Case Study 2

Sonia Mazzi

13/11/2018

Data Science with

Please, go to

https://github.com/sonjam111/NICE

and download the file

mbta.xlsx

Tidy data

> *"Happy families are all alike; every unhappy family is unhappy in its own way." Leo Tolstoy*

- Hadley Wickham in “Tidy Data” defines the three qualities of tidy data which standardise the process of dealing with any data set:

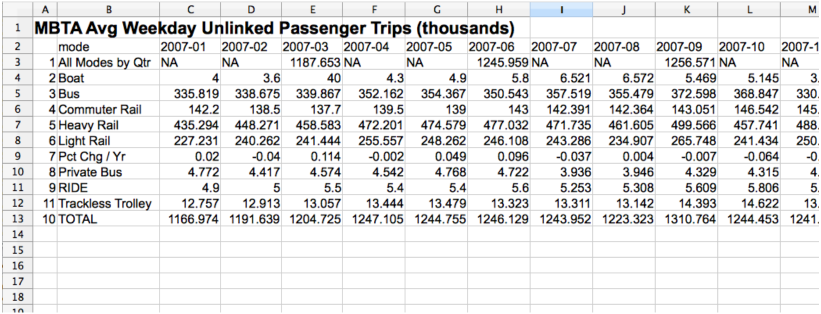
1. Each variable forms a column.
2. Each observation forms a row.
3. Each type of observational unit forms a table.

- These three qualities are what the end product of the data tidying process must possess.
- *Messy data* is data which is not tidy.
- Applying the tidy data criteria standardises the structure of a data set, making exploration and analysis of data easier and less error-prone.

CASE STUDY Massachusetts Bay Transport Authority (MBTA) data from an excel file

Data on transportation in Boston, USA: monthly averages of weekday number of passengers (in thousands) by mode of transportation.

A snapshot of part of the data (not all columns are included), in excel format, is below.



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	**MBTA Avg Weekday Unlinked Passenger Trips (thousands)**												
2		mode	2007-01	2007-02	2007-03	2007-04	2007-05	2007-06	2007-07	2007-08	2007-09	2007-10	2007-1
3	1	All Modes by Qtr	NA	NA	1187.653	NA	NA	1245.959	NA	NA	1256.571	NA	NA
4	2	Boat	4	3.6	40	4.3	4.9	5.8	6.521	6.572	5.469	5.145	3.
5	3	Bus	335.819	338.675	339.867	352.162	354.367	350.543	357.519	355.479	372.598	368.847	330.
6	4	Commuter Rail	142.2	138.5	137.7	139.5	139	143	142.391	142.364	143.051	146.542	145.
7	5	Heavy Rail	435.294	448.271	458.583	472.201	474.579	477.032	471.735	461.605	499.566	457.741	488.
8	6	Light Rail	227.231	240.262	241.444	255.557	248.262	246.108	243.286	234.907	265.748	241.434	250.
9	7	Pct Chg / Yr	0.02	-0.04	0.114	-0.002	0.049	0.096	-0.037	0.004	-0.007	-0.064	-0.
10	8	Private Bus	4.772	4.417	4.574	4.542	4.768	4.722	3.936	3.946	4.329	4.315	4.
11	9	RIDE	4.9	5	5.5	5.4	5.4	5.6	5.253	5.308	5.609	5.806	5.
12	11	Trackless Trolley	12.757	12.913	13.057	13.444	13.479	13.323	13.311	13.142	14.393	14.622	13.
13	10	TOTAL	1166.974	1191.639	1204.725	1247.105	1244.755	1246.129	1243.952	1223.323	1310.764	1244.453	1241.
14													
15													
16													
17													
18													

- 4 variables: transportation mode, year, month, and monthly weekday average number of trips.
- The first row in the excel sheet is a title, so we skip this row when reading the data in.
- The NA character is “NA”, not blank space which is the default value.

Do this first

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
mbta = pd.read_excel("mbta.xlsx", skiprows = [0], na_values = "NA")
```

```
#show the first 6 rows
print(mbta.head(6))
```

```
##    Unnamed: 0              mode  2007-01   ...     2011-08  2011-09  2011-10
## 0           1  All Modes by Qtr      NaN   ...         NaN  1290.549      NaN
## 1           2              Boat    4.000   ...       6.733     5.003    4.484
## 2           3               Bus  335.819   ...     353.793   388.271  398.456
## 3           4     Commuter Rail  142.200   ...     130.616   136.901  128.720
## 4           5        Heavy Rail  435.294   ...     508.145   550.137  554.932
## 5           6        Light Rail  227.231   ...     220.164   244.949  237.768
##
## [6 rows x 60 columns]
```

```
#show the last 5 rows
print(mbta.tail())
```

```
##     Unnamed: 0               mode    ...      2011-09   2011-10
## 6            7       Pct Chg / Yr    ...        0.043     0.032
## 7            8        Private Bus    ...        2.843     2.967
## 8            9               RIDE    ...        8.318     8.598
## 9           10  Trackless Trolley    ...       12.332    12.297
## 10          11              TOTAL    ...     1348.754  1348.222
##
## [5 rows x 60 columns]
```

```
#display all the column names
print(mbta.columns)
```

```
## Index(['Unnamed: 0', 'mode', '2007-01', '2007-02', '2007-03', '2007-04',
##        '2007-05', '2007-06', '2007-07', '2007-08', '2007-09', '2007-10',
##        '2007-11', '2007-12', '2008-01', '2008-02', '2008-03', '2008-04',
##        '2008-05', '2008-06', '2008-07', '2008-08', '2008-09', '2008-10',
##        '2008-11', '2008-12', '2009-01', '2009-02', '2009-03', '2009-04',
##        '2009-05', '2009-06', '2009-07', '2009-08', '2009-09', '2009-10',
##        '2009-11', '2009-12', '2010-01', '2010-02', '2010-03', '2010-04',
##        '2010-05', '2010-06', '2010-07', '2010-08', '2010-09', '2010-10',
##        '2010-11', '2010-12', '2011-01', '2011-02', '2011-03', '2011-04',
##        '2011-05', '2011-06', '2011-07', '2011-08', '2011-09', '2011-10'],
##       dtype='object')
```

- 1st column enumerates rows. Rows are identified by mode of transportation. 1st column is unnecessary.
- 1st row is a quarterly aggregation. Not needed.
- The last row (11th) has totals. Not needed.
- 7th row has % change in the year. Not needed.

```
#drop 1st, 7th and 11th rows and the first column
mbta.drop(index = [0,6,10], columns = 'Unnamed: 0', inplace = True)
print(mbta)
```

```
##                   mode  2007-01  2007-02   ...     2011-08  2011-09  2011-10
## 1                 Boat    4.000    3.600   ...       6.733    5.003    4.484
## 2                  Bus  335.819  338.675   ...     353.793  388.271  398.456
## 3        Commuter Rail  142.200  138.500   ...     130.616  136.901  128.720
## 4           Heavy Rail  435.294  448.271   ...     508.145  550.137  554.932
## 5           Light Rail  227.231  240.262   ...     220.164  244.949  237.768
## 7          Private Bus    4.772    4.417   ...       2.655    2.843    2.967
## 8                 RIDE    4.900    5.000   ...       8.071    8.318    8.598
## 9   Trackless Trolley    12.757   12.913   ...      11.091   12.332   12.297
##
## [8 rows x 59 columns]
```

- Variables are mode of transportation, year, month and monthly average number of passengers.
- All column names, except for the first one, `mode`, are values of year and month combined.
- To correct this we use the `melt()` function

```
# melt all column name values, except the first one (the index),
#into the "year_month" column with the corresponding values
# in the column "NrPassengers"
#
mbta2 = mbta.melt('mode', var_name = "year_month", value_name = "NrPassengers")
print(mbta2.head(10))
```

```
##                    mode year_month  NrPassengers
## 0                  Boat    2007-01         4.000
## 1                   Bus    2007-01       335.819
## 2         Commuter Rail    2007-01       142.200
## 3            Heavy Rail    2007-01       435.294
## 4            Light Rail    2007-01       227.231
## 5           Private Bus    2007-01         4.772
## 6                  RIDE    2007-01         4.900
## 7     Trackless Trolley    2007-01        12.757
## 8                  Boat    2007-02         3.600
## 9                   Bus    2007-02       338.675
```

- `year_month` has values of 2 variables. Keep the year in one column and month in another column.
- We separate them using `str.split()`.

```
# new data frame with split value columns
new = mbta2['year_month'].str.split('-', n = 1, expand = True)
mbta3 = mbta2
```

```
# making separate columns from new data frame
mbta3['year'] = new[0]
mbta3['month'] = new[1]
```

```
#drop year_month
mbta3.drop(columns = ['year_month'], inplace = True)
print(mbta3.head())
```

```
##             mode  NrPassengers  year month
## 0           Boat         4.000  2007    01
## 1            Bus       335.819  2007    01
## 2  Commuter Rail       142.200  2007    01
## 3     Heavy Rail       435.294  2007    01
## 4     Light Rail       227.231  2007    01
```

```
print(mbta3.dtypes)
```

```
## mode                object
## NrPassengers    float64
## year                object
## month               object
## dtype: object
```

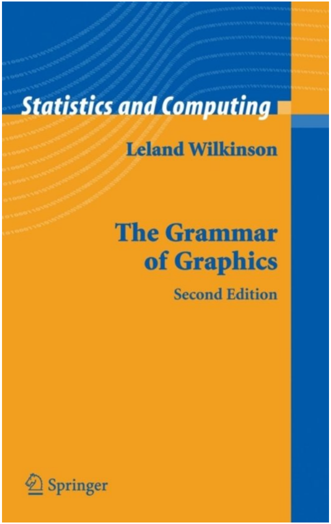
```
#make columns year and month numeric
mbta3['year'] = pd.to_numeric(mbta3['year'])
mbta3['month'] = pd.to_numeric(mbta3['month'])
print(mbta3.dtypes)
```

```
## mode                object
## NrPassengers    float64
## year                  int64
## month                 int64
## dtype: object
```

- The data is tidy and ready to be explored.

Visualisation using `plotnine`

- `plotnine` is a Grammar-of-Graphics-inspired package written by Hassan Kibirige.
- It brings the advantages of `R`'s `ggplot2` to `Python`: less coding and more meaningful syntax.



Statistics and Computing

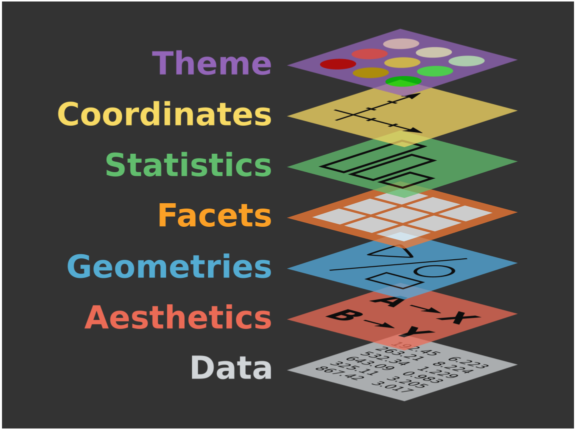
Leland Wilkinson

The Grammar of Graphics

Second Edition

Springer

"A grammar of graphics is a tool that enables us to concisely describe the components of a graphic. Such a grammar allows us to move beyond named graphics (e.g., the "scatterplot") and gain insight into the deep structure that underlies statistical graphics." H.Wickham in 'A Layered Grammar of Graphics'.



Using `plotnine` to visualise data

- The function `ggplot()`, in the package `plotnine`, is used to visualise data.
- The basic use is

(ggplot(myData, aes = (myMapping)) + myGeometryLayer)

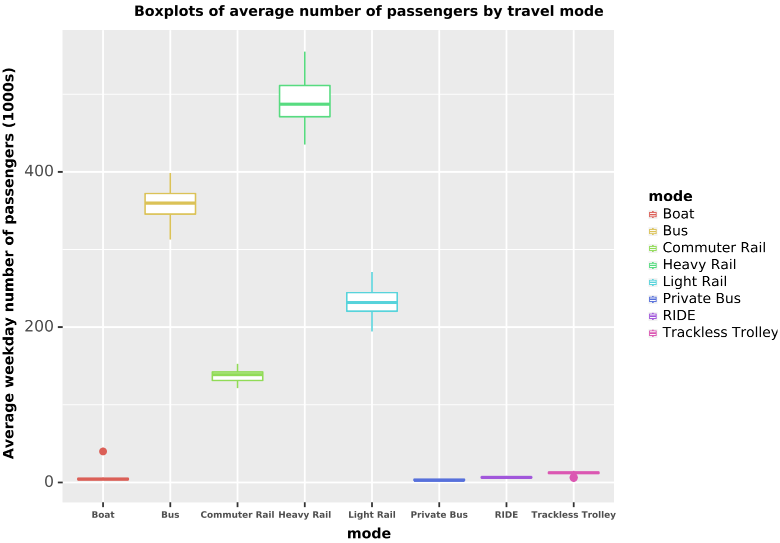
- myData: data frame with variables to use in plot.
- myMapping: mapping from the data to the aesthetics (visual dimension) in the graph. For example, the mapping can be `x = Varx, y = Vary` for a scatter plot of `Vary` vs. `Varx`.
- myGeometryLayer: specify what you want, points, lines, boxes, etc. e.g.: `geom_point()` for a scatter plot, `geom_line` for a line plot, etc.
- One can add many layers to the basic ggplot object created with the `ggplot()` function.

In order to access the functions in `plotnine` first we execute

```
from plotnine import *
```

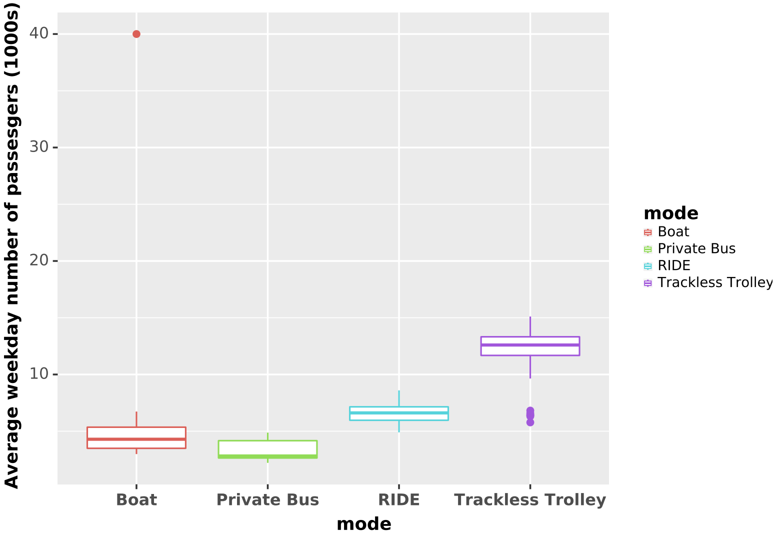
- In `plotnine` a plot is an object which can be re-called and modified.

```
fig = (
ggplot(mbta3, aes(x = 'mode', y = 'NrPassengers', color = 'mode'))
 + geom_boxplot()
 + ylab("Average weekday number of passengers (1000s)")
 + ggtitle("Boxplots of average number of passengers by travel mode")
 + theme(axis_text_x=element_text(rotation=0,ha='center',size=5,weight='bold'))
 + theme(title = element_text(size=8, weight='bold'))
 + theme(legend_text = element_text(size = 7.8))
 + theme(legend_key_size=5)
 + theme(subplots_adjust={'right': 0.8})
 )
#print(fig)#if you want to view the figure
```


Boxplots of average number of passengers by travel mode
Average weekday number of passengers (1000s)
0
200
400
Boat
Bus
Commuter Rail
Heavy Rail
Light Rail
Private Bus
RIDE
Trackless Trolley
mode
mode
Boat
Bus
Commuter Rail
Heavy Rail
Light Rail
Private Bus
RIDE
Trackless Trolley

- Most trips were made by heavy rail, bus, light rail and commuter rail, in descending order.
- Number of passengers are on a different scale for boat, private bus and trackless trolley. Plot them separately

```
aux = mbta3[
      mbta3['mode'].isin(['Boat', 'Private Bus', 'RIDE', 'Trackless Trolley'])
      ]
#
fig = (
ggplot(aux, aes(x = 'mode' , y = 'NrPassengers', color = 'mode'))
 + geom_boxplot()
 + ylab("Average weekday number of passesgers (1000s)")
 + theme(axis_text_x=element_text(rotation=0,ha='center',size=9,weight='bold'))
 + theme(title = element_text(size=10, weight='bold'))
 + theme(legend_text = element_text(size = 7.8))
 + theme(legend_key_size=5)
 + theme(subplots_adjust={'right': 0.8})
 )
#print(fig)
```



Average weekday number of passesgers (1000s)
40
30
20
10
Boat
Private Bus
RIDE
Trackless Trolley
mode
mode
Boat
Private Bus
RIDE
Trackless Trolley

- There is a very large observation for Boat. Let us find out when it was observed.

```
#pb contains NrPassengers by boat only
pb = mbta3.loc[mbta3['mode'] =='Boat', 'NrPassengers']
x = mbta3.loc[
     (mbta3['mode'] == 'Boat') &  (mbta3['NrPassengers'] == max(pb)), : ]
print(x)
```

```
##     mode  NrPassengers  year  month
## 16  Boat          40.0  2007      3
```

- The unusual observation occurred in March 2007.

- Look at the distribution of the other values of number of passengers who traveled by boat

```
pb1 = pb[pb < max(pb)]
print(pb1.describe())
```

```
## count    57.000000
## mean      4.455123
## std       1.134642
## min       2.985000
## 25%       3.488000
## 50%       4.285000
## 75%       5.189000
## max       6.733000
## Name: NrPassengers, dtype: float64
```

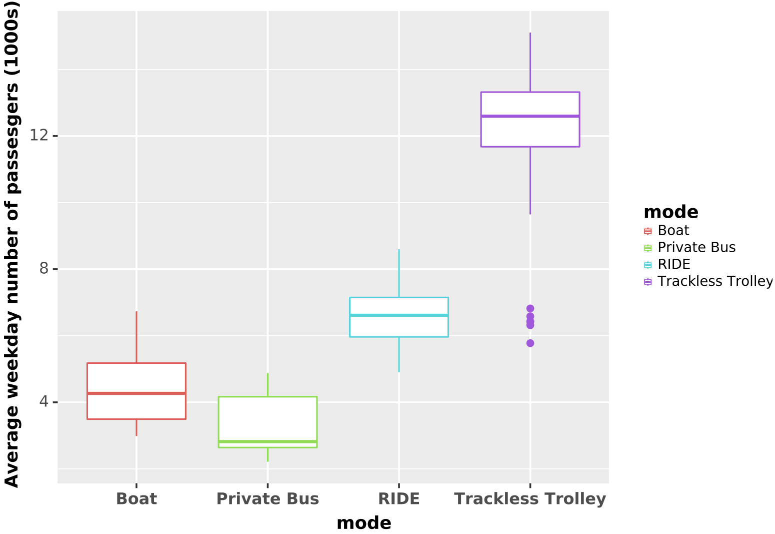
- No big event happened in Boston in March 2007.
- We conclude that it's quite likely the person who entered the data added an extra zero.
- Change this value to 4. Check with data originators and report.

```
mbta3.loc[
    (mbta3['mode'] == 'Boat') & (mbta3['NrPassengers'] == 40) ,
    'NrPassengers' ] = 4
#
x=mbta3.loc[mbta3['mode'] == 'Boat', 'NrPassengers']
print(x.describe())
```

```
## count    58.000000
## mean      4.447276
## std       1.126232
## min       2.985000
## 25%       3.494000
## 50%       4.268000
## 75%       5.178000
## max       6.733000
## Name: NrPassengers, dtype: float64
```

- Let us see the box plots again

```
aux = mbta3[
      mbta3['mode'].isin(['Boat', 'Private Bus', 'RIDE', 'Trackless Trolley'])
      ]
#
fig = (
ggplot(aux, aes(x = 'mode' , y = 'NrPassengers', color = 'mode'))
 + geom_boxplot()
 + ylab("Average weekday number of passesgers (1000s)")
 + theme(axis_text_x=element_text(rotation=0,ha='center',size=9,weight='bold'))
 + theme(title = element_text(size=10, weight='bold'))
 + theme(legend_text = element_text(size = 7.8))
 + theme(legend_key_size=5)
 + theme(subplots_adjust={'right': 0.8})
)
#print(fig)
```


Average weekday number of passesgers (1000s)
12
8
4
Boat
Private Bus
RIDE
Trackless Trolley
mode
mode
Boat
Private Bus
RIDE
Trackless Trolley

- There are some unusually low values for the number of passengers travelling by trackless trolley.

```
#this is all the data for trackless trolley only
trtr = mbta3.loc[mbta3['mode'] == 'Trackless Trolley', :]
aux = trtr.sort_values('NrPassengers')
print(aux.head(n=10))
```

```
##                    mode  NrPassengers  year  month
## 383  Trackless Trolley         5.777  2010     12
## 351  Trackless Trolley         6.316  2010      8
## 375  Trackless Trolley         6.415  2010     11
## 359  Trackless Trolley         6.436  2010      9
## 343  Trackless Trolley         6.584  2010      7
## 367  Trackless Trolley         6.819  2010     10
## 207  Trackless Trolley         9.645  2009      2
## 439  Trackless Trolley        11.060  2011      7
## 447  Trackless Trolley        11.091  2011      8
## 391  Trackless Trolley        11.104  2011      1
```

- The unusually low observations for trackless trolley occurred in the second semester of 2010.
- Don’t change or delete, but be aware.

- We will plot the data on numbers of passengers against time.
- Create a new variable, date, of “datetime” type.

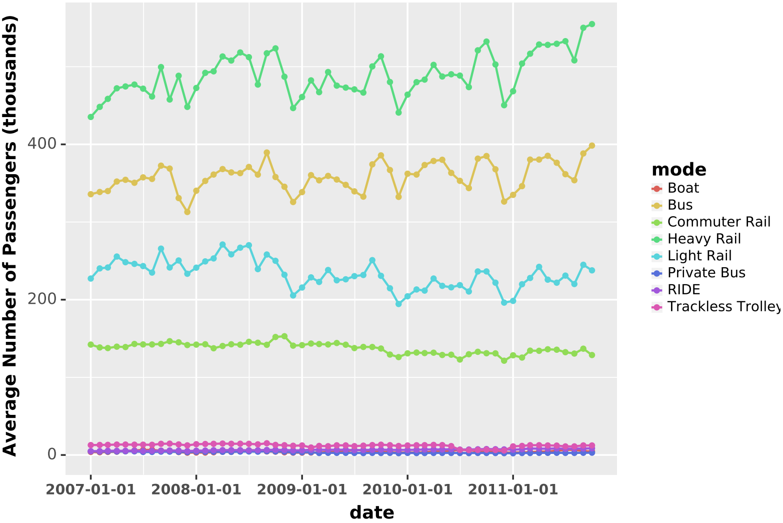
```
#create a column named day with 1s
mbta3 = mbta3.assign(day = 1)
mbta3['date'] = pd.to_datetime(mbta3[['year', 'month', 'day']])
print(mbta3.head())
```

```
##             mode  NrPassengers  year  month  day       date
## 0           Boat         4.000  2007      1    1 2007-01-01
## 1            Bus       335.819  2007      1    1 2007-01-01
## 2  Commuter Rail       142.200  2007      1    1 2007-01-01
## 3     Heavy Rail       435.294  2007      1    1 2007-01-01
## 4     Light Rail       227.231  2007      1    1 2007-01-01
```

```
print(mbta3.dtypes)
```

```
## mode                    object
## NrPassengers           float64
## year                     int64
## month                    int64
## day                      int64
## date            datetime64[ns]
## dtype: object
```

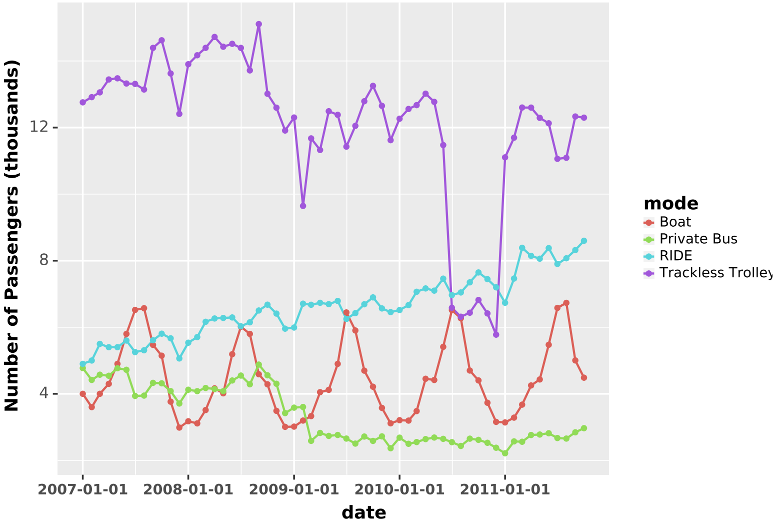
```
fig = (
  ggplot(mbta3, aes(x = 'date', y = 'NrPassengers', color = 'mode')) +
  geom_line(size=0.7) +
  geom_point(size=1)+
  ylab("Average Number of Passengers (thousands)")  +
  theme(axis_text_x=element_text(rotation=0,ha='center',size=8,weight='bold'))
  theme(title = element_text(size=10, weight='bold')) +
  theme(legend_text = element_text(size = 7.8)) +
  theme(legend_key_size=5) +
  theme(subplots_adjust={'right': 0.8})
  )
#print(fig)
```



Average Number of Passengers (thousands)
0
200
400
2007-01-01
2008-01-01
2009-01-01
2010-01-01
2011-01-01
date
mode
Boat
Bus
Commuter Rail
Heavy Rail
Light Rail
Private Bus
RIDE
Trackless Trolley

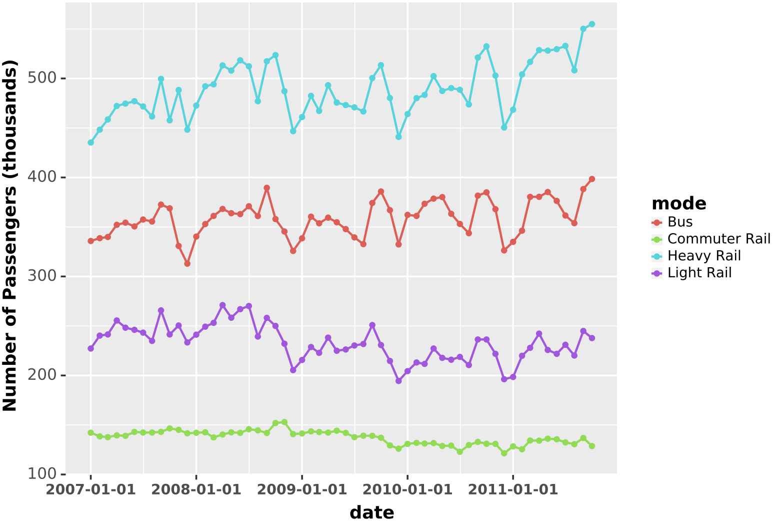
- Different scales for the data corresponding to the number of passengers travelling by boat, car, trackless trolley and private bus.

```
#aux has already been defined but here it goes again
aux = mbta3[ mbta3['mode'].isin(['Boat', 'Private Bus', 'RIDE',
         'Trackless Trolley']) ]
fig = (
  ggplot(aux, aes(x = 'date', y = 'NrPassengers', color = 'mode')) +
  geom_line(size=0.7) +
  geom_point(size=1)+
  ylab("Number of Passengers (thousands)")  +
  theme(axis_text_x=element_text(rotation=0,ha='center',size=8,weight='bold'))
  theme(title = element_text(size=10, weight='bold')) +
  theme(legend_text = element_text(size = 7.8)) +
  theme(legend_key_size=5) +
  theme(subplots_adjust={'right': 0.8})
 )
#print(fig)
```


Number of Passengers (thousands)
12
8
4
2007-01-01
2008-01-01
2009-01-01
2010-01-01
2011-01-01
date
mode
Boat
Private Bus
RIDE
Trackless Trolley

- Strong seasonal component in number of passengers travelling by boat.
- The use of RIDE seems to be steadily increasing during time.
- The use of private bus and trackless trolley had a sharp decrease since 2009 and something unusual made the use of trackless trolley dramatically decrease in the second half of 2010.

```
aux2 = mbta3[ mbta3['mode'].isin(['Bus', 'Commuter Rail', 'Heavy Rail',
         'Light Rail']) ]
#
fig = (
  ggplot(aux2, aes(x = 'date', y = 'NrPassengers', color = 'mode')) +
  #make lines thicker
  geom_line(size=0.7) +
  #make points bigger
  geom_point(size=1)+
  ylab("Number of Passengers (thousands)")  +
  theme(axis_text_x=element_text(rotation=0,ha='center',size=8,weight='bold'))
  theme(title = element_text(size=10, weight='bold')) +
  theme(legend_text = element_text(size = 7.8)) +
  theme(legend_key_size=5) +
  theme(subplots_adjust={'right': 0.8})
 )
#print(fig)
```


Number of Passengers (thousands)
500
400
300
200
100
2007-01-01
2008-01-01
2009-01-01
2010-01-01
2011-01-01
date
mode
Bus
Commuter Rail
Heavy Rail
Light Rail

- The number of passengers travelling by light and commuter rail has decreased since 2009.
- There seems to be an upwards trend on number of passengers travelling by heavy rail and perhaps by bus as well.

Please, give us your comments in

http://www.smartsurvey.co.uk/s/BS0FU/

B S ZERO F U