post01-joan-dai

Joan Dai

10/16/2017

Post 01 Homework

As someone completely new to coding, learning R as a language was not easy to say the least. For this project, I thought I would delve deeper into data cleaning and visualization, using `tidyr`, `ggplot` and base R to experiment with some simple data I have worked with previously on Excel.

This past year, I had a chance to study abroad in Japan. One of the things I became aware of when I was abroad was the price of produce compared to that in the United States! Whereas a whole 5-pound watermelon costs around 3 dollars here, it could cost at least as much as 1000 yen in Japan (approx. 10 dollars). After reading this article, I became interested in Japan's engel coefficient, and how it has increased over time despite deflationary pressures in the bigger economy. *By the way, the engel coefficient is the proportion of income spent on food. Engel's law says that as income increases, the proportion of income spent on food decreases even though actual expenditure increases.*

I want to plot the Engel Coefficient over time between income groups. Data will be taken from Statistics Bureau of Japan for years between 2003 and 2016 for each income quintile group. I have compiled it into an excel sheet called "income_deciles_engel". It can be found in the `data` file.

```
library(readr)
library(ggplot2)
library(tidyr)
```

Let's read in the data.

```
engel <- read_csv("~/stat133/stat133-hws-fall17/post01/data/income_deciles_engel.csv")
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## Parsed with column specification:
## cols(
##   X1 = col_integer(),
##   I = col_double(),
##   II = col_double(),
##   III = col_double(),
##   IV = col_double(),
##   V = col_double()
## )
```

```
engel <- data.frame(engel)
names(engel) <- c("year", "I", "II", "III", "IV", "V")
engel
```

```
##    year    I   II  III   IV    V
## 1  2016 27.9 27.8 27.2 25.7 22.9
## 2  2015 27.0 27.1 26.1 24.7 22.9
## 3  2014 26.3 25.8 24.9 24.1 21.6
## 4  2013 25.4 25.7 24.5 23.8 21.2
## 5  2012 25.0 25.8 25.0 23.1 21.5
## 6  2011 25.8 25.8 24.6 23.4 21.3
## 7  2010 25.7 24.8 24.4 23.3 20.9
## 8  2009 25.9 25.4 24.0 23.2 21.2
## 9  2008 25.6 24.4 24.2 23.2 21.1
## 10 2007 24.9 24.8 23.8 23.3 20.5
## 11 2006 25.1 25.2 24.0 23.4 20.5
## 12 2005 24.7 24.9 23.8 22.7 20.3
## 13 2004 25.0 25.4 23.8 23.2 20.4
## 14 2003 25.0 24.8 24.4 22.9 20.8
```

Unfortunately, this data is not tidy and we cannot really work with it yet. Tidy data is when there is an observation in every row, variables in every column. Furthermore, each value is in a cell. `ggplot` works best with tidy data, so we will need to clean it up before anything else. One way to do this is by using `tidyr`. We have not used this in class yet, so it should be interesting to learn. `tidyr` has two main functions - `gather()` and `spread()`.

`gather(data, key, value, ..., na.rm = FALSE, convert = FALSE, factor_key = FALSE)`

- `gather()` is a function to turn wide data into long data. It gets rid of multiple columns (these will be the variables you list at the end of the key and value arguments) and condenses it to a single variable (`key` = `varname`).

For our data set, this is what we can do.

```
engel <- gather(engel, key = quintiles, value = observation, I, II, III, IV, V)
head(engel)
```

```
##   year quintiles observation
## 1 2016         I        27.9
## 2 2015         I        27.0
## 3 2014         I        26.3
## 4 2013         I        25.4
## 5 2012         I        25.0
## 6 2011         I        25.8
```

For the sake of learning `spread()` function, we can use it to reverse what we did. You can see below that it is back to looking like the original data set.

`spread(data, key, value, fill = NA, convert = FALSE, drop = TRUE, sep = NULL)`

- `spread()` is a function to turn long data into wide data.

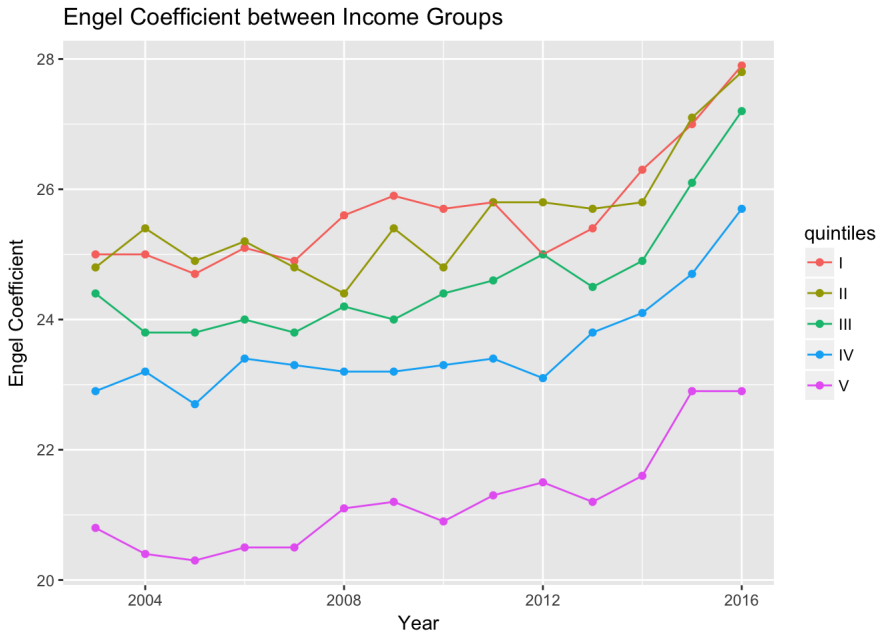
```
engel_wide <- spread(engel, key = quintiles, value = observation)
head(engel_wide)
```

```
##   year    I   II  III   IV    V
## 1 2003 25.0 24.8 24.4 22.9 20.8
## 2 2004 25.0 25.4 23.8 23.2 20.4
## 3 2005 24.7 24.9 23.8 22.7 20.3
## 4 2006 25.1 25.2 24.0 23.4 20.5
## 5 2007 24.9 24.8 23.8 23.3 20.5
## 6 2008 25.6 24.4 24.2 23.2 21.1
```

Now that we have our long data `engel`, let's try visualizing it with `ggplot`. Previously, we have mapped simple data to show one-to-one relationships (ex. scatterplots). I want to now experiment with different ways to visualize the data so that it provides a more holistic view. I will also be experimenting with different geom layers in ggplot.

Examples

```
ggplot(engel, aes(x = year, y = observation)) +
  geom_point(aes(color = quintiles)) + geom_line(aes(col = quintiles)) + xlab("Year") + ylab("Engel Coefficient")
+ ggtitle("Engel Coefficient between Income Groups")
```



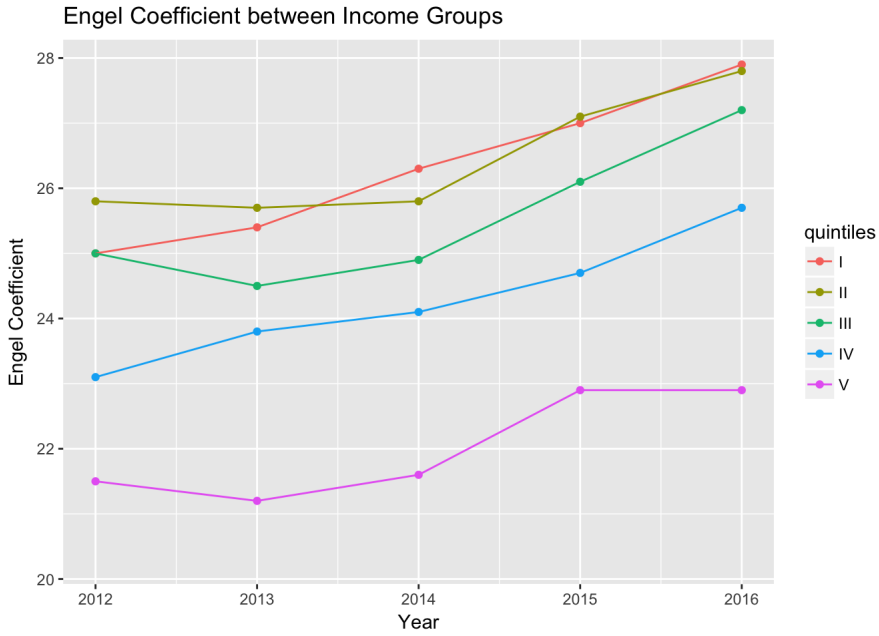
This line graph shows us the relative position of each income group's engel coefficient in relation to the others. The increasing trend appears to be a lot stronger in the lowest 3 income groups than the highest 2.

Suppose we are only interested in the change in engel coefficient within the last four years. We could limit the time frame we see by using `scale_x_continuous()`. Notice that R gives us a warning message since not all our data points are represented in the graph; we are only seeing a subset of it.

```
ggplot(engel, aes(x = year, y = observation)) +
  geom_point(aes(color = quintiles)) + geom_line(aes(col = quintiles)) + scale_x_continuous(limits = c(2012,2016))
+ xlab("Year") + ylab("Engel Coefficient") + ggtitle("Engel Coefficient between Income Groups")
```

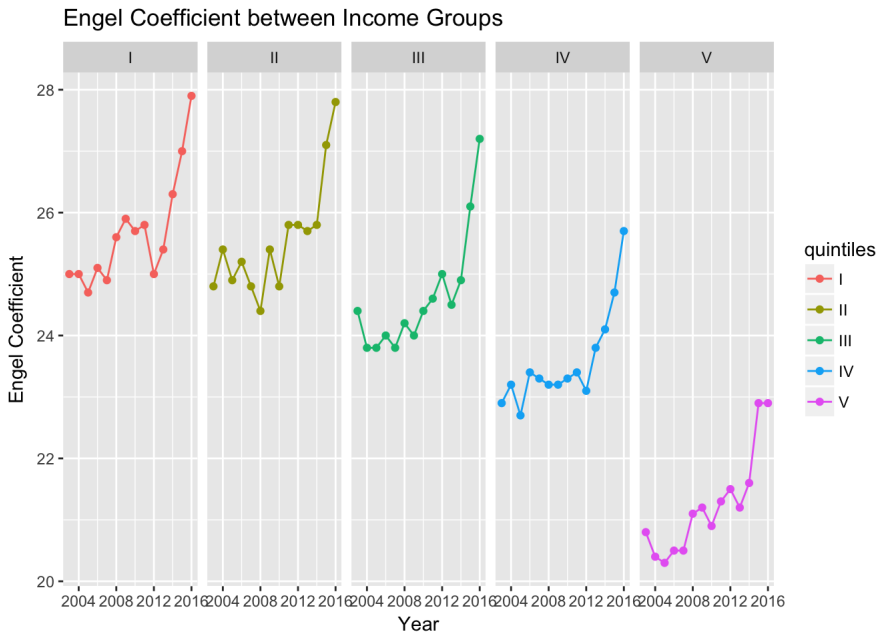
```
## Warning: Removed 45 rows containing missing values (geom_point).
```

```
## Warning: Removed 45 rows containing missing values (geom_path).
```



Faceting gives us...

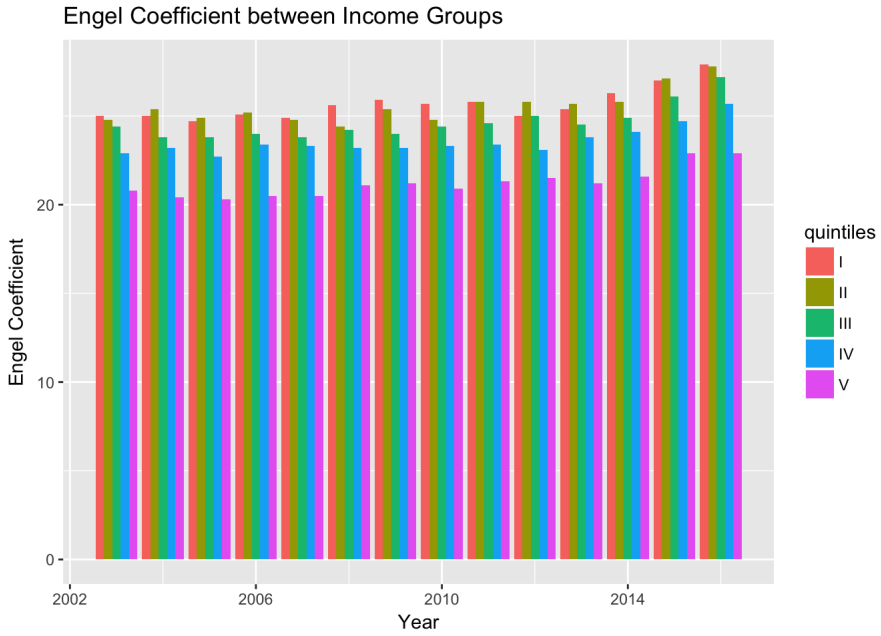
```
ggplot(engel, aes(x = year, y = observation)) +
  geom_point(aes(colour = quintiles)) + geom_line(aes(col = quintiles)) + facet_grid(~quintiles) +xlab("Year") + y
lab("Engel Coefficient") + ggtitle("Engel Coefficient between Income Groups")
```



If we facet by quintiles, we can see clearly the difference in engel coefficients between the income groups (as well as the general trend upwards in recent years).

What if we were to try using side-by-side barplots?

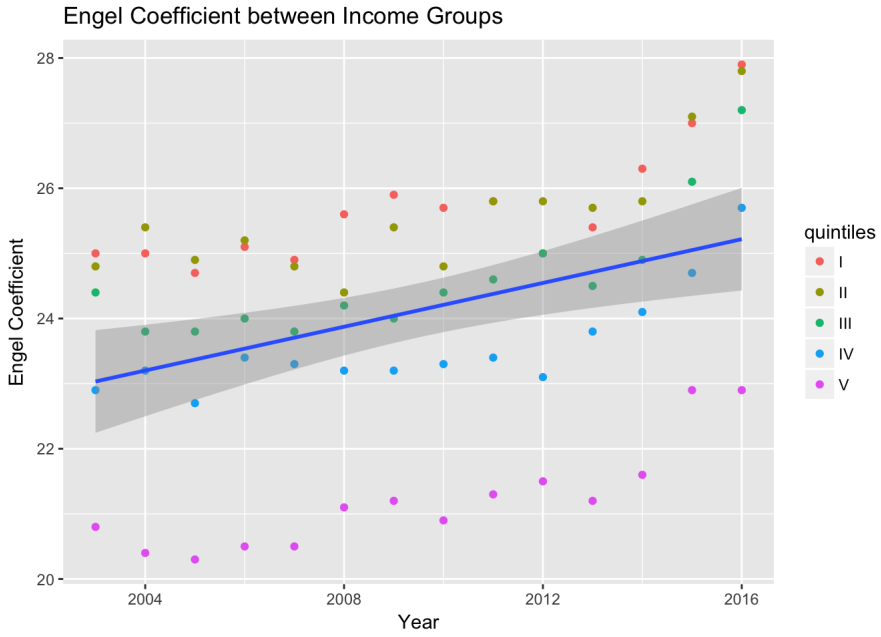
```
ggplot(engel, aes(x = year, y = observation, fill = quintiles)) +
  geom_bar(stat = "identity", position = "dodge") +xlab("Year") + ylab("Engel Coefficient") + ggtitle("Engel Coeff
icient between Income Groups")
```



This is not a very good way to visualize our data - it is much too noisy. The colors are clashing too, and it's overall very difficult to glean information from this chart. This definitely goes to show how important the way one presents data makes a difference in how the audience interprets it!

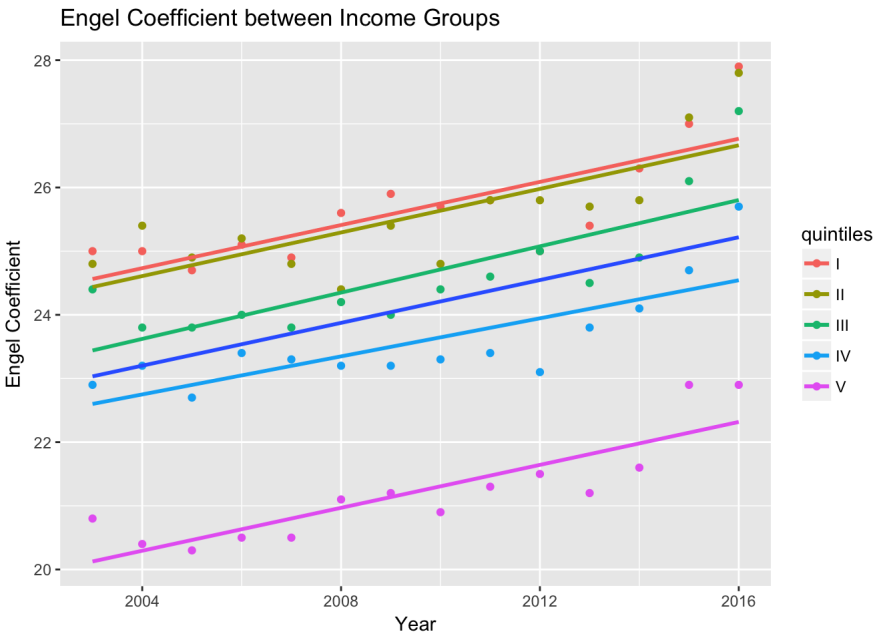
If we are interested in trying to predict the engel coefficient in the future, we could make a linear model.

```
ggplot(engel, aes(x = year, y = observation)) +
  geom_point(aes(colour = quintiles)) + geom_smooth(method = "lm") +xlab("Year") + ylab("Engel Coefficient") + ggt
itle("Engel Coefficient between Income Groups")
```



Alternatively, we can make linear models for each quintile group (and also include the previous general regression line). The slopes appear to be very similar across all income groups, but now we can see that they are not parallel. This shows that we can add multiple `geom_smooth()` aesthetic layers, just like how we can add multiple geom layers. How neat is that?

```
ggplot(engel, aes(x = year, y = observation, col = quintiles)) +
  geom_point() + geom_smooth(method = "lm", se = F) + geom_smooth(method = "lm", se = F, aes(group = 1))+ xlab("Ye
ar") + ylab("Engel Coefficient") + ggtitle("Engel Coefficient between Income Groups")
```



Now suppose we want to know the breakdown of expenditures of the bottom quintile last year. I have compiled the data again into an excel file called “expenditures_bottom20.csv”. This data represents monthly expenditures.

```
expenditures_bottom20 <- read_csv("~/stat133/stat133-hws-fall17/post01/data/expenditures_bottom20.csv")
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## Parsed with column specification:
## cols(
##   X1 = col_character(),
##   Food = col_integer(),
##   Housing = col_integer(),
##   Clothing = col_integer(),
##   Medical = col_integer(),
##   Transportation = col_integer(),
##   Education = col_integer(),
##   Recreation = col_integer(),
##   Other = col_integer(),
##   Unaccounted = col_integer()
## )
```

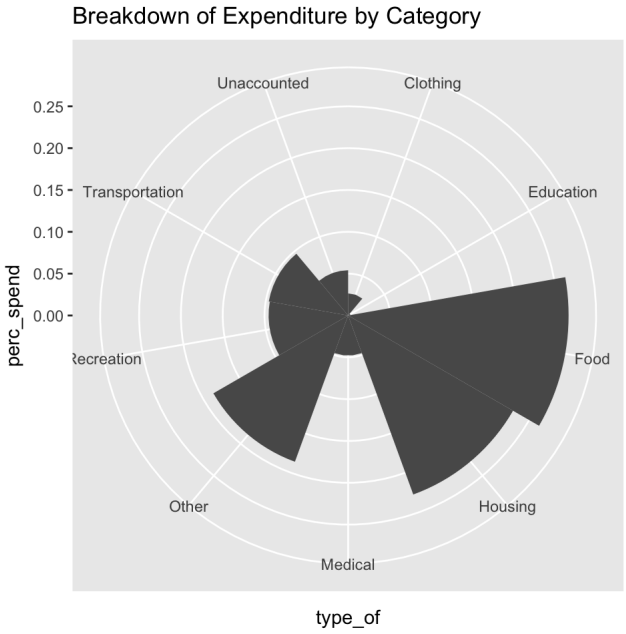
```
expenditures_bottom20
```

```
## # A tibble: 1 x 10
##      X1  Food Housing Clothing Medical Transportation Education
##   <chr> <int>   <int>    <int>   <int>          <int>     <int>
## 1     I 37346   32257     3738    6782          13649       407
## # ... with 3 more variables: Recreation <int>, Other <int>,
## #   Unaccounted <int>
```

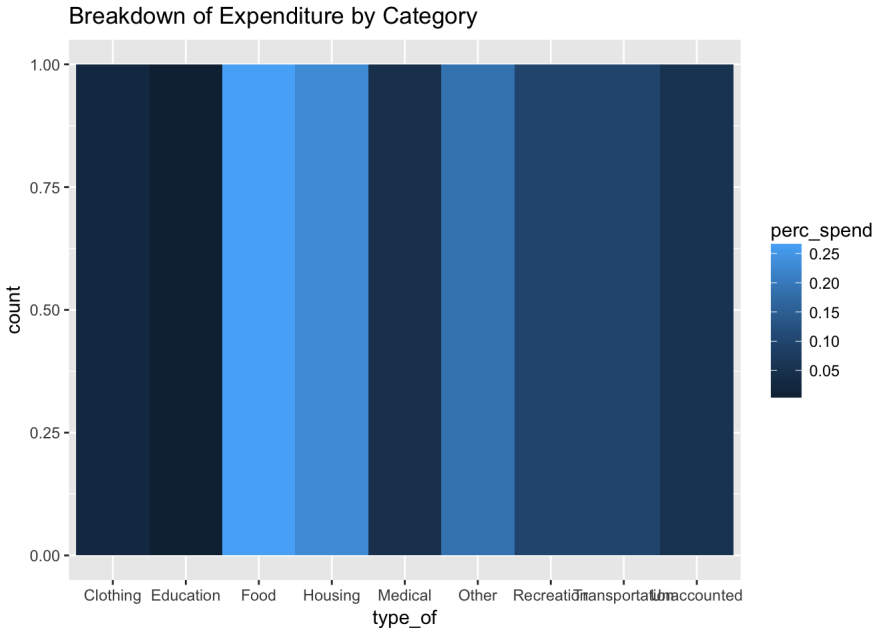
```
spending <- gather(expenditures_bottom20, key = type_of, value = perc_spend, Food, Housing, Clothing, Medical, Tra
nsportation, Education, Recreation, Other, Unaccounted)
spending[,3] <- spending[,3]/141667 #where 141667 is the average monthly income.

spending
```

```
## # A tibble: 9 x 3
##      X1        type_of  perc_spend
##   <chr>          <chr>       <dbl>
## 1     I           Food 0.263618203
## 2     I        Housing 0.227695935
## 3     I       Clothing 0.026385820
## 4     I        Medical 0.047872829
## 5     I Transportation 0.096345656
## 6     I      Education 0.002872934
## 7     I     Recreation 0.095110364
## 8     I          Other 0.186041915
## 9     I    Unaccounted 0.054056343
```

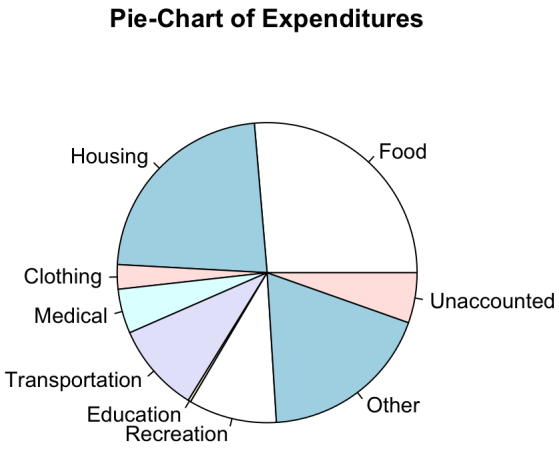
```
barexp1 <- ggplot(spending, aes(x = type_of, y = perc_spend)) + geom_bar(width = 1, stat = "identity")
pie1 <- barexp1 + coord_polar(theta = "x") + ggtitle("Breakdown of Expenditure by Category")
pie1
```




```
barexp2 <- ggplot(spending, aes(x = type_of, fill = perc_spend)) + geom_bar(width = 1)+ ggtitle("Breakdown of Expe
nditure by Category")
barexp2
```




```
pie(spending$perc_spend, labels = spending$type_of, main = "Pie-Chart of Expenditures")
```



From these depictions, one can tell that the bottom quintile spends quite a bit of their expenditures on food - at least 25%! Furthermore, this value is gradually increasing over time despite slacking wage growth and deflationary pressures. To give a comparison, the Engel Coefficient for the United States is somewhere around 10%.

While scatter graphs and line graphs previously also represented similar information, one can gain a much more holistic idea of expenditures when presented with a pie-chart. I resorted to making the pie-chart through base R since I had some difficulties with `ggplot`.

Discussion & Conclusion

The takeaway message for this post is that R can help us visualize our data in many ways. We have touched upon some of these strategies in class, so I wanted to expand on that and explore what other types of charts R is capable of making. At the same time, data visualization is crucial to conveying to the audience ones' findings. Two charts may give the same information, but one may be easier to understand - as you may have seen by some graphics above. Alternatively, two charts utilizing the same information may also be able to provide a whole new perspective. Therefore, as data scientists we must always keep in mind who the audience is and manipulate the data to find the most efficient and clear way to present it.

References

Engels Food Gauge Hit 29-Year-High

Family Income and Expenditure Survey from 2003-2016 *This link is for 2003; the data for the other 13 years can be found on the same website.*

Datacamp

tidyr

stackoverflow

Pie-Charts

Some of these sources I used more than once - especially the data from the Family Income & Expenditure Survey and Datacamp.