post01-jean-ji

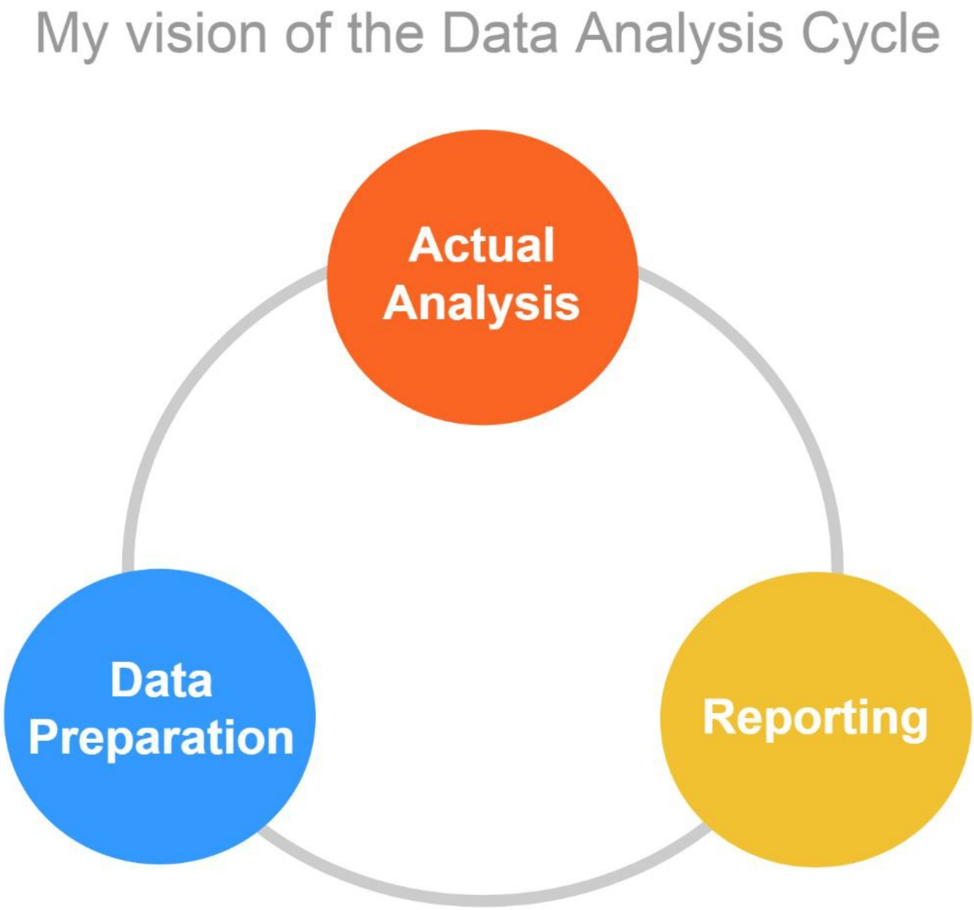
Jean Ji

October 31, 2017

Data Wrangling 201: how I applied data wrangling methods to my water data project

Introduction

You may have learned about "data wrangling" from our Statistics 133 class, and saw how it is part of our Data Analysis Cycle. In fact, data wrangling is a crucial step where analysts identify the "gaps" between the existing data formats and the desired formats for analyses. Today, I am going to convince you the importance of data wrangling and provide you with a concrete example of how I applied data wrangling techniques to help me understand a dataset better.



The Data Aanalysis Cycle

Background

Before we jump into the weeds of data wrangling techniques, let us review some background knowledge about what data wrangling is and how we have been using it in our class. First of all, the definition of data wrangling is quite general, meaning "the activity that you do on the raw data to make it 'clean' enough to input to your analytical algorithm" (Springboard.com). What this definition meant by "clean" data is actually something that we have been working on in class. For example, we learned to detect and correct any structurally-inconsistent records of data by unifying these data's units on the Excel spreadsheet before we load them into R. In addition, we have also learned about the coercion rule in R, where a vector in R is composed of data of the same type (e.g. numeric, logical, character etc.).

Name	Gender	Homeland	Birthyr	Mass	Height	Jedi
Luke	male	Tatooine	-99999	77kg	1.72m	yes
Leia	female	Alderaan	19BBY	49kg	1.50m	no
Obi-Wan	male	Stewjon	57BBY	77kg	1.82m	yes
Han	MALE	Corellia	29BBY	80kg	1.80m	no
Anakin	male	Tatooine	41.9BBY	84kg	1.88m	NA
Amidala	female	Naboo	46BBY	45kg	1.65m	no

there's still a lot of work to do
to have data ready to be analyzed

How to prepare a data table by unifying units

While the process of data wrangling is time-consuming and sometimes discouraging, especially when we thought we sign up for this class to learn data analysis but we actually spend more than half of our time just cleaning our data, this is normal! In O'Reilly's 2016 Data Science Salary Survey, close to 60% of data scientists spend a significant amount of their time cleaning their data. And, these are data scientists who have a wealth of knowledge and experience with coding and data analysis, which shows how data wrangling is an inevitable and crucial part of any Data Analysis Cycle.

stock images of data analysts

Now that we have a refresher of our experience with data wrangling and some contexts of its importance, we are ready to move on and learn more about the techniques that help us become better at processing our data! I am going to share a data analysis project that I am currently working on and try to implement some data wrangling techniques to my dataset.

Examples

Recently, I have been assigned a project of understanding our school's water usage data from 1975s to 2016 for my on-campus job with the Office of Sustainability and Energy. This dataset, stored on an Excel spreadsheet, has been processed by the previous data analyst, therefore, it contains metrics of measurements that are unfamiliar to me. In addition, the color coding on this sheet is confusing and seems to highlight important information that would require further investigations into what each column actually means and how these metrics were calculated.

Account N	Start Date	End Date	Ccf
44533271	1/4/2016	1/28/2016	0
10373600(	1/5/2016	3/4/2016	9
56124100(	1/7/2016	3/9/2016	0
57120800(	1/7/2016	3/8/2016	0
14320400(	1/8/2016	2/9/2016	547
14320400(	1/8/2016	2/9/2016	24
34965400(	1/8/2016	3/10/2016	3
53341100(	1/8/2016	3/10/2016	40
14859900(	########	2/10/2016	2
14859900(	########	2/10/2016	1,219
14860000(	########	2/10/2016	24
14860000(	########	2/10/2016	17
14860100(	########	2/10/2016	515
14860100(	########	2/10/2016	1,121
14860200(	########	2/10/2016	1,515

Captures of dataset's unknown measurement

I approached the first issue of understanding unfamiliar metrics by contacting the previous data analyst to understand his metrics. He explained to me that the metric, "Ccf" means the "hundred cubic feet". Therefore, that column stores water usage data from different university accounts in the unit of hundred cubic feet. In order to make this information more accessible to others, I decided to change the units of these columns with "Ccf" to "Hundreds of cubic feet".

	StartDate	EndDate	Hundreds of cubic feet
1	1/4/2016	1/28/2016	0
2	1/5/2016	3/4/2016	9
3	1/7/2016	3/9/2016	0
4	1/7/2016	3/8/2016	0
5	1/8/2016	2/9/2016	547
6	1/8/2016	2/9/2016	24
7	1/8/2016	3/10/2016	3
8	1/8/2016	3/10/2016	40
9	1/11/2016	2/10/2016	2
10	1/11/2016	2/10/2016	1219
11	1/11/2016	2/10/2016	24
12	1/11/2016	2/10/2016	17
13	1/11/2016	2/10/2016	515
14	1/11/2016	2/10/2016	1121
15	1/11/2016	2/10/2016	1515

Capture of the changed column name

The second step that I did was to understand the "Data Over Time" tab of this dataset, which supposedly tracks the University's water usage trend from 1975 to 2016, broken down by months. I want to verify how the total water usage is calculated, and I hypothesized that they are calculated by totalling the water usage data (measured in "Hundreds of cubic feet") from all the university accounts, and creating a time series data for every month from 1975 to 2016.

Year	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Annual Total	*Annual: gals*
1975	90641	93322	100767	96790	120802	122747	131955	125026	113096	113323	101672	88655	1298796	*971,499,408*
1976	93146	94061	103488	111728	134968	132382	129212	116395	106913	109720	102948	93653	1328614	*993,803,272*
1977	96389	80500	76487	69312	69348	62429	69194	73973	65423	68275	64615	57304	853249	*638,230,252*
1978	56124	56387	66368	68625	87552	87778	93855	97988	99056	103103	81682	69688	968206	*724,218,088*
1979	70627	66684	68371	68247	85845	85257	122070	148432	101384	85454	75119	65491	1042981	*780,149,788*
1980	70122	69115	77131	80954	84351	93139	100324	85802	86881	94103	80335	72447	994704	*744,038,592*
1981	77401	78746	93507	100338	112736	113256	117097	**32033**	60389	96644	90600	79362	1052109	*786,977,532*
1982	83529	84842	82561	96471	104972	101080	108640	107191	106393	108947	92640	81155	1158421	*866,498,908*
1983	86115	85695	94809	93415	105064	104666	58752	86040	109620	111253	87932	75201	1098562	*821,724,376*
1984	75478	72711	88614	104998	95884	113477	**164186**	**186199**	**182970**	**172755**	**201974**	**173185**	1632431	*1,221,058,388*
1985	131484	112380	113682	125284	131838	149094	136911	145745	123787	117456	115921	121499	1525081	*1,140,760,588*
1986	128591	131680	152605	142463	106822	79569	105374	121716	129924	143583	125478	105375	1473180	*1,101,938,640*
1987	93197	114531	123754	107307	87598	63807	84732	123307	149741	115016	103878	87728	1254596	*938,437,808*
1988	97586	101235	113827	103405	97128	94098	101984	100056	95110	99379	85537	67546	1156891	*865,354,468*
1989	73429	64995	76427	81307	118307	118725	116065	119270	118770	116780	100263	102752	1207090	*902,903,320*
1990	89301	70120	74946	64994	74862	87332	99706	107184	103071	103791	87983	73835	1037125	*775,769,500*
1991	74475	72917	77885	79884	83490	81567	86745	88672	90622	88417	94042	74278	992994	*742,759,512*
1992	71024	73241	81823	85863	89594	86615	92757	93865	97328	98533	82580	65966	1019189	*762,353,372*
1993	66304	63862	72063	75518	77835	75364	81500	86153	81898	79775	70515	57511	888298	*664,446,904*
1994	66362	71814	81685	83670	81759	83996	89618	95667	103719	101126	83276	70774	1013466	*758,072,568*
1995	77855	76372	85072	86302	89664	99500	105487	107460	100848	95911	84263	77559	1086293	*812,547,164*
1996	85478	76494	81280	88824	88177	85568	96106	105829	105376	101455	83572	69870	1068029	*798,885,692*
1997	83017	79971	94508	99289	104860	99691	106396	107313	102338	102595	90980	80159	1151117	*861,035,516*
1998	79401	76103	75044	75409	83961	85408	88196	90776	89066	85721	79667	66887	975639	*729,777,972*
1999	61243	60399	66625	66757	74385	72678	76708	93561	102264	103396	82117	76729	936862	*700,772,776*
2000	78497	77066	85432	86284	89157	90736	100218	100426	95997	98898	78717	79555	1060983	*793,615,284*
2001	82876	71937	81741	89013	94752	98539	102899	105165	96029	93416	80230	71831	1068428	*799,184,144*

Capture of “Data Over Time” sheet

In order to verify my hypothesis, I investigated the previous data analyst’s working tab, where he calculated the monthly total across all accounts. Upon closer investigation, I realized that his working data only spanned the time period from January to December 2016. In this case, I presumed that the monthly water usage data from Year 1975 to 2015 were drawn from other datasets. This means that my investigation applies solely to 2016’s monthly water usage data.

Discussion

Once I loaded the dataset into R, it displayed the data frame of the “Data Over Time” worksheet. Since I would like to verify the total water usage in 2016, I manipulated the data frame to create a new data frame, named “waterdataver2” which contains the columns: “Year”, “Annual Total (hundreds cubic feet)”, and “Annual Total (gallons)”.

```
#Code chunk of loading "dplyr" and reading the water data csv file onto R
library(dplyr)
waterdata <-read.csv ("C:/Users/Jean Ji/stat133/stat133-hws-fall17/post01/data/waterdata.csv", stringsAsFactors = FALSE)
```

Code chunk for loading the water data

```
#Code chunk of creating the a new table for the water data to include columns: Year, Annual total
#water usage, measured in hundred cubic feet and in gallons
waterdataver2 <- select (waterdata, "Year" = 1,
                          "Annual Total (hundreds cubic feet)" = 14,
                          "Annual Total (gallons)" = 15)
```

Code chunk of creating waterdataver2

	Year	Annual Total (hundreds cubic feet)	Annual Total (gallons)
1	1975	1298796.0	971,499,408
2	1976	1328614.0	993,803,272
3	1977	853249.0	638,230,252
4	1978	968206.0	724,218,088
5	1979	1042981.0	780,149,788
6	1980	994704.0	744,038,592
7	1981	1052109.0	786,977,532
8	1982	1158421.0	866,498,908
9	1983	1098562.0	821,724,376
10	1984	1632431.0	1,221,058,388
11	1985	1525081.0	1,140,760,588
12	1986	1473180.0	1,101,938,640
13	1987	1254596.0	938,437,808
14	1988	1156891.0	865,354,468
15	1989	1207090.0	902,903,320
16	1990	1037125.0	775,769,500
17	1991	992994.0	742,759,512
18	1992	1019189.0	762,353,372

Capture of waterdataver2

Since I only have access to the 2016's calculations of water usage, I created a new data frame for the monthly water usage, from January to September 2016. This new table is called "monthlywaterdata".

```
#Code chunk of creating a table of monthly water data from January to September 2016
monthlywaterdata <- read.csv ("C:/Users/Jean Ji/stat133/stat133-hws-fall17/post01/data/2016monthlywaterdata.csv", stringsAsFactors = FALSE)

monthlywaterdata <- select (monthlywaterdata, "StartDate" = "Start.Date", "EndDate" = "End.Date", "Hundreds of cubic feet" = "Ccf")
```

Code chunk of creating the monthly water data table

As mentioned earlier, the monthly water data table only contains data from January to September, it does not capture the range of the entire Year 2016. This leads to discrepancy between the monthly water data and the annual water data, waterdataver2. Furthermore, the sum of water usage across the 9 months are 698549 hundreds of cubic feet, which is approximately 0.5 billion gallons of water.

```
#Code chunk of creating a table of monthly water data from January to September 2016
monthlywaterdata <- read.csv ("C:/Users/Jean Ji/stat133/stat133-hws-fall17/post01/data/2016monthlywaterdata.csv", stringsAsFactors = FALSE)

monthlywaterdata <- select (monthlywaterdata, "StartDate" = "Start.Date", "EndDate" = "End.Date", "Hundreds of cubic feet" = "Ccf")

#Code chunk of adding a new column that shows the sum of the water usage in hundreds of cubic feet and in gallons
hundreds_of_cubic_feet <- as.double(monthlywaterdata$`Hundreds of cubic feet`)
#Turning the water usage into real numbers introduced coercion rule which replaced the missing values with "NA"
#Code chunk for turning 'NA' to '0'
hundreds_of_cubic_feet[is.na(hundreds_of_cubic_feet)] <- 0
#Summing up the 'hundreds of cubic feet' to obtain the total usage and adding a new column to the table
monthlywaterdata <- mutate (monthlywaterdata, 'TotalUsage(Hundreds of cubic feet)' = sum(hundreds_of_cubic_feet))
#Converting the usage in 'hundreds of cubic feet' to gallons
monthlywaterdata <- mutate (monthlywaterdata, 'TotalUsage(Gallons)'= `TotalUsage(Hundreds of cubic feet)` * 748.052)
```

Code chunk for calculating the sum and coverting water usage to gallons

This is what the montly water data looks like:

	StartDate	EndDate	Hundreds of cubic feet	TotalUsage(Hundreds of cubic feet)	TotalUsage(Gallons)
1	1/4/2016	1/28/2016	0	698549	522550977
2	1/5/2016	3/4/2016	9	698549	522550977
3	1/7/2016	3/9/2016	0	698549	522550977
4	1/7/2016	3/8/2016	0	698549	522550977
5	1/8/2016	2/9/2016	547	698549	522550977
6	1/8/2016	2/9/2016	24	698549	522550977
7	1/8/2016	3/10/2016	3	698549	522550977
8	1/8/2016	3/10/2016	40	698549	522550977
9	1/11/2016	2/10/2016	2	698549	522550977
10	1/11/2016	2/10/2016	1219	698549	522550977
11	1/11/2016	2/10/2016	24	698549	522550977
12	1/11/2016	2/10/2016	17	698549	522550977
13	1/11/2016	2/10/2016	515	698549	522550977
14	1/11/2016	2/10/2016	1121	698549	522550977
15	1/11/2016	2/10/2016	1515	698549	522550977

Capture of the monthly water table with total usage in hundreds of cubic feet and in gallons

In the process of trying to verify the 2016 water usage data for the table: waterdataver2, I realized that there was missing data for October, November and December 2016. These three pieces of data were not present in the working tabs of the Excel workbook, but were represented in the waterdataver2 sheet. While the origin of these pieces of data remained mysterious, they are important in the calculation of annual total water usage.

```
#Filtering to see the annual total usage of Year 2016
waterdataver3 <- filter (waterdataver2, Year == 2016)
```

Code chunk for creating 2016 total water usage

	Year	Annual Total (hundreds cubic feet)	Annual Total (gallons)
1	2016	813363.3	608,395,783

Capture of 2016 total water usage from “Data Over Time” sheet; when compared to the previous figure, we can note the discrepancy in the total usage

Conclusion

In attempting to conduct data analysis on our school’s water usage data, I ran into multiple issues that needed further data processing and exploratory data analyses. As we learned from class, data wrangling is such an essential component of data analysis, and sometimes it is the most time-consuming one too.

While many of us would prefer a linear and straightforward process to data analysis, the reality of it is not as glamorous as I once thought. However, I am glad to learn about the different techniques that I can apply to process data and to turn it into useful formats for analysis.

References

1) A comprehensive introduction to data wrangling, URL: https://www.springboard.com/blog/data-wrangling/

2) o’Reiley’s 2016 Data Science Salary Survey (report requires payment to access but is referenced in above linked, so I added it to my reference list), URL: http://www.oreilly.com/data/free/2016-data-science-salary-survey.csp

3) UC Berkeley’s water data from 1975 to 2016

4) R Studio’s tutorial on data wrangling, URL: http://www.oreilly.com/data/free/2016-data-science-salary-survey.csp

5) Stack overflow’s post on turning ‘NA’ to ‘0’, URL: https://stackoverflow.com/questions/8161836/how-do-i-replace-na-values-with-zeros-in-an-r-dataframe

6) R Markdown tutorial, URL: https://www.markdowntutorial.com/lesson/4/

7) How to insert pictures in R Markdown file, URL: https://stackoverflow.com/questions/25166624/insert-picture-table-in-rmarkdown