

# Amazon Spending

Code ▾

Hide

```
setwd("C:/Users/Branch/Documents/misc. code and homework/DSC680/Amazon project")
getwd()
```

```
[1] "C:/Users/Branch/Documents/misc. code and homework/DSC680/Amazon project"
```

Hide

```
install.packages("tidyverse")
library(tidyverse)
install.packages("lubridate")
library(lubridate)
install.packages("plotly")
library(plotly)
install.packages("moderndive")
library(moderndive)
install.packages("skimr")
library(skimr)
install.packages("ISLR")
library(ISLR)
install.packages("ggplot2")
library(ggplot2)
# The ones below here are for ARIMA
install.packages("rio")
install.packages("forecast")
install.packages("tseries")
library(rio)
library(forecast)
library(tseries)
install.packages("anytime")
library(anytime)
```

Hide

```
data <- read.csv('cl_Amazon_both.csv',stringsAsFactors = FALSE)
head(data)
```

	OrderDate <chr>	Year <int>	DayOfWeek <chr>	Month <chr>	OrderID <chr>	▶
1	7/18/2006	2006	Tue	Jul	104-1054314-4386325	
2	5/21/2007	2007	Mon	May	002-4639196-3464828	
3	6/22/2009	2009	Mon	Jun	105-6904728-6779445	
4	7/22/2009	2009	Wed	Jul	105-0221596-1891453	
5	7/22/2009	2009	Wed	Jul	105-9427902-2770610	
6	11/1/2009	2009	Sun	Nov	102-5268790-8011454	

6 rows | 1-6 of 32 columns

Hide

```
df = subset(data, select = -c(Website, OriginalListPrice))
head(df)
```

	OrderDate <chr>	Year <int>	DayOfWeek <chr>	Month <chr>	OrderID <chr>	▶
1	7/18/2006	2006	Tue	Jul	104-1054314-4386325	
2	5/21/2007	2007	Mon	May	002-4639196-3464828	
3	6/22/2009	2009	Mon	Jun	105-6904728-6779445	
4	7/22/2009	2009	Wed	Jul	105-0221596-1891453	
5	7/22/2009	2009	Wed	Jul	105-9427902-2770610	
6	11/1/2009	2009	Sun	Nov	102-5268790-8011454	

6 rows | 1-6 of 30 columns

Hide

```
summary(df)
```

OrderDate	Year	DayOfWeek	Month	OrderID
Length:2020	Min. :2006	Length:2020	Length:2020	Length:2020
Class :character	1st Qu.:2017	Class :character	Class :character	Class :character
Mode :character	Median :2018	Mode :character	Mode :character	Mode :character
	Mean :2018			
	3rd Qu.:2020			
	Max. :2021			

Title	Category	ASIN_ISBN	UNSPSCCode	ReleaseDate
Length:2020	Length:2020	Length:2020	Min. :10000000	Length:2020
Class :character	Class :character	Class :character	1st Qu.:43230000	Class :character
Mode :character	Mode :character	Mode :character	Median :52150000	Mode :character
			Mean :47072994	
			3rd Qu.:54100000	
			Max. :64151505	
			NA's :134	

Condition	Seller	ListPriceCorrected	PurchasePricePerUnit	SavingsAmount
Length:2020	Length:2020	Min. : 1.25	Min. : 0.89	Min. : 0.00
Class :character	Class :character	1st Qu.: 10.21	1st Qu.: 7.99	1st Qu.: 0.00
Mode :character	Mode :character	Median : 18.88	Median : 12.99	Median : 2.92
		Mean : 33.01	Mean : 21.76	Mean : 11.25
		3rd Qu.: 36.04	3rd Qu.: 23.83	3rd Qu.: 11.72
		Max. :738.00	Max. :350.89	Max. :404.01

SavingsPercent	Quantity	PaymentInstrumentType	ShipmentDate	ShippingName
Min. :0.0000	Min. : 1.000	Length:2020	Length:2020	Length:2020
1st Qu.:0.0000	1st Qu.: 1.000	Class :character	Class :character	Class :character
Median :0.2000	Median : 1.000	Mode :character	Mode :character	Mode :character
Mean :0.2422	Mean : 1.086			
3rd Qu.:0.4200	3rd Qu.: 1.000			
Max. :0.9700	Max. :12.000			

ShippingCity	ShippingState	ShippingZip	OrderStatus	CarrierName
Length:2020	Length:2020	Min. :54660	Length:2020	Length:2020
Class :character	Class :character	1st Qu.:54660	Class :character	Class :character
Mode :character	Mode :character	Median :54660	Mode :character	Mode :character
		Mean :55052		
		3rd Qu.:54660		
		Max. :92660		
		NA's :15		

ItemSubtotal	ItemSubtotalTax	ItemTotal	BuyerName	PercentOffCategory
Min. : 0.89	Min. : 0.0000	Min. : 0.940	Length:2020	Length:2020
1st Qu.: 8.29	1st Qu.: 0.0000	1st Qu.: 8.555	Class :character	Class :character
Median : 13.99	Median : 0.3800	Median : 14.640	Mode :character	Mode :character
Mean : 22.53	Mean : 0.7704	Mean : 23.303		
3rd Qu.: 24.98	3rd Qu.: 0.9300	3rd Qu.: 25.863		
Max. :350.89	Max. :16.8200	Max. :350.890		

[Hide](#)

```
str(df)
```

```
'data.frame': 2020 obs. of 30 variables:
 $ OrderDate      : chr  "7/18/2006" "5/21/2007" "6/22/2009" "7/22/2009" ...
 $ Year           : int   2006 2007 2009 2009 2009 2009 2009 2010 2010 2010 ...
 $ DayOfWeek      : chr   "Tue" "Mon" "Mon" "Wed" ...
 $ Month          : chr   "Jul" "May" "Jun" "Jul" ...
 $ OrderID        : chr   "104-1054314-4386325" "002-4639196-3464828" "105-6904728-6779445" "105-0221596-1
891453" ...
 $ Title          : chr   "SSX On Tour - PlayStation 2" "Quick & Easy Scrapbook Pages" "Glenn Beck's Commo
n Sense: The Case Against an Out-of-Control Government, Inspired by Thomas Paine" "Why Hospitals Should Fly: Th
e Ultimate Flight Plan to Patient Safety and Quality Care" ...
 $ Category       : chr   "CONSOLE_VIDEO_GAMES" "ABIS_BOOK" "ABIS_BOOK" "ABIS_BOOK" ...
 $ ASIN_ISBN      : chr   "B000AOEUOI" "1892127202" "1439168571" "974386065" ...
 $ UNSPSCCode     : int   43230000 55101500 55101500 55101500 55101500 55111514 39122200 40161500 55111514
55101500 ...
 $ ReleaseDate    : chr   "2006-09-08T00:00:01" "2003-04-01T00:00" "2009-06-16T00:00:01" "2008-01-15T00:00
:01" ...
 $ Condition      : chr   "new" "new" "new" "new" ...
 $ Seller         : chr   "Amazon.com" "Amazon.com" "Amazon.com" "Second River Healthcare - Publisher" ...
 $ ListPriceCorrected : num  20 23 12 25 27.9 ...
 $ PurchasePricePerUnit : num  18.99 15.63 6.59 25 18.45 ...
 $ SavingsAmount   : num   1 7.36 5.4 0 9.5 ...
 $ SavingsPercent   : num   0.05 0.32 0.45 0 0.34 0.3 0.3 0.34 0.69 0 ...
 $ Quantity        : int    1 1 2 1 1 1 1 2 1 1 ...
 $ PaymentInstrumentType: chr   "Visa" "Discover" "Discover" "Discover" ...
 $ ShipmentDate    : chr   "8/8/2006" "5/22/2007" "6/22/2009" "7/23/2009" ...
 $ ShippingName     : chr   "husband" "husband" "wife" "wife" ...
 $ ShippingCity     : chr   "WI_town" "WI_town" "WI_town" "WI_town" ...
 $ ShippingState    : chr   "WI" "WI" "WI" "WI" ...
 $ ShippingZip      : int   54660 54660 54660 54660 54660 54660 54660 54660 54660 54660 ...
 $ OrderStatus      : chr   "Shipped" "Shipped" "Shipped" "Shipped" ...
 $ CarrierName      : chr   "USPS" "UPS" "USPS" "USPS" ...
 $ ItemSubtotal     : num   19 15.6 13.2 25 18.4 ...
 $ ItemSubtotalTax   : num   0 0 0 0 0 0 0 0 0 ...
 $ ItemTotal        : num   19 15.6 13.2 25 18.4 ...
 $ BuyerName        : chr   "husband" "husband" "wife" "wife" ...
 $ PercentOffCategory : chr   "Up to 10%" "31-40%" "41-50%" "0" ...
```

[Hide](#)

```
class(df$listPriceCorrected)
```

```
[1] "numeric"
```

from <https://stackoverflow.com/questions/16384933/create-an-id-row-number-column>  
(<https://stackoverflow.com/questions/16384933/create-an-id-row-number-column>)

Hide

```
df$ID <- 1:nrow(df)
head(df)
```

	OrderDate <chr>	Year <int>	DayOfWeek <chr>	Month <chr>	OrderID <chr>
1	7/18/2006	2006	Tue	Jul	104-1054314-4386325
2	5/21/2007	2007	Mon	May	002-4639196-3464828
3	6/22/2009	2009	Mon	Jun	105-6904728-6779445
4	7/22/2009	2009	Wed	Jul	105-0221596-1891453
5	7/22/2009	2009	Wed	Jul	105-9427902-2770610
6	11/1/2009	2009	Sun	Nov	102-5268790-8011454

6 rows | 1-6 of 31 columns

from <https://stackoverflow.com/questions/22286419/move-a-column-to-first-position-in-a-data-frame>  
(<https://stackoverflow.com/questions/22286419/move-a-column-to-first-position-in-a-data-frame>)

Hide

```
df <- df[,c(31, 1:30)]
head(df)
```

	ID <int>	OrderDate <chr>	Year <int>	DayOfWeek <chr>	Month <chr>	OrderID <chr>
1	1	7/18/2006	2006	Tue	Jul	104-1054314-4386325
2	2	5/21/2007	2007	Mon	May	002-4639196-3464828
3	3	6/22/2009	2009	Mon	Jun	105-6904728-6779445
4	4	7/22/2009	2009	Wed	Jul	105-0221596-1891453
5	5	7/22/2009	2009	Wed	Jul	105-9427902-2770610
6	6	11/1/2009	2009	Sun	Nov	102-5268790-8011454

6 rows | 1-7 of 31 columns

Hide

```
View(df)
```

from <https://www.earthdatascience.org/courses/earth-analytics/time-series-data/summarize-time-series-by-month-in-r/>  
(<https://www.earthdatascience.org/courses/earth-analytics/time-series-data/summarize-time-series-by-month-in-r/>)

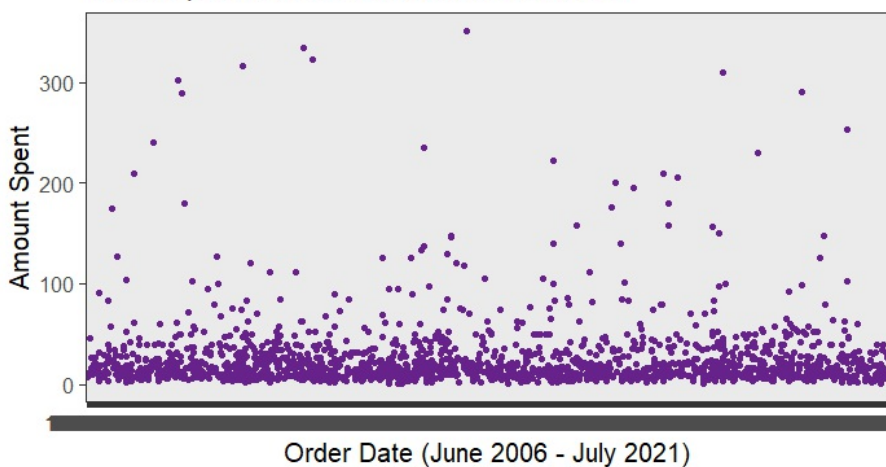
If this plot doesn't work, check the ggplot2 box under packages. Has to do with newer version of R.

Hide

```
ggplot(df, aes(x = OrderDate, y = ItemTotal)) +
  geom_point(color = "darkorchid4") +
  labs(title = "Amazon Spending by One Family",
        subtitle = "Includes purchases made with Amazon rewards",
        y = "Amount Spent",
        x = "Order Date (June 2006 - July 2021)") + theme_bw(base_size = 15)
```

## Amazon Spending by One Family

Includes purchases made with Amazon rewards



Hide

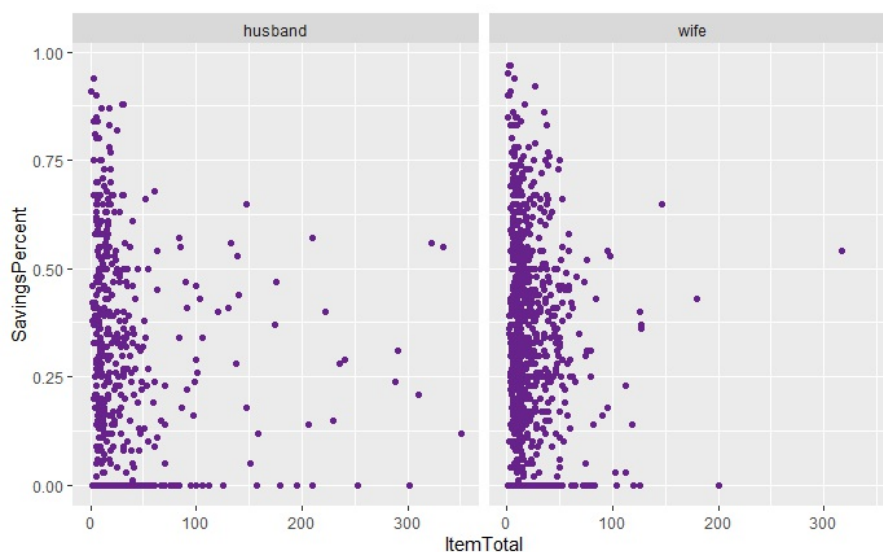
```
ggplot(df, aes(x = OrderDate, y = ItemTotal)) +
  geom_point(color = "darkorchid4") +
  facet_wrap(~ Year) +
  labs(title = "Family Spending on Amazon Across the Year",
        subtitle = "Includes purchases made with Amazon Rewards",
        y = "Total Spent, in USD"
  )
```

Hide

```
ggplot(df, aes(x = OrderDate, y = ItemTotal)) +
  geom_point(color = "darkorchid4") +
  facet_wrap(~ BuyerName) +
  labs(title = "Amazon Spending by One Family",
        subtitle = "Includes purchases made with Amazon rewards",
        y = "Amount Spent",
        x = "Order Date (June 2006 - July 2021)") + theme_bw(base_size = 15)
```

Hide

```
ggplot(df, aes(x = ItemTotal, y = SavingsPercent)) +
  geom_point(color = "darkorchid4") +
  facet_wrap(~ BuyerName)
```



Hide

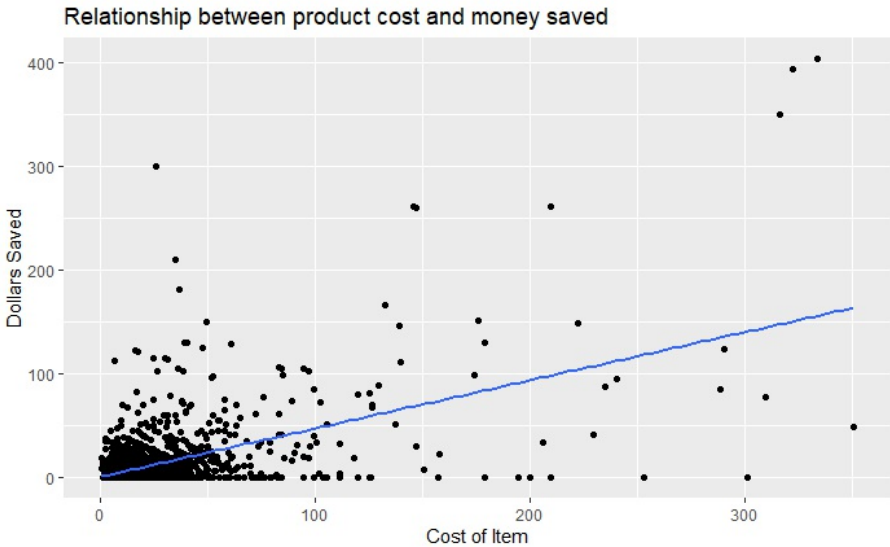
```
labs(title = "Amazon Spending by One Family",
      subtitle = "Includes purchases made with Amazon rewards",
      y = "Percent Off List Price",
      x = "Purchase Price in USD") + theme_bw(base_size = 15)
```

NULL

from <https://moderndive.com/5-regression.html>  
(<https://moderndive.com/5-regression.html>)

Hide

```
ggplot(df, aes(x = ItemTotal, y = SavingsAmount)) +  
  geom_point() +  
  labs(x = "Cost of Item", y = "Dollars Saved",  
       title = "Relationship between product cost and money saved") +  
  geom_smooth(method = "lm", se = FALSE)
```



from <https://a-little-book-of-r-for-time-series.readthedocs.io/en/latest/src/timeseries.html> (<https://a-little-book-of-r-for-time-series.readthedocs.io/en/latest/src/timeseries.html>)

Hide

```
timeseries <- ts(df)  
timeseries
```

Time Series:  
Start = 1  
End = 2020  
Frequency = 1

	ID	OrderDate	Year	DayOfWeek	Month	OrderID	Title	Category	ASIN_ISBN	UNSPSCCode	ReleaseDate
1	1	572	2006	6	6	59	1295	80	268	43230000	65
2	2	461	2007	2	9	7	1145	2	100	55101500	45
3	3	519	2009	2	7	92	648	2	32	55101500	81
4	4	582	2009	7	6	82	1529	2	183	55101500	72
5	5	582	2009	7	6	95	889	2	138	55101500	74
6	6	101	2009	4	10	33	1385	4	270	55111514	58
7	7	162	2009	4	10	94	195	334	293	39122200	1
8	8	405	2010	4	1	51	552	352	262	40161500	1
9	9	472	2010	5	9	15	1433	4	454	55111514	90
10	10	707	2010	6	12	34	586	44	160	55101500	38
11	11	707	2010	6	12	21	566	2	159	55101500	34
12	12	56	2010	2	11	39	759	114	323	26121600	1
13	13	180	2010	4	10	72	1	1	297	NA	1
14	14	129	2010	4	10	76	632	338	474	27110000	1
15	15	238	2010	2	3	67	1	1	312	NA	1
16	16	339	2011	3	8	79	1088	366	422	52160000	94
17	17	351	2011	3	8	60	1505	80	335	43230000	69
18	18	351	2011	3	8	60	1120	365	492	60141100	96
19	19	351	2011	3	8	80	1	1	240	NA	1
20	20	541	2011	7	7	66	299	2	184	55101500	93
21	21	541	2011	7	7	66	295	2	45	55101500	102
22	22	541	2011	7	7	66	1388	2	185	55101500	83
23	23	668	2011	7	12	69	1347	332	206	52150000	1
24	24	668	2011	7	12	69	197	363	385	25000000	1
25	25	89	2011	4	11	88	320	4	558	55111514	113
26	26	115	2011	5	10	18	326	4	549	55111514	114
27	27	146	2011	3	10	62	879	80	480	43230000	110
28	28	146	2011	3	10	62	1	1	507	NA	1
29	29	49	2012	3	5	63	122	100	572	43211619	1
30	30	49	2012	3	5	63	895	22	223	25170000	1
31	31	21	2012	1	5	71	1308	4	320	55111514	112
32	32	293	2012	4	4	81	1227	345	237	60141000	1

	Condition	Seller	ListPriceCorrected	PurchasePricePerUnit	SavingsAmount	SavingsPercent	Quantity
1	1	31	19.99	18.99	1.00	0.05	1
2	1	31	22.99	15.63	7.36	0.32	1
3	1	31	11.99	6.59	5.40	0.45	2

4	1	515	25.00	25.00	0.00	0.00	1
5	1	31	27.95	18.45	9.50	0.34	1
6	1	31	14.98	10.49	4.49	0.30	1
7	1	31	31.99	22.50	9.49	0.30	1
8	1	54	18.99	12.50	6.49	0.34	2
9	1	587	39.99	12.45	27.54	0.69	1
10	1	386	19.70	19.70	0.00	0.00	1
11	1	654	14.25	1.41	12.84	0.90	1
12	1	286	99.99	17.35	82.64	0.83	1
13	1	31	58.80	35.86	22.94	0.39	1
14	1	1	14.99	8.99	6.00	0.40	2
15	1	31	39.99	31.90	8.09	0.20	1
16	1	31	399.99	350.89	49.10	0.12	1
17	1	216	29.99	14.14	15.85	0.53	1
18	1	8	24.99	20.75	4.24	0.17	1
19	1	89	39.99	2.43	37.56	0.94	1
20	1	31	19.97	17.07	2.90	0.15	1
21	1	31	17.95	15.34	2.61	0.15	1
22	1	31	24.95	24.95	0.00	0.00	1
23	1	31	25.99	12.18	13.81	0.53	1
24	1	31	39.99	19.57	20.42	0.51	1
25	1	1	42.99	16.75	26.24	0.61	1
26	1	245	49.99	16.89	33.10	0.66	1
27	1	31	59.95	39.99	19.96	0.33	1
28	1	31	200.00	119.99	80.01	0.40	1
29	1	1	21.95	10.95	11.00	0.50	1
30	1	31	22.67	12.28	10.39	0.46	1
31	1	31	69.99	34.99	35.00	0.50	1
32	1	31	19.99	14.91	5.08	0.25	1

	PaymentInstrumentType	ShipmentDate	ShippingName	ShippingCity	ShippingState	ShippingZip
1	5	753	3	5	4	54660
2	1	535	3	5	4	54660
3	1	596	10	5	4	54660
4	1	668	10	5	4	54660
5	1	668	10	5	4	54660
6	1	144	10	5	4	54660
7	1	195	10	5	4	54660
8	1	470	3	5	4	54660
9	1	545	3	5	4	54660
10	1	814	10	5	4	54660
11	1	814	10	5	4	54660
12	1	68	3	5	4	54660
13	1	212	3	5	4	54660
14	1	157	3	5	4	54660
15	3	280	3	5	4	54660
16	2	395	3	5	4	54660
17	1	415	3	5	4	54660
18	1	415	3	5	4	54660
19	1	415	3	5	4	54660
20	1	685	10	5	4	54660
21	1	685	10	5	4	54660
22	1	685	10	5	4	54660
23	1	769	3	5	4	54660
24	1	769	3	5	4	54660
25	1	106	3	5	4	54660
26	1	135	3	5	4	54660
27	1	176	3	5	4	54660
28	1	176	3	5	4	54660
29	4	61	3	5	4	54660
30	4	61	3	5	4	54660
31	4	23	3	5	4	54660
32	4	355	3	5	4	54660

	OrderStatus	CarrierName	ItemSubtotal	ItemSubtotalTax	ItemTotal	BuyerName	PercentOffCategory
1	1	27	18.99	0.00	18.99	1	11
2	1	26	15.63	0.00	15.63	1	4
3	1	27	13.18	0.00	13.18	2	5
4	1	27	25.00	0.00	25.00	2	1
5	1	27	18.45	0.00	18.45	2	4
6	1	27	10.49	0.00	10.49	2	3
7	1	27	22.50	0.00	22.50	2	3
8	1	27	25.00	0.00	25.00	1	4
9	1	27	12.45	0.00	12.45	1	7
10	1	27	19.70	0.00	19.70	2	1
11	1	27	1.41	0.00	1.41	2	9
12	1	27	17.35	0.00	17.35	1	9
13	1	27	35.86	0.00	35.86	1	4
14	1	26	17.98	0.00	17.98	1	4
15	1	27	31.90	0.00	31.90	1	2
16	1	27	350.89	0.00	350.89	1	2
17	1	27	14.14	0.00	14.14	1	6
18	1	27	20.75	0.00	20.75	1	2
19	1	26	2.43	0.00	2.43	1	10
20	1	27	17.07	0.00	17.07	2	2
21	1	27	15.34	0.00	15.34	2	2
22	1	27	24.95	0.00	24.95	2	1
23	1	27	12.18	0.00	12.18	1	6
24	1	27	19.57	0.00	19.57	1	6
25	1	27	16.75	0.00	16.75	1	7
26	1	27	16.89	0.00	16.89	1	7
27	1	27	39.99	0.00	39.99	1	4
28	1	27	119.99	0.00	119.99	1	4
29	1	12	10.95	0.00	10.95	1	5
30	1	12	12.28	0.00	12.28	1	5
31	1	26	34.99	0.00	34.99	1	5

```
32      1      27      14.91      0.00      14.91      1      3
[ reached getOption("max.print") -- omitted 1988 rows ]
```

That was not what I wanted. Try putting OrderDate into its own .csv file, read it into R store it as a time series

Hide

```
orderdates <- read.csv('OrderDate_only.csv',stringsAsFactors = FALSE)
head(orderdates)
```

	OrderDate <chr>	Quantity <int>
1	7/18/2006	1
2	5/21/2007	1
3	6/22/2009	2
4	7/22/2009	1
5	7/22/2009	1
6	11/1/2009	1
6 rows		

from <https://www.statmethods.net/input/dates.html>  
(<https://www.statmethods.net/input/dates.html>)

Hide

```
dates <- as.Date(orderdates$OrderDate, "%m/%d/%Y")
head(dates)
```

```
[1] "2006-07-18" "2007-05-21" "2009-06-22" "2009-07-22" "2009-07-22" "2009-11-01"
```

Hide

```
ordertimeseries <- ts(dates)
head(ordertimeseries)
```

```
[1] 13347 13654 14417 14447 14447 14549
```

Hide

```
ts_with_frequency <- ts(dates, start=c(2006,7), frequency = 365)
ts_with_frequency
```

```

Time Series:
Start = c(2006, 7)
End = c(2011, 201)
Frequency = 365

[1] 13347 13654 14417 14447 14447 14549 14577 14717 14756 14859 14859 14893 14920 14934 14970
[16] 15045 15052 15052 15052 15154 15154 15154 15231 15231 15277 15295 15304 15304 15346 15346
[31] 15359 15389 15389 15457 15457 15457 15457 15491 15514 15515 15525 15525 15526 15526 15636
[46] 15636 15636 15641 15642 15642 15642 15642 15646 15646 15646 15646 15646 15649 15653
[61] 15653 15653 15654 15660 15660 15660 15666 15666 15666 15666 15666 15666 15667 15667 15667
[76] 15667 15670 15697 15697 15697 15697 15697 15698 15698 15706 15713 15755 15755 15755 15766 15767
[91] 15767 15767 15773 15773 15773 15773 15798 15798 15798 15798 15844 15860 15860 15866 15866
[106] 15869 15869 15869 15870 15870 15870 15899 15899 15899 15913 15913 15913 15913 15913 15913
[121] 15913 15913 15913 15913 15916 15917 15919 15919 15919 15923 15923 15923 15930 15976 15976
[136] 16008 16010 16010 16015 16015 16030 16030 16030 16030 16030 16030 16030 16030 16030 16030
[151] 16030 16030 16030 16036 16036 16036 16037 16037 16037 16038 16038 16038 16038 16038 16041
[166] 16041 16041 16042 16042 16047 16047 16047 16047 16047 16047 16105 16115 16116 16116 16116
[181] 16116 16116 16116 16116 16116 16116 16145 16145 16145 16145 16166 16172 16172 16172 16181
[196] 16220 16220 16220 16244 16244 16249 16272 16272 16272 16272 16286 16286 16286 16286 16293
[211] 16293 16314 16314 16314 16316 16326 16326 16326 16343 16355 16356 16356 16357 16357 16357
[226] 16357 16357 16357 16357 16357 16357 16383 16383 16383 16399 16402 16402 16402 16403 16403
[241] 16403 16403 16403 16403 16403 16403 16403 16409 16409 16409 16412 16412 16413 16413 16413
[256] 16424 16430 16480 16480 16480 16480 16480 16492 16492 16492 16492 16492 16504 16505 16505
[271] 16505 16530 16534 16534 16534 16541 16544 16588 16602 16602 16602 16602 16604 16637 16637
[286] 16637 16637 16656 16664 16664 16664 16697 16697 16697 16700 16704 16707 16710 16710 16710
[301] 16712 16712 16713 16713 16713 16718 16718 16718 16718 16718 16719 16764 16764 16764 16764
[316] 16764 16765 16765 16766 16766 16766 16767 16767 16767 16767 16767 16768 16768 16768 16768
[331] 16771 16771 16771 16771 16771 16771 16771 16771 16774 16774 16774 16774 16778 16778 16778
[346] 16778 16778 16778 16778 16778 16778 16778 16778 16778 16798 16798 16802 16802 16802 16802
[361] 16802 16812 16812 16816 16816 16816 16816 16816 16816 16821 16821 16822 16822 16823 16823
[376] 16823 16823 16826 16826 16826 16826 16826 16826 16826 16838 16838 16838 16844 16844 16844
[391] 16854 16854 16854 16858 16858 16858 16858 16899 16899 16899 16899 16899 16899 16899 16920
[406] 16920 16920 17003 17003 17016 17016 17049 17049 17049 17049 17049 17049 17068 17068 17068
[421] 17107 17107 17107 17107 17107 17110 17115 17115 17123 17123 17125 17125 17126 17128 17129
[436] 17129 17129 17129 17129 17129 17129 17129 17129 17129 17129 17130 17130 17130 17130 17130
[451] 17130 17132 17132 17132 17133 17133 17133 17133 17133 17135 17135 17135 17135 17135 17135
[466] 17135 17137 17137 17137 17137 17137 17141 17142 17142 17142 17142 17142 17143 17143 17143
[481] 17143 17143 17143 17146 17146 17148 17148 17148 17148 17150 17150 17150 17150 17150 17150
[496] 17153 17155 17159 17177 17197 17203 17203 17204 17204 17204 17204 17204 17205 17206 17206
[511] 17206 17213 17213 17213 17213 17213 17213 17213 17215 17215 17218 17218 17221 17221 17221
[526] 17221 17221 17225 17226 17227 17230 17230 17231 17231 17231 17231 17231 17232 17238 17243
[541] 17243 17243 17246 17246 17246 17247 17247 17257 17269 17272 17272 17272 17272 17272 17272
[556] 17274 17287 17287 17291 17291 17298 17301 17302 17302 17302 17331 17332 17332 17332 17332
[571] 17332 17332 17332 17332 17335 17339 17339 17339 17339 17339 17342 17342 17342 17342 17342
[586] 17346 17346 17352 17352 17352 17352 17352 17352 17352 17352 17352 17352 17353 17353 17353
[601] 17353 17353 17353 17358 17358 17358 17358 17358 17358 17358 17358 17359 17359 17359 17361
[616] 17362 17362 17369 17370 17374 17374 17374 17377 17377 17380 17384 17385 17385 17393 17393
[631] 17393 17393 17393 17393 17393 17396 17398 17398 17398 17398 17398 17398 17398 17398 17398
[646] 17398 17401 17401 17412 17412 17412 17413 17414 17415 17415 17415 17415 17417 17417 17417
[661] 17417 17429 17429 17429 17429 17429 17432 17432 17433 17436 17436 17436 17436 17437 17437
[676] 17437 17437 17437 17440 17440 17440 17444 17444 17450 17451 17454 17454 17458 17460 17460
[691] 17473 17475 17475 17476 17476 17476 17477 17478 17485 17485 17486 17489 17489 17493 17493
[706] 17493 17493 17493 17493 17493 17495 17495 17495 17495 17495 17495 17495 17495 17495 17495
[721] 17497 17497 17497 17497 17497 17497 17499 17499 17501 17501 17502 17502 17502 17502 17502
[736] 17505 17505 17507 17507 17507 17507 17507 17507 17507 17513 17513 17513 17513 17513 17515
[751] 17515 17525 17525 17525 17525 17527 17527 17529 17529 17531 17531 17534 17541 17542 17549 17549
[766] 17549 17549 17549 17549 17550 17550 17553 17553 17555 17555 17555 17556 17556 17556 17558
[781] 17558 17558 17558 17558 17558 17567 17570 17570 17570 17571 17575 17575 17577 17577 17586
[796] 17587 17590 17590 17590 17590 17591 17591 17591 17592 17592 17592 17595 17595 17595 17596
[811] 17599 17599 17608 17609 17609 17612 17612 17612 17612 17615 17615 17631 17631 17631 17631
[826] 17632 17632 17634 17640 17640 17644 17644 17644 17654 17668 17668 17668 17668 17669 17669
[841] 17670 17670 17670 17677 17677 17678 17680 17686 17686 17687 17691 17691 17692 17692 17693
[856] 17695 17695 17699 17700 17700 17700 17700 17700 17708 17708 17708 17710 17713 17716 17716
[871] 17716 17717 17719 17720 17720 17720 17720 17720 17720 17721 17723 17723 17723 17723 17723
[886] 17723 17723 17723 17723 17723 17725 17725 17725 17729 17729 17729 17733 17733 17751 17751
[901] 17753 17757 17757 17757 17757 17757 17763 17763 17763 17763 17763 17764 17765 17765 17771
[916] 17771 17771 17771 17775 17781 17781 17781 17787 17787 17788 17788 17789 17790 17794 17794
[931] 17794 17798 17798 17799 17811 17811 17811 17811 17815 17824 17824 17826 17835 17836 17836
[946] 17836 17836 17837 17837 17838 17838 17839 17844 17850 17850 17850 17850 17852 17852 17853
[961] 17853 17855 17855 17855 17857 17857 17857 17857 17858 17859 17859 17859 17862 17862 17863
[976] 17866 17866 17868 17868 17868 17868 17868 17868 17868 17868 17868 17868 17871 17871 17871
[991] 17871 17874 17875 17875 17875 17875 17875 17875 17876 17876 17876 17876 17876 17876 17876
[ reached getOption("max.print") -- omitted 1020 entries ]

```

Hide

# Ouch! This assumes that each order is a single month and stretches the time series out into the future!

from <https://www.tutorialspoint.com/how-to-convert-a-data-frame-column-to-date-that-contains-integer-values-in-r>  
(<https://www.tutorialspoint.com/how-to-convert-a-data-frame-column-to-date-that-contains-integer-values-in-r>)

Hide

```

orderdates_try <- read.csv('OrderDate_only.csv',stringsAsFactors = FALSE)
dates_as_dates <- data.frame(orderdates_try)
dates_as_dates <- transform(dates_as_dates, OrderDate=as.Date(as.character(OrderDate), "%Y%m%d"))
dates_as_dates      # Why does it change the date to 'NA'?

```



OrderDate <date>	Quantity <int>
<NA>	1
<NA>	1
<NA>	2
<NA>	1
<NA>	1
<NA>	1
<NA>	1
<NA>	2
<NA>	1
<NA>	1
<NA>	1

1-10 of 2,020 rows

Previous 1 2 3 4 5 6 ... 100 Next

from <https://moderndive.com/5-regression.html>  
(<https://moderndive.com/5-regression.html>)

Hide

```
evals_ch6 <- df %>%
  select(ID, SavingsPercent, Category, BuyerName, ItemTotal)
glimpse(evals_ch6)
```

Hide

```
evals_ch6 %>%
  get_correlation(formula = SavingsPercent ~ ItemTotal)
```

from <https://moderndive.com/6-multiple-regression.html>  
(<https://moderndive.com/6-multiple-regression.html>)

Hide

```
ggplot(evals_ch6, aes(x = ItemTotal, y = SavingsPercent, color = BuyerName)) +
  geom_point() +
  labs(x = "Item Total in USD", y = "Percent Savings", color = "Purchaser") +
  geom_smooth(method = "lm", se = FALSE)
```

Hide

```
ggplot(df, aes(x = ItemTotal)) +
  geom_histogram(binwidth = 5, boundary = 0.4, color = "white") +
  labs(x = "Item Cost",
       title = "Distribution of Items Purchased, n=2020")
```

from [http://rstudio-pubs-static.s3.amazonaws.com/374830\\_ab4ac8951c94411aa9abfe5f561cef9](http://rstudio-pubs-static.s3.amazonaws.com/374830_ab4ac8951c94411aa9abfe5f561cef9)  
([http://rstudio-pubs-static.s3.amazonaws.com/374830\\_ab4ac8951c94411aa9abfe5f561cef9](http://rstudio-pubs-static.s3.amazonaws.com/374830_ab4ac8951c94411aa9abfe5f561cef9))

Hide

```
ggplot(df, aes(OrderDate, ItemTotal)) + geom_line() + ylab("Item Cost in USD") +
  xlab("date")
```

from <https://statisticsglobe.com/error-discrete-value-supplied-to-continuous-scale-in-r> (<https://statisticsglobe.com/error-discrete-value-supplied-to-continuous-scale-in-r>)

Hide

```
data_new <- df[df$Quantity != "8+", ]
ggplot(data_new, aes(y=Quantity, x = DayOfWeek)) +
  geom_boxplot(aes(group=DayOfWeek)) +
  #geom_point() +
  #scale_x_continuous(breaks=seq(0,7,1)) +
  ggtitle("Order Count by Day")
```

Hide

```
ggplot(df, aes(OrderDate, Quantity)) + geom_line() + ylab("Count of Items Ordered") +
  xlab("date")
```

from <https://otexts.com/fpp2/forecasting-decomposition.html>  
(<https://otexts.com/fpp2/forecasting-decomposition.html>)

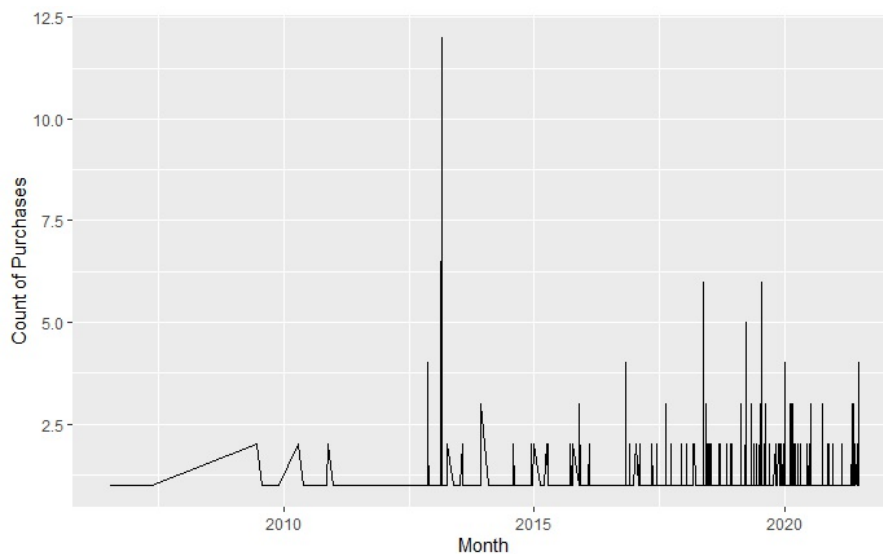
Hide

```
fit <- stl(df, t.window=13, s.window="periodic",
  robust=TRUE)
fit %>% seasadj() %>% naive() %>%
  autoplot() + ylab("New orders index") +
  ggtitle("Naive forecasts of seasonally adjusted data")
```

from <https://youtu.be/UekOBfpu8m8>  
(<https://youtu.be/UekOBfpu8m8>)

Hide

```
data1 <- import("cl_Amazon_both_for_ARIMA.csv")
data1$DATES <- anydate(data1$OrderDate)
ggplot(data1, aes(DATES, Quantity)) + geom_line() + scale_x_date('Month') + ylab("Count of Purchases") + xlab("
")
```



#This did not work # from <https://otexts.com/fpp2/arima-r.html> (<https://otexts.com/fpp2/arima-r.html>)

Hide

```
data3 <- read.csv('unique_order_dates.csv', stringsAsFactors = FALSE)
data3_ts <- ts(data3)
```

Hide

```
data3_ts %>% stl(s.window='periodic') %>% seasadj() -> eeadj
```

```
Error in na.fail.default(as.ts(x)) : missing values in object
```

try using padr library

from <https://cran.r-project.org/web/packages/padr/vignettes/padr.html>  
(<https://cran.r-project.org/web/packages/padr/vignettes/padr.html>)

trying to fill in dates that have no entries; 'padding' the time series?

Hide

```
install.packages("padr")
library(padr)
```

Hide

```

dates_as_dates %>%
  thicken('day') %>%
  group_by(OrderDate) %>%
  summarise(item_total = sum(ItemTotal)) %>%
  pad() %>%
  fill_by_value() %>%
  ggplot(aes(OrderDate, item_total)) + geom_line()

```

There are NA values in the column OrderDate.  
Returned dataframe contains original observations, with NA values for OrderDate and OrderDate\_day.  
Error in seq.int(0, to0 - from, by) : 'to' must be a finite number

Hide

```

#UniqueOrderDate <- as.Date(data3$UniqueOrderDate, "%m/%d/%Y") # doesn't work
data3$UniqueOrderDate <- as.Date(data3$UniqueOrderDate)
head(data3)

```

Hide

```
data4 <- data3 %>% thicken('week')
```

```

Error in round_down_core(a, b) :
  function 'Rcpp_precious_remove' not provided by package 'Rcpp'

```

try purrr, from <https://algotech.netlify.app/blog/purrr-operly-fitting-multiple-time-series-model/>  
(<https://algotech.netlify.app/blog/purrr-operly-fitting-multiple-time-series-model/>)

Hide

```
install.packages("magrittr")
```

```
Error in install.packages : Updating loaded packages
```

Hide

```
library(magrittr)
```

```

Attaching package: 'magrittr'

The following object is masked from 'package:rlang':

  set_names

The following object is masked from 'package:purrr':

  set_names

The following object is masked from 'package:pastecs':

  extract

```

Hide

```
install.packages("tidymodels")
```

```
Restarting R session...
```

try to fill in the missing dates, from  
<https://community.rstudio.com/t/how-to-fill-missing-dates-in-dataset/70517/6> (<https://community.rstudio.com/t/how-to-fill-missing-dates-in-dataset/70517/6>)

Hide

```

install.packages("tibble")
install.packages("tidyr")
library(tibble)
library(tidyr)

```

Hide

```
install.packages("dplyr")
```

```
Error in install.packages : Updating loaded packages
```

[Hide](#)

```
library(dplyr)
```

```
package 'dplyr' was built under R version 4.0.5  
Attaching package: 'dplyr'
```

```
The following objects are masked from 'package:stats':
```

```
filter, lag
```

```
The following objects are masked from 'package:base':
```

```
intersect, setdiff, setequal, union
```

[Hide](#)

```
data5 <- read.csv('indiv_order_dates_for_Python.csv', stringsAsFactors = FALSE)  
head(data5)
```

	OrderDate <chr>	Quantity <int>	Combining_Duplicates <dbl>	X <chr>
1	7/18/2006	1	18.99	
2	5/21/2007	1	15.63	
3	6/22/2009	2	13.18	
4	7/22/2009	1	43.45	
5	11/1/2009	1	10.49	
6	11/29/2009	1	22.50	
6 rows				

[Hide](#)

```
tibble5 <- tibble(data5$OrderDate, data5$Quantity, data5$Combining_Duplicates) %>%  
  mutate(OrderDate = mdy(OrderDate))
```

```
Error in tibble(data5$OrderDate, data5$Quantity, data5$Combining_Duplicates) %>% :  
  could not find function "%>%"
```

I am DONE trying to get this time series wrangled into a regular time series in R!

Time to try Python!