

JANUARY 2018



APPLIED ENGINEERING DATA ANALYSIS, OPTIMIZATION AND VISUALIZATION

Basic R

JOSHUA RHODES, PHD

Research Fellow/Adjunct Professor, The University of Texas at Austin



The University of Texas at Austin

WHAT STARTS HERE CHANGES THE WORLD

What is R?



Git* ready



*more on this later

Recommended playlist for coding*



PLAYLIST

Code For Hours

#programming #coding #focus #intense #studying #power #productivity #stu
#chill #notabug #feature -- My personal playlist for CompSci Coding/Learning S

Created by: **newtron54** • 727 songs, 51 hr 23 min

PAUSE FOLLOW ...

*optional

R you done? (yes)

- R is a system for statistical computation and graphics. It consists of a language plus a run-time environment with graphics, a debugger, access to certain system functions, and the ability to run programs stored in script files.
- FAQ: <https://cran.r-project.org/doc/FAQ/R-FAQ.html#What-is-R 003f>

R is a computer language that I find to be *easy* to learn/use

- Runs on all platforms
- Free
- Open source and well documented
- Lots of TACC support
- Good data visualization
- <https://stackoverflow.com/questions/tagged/r>

StackExchange

Search on Interpersonal Skills...

1

WHAT STARTS HERE CHANGES THE WORLD

StackExchange

Search on Personal Finance & Money...

personal finance & MONEY

QUESTIONS TAGS US

How far removed from source does one need to be to avoid Insider Trading?

As I tried to research and ponder [this question](#), I couldn't help but wonder:

How many steps (i.e., persons) removed does one need to be from a source of material/non-public information regarding a company to avoid accusations of [Insider trading](#)?

- For example, if the CEO tells a trader that tells his brother who tells his barber who tells his

Find a prime factor of 7999973 without a calculator

How would you go about finding prime factors of a number like 7999973? I have trivial knowledge about divisor-searching algorithms.

asked
viewed

Answers usually contain the code you need you just might have to Google it multiple times

stack overflow Questions Developer Jobs Tags Users Search...

I want to sort a data.frame by multiple columns. For example, with the data.frame below I would like to sort by column `z` (descending) then by column `b` (ascending):

```
dd <- data.frame(b = factor(c("Hi", "Med", "Hi", "Low"), levels = c("Low", "Med", "Hi"), ordered = TRUE), x = c("A", "D", "A", "C"), y = c(8, 3, 9, 9), z = c(1, 1, 1, 2))
dd
  b x y z
1 Hi A 8 1
2 Med D 3 1
3 Hi A 9 1
4 Low C 9 2
```

r sorting dataframe r-faq

share improve this question edited Jun 4 at 18:27 asked Aug 18 '09 at 21:33

 Jaap 39.5k 15 81 89

 Christopher DuBois 15.6k 20 54 90

- 1 Does anyone want to add an answer to do this programmatically. Say I want to choose one or more columns by position or column name. – [rmf](#) Dec 12 '15 at 20:07
- 2 @Roy added a couple of examples here: stackoverflow.com/a/35233592/1908452 – [info_seeker](#) Feb 5 '16 at 21:12

[add a comment](#)

15 Answers

active oldest votes

▲ You can use the `order()` function directly without resorting to add-on tools -- see this simpler answer which uses a trick right from the top of the `example(order)` code:

1279

```
R> dd[with(dd, order(-z, b)), ]
  b x y z
4 Low C 9 2
2 Med D 3 1
1 Hi A 8 1
3 Hi A 9 1
```

I recommend using the RStudio platform

- The free version is good for this class
 - And likely for your research
- Get it here: <https://www.rstudio.com/>

Crash course in programming using R

- Essentially a lot of engineering coding looks like this:
 - Read in some stuff from a file
 - Clean, sort, merge data
 - Do some analysis (business calls this “value add”)
 - Learn
 - Visualize the data
 - Write some other file

It rarely (never) goes that smooth the first time

- Many intermediate steps
 - Data cleaning
 - Data aggregation
 - Intermediate visualization

Scripts

```

1 clean_merge_EIA_860_923_data.R * make_EIA_data_lkert_chart.R * make_lkert_cap_fac.R * make_it_wind_chart_DELL.R *
2
3 <- read.csv('ercot_prices_other.csv')
4
5 it_col <- rgb(110/255,235/255,131/255)
6 col <- rgb(27/255,231/255,255/255)
7
8 (file = 'ercot_wind_v_it.pdf', height = 6, width = 9)
9
10 (mar = c(5.1, 4.1, 4.1, 4.1), oma = c(1,2,2,2))
11
12 it(e1$ercot_it ~ e1$Year, type = 'l', las = 1, ylab = '', xlab = 'Year', lwd = 3, col = it_col, yaxt = 'n', cex.lab = 1.5)
13 (side = 2, at = seq(from = 0, to = 3500, by = 500), col.ticks = it_col, col.axis = it_col, las = 1)
14 (side = 2, text = 'Number of ERCOT IT systems', line = 4, cex = 1.5)
15
16 (new = T)
17
18 it(e1$wind ~ e1$Year, type = 'l', col = wind.col, lwd = 3, vlab = '', vaxt = 'n', xaxt = 'n', ylim = c(0,18000))
12:126 make_it_wind_chart_DELL() R Script +

```

Environment

Console

```

Lasy All Other Solar Photovoltaic
[13] Landfill Gas Petroleum Coke
[15] Flywheels Wood/Wood Waste Biomass
[17] Hydroelectric Pumped Storage Other Gases
[19] Nuclear Other Natural Gas
[21] Geothermal Coal Integrated Gasification Combined Cycle
[23] Natural Gas with Compressed Air Storage Solar Thermal without Energy Storage
[25] Solar Thermal with Energy Storage Offshore Wind Turbine
26 Levels: All Other Batteries Coal Integrated Gasification Combined Cycle ... Wood/Wood Waste Biomass
> wind <- sum(m1$Net.Generation..Megawatthours.[m1$Technology == 'Onshore Wind Turbine'])
> sum(m1$Net.Generation..Megawatthours.[m1$Technology == 'Onshore Wind Turbine'])
[1] 226392754
> sum(m1$Net.Generation..Megawatthours.[m1$Technology == 'Onshore Wind Turbine' & m1$Operating.Year.avg > 2006])
[1] 200186313
> wind <- sum(m1$Net.Generation..Megawatthours.[m1$Technology == 'Onshore Wind Turbine' & m1$Operating.Year.avg > 2006])
> sum(m1$Net.Generation..Megawatthours.[m1$Technology == 'Onshore Wind Turbine' & m1$Operating.Year.avg > 2006])*23
[1] 4604285194
> sum(m1$Nameplate.Capacity[m1$Technology == 'Solar Photovoltaic', & m1$Operating.Year.avg == 2015])
Error: unexpected '&' in "sum(m1$Nameplate.Capacity[m1$Technology == 'Solar Photovoltaic', &
> sum(m1$Nameplate.Capacity[m1$Technology == 'Solar Photovoltaic' & m1$Operating.Year.avg == 2015])
Error: unexpected '=' in "sum(m1$Nameplate.Capacity[m1$Technology == 'Solar Photovoltaic' & m1$Operating.Year.avg ==
> sum(m1$Nameplate.Capacity[m1$Technology == 'Solar Photovoltaic' & m1$Operating.Year.avg == 2015])
[1] 2637.4
> sum(m1$Net.Generation..Megawatthours.[m1$Technology == 'Onshore Wind Turbine' & m1$Operating.Year.avg > 2005])
[1] 207655018
>

```

Command Line

Directory/plots

Help/etc.

Reading in some data

- A lot of data we use comes from excel files or CSV files
- There are others (non-exhaustive list):
 - Tab delimited
 - JSON (web)
 - XML (Extensible Markup Language, structured)
 - HTML (Hypertext Markup Language, web, structured)

R also has packages to read in data stored in other software formats

Package ‘foreign’

June 21, 2017

Priority recommended

Version 0.8-69

Date 2017-06-21

Title Read Data Stored by 'Minitab', 'S', 'SAS', 'SPSS', 'Stata',
'Systat', 'Weka', 'dBase', ...

rplexos v1.1.11 [Other versions](#)

by [Jef Daniels](#)

Read and Analyze 'PLEXOS' Solutions

Efficiently read and analyze 'PLEXOS' solutions by converting them into 'SQLite' databases divided into different time partitions, as well as the comparison across different scenarios. Energy Exemplar (see <<http://energyexemplar.com/software/plexos-desktop-edition>> for n

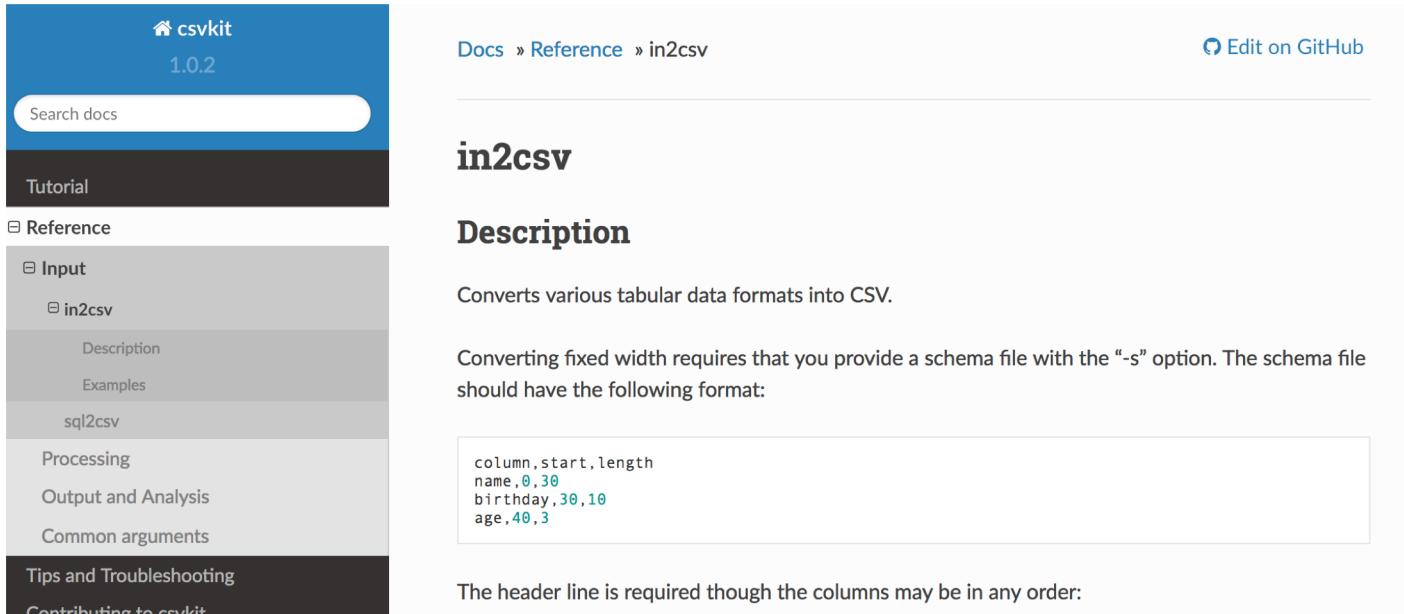
I find that 90% of my work is CSV, or Excel -> CSV

- Let's use an (I think) interesting example*
- Open Excel, grab some columns, save as CSV
- “gen_ercot_try.csv” file on box/board/whatever

The screenshot shows a Microsoft Excel window with a table of data. The table has columns for Interconnecting Entity, Point of Interconnection (POI), County, Fuel, Capacity to Grid (MW), and Projected COD Month/Year. A context menu is open over the table, showing options like Insert, Delete, Format, and Clear. Below the table, a 'Save As' dialog is open, showing the file name 'gen_ercot_try' and the file format 'CSV UTF-8 (Comma delimited) (.csv)'. The background shows a sidebar with 'data_analysis_examples' and a file tree with various presentation files.

Interconnecting Entity	Point of Interconnection (POI)	County	Fuel	Capacity to Grid (MW)	Projected COD Month/Year (as specified by the resource developer)
Irvin			WIND	7	12/2017 Incom
Irvin			WIND	6	12/2017 Incom
Nease			WIND	0	12/2017 Compe
Nease			WIND	0	12/2017 Compe
Leake			WIND	0	12/2017 Compe
Leake			WIND	7	12/2017 Compe
Leake			WIND	0	12/2017 Compe
Leake			WIND	32	2/2018 Incom
Leake			WIND	0	2/2018 Compe
Leake			WIND	0	3/2018 Incom
Leake			WIND	9	3/2018 Incom
Leake			WIND	0	3/2018 Incom
Leake			WIND	44	3/2018 Compe
Leake			WIND	0	3/2018 Compe
Leake			WIND	0	3/2018 Incom
Leake			WIND	44	3/2018 Compe
Leake			WIND	0	3/2018 Compe
Leake			WIND	100	10/2018 Incom
Leake			WIND	152	11/2018 Incom
Leake			WIND	102	12/2018 Incom
Leake			WIND	300	12/2018 Incom
Leake			WIND	400	12/2018 Incom
Leake			SOLAR	150	12/2018 Incom
Leake			WIND	300	12/2018 Incom
Leake			WIND	400	12/2018 Incom
Leake			WIND	122	12/2018 Incom
Leake			WIND	200	12/2018 Incom
Leake			WIND	200	12/2018 Compe
Leake			SOLAR	50	1/2019 Incom
Leake			SOLAR	45	1/2019 Incom
Leake			SOLAR	200	1/2019 Incom
Longroad E&I	Zapata	WIND	400		
Longroad E&I	Andrews	SOLAR	150		
SunChase P	Jim Hogg	WIND	300		
SunChase P	Crockett	WIND	400		
SunChase P	Hale	WIND	122		
SunChase P	Hale	WIND	200		
IA Table	La Salle	WIND	200		
IA Table	Schleicher	SOLAR	50		
IA Table	Hopkins	SOLAR	45		

Aside: there are command line (& R) packages that can auto convert Excel to CSV if you need to do that (be careful)



The screenshot shows a documentation page for the `in2csv` package within the `csvkit` library. The top navigation bar includes the `csvkit` logo, version 1.0.2, a search bar, and links for `Tutorial`, `Reference`, `Input`, `in2csv`, `sql2csv`, `Processing`, `Output and Analysis`, `Common arguments`, `Tips and Troubleshooting`, and `Contributing to csvkit`. The main content area has a breadcrumb trail: [Docs](#) » [Reference](#) » `in2csv`. On the right, there's a link to [Edit on GitHub](#). The title is **in2csv**. The **Description** section states: "Converts various tabular data formats into CSV." Below this, it says: "Converting fixed width requires that you provide a schema file with the `-s` option. The schema file should have the following format:" followed by a code block:

```
column,start,length
name,0,30
birthday,30,10
age,40,3
```

The footer note states: "The header line is required though the columns may be in any order."

Another aside on file/folder/directory hygiene

- NEVER use spaces in directory or file names
- my awesome file.csv -> my_awesome_file.csv
- Some systems (TACC) have trouble with spaces
- This will save you hours of headaches
 - You're welcome

REMINDER: On file names – **VERY IMPORTANT!!!!**

- If you send me (turn in a file named) this:
 - assignment.doc, or hwk3.csv
 - I will NOT even open it, you get 0%
- Always name files so you (and I) know what they are
 - RHODES_HWK3.pdf OR
 - RHODES_midterm_dataset.csv
- This way, I know who it is from and what it is w/o opening it

Navigate to folder with “gen_ercot_try.csv” file

R is a collaborative project with many contributors.

Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.

Type 'q()' to quit R.

```
> setwd('/Users/jdr2823/Box Sync/Teaching/Spring2018/Data_class/Lectures/data_analysis_examples')
> dir()
[1] "gen_ercot_try.csv"
> |
```

setwd('you/are/here') = set working directory to wherever you want to be

Reading in CSV data

```
> setwd('/Users/jdr2823/Box Sync/Teaching/Spring2018/Data_class/Lectures/data_analysis_examples')
> dir()
[1] "gen_ercot_try.csv"
> gen <- read.csv("gen_ercot_try.csv")
> head(gen)
  County Fuel cap_MW
1 Pecos WIND    7
2 Nolan WIND   6
3 Coke WIND    0
4 Taylor WIND   0
5 Taylor WIND   0
6 Nolan WIND    7
> |
```

- `head(x, n = 5)` is a function that shows you the first n (5 is default) rows of something
- `tail(x)` shows the last

There are other ways to read in data

- `read.table` – the actual call being made (`read.csv` is a wrapper, but helpful)
- `fread = 'fast read'` from `package(data.table)`
- If `read.csv` takes too long, look at faster options – they exist

“data” is plural

- Datum or dataset is singular
 - Correct: The data are
 - Incorrect: The data is
- The latter is akin to saying “the apples is”
 - Not correct, don’t do it

You should know your data, or at least have an idea of what your data are

- This dataset shows how much capacity of each type of power plant is trying to connect to the grid in each county in Texas (ERCOT)
- 3 rows, different types
 - County = string
 - FUEL = string
 - Cap_MW = integer

```
> setwd('/Users/jdr2823/Box Sync/Teaching')
> dir()
[1] "gen_ercot_try.csv"
> gen <- read.csv("gen_ercot_try.csv")
> head(gen)
  County Fuel cap_MW
1 Pecos WIND    7
2 Nolan WIND   6
3 Coke WIND    0
4 Taylor WIND   0
5 Taylor WIND   0
6 Nolan WIND    7
> |
```

Code: summary(x) is a good way to look at your data quickly

```
> summary(gen)
  County        Fuel      cap_MW
Pecos : 9    GAS  :10   Min.  : -27.0
Nolan : 8    SOLAR:84  1st Qu.: 90.0
Castro : 7   WIND :93  Median :180.0
Hidalgo : 7                    Mean   :179.4
Taylor  : 7                    3rd Qu.:202.0
Crockett: 6                  Max.   :1156.0
(Other) :143
> |
```

Wait, -27 MW? That doesn't seem to make sense... Go back and check, it does!

.64	20INR0030	Rustler Solar	Cypress Creek Renewables	tap 138kV 5827 Moore - 5895 Pearsall	Frio	SOLAR	113	6/2020	Incomplete
.65	20INR0002	Lake Spring Solar	8 Minute Energy	79501 Ogallala 345kV	Castro	SOLAR	400	6/2020	Incomplete
.66	19INR0030	San Bernard Solar	Sunchase Power	43340 Tx Gulf Sulphur 138kV	Wharton	SOLAR	100	6/2020	Incomplete
.67	20INR0025	Is 195	Innovative Solar	tap 138kV 5225 Hondo - 5813 Pearson	Medina	SOLAR	30	7/2020	Incomplete
.68	20INR0024	Is 177	Innovative Solar	5817 FerrisSw 69kV	Uvalde	SOLAR	25	7/2020	Incomplete
.69	20INR0021	Is 341	Innovative Solar	tap 69kV 3648 Granger - 3649 Taylor	Williamson	SOLAR	20	7/2020	Incomplete
.70	20INR0019	Trinity Hills Wind repowering	Bp	150751 & 150752 Trinity Hills 34.5kV	Young	WIND	-27	7/2020	Incomplete
.71	20INR0041	Watusi Solar	Cypress Creek Renewables		Menard	SOLAR	195	8/2020	Incomplete
.72	20INR0036	Stillwater Solar	Cypress Creek Renewables	tap 345kV 8455 LONHILL - 8955 NEDIN	Nueces	SOLAR	216	8/2020	Incomplete
.73	20INR0035	Angus Solar	Cypress Creek Renewables	tap 138kV 177 Bosque - 181 Cayote	Bosque	SOLAR	188	8/2020	Incomplete
.74	20INR0034	Cargil Solar	Cypress Creek Renewables	tap 69kV 8231 Uvalde - 8672 LaPryor	Zavala	SOLAR	98	8/2020	Incomplete
.75	20INR0023	Is 207	Innovative Solar	1906 Venus S 345kV	Ellis	SOLAR	60	8/2020	Incomplete
.76	20INR0022	Is 305	Innovative Solar	tap 138kV 6135 Paducah - 6320 Tardis	Cottle	SOLAR	30	8/2020	Incomplete
.77	20INR0016	Is 165	Innovative Solar	tap one 345kV 60501 Tesla - 60500 Edith Clark	Hardeman	SOLAR	45	8/2020	Incomplete
.78	20INR0015	Is 145	Innovative Solar	Tap 138kV 6135 Paducah Clare - 6117 E Mund	Cottle	SOLAR	45	8/2020	Incomplete
.79	20INR0014	Is 144	Innovative Solar	6131 Padr 69kV	Cottle	SOLAR	60	8/2020	Incomplete
.80	20INR0013	Is 137	Innovative Solar	tap 138kV 8234 Uvalde - 8241 Razorback	Uvalde	SOLAR	20	8/2020	Incomplete
.81	20INR0012	Is 75	Innovative Solar	Tap 138kV 6161 Paint Creek - 6200 Ft Phantom	Haskell	SOLAR	40	8/2020	Incomplete
.82	19INR0099b	Kontiki Wind	TriGlobal	59903 Bearkat 345kV	Glasscock	WIND	255	9/2020	Incomplete

Always be checking your data as you go

- If you make a mistake or your data are bad (not what you thought) they will propagate through your analysis
- Or you will realize it later and have wasted time
- Just check in every once in a while

In R we use data.frames a lot

- Fast and easy for most applications
- Data stored in RAM, so size limitations
 - “big data” packages for large size issues
 - Or figure out how to get code working with sample and use TACC’s TBs of RAM
- `read.csv` automatically creates a dataframe
 - For some applications you might want lists, matrices, ect.

You call a specific column of a data frame using the ‘\$’ operator: dataframe\$column

```
> summary(gen$County)
```

	Andrews	Armstrong	Baylor	Bee	Borden	Bosque	Brazoria	Brewster	Briscoe
	4	1	1	1	4	1	4	5	2
Brooks	1	Calhoun	Callahan	Cameron	Castro	Cherokee	Childress	Clay	Coke
	1	1	1	3	7	1	3	2	1
Concho	1	Cottle	Crane	Crockett	Crosby	Culberson	Deaf Smith	Dickens	Eastland
	3	3	3	6	2	1	2	1	1
Ector	1	Ellis	Fannin	Foard	Fort Bend	Frio	Glasscock	Gray	Grayson
	1	1	1	1	2	1	2	2	1
Hale	2	Hall	Hardeman	Harris	Haskell	Hidalgo	Hill	Hopkins	Hunt
	1	1	4	1	1	7	1	1	1
Irion	1	Jackson	Jim Hogg	Jim Wells	Kaufman	Kenedy	Kent	Knox	La Salle
	1	1	1	1	2	1	2	3	1
Limestone	2	Matagorda	Maverick	Medina	Menard	Mills	Mitchell	Nolan	Nueces
	1	1	1	1	3	1	1	8	1
Pecos	9	Randall	Reeves	Refugio	Runnels	San Patricio	Schleicher	Scurry	Starr
	2	2	1	1	1	3	1	2	2
Sterling	5	Taylor	Upton	Uvalde	Val Verde	Victoria	Webb	Wharton	Wilbarger
	7	7	3	2	1	3	1	2	1
Willacy	4	Williamson	Winkler	Young	Zapata	Zavala			
	1	1	3	1	2	3			

```
> max(gen$County)
```

```
Error in Summary.factor(cc(64L, 62L, 18L, 74L, 74L, 62L, 62L, 73L, 5L, :
  'max' not meaningful for factors
```

```
> max(gen$cap_MW)
```

```
[1] 1156
```

```
> |
```

Some helpful ways to slice ‘n dice data

```

> max(gen$cap_MW)
[1] 1156
> min(gen$cap_MW)
[1] -27
> mean(gen$cap_MW)
[1] 179.4011
> gen[gen$County == 'Menard',]
  County Fuel cap_MW
98 Menard WIND  200
133 Menard WIND  250
168 Menard SOLAR 195
> aggregate(x = gen$cap_MW, by = list(gen$Fuel), FUN = sum)
  Group.1   x
1     GAS 3554
2    SOLAR 13619
3    WIND 16375
> ?aggregate
> aggregate(x = gen[gen$County == 'Menard',], by = list(gen$Fuel), FUN = sum)
Error in aggregate.data.frame(x = gen[gen$County == "Menard", ], by = list(gen$Fuel), :
  arguments must have same length
> aggregate(x = gen[gen$County == 'Menard',], by = list(gen[gen$County == 'Menard',]$Fuel), FUN = sum)
Error in Summary.factor(59L, na.rm = FALSE) :
  'sum' not meaningful for factors
> aggregate(x = gen[gen$County == 'Menard',]$cap_MW, by = list(gen[gen$County == 'Menard',]$Fuel), FUN = sum)
  Group.1   x
1    SOLAR 195
2    WIND 450
> Menard <- gen[gen$County == 'Menard',]
> aggregate(x = Menard$cap_MW, by = list(Menard$Fuel), FUN = sum)
  Group.1   x
1    SOLAR 195
2    WIND 450
> |

```

R: Compute Summary Statistics of Data Subsets ▾ [Find in Topic](#)

aggregate {stats}

Compute Summary Statistics of Data Subsets

Description

Splits the data into subsets, computes summary statistics for each, and returns the result in a

Usage

```

aggregate(x, ...)

## Default S3 method:
aggregate(x, ...)

## S3 method for class 'data.frame'
aggregate(x, by, FUN, ..., simplify = TRUE, drop = TRUE)

## S3 method for class 'formula'
aggregate(formula, data, FUN, ...,
          subset, na.action = na.omit)

## S3 method for class 'ts'
aggregate(x, nfrequency = 1, FUN = sum, ndeltat = 1,
          ts.eps = getOption("ts.eps"), ...)

```

Arguments

x	an R object.
by	a list of grouping elements, each as long as the variables in the data frame x before use.

read.csv can also scrape web data

```
> data <- read.csv('https://www.wunderground.com/weatherstation/WXDailyHistory.asp?ID=KTXAUSTI942&month=1&day=16&year=2018&format=1', row.names = NULL)
> head(data)
  row.names Time TemperatureF DewpointF PressureIn WindDirection WindDirectionDegrees WindSpeedMPH
1 2018-01-16 00:08:00 36.9        33.7    30.48      ESE             113                 4                  9
2 <br> NA          NA        NA      NA           NA             NA                NA
3 2018-01-16 00:13:00 36.7        33.2    30.49      ESE             113                 8                  10
4 <br> NA          NA        NA      NA           NA             NA                NA
5 2018-01-16 00:26:00 36.5        33.3    30.50      SSE             158                 9                  9
6 <br> NA          NA        NA      NA           NA             NA                NA
  WindSpeedGustMPH Humidity HourlyPrecipIn Conditions Clouds dailyrainin SoftwareType DateUTC.br.
1          88   0.06        NA        NA   0.00  myAcuRite 2018-01-16 06:08:00            NA
2          NA     NA        NA        NA     NA            NA            NA
3          87   0.05        NA        NA   0.00  myAcuRite 2018-01-16 06:13:00            NA
4          NA     NA        NA        NA     NA            NA            NA
5          88   0.06        NA        NA   0.02  myAcuRite 2018-01-16 06:26:00            NA
6          NA     NA        NA        NA     NA            NA            NA
> |
```

I find the best way to code in R is to write function scripts

R markdowns and notebooks are cool, but won't work on TACC