# Data Structure

## Contents



## Small Data

The following simple dataset contains GDP (in millions of dollars) of three countries for 1960 and 2017. (Source: Worldbank

```
gdp = data.frame(c('USA','China','Japan'),
           c(543300, 59716, 44307),
           c(19390604, 12237700, 4872136))
names(gdp) = c('Country','1960','2017')
gdp
```

```
##   Country   1960     2017
## 1     USA 543300 19390604
## 2   China  59716 12237700
## 3   Japan  44307  4872136
```

### Wide to Tall

Above data has a wide format. This data is made tidy by converting to a tall format

```
library(tidyr)
gdp_tall =
  gdp %>%
  gather('Year','GDP',2:3)
gdp_tall
```

```
##   Country Year      GDP
## 1     USA 1960   543300
## 2   China 1960    59716
## 3   Japan 1960    44307
## 4     USA 2017 19390604
## 5   China 2017 12237700
## 6   Japan 2017  4872136
```

library(tidyr) has introduced two new verbs that perform pretty much the same function.

```
library(tidyr)
gdp_tall =
  gdp %>%
  pivot_longer(cols = 2:3, names_to = 'Year',values_to = 'GDP')
gdp_tall
```

```
## # A tibble: 6 x 3
##   Country Year       GDP
##   <chr>   <chr>    <dbl>
## 1 USA     1960    543300
## 2 USA     2017  19390604
## 3 China   1960     59716
## 4 China   2017  12237700
## 5 Japan   1960     44307
## 6 Japan   2017   4872136
```

### Tall to Wide

Although not common, there are circumstances when a wide format may be called for.

```
gdp_wide =
  gdp_tall %>%
  spread('Year','GDP')
gdp_wide
```

```
## # A tibble: 3 x 3
##   Country `1960`   `2017`
##   <chr>    <dbl>    <dbl>
## 1 China    59716 12237700
## 2 Japan    44307  4872136
## 3 USA     543300 19390604
```

Using pivot_wider() instead of spread()

```
gdp_wide =
  gdp_tall %>%
  pivot_wider(names_from = 'Year',values_from  = 'GDP')
gdp_wide
```

```
## # A tibble: 3 x 3
##   Country `1960`   `2017`
##   <chr>    <dbl>    <dbl>
## 1 USA     543300 19390604
## 2 China    59716 12237700
## 3 Japan    44307  4872136
```

## Survey Data

Survey data usually has a wide format, with each row corresponding to a respondent and each column containing responses to each question. Unfortunately, many functions only operate on data that is in a tall format, which Hadley Wickham refers to as a 'tidy' format. (I would caution you against the logical assumption that data that is wide, is untidy. This may not always be true.)

Let us begin by simulating some survey data that includes responses to five survey items designed to measure coupon proneness. Respondents select a number to indicate their level of agreement where a small number

indicates strongly disagree and a high number indicates srongly agree. The first two items are measured on a 1-5 scale and the next three on a 1-7 scale.

Here are the five items in the Coupon proneness scale
1. Redeeming coupons makes me feel good.
2. I enjoy clipping coupons out of the newspapers.
3. When I use coupons, I feel that I am getting a good deal.
4. I enjoy using coupons, regardless of the amount I save while doing so.
5. Beyond the money I save, redeeming coupons gives me pleasure.

```
set.seed(617)
coupons = data.frame(id = 1:100,
                     c1 = round(runif(100,1,5),0),
                     c2 = round(runif(100,1,5),0),
                     c3 = round(runif(100,1,7),0),
                     c4 = round(runif(100,1,7),0),
                     c5 = round(runif(100,1,7),0))
head(coupons)
```

```
##   id c1 c2 c3 c4 c5
## 1  1  3  4  2  5  4
## 2  2  4  4  3  5  5
## 3  3  5  3  3  6  5
## 4  4  2  3  3  5  7
## 5  5  4  5  3  3  6
## 6  6  2  4  4  6  4
```

As noted above, certain functions and techniques only work with data in a tall format. To illustrate, it is not possible to generate a side-by-side bar chart for the coupon data using ggplot2 as the latter requires the data to be in a tall format. So, let us first convert the data to a tall format.

## Wide to Tall

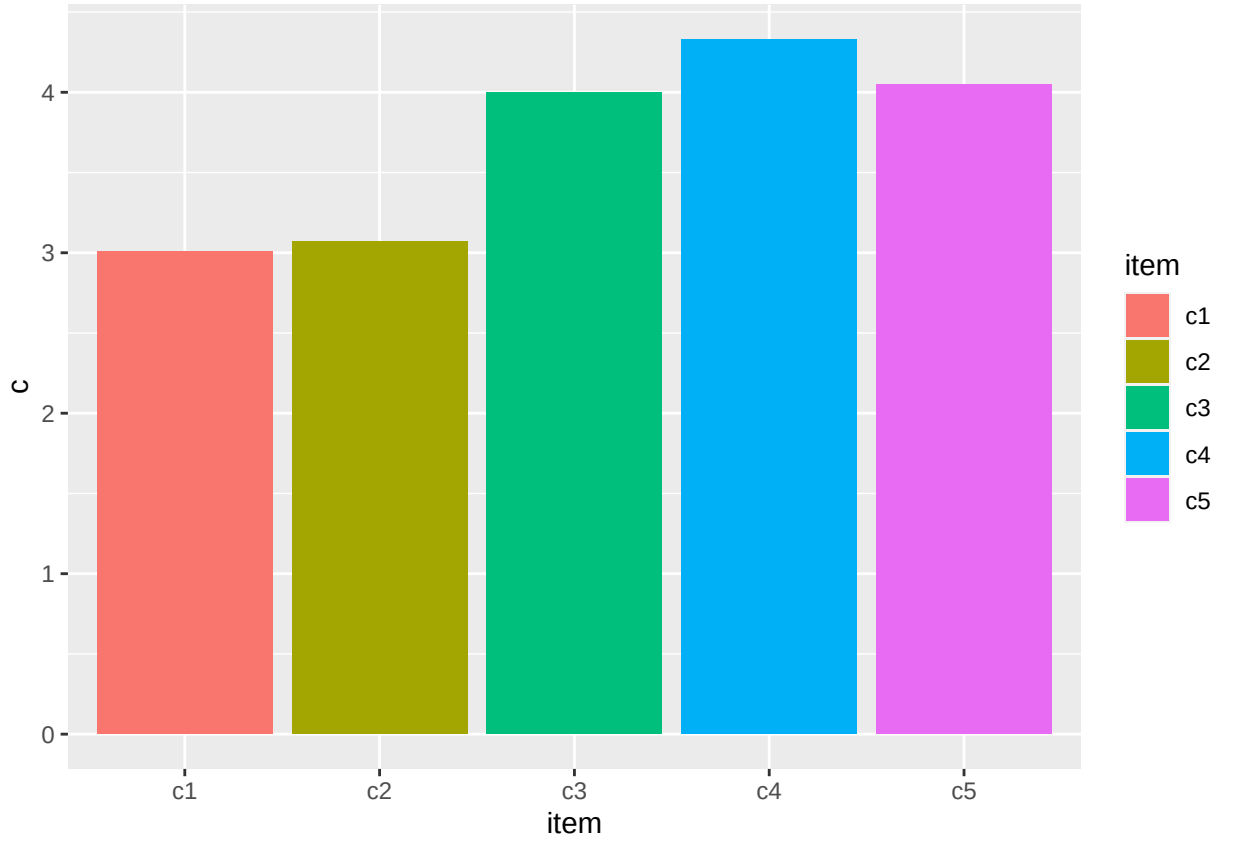
```
library(tidyr)
coupon_tall = gather(coupons,key = 'item',value = 'c',2:6)
head(coupon_tall, 10)
```

```
##    id item c
## 1   1   c1 3
## 2   2   c1 4
## 3   3   c1 5
## 4   4   c1 2
## 5   5   c1 4
## 6   6   c1 2
## 7   7   c1 1
## 8   8   c1 2
## 9   9   c1 5
## 10 10   c1 4
```

## Chart

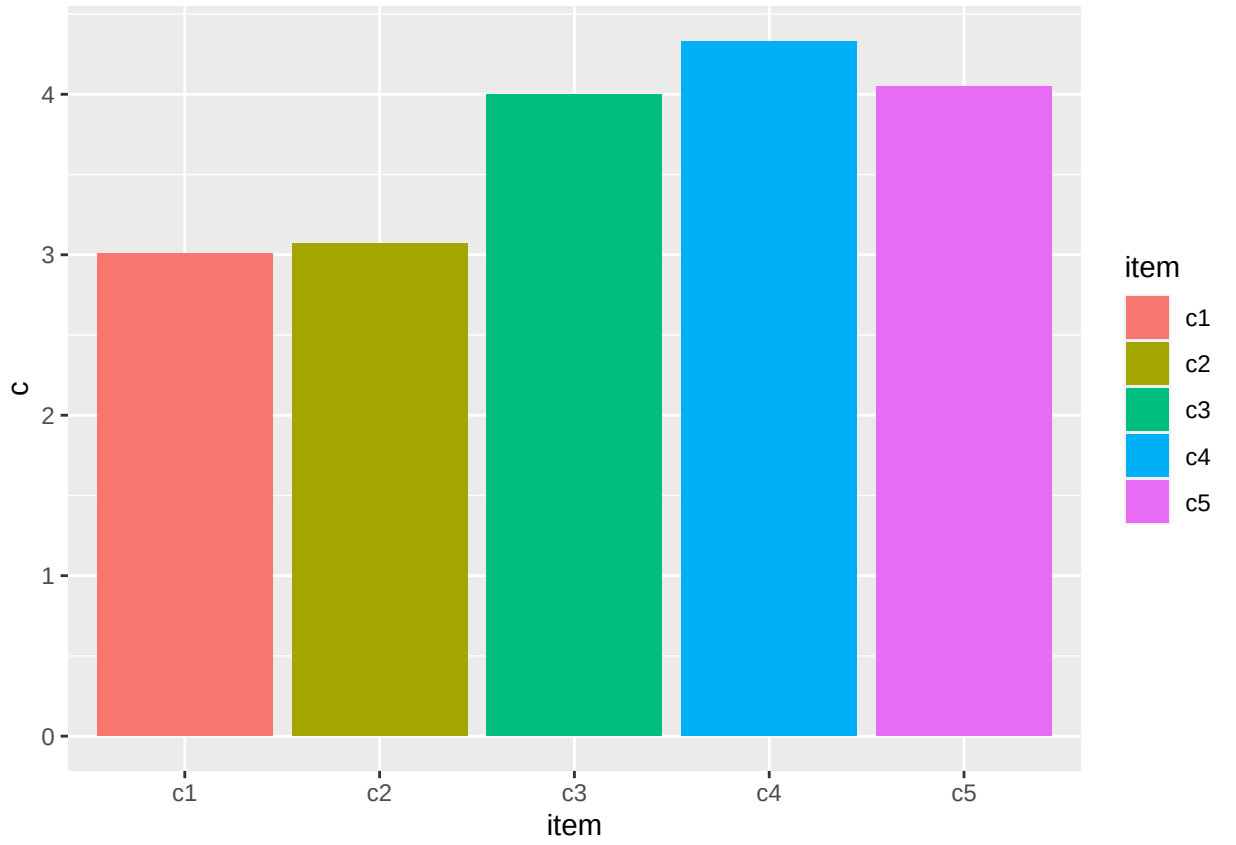
Now, let us create a bar chart.

```
library(ggplot2)
ggplot(data=coupon_tall,aes(x=item,y=c,fill=item))+
  geom_bar(stat='summary',position='dodge',fun='mean')
```




We can also use piped operations to seamlessly connect restructuring and plotting. Also, we will use pivot_longer() instead of gather()

```
coupons %>%
  pivot_longer(cols = c1:c5, names_to = 'item',values_to='c')%>%
  ggplot(aes(x=item,y=c,fill=item))+
  geom_bar(stat='summary',position='dodge',fun='mean')
```




## Crime Data

Let us examine a dataset on [crime](https://www.ucrdatatool.gov/Search/Crime/State/RunCrimeTrendsInOneVar.cfm (downloaded on July 24, 2018). Note the wide format of the data does not lend itself to many functions such as ggplot()

```
data = read.csv('C:/Users/vlala/Downloads/CrimeTrendsInOneVar.csv',header = T,skip = 4,nrows=55)
tail(data)
```

```
##    Year Alabama Alaska Arizona Arkansas California Colorado Connecticut
## 50 2009   21194   4424   28128    14905     174934    17022       10588
## 51 2010   18363   4537   26528    14711     164133    16339       10083
## 52 2011   20166   4416   26789    14173     154943    16085        9889
## 53 2012   21693   4412   28077    13851     160944    15951       10183
## 54 2013   20834   4709   27576    13705     154739    16099        9439
## 55 2014   20727   4684   26916    14243     153709    16554        8522
##    Delaware District.of.Columbia Florida Georgia Hawaii Idaho Illinois Indiana
## 50     5713                8089  113541   42073   3550  3805    64185   21455
## 51     5608                8026  101969   39068   3603  3464    57132   20983
## 52     5144                7433   98198   36762   3465  3202    54523   21619
## 53     5048                7866   94087   37675   3378  3348    53556   22544
## 54     4633                8415   91993   37519   3585  3471    51956   23627
## 55     4576                8199  107521   38097   3680  3468    47663   24099
##    Iowa Kansas Kentucky Louisiana Maine Maryland Massachusetts Michigan
## 50 8485  11460    11000     28878  1580    33625         30503    49825
## 51 8191  10602    10604     25241  1621    31607         30737    48693
```

```
## 52 7883  10209    10465     25373  1638    28817         28232    43731
## 53 8167  10292     9852     22839  1626    28086         27047    44962
## 54 8443   9928     9280     24127  1761    28235         27264    44757
## 55 8497  10123     9340     23934  1700    26661         26399    42348
##    Minnesota Mississippi Missouri Montana Nebraska Nevada New.Hampshire
## 50     12874        8451    29513    2798     5199  18639          2125
## 51     12515        7999    27440    2733     5093  17929          2204
## 52     12323        8009    26888    2755     4672  15452          2864
## 53     12419        7769    27189    2803     4802  16763          2841
## 54     12710        8303    26216    2924     4949  16888          2952
## 55     12505        8338    26856    3313     5275  18045          2602
##    New.Jersey New.Mexico New.York North.Carolina North.Dakota  Ohio Oklahoma
## 50      27113      12709    75110          37946         1723 38305    18560
## 51      27055      12147    76492          34679         1548 36306    18100
## 52      27203      11904    77463          33421         1699 35218    17311
## 53      25727      11660    79535          34464         1723 34827    18102
## 54      25748      12990    77563          33587         1979 33722    17187
## 55      23346      12459    75398          32767         1960 33030    15744
##    Oregon Pennsylvania Rhode.Island South.Carolina South.Dakota Tennessee
## 50   9968        48188         2678          30799         1777     41933
## 51   9648        46612         2709          27923         2196     38909
## 52   9643        46189         2586          27894         2105     38895
## 53   9638        45384         2657          26474         2701     41213
## 54   9536        42825         2710          24263         2733     38063
## 55   9224        40164         2313          24052         2786     39848
##     Texas Utah Vermont Virginia Washington West.Virginia Wisconsin Wyoming
## 50 121684 5998     837    18195      22412          5554     14650    1196
## 51 113231 5925     820    17184      21138          5586     14167    1117
## 52 104734 5547     925    16014      20152          5497     14268    1245
## 53 106475 5939     891    15676      20553          5943     16254    1161
## 54 108757 6644     775    16355      20223          5657     16118    1212
## 55 109414 6346     622    16340      20136          5588     16714    1142
```

### Tidy Data

```
data %>%
  pivot_longer(cols = Alabama:Wyoming, names_to = 'State', values_to = 'Number_of_Violent_Crimes')%>%
  head()
```

```
## # A tibble: 6 x 3
##    Year State      Number_of_Violent_Crimes
##   <int> <chr>                         <int>
## 1  1960 Alabama                        6097
## 2  1960 Alaska                          236
## 3  1960 Arizona                        2704
## 4  1960 Arkansas                       1924
## 5  1960 California                    37558
## 6  1960 Colorado                       2408
```

The same process but using gather()

```
data %>%
  gather('State','Number_of_Violent_Crimes',2:52)%>%
  head()
```

```
##   Year   State Number_of_Violent_Crimes
```

```
## 1 1960 Alabama                    6097
## 2 1961 Alabama                    5564
## 3 1962 Alabama                    5283
## 4 1963 Alabama                    6115
## 5 1964 Alabama                    7260
## 6 1965 Alabama                    6916
```

## Chart

Tidy data can be easily summarized and then plotted using ggplot2. Chart below depicts average crime from 1960-2014 for each State.

```
library(tidyr); library(dplyr); library(ggplot2)
data %>%
  pivot_longer(cols = Alabama:Wyoming, names_to = 'State', values_to = 'Number_of_Violent_Crimes')%>%
  group_by(State)%>%
  summarize(AverageViolentCrime = mean(Number_of_Violent_Crimes,na.rm=T))%>%
  ggplot(aes(x=reorder(State,X = AverageViolentCrime), y=AverageViolentCrime,fill=AverageViolentCrime))+
  geom_col()+scale_fill_continuous(low='white',high='red')+xlab('State')+ylab('Crime')+
  theme(axis.text.y = element_text(size = 6, hjust = .5, vjust = .5, face = "plain"))+
  coord_flip()
```