t	a tab	

General remarks

- Start simple and expand your program incrementally.
- Keep it simple. Do not overengineer the problem.
- Do not repeat yourself. Code duplication implies bug reuse.
- Limit the number of iterations for test runs. Use print statements to inspect objects.
- Write tests to verify things work as intended.
- If the web page cannot be navigated easily or has hidden javascript, look into Selenium (`library(rselenium)`).
- If you scraper requires complex monitoring/validation procedures or threading for performance, look into Python.

Assignment

- Phillippine Statistics Authority *Good Governance Index*.
- Available at `http://nap.psa.gov.ph/ggi/default.asp`.
- Get all GGI data tables for all municipalities.
- Save them in a local data file for further analysis.
- Try for yourself. How would you go about this?

Assignment

- Submission deadline is next Monday, January 28.
- Submit code only, no data.
- Comment your code or submit a short description alongside explaining it.
- Add a short statement regarding your contribution if solved in a group.
- A proof-of-concept restricted to the first 30 municipalities is fine.
- Accounts for 20% of the final grade.
- I will provide a solution in R and Python after Monday.

Final remarks

- Sometimes small programs can go a long way.
- Do not lose sight of your ultimate goal. Time is valuable.
- Do not engage in perfectionism, focus on GTD.
- Identify everyday tasks that you can optimize.
- It might even be fun.

Appendix

Why Python?

```
>>> import this
The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
>>>
```

What (else) can Python be used for?

- Almost anything you can do in Stata, R, Gauss, Matlab or similar software. Library support is growing.
- Data management, analysis, numerics, graphs, structural modelling etc. (e.g. `scipy`, `pandas`, `numpy`, `matplotlib`, `seaborn`).
- Symbolic math (e.g. `sympy`, `Sage`).
- Geospatial work (e.g. `QGIS`).
- Text analysis and language processing (e.g. `nltk`, `spacy`, `gensim`).
- Create a website or blog (e.g. `django`, `hyde`, `sphinx`).
- Directly access many APIs (e.g. Twitter).
- Automate pretty much anything (e.g. experiments, data collection).

- In recent years, R has been extended to many of these domains.

Python resources

- Relevant modules.
  - `requests`, `bs4/BeautifulSoup`, `mechanize/mechanicalsoup`, `selenium`
  - `Scrapy` provides a complete framework for more complex projects.
  - `csv`, `re`, `pickle`, `pprint`, `pandas`, `random`, `itertools`, `pickle`, ...
- Learning the language.
  - *A byte of Python* is free. *The Quick Python Book* or *Dive into Python* offer a denser treatment.
  - O'Reilly: *Learning Python/Programming Python/Pocket reference*, *Web scraping with Python*.
  - *Automate the boring stuff with Python* for inspiration.
  - Plenty of video lectures and courses online. Stackoverflow helps.
  - (For Git: *ProGit* is free and really all you need.)
- Read about basic types, syntax, look at a few examples, then just have a go.