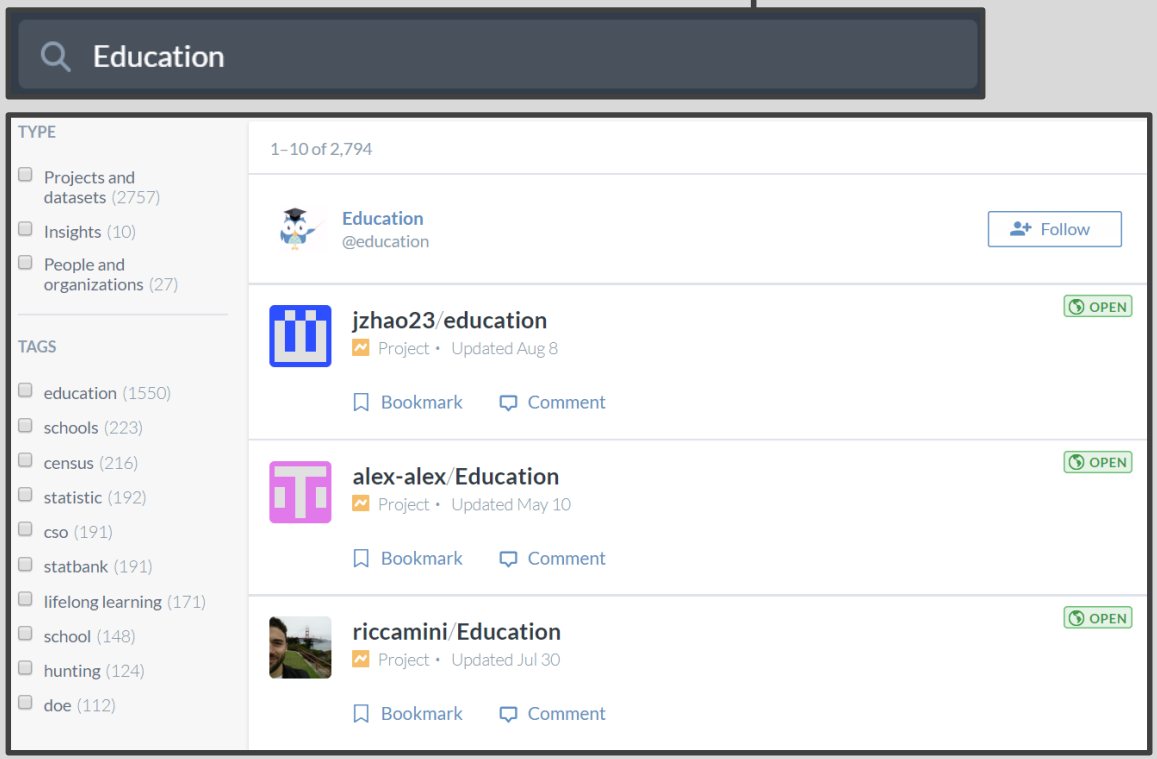
Data Import

Finding Data

- Built-in Datasets in R Packages
  - Example: NYC Flights
    - `>library(nycflights13)`
    - 5 Different Data Sets
  - More Comprehensive List
    - Vincent Arel-Bundock
    - Link
    - Packages
    - Data Name
    - Variable Information
    - CSV Links for Download
    - DOC Links for Details

Finding Data

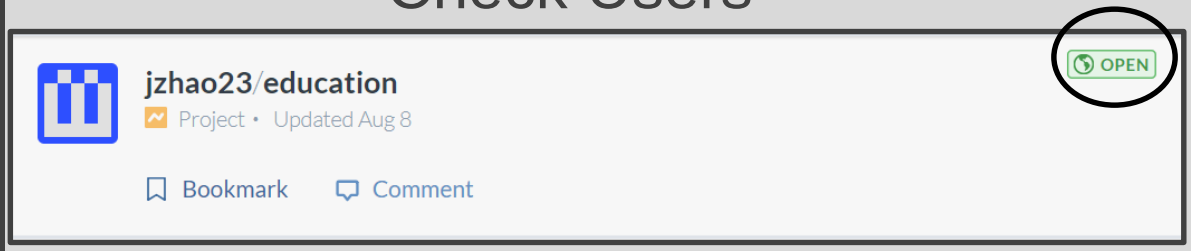
- Online Websites
  - Data.World
    - Requires Sign-up
    - Search for Topic

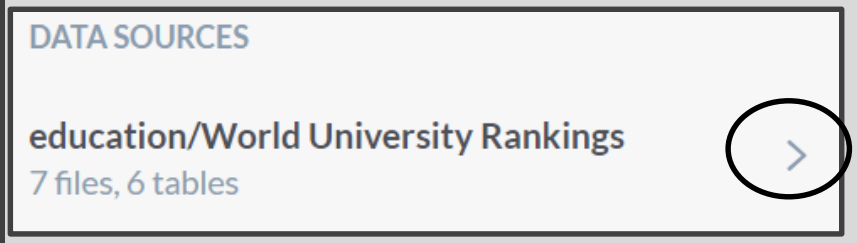


Finding Data

- Online Websites
  - Data.World
    - Select Projects/Datasets

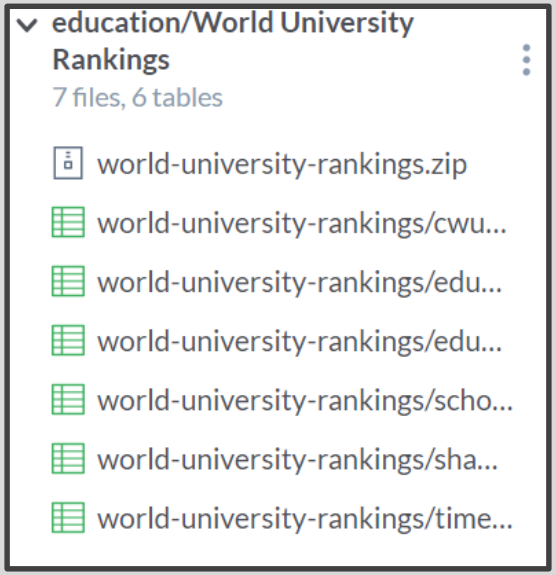
    - Check Users

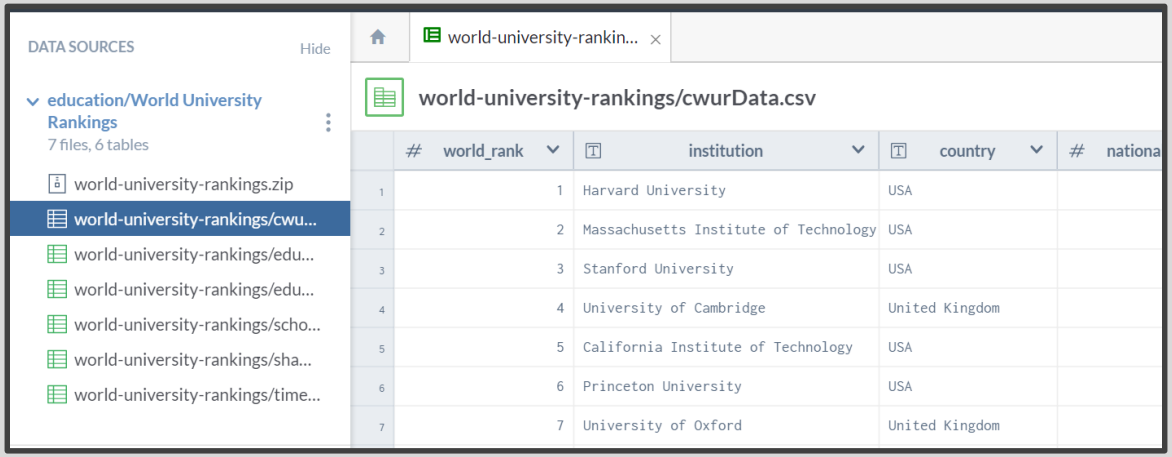




Finding Data

- Online Websites
  - Data.World
    - Inspect Data





	# world_rank	institution	country	# nationa
1	1	Harvard University	USA	
2	2	Massachusetts Institute of Technology	USA	
3	3	Stanford University	USA	
4	4	University of Cambridge	United Kingdom	
5	5	California Institute of Technology	USA	
6	6	Princeton University	USA	
7	7	University of Oxford	United Kingdom	

Finding Data

- Online Websites
  - Data.World
    - Read Data Dictionary

DOCUMENTS (1) Hide

Data dictionary

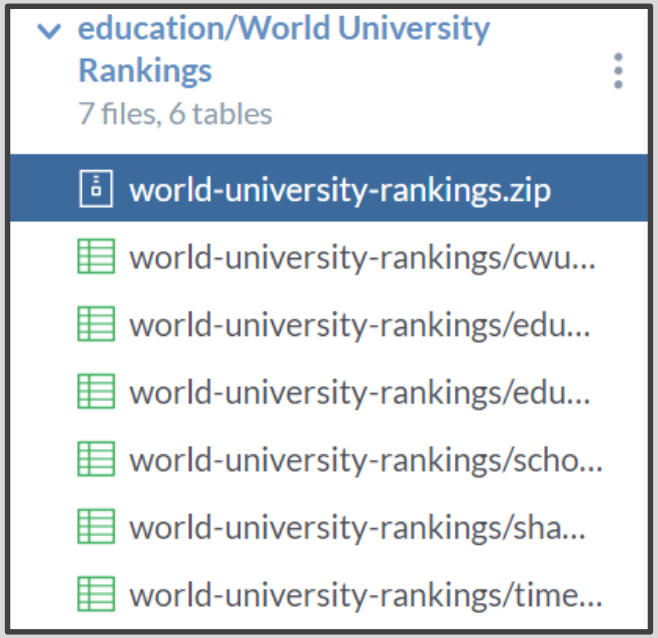
world-university-rankings/cwurData.csv

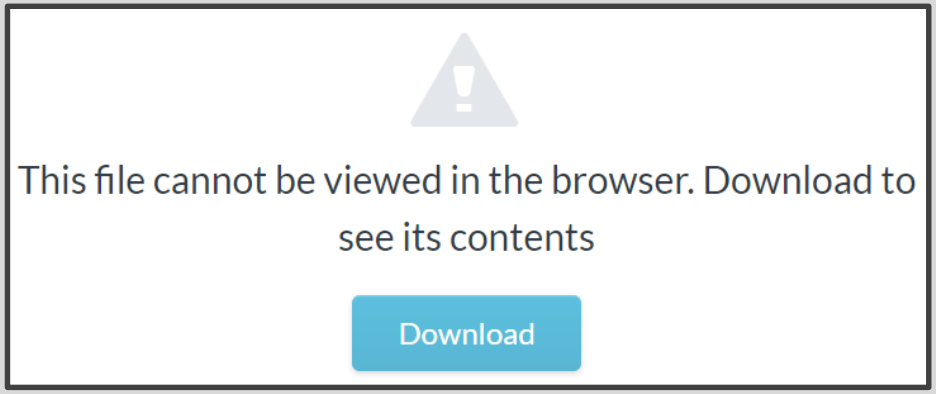
Request more info

- # world_rank
- institution
- country
- # national_rank
- # quality_of_education
- # alumni_employment
- # quality_of_faculty
- # publications
- # influence
- # citations
- # broad_impact
- # patents
- # score
- year

Finding Data

- Online Websites
  - Data.World
    - Download .zip Folder



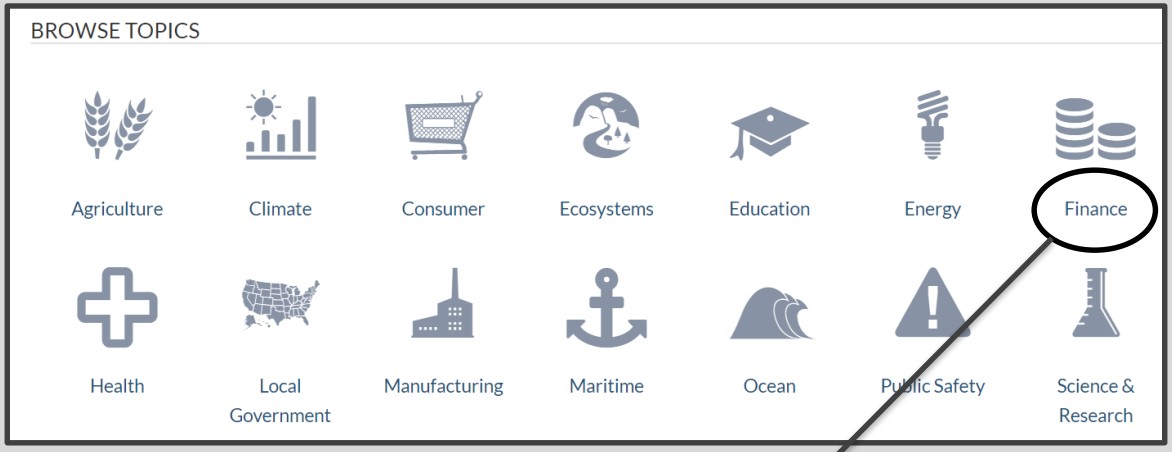


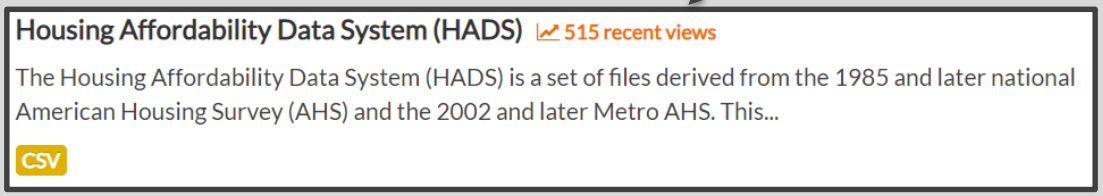
Finding Data

- Online Websites
  - Data.Gov
    - Logo… So Hot Right Now

DATA.GOV

    - Topics List





Finding Data

- Online Websites
  - Data.Gov
    - Check Description

Housing Affordability Data System (HADS)

Metadata Updated: March 8, 2017

The Housing Affordability Data System (HADS) is a set of files derived from the 1985 and later national American Housing Survey (AHS) and the 2002 and later Metro AHS. This system categorizes housing units by affordability and households by income, with respect to the Adjusted Median Income, Fair Market Rent (FMR), and poverty income. It also includes housing cost burden for owner and renter households. These files have been the basis for the worst case needs tables since 2001. The data files are available for public use, since they were derived from AHS public use files and the published income limits and FMRs. These dataset give the community of housing analysts the opportunity to use a consistent set of affordability measures.

Access & Use Information

Public: This dataset is intended for public access and use.

License: No license information was provided. If this work was prepared by an officer or employee of the United States government as part of that person's official duties it is considered a U.S. Government Work.

Downloads & Resources

Comma Separated Values File

hads.html

Link is ok ★☆☆☆☆ Openness score

Download

Finding Data

- Online Websites
  - Data.Gov
    - Find Documentation

Download the HADS documentation file (*.pdf, 159 KB)

The Housing Affordability Data System (HADS) is a set of housing-unit level datasets that measures the affordability of housing *units* and the housing cost burdens of *households*, relative to area median incomes, poverty level incomes, and Fair Market Rents. The purpose of these datasets is to provide housing analysts with consistent measures of affordability and burdens over a long period. The datasets are based on the American Housing Survey (AHS) national files from 1985 through 2009 and the metropolitan files from 2002 through 2009. Users can link records in HADS files to AHS records, allowing access to all of the AHS variables.

- Important Info About Data
  - Purpose of Data
  - Survey Data
  - Two Sets of Files
  - Years Included

Finding Data

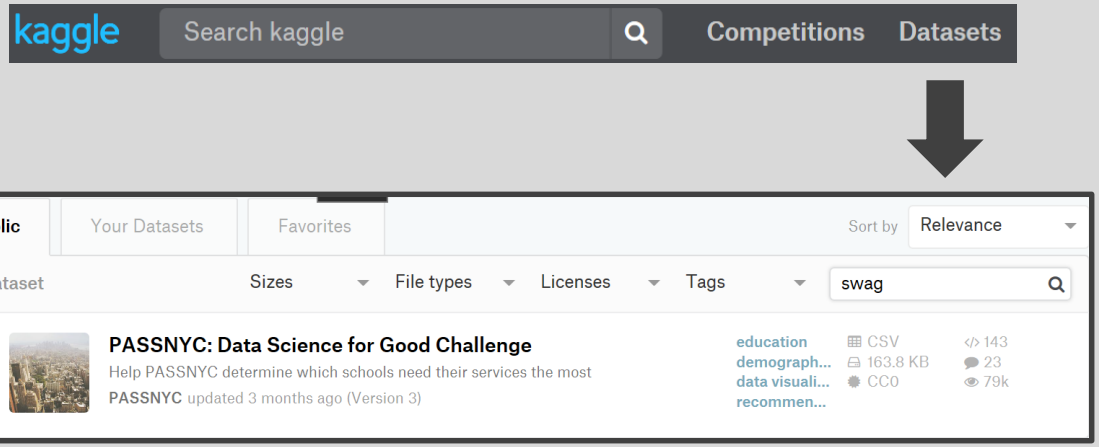
- Online Websites
  - Data.Gov
    - Download Links

HADS Data derived from AHS National Data		
Year	ASCII version	SAS version
2013	*.zip (11.3 MB)	*.zip (18.8 MB)
2011	*.zip (22.3 MB)	*.zip (28.6 MB)

HADS Data derived from AHS Metro Data		
Year	ASCII version	SAS version
2013	*.zip (9.4 MB)	*.zip (12.3 MB)
2009	Seattle Data (654 KB)	Seattle Data (727 KB)

Finding Data

- Online Websites
  - Kaggle
    - Requires Sign-up
    - Check Datasets



Finding Data

- Online Websites
  - Kaggle
    - Requires Sign-up
    - Overview and Question

Data | Overview | Kernels | Discussion | Activity

Overview

PASSNYC is a not-for-profit organization that facilitates a collective impact that is dedicated to broadening educational opportunities for New York City's talented and underserved students. New York City is home to some of the most impressive educational institutions in the world, yet in recent years, the City's specialized high schools - institutions with historically transformative impact on student outcomes - have seen a shift toward more homogeneous student body demographics.

PASSNYC uses public data to identify students within New York City's under-performing school districts and, through consulting and collaboration with partners, aims to increase the diversity of students taking the Specialized High School Admissions Test (SHSAT). By focusing efforts in under-performing areas that are historically underrepresented in SHSAT registration, we will help pave the path to specialized high schools for a more diverse group of students.

Problem Statement

PASSNYC and its partners provide outreach services that improve the chances of students taking the SHSAT and receiving placements in these specialized high schools. The current process of identifying schools is effective, but PASSNYC could have an even greater impact with a more informed, granular approach to quantifying the potential for outreach at a given school. Proxies that have been good indicators of these types of schools include data on English Language Learners, Students with Disabilities, Students on Free/Reduced Lunch, and Students with Temporary Housing.

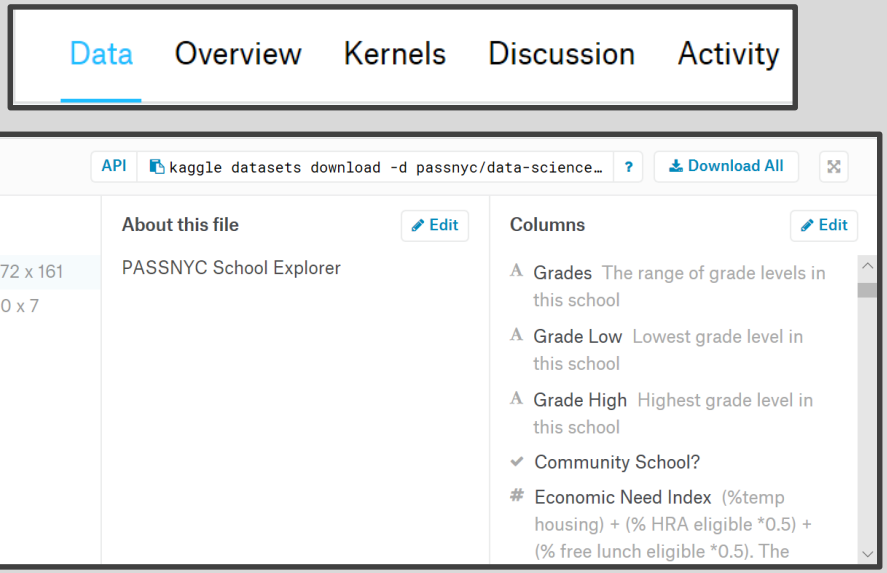
Part of this challenge is to assess the needs of students by using publicly available data to quantify the challenges they face in taking the SHSAT. The best solutions will enable PASSNYC to identify the schools where minority and underserved students stand to gain the most from services like after school programs, test preparation, mentoring, or resources for parents.

Submissions for the Main Prize Track will be judged based on the following general criteria:

- **Performance** - How well does the solution match schools and the needs of students to PASSNYC services? PASSNYC will not be able to live test every submission, so a strong entry will clearly articulate why it is effective at tackling the problem.
- **Influential** - The PASSNYC team wants to put the winning submissions to work quickly. Therefore a good entry will be easy to understand and will enable PASSNYC to convince stakeholders where services are needed the most.
- **Shareable** - PASSNYC works with over 60 partner organizations to offer services such as test preparation, tutoring, mentoring, extracurricular programs, educational consultants, community and student groups, trade associations, and more. Winning submissions will be able to provide convincing insights to a wide subset of these organizations.

Finding Data

- Online Websites
  - Kaggle
    - Requires Sign-up
    - Data Info and Download



File Types

- Read Chapter 8
  - Package for Importing

```
>library(readr)
```

  - Functions for Loading Data

- File Types
  - Different Delimiters
    - Comma, Tab, Space, Semicolon, Period
  - Different File Types
    - CSV – Comma
    - XLSX or XLS – Tab
    - TXT – Anything Possible
    - HTML – Anything Possible
  - Inspect Raw Data File

Data Import

- Importing CSV – Most Common
  - read_csv()

```{r}
UniRank=read_csv(file="D:/Mario Documents/UNC/STOR 320/STOR320_WEBSITE/Lecture/Lecture 11/Example/cwurData.csv",
                 col_names=T)
glimpse(UniRank)
```

```
Observations: 2,198
Variables: 14
$ world_rank           <int> 1, 2, 3, 4, 5, 6, 7, 8, 9...
$ institution          <chr> "Harvard University", "Ma...
$ country              <chr> "USA", "USA", "USA", "Uni...
$ national_rank        <int> 1, 2, 3, 1, 4, 5, 2, 6, 7...
$ quality_of_education <int> 7, 9, 17, 10, 2, 8, 13, 1...
$ alumni_employment    <int> 9, 17, 11, 24, 29, 14, 28...
$ quality_of_faculty   <int> 1, 3, 5, 4, 7, 2, 9, 12, ...
$ publications         <int> 1, 12, 4, 16, 37, 53, 15,...
$ influence            <int> 1, 4, 2, 16, 22, 33, 13, ...
$ citations            <int> 1, 4, 2, 11, 22, 26, 19, ...
$ broad_impact         <int> NA, NA, NA, NA, NA, NA, N...
$ patents              <int> 5, 1, 15, 50, 18, 101, 26...
$ score                <dbl> 100.00, 91.67, 89.50, 86....
$ year                 <int> 2012, 2012, 2012, 2012, 2...
```

- File Path Requires “/”
- Auto Use of Column Names
- Autodetects Variable Types

Data Import

- Importing CSV – Most Common

```
SHSAT=read_csv(file="Example/D5 SHSAT Registrations and Testers.csv")
```

```
## Parsed with column specification:
## cols(
##   DBN = col_character(),
##   `School name` = col_character(),
##   `Year of SHST` = col_integer(),
##   `Grade level` = col_integer(),
##   `Enrollment on 10/31` = col_integer(),
##   `Number of students who registered for the SHSAT` = col_integer(),
##   `Number of students who took the SHSAT` = col_integer()
## )
```

Autodetect Info

```
glimpse(SHSAT)
```

```
## Observations: 140
## Variables: 7
## $ DBN                                          <chr> "05M046", "0...
## $ `School name`                                <chr> "P.S. 046 Ar...
## $ `Year of SHST`                               <int> 2013, 2014, ...
## $ `Grade level`                                <int> 8, 8, 8, 8, ...
## $ `Enrollment on 10/31`                        <int> 91, 95, 73, ...
## $ `Number of students who registered for the SHSAT` <int> 31, 26, 21, ...
## $ `Number of students who took the SHSAT`      <int> 14, 7, 10, 8...
```

Data Import

- Other Types
  - read_delim() for General
  - XLS or XLSX

```
>library(readxl)
```

- Always Check Tibble After Import
- Observe That Variable Types are What You Want
- Check Missing Values
  - See if NA's are Appropriately Recorded
  - Too Many NA's
  - Not Enough NA's
  - Crosscheck Raw Data and Data Documentation

Example

- HADS Data From Data.Gov

```{r,message=F}
Housing=read_csv(file="Example/thads2013n.txt")
head(Housing,5)
```

```
Housing=read_csv(file="Example/thads2013n.txt")
head(Housing,5)
```

```
## # A tibble: 5 x 99
##   CONTROL  AGE1 METRO3 REGION  LMED   FMR   L30   L50   L80  IPOV BEDRMS
##   <chr>   <int> <chr>  <chr>  <int> <int> <int> <int> <int> <int>  <int>
## 1 '10000~    82 '3'    '1'    73738   956 15738 26213 40322 11067      2
## 2 '10000~    50 '5'    '3'    55846  1100 17165 28604 45744 24218      4
## 3 '10000~    53 '5'    '3'    55846  1100 13750 22897 36614 15470      4
## 4 '10000~    67 '5'    '3'    55846   949 13750 22897 36614 13964      3
## 5 '10000~    26 '1'    '3'    60991   737 14801 24628 39421 15492      2
## # ... with 88 more variables: BUILT <int>, STATUS <chr>, TYPE <int>,
## #   VALUE <int>, VACANCY <int>, TENURE <chr>, NUNITS <int>, ROOMS <int>,
## #   WEIGHT <dbl>, PER <int>, ZINC2 <int>, ZADEQ <chr>, ZSMHC <int>,
## #   STRUCTURETYPE <int>, OWNRENT <chr>, UTILITY <dbl>, OTHERCOST <dbl>,
## #   COST06 <dbl>, COST12 <dbl>, COST08 <dbl>, COSTMED <dbl>, TOTSAL <int>
```

Example

- HADS Data From Data.Gov

```
Housing2=read_csv(file="Example/thads2013n.txt") %>%
          select(METRO3,REGION,VALUE,ASSISTED)
head(Housing2,5)
```

```
## # A tibble: 5 x 4
##   METRO3 REGION   VALUE ASSISTED
##   <chr>  <chr>    <int>    <int>
## 1 '3'    '1'      40000       -9
## 2 '5'    '3'     130000       -9
## 3 '5'    '3'     150000       -9
## 4 '5'    '3'     200000       -9
## 5 '1'    '3'         -6        0
```

That is to say, using DEGREE, METRO or METRO3, and REGION variables. METRO and METRO3 indicate whether a unit is in a central city, suburb, or outside a metropolitan area. Further subdivision is available for some survey years.

Errors or Missing Should Become NA

Housing cost burden is simply a household's monthly housing cost divided by its monthly income[12]. In particular, note that we *do not* use mortgage payment assumptions discussed in the "Housing Costs" section above when calculating burden.[13] Households with zero or negative income are given the special code of BURDEN = -1. Vacant units, not being households, have missing values for BURDEN.

URL to R

- Benefit: Don’t Need Data on PC
- Problem: Links Change
- Example

Music CSV Library

From the CORGIS Dataset Project

By Ryan Whitcomb (rwhit94@vt.edu)
Version 1, created 5-18-16
Tags: music, songs, artists, creativity, media

Overview

This library comes from the Million Song Dataset, which used a company called the Echo Nest to derive data points about one million popular contemporary songs. The Million Song Dataset is a collaboration between the Echo Nest and LabROSA, a laboratory working towards intelligent machine listening. The project was also funded in part by the National Science Foundation of America (NSF) to provide a large data set to evaluate research related to algorithms on a commercial size while promoting further research into the Music Information Retrieval field. The data contains standard information about the songs such as artist name, title, and year released. Additionally, the data contains more advanced information; for example, the length of the song, how many musical bars long the song is, and how long the fade in to the song was.

Downloads

Download all of the following files.

1. music.csv

Downloads

Download all of the following files.

1. mu

Open in new tab
Open in new window
Open in new InPrivate window
Save target as
Copy link
Add to reading list
Search the web for "music.csv"
Ask Cortana about "music.csv"
View source
Inspect element

Field Descriptio

Key	Comment
artist.hotttnesss	
artist.id	
artist.name	
artist_mbtags	
artist_mbtags_count	

URL to R

- Example

Field Descriptions

Key	List of...	Comment	Example Value
artist.hotttnesss	Real number		0.401997543
artist.id	String		"ARD7TVE1187B99BFB1"
artist.name	String		"Casual"
artist_mbtags	String		""
artist_mbtags_count	Real number		0.0

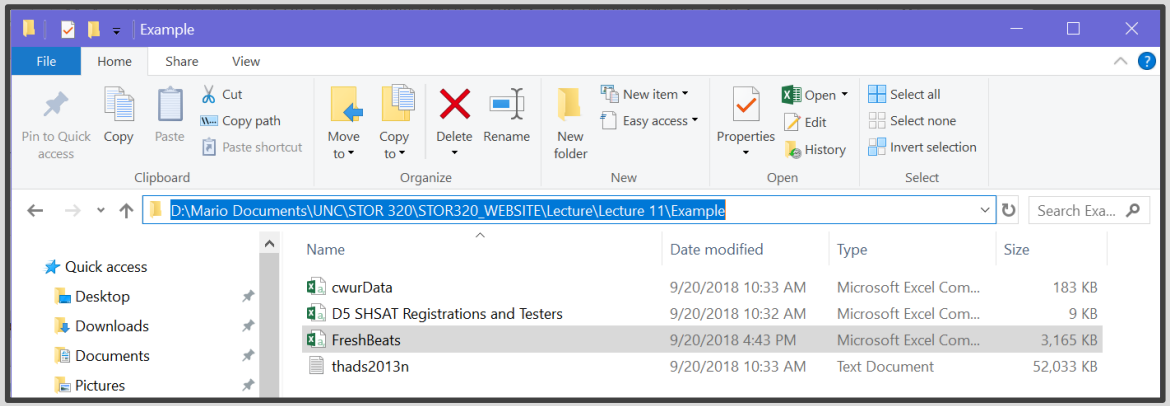
```
FreshBeats=read_csv(url("https://think.cs.vt.edu/corgis/csv/music/music.csv?forcedownloa
d=1"))

FreshBeats %>%
  filter(artist.name=="Word") %>%
  arrange(desc(artist.hotttnesss)) %>%
  select(artist.hotttnesss,artist.name,title)
```

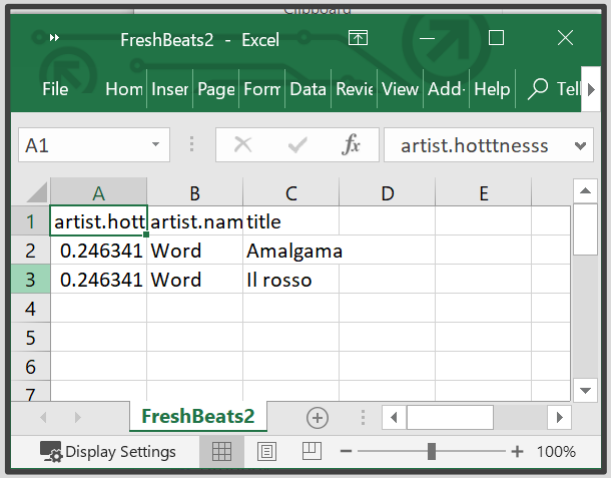
```
## # A tibble: 2 x 3
##   artist.hotttnesss artist.name title
##               <dbl> <chr>       <chr>
## 1             0.246 Word        Amalgama
## 2             0.246 Word        Il rosso
```

Writing Data

- write_csv()
  - Saves R Tibble to Computer

```{r}
setwd("Example")
write_csv(FreshBeats,"FreshBeats.csv")
```





Tibbles and Bits

- Read Chapter 7
  - Tibbles
  - Tribbles
  - You Tibble When You Tribble
- Subsetting Info

```
DATA=tribble(
  ~x, ~y, ~z,
  #--/--/--
  "a",2,3.6,
  "b",1,8.5
)
DATA
```

```
## # A tibble: 2 x 3
##   x         y     z
##   <chr> <dbl> <dbl>
## 1 a         2   3.6
## 2 b         1   8.5
```

```
#Extract by Variable Name
DATA$x
```

```
## [1] "a" "b"
```

```
DATA$":("
```

```
## [1] 3.6 8.5
```

```
DATA[["y"]]
```

```
## [1] 2 1
```

```
DATA[,c("x",":(")]
```

```
## # A tibble: 2 x 2
##   x       `:(`
##   <chr> <dbl>
## 1 a       3.6
## 2 b       8.5
```

```
#Extract by Location
DATA[[1]]
```

```
## [1] "a" "b"
```

```
DATA[,3]
```

```
## # A tibble: 2 x 1
##    `:(`
##   <dbl>
## 1   3.6
## 2   8.5
```

```
DATA[2,]
```

```
## # A tibble: 1 x 3
##   x         y  `:(`
##   <chr> <dbl> <dbl>
## 1 b         1   8.5
```

```
DATA[2,2:3]
```

```
## # A tibble: 1 x 2
##       y  `:(`
##   <dbl> <dbl>
## 1     1   8.5
```

Closing

Disperse and Make Reasonable Decisions