PSY6422 Data Management & Visualisation: Module Project

220225649

2023-05-16

Project Focus

For the module project, I wanted to create a visualisation which would present data about an important issue, that could be presented in the context of something seemingly unusual but interesting. While considering what topic could be suitable, I remembered seeing a tweet from someone who had worked out how many Freddos an hour Doctors could afford in 2023, compared to 2010 (https://twitter.com/Huw_Corness/status/1609957206757081090?cxt=HHwWhICwxZbj29csAAAA), in light of the current cost of living crisis. This inspired me. I decided to focus on the Freddo Index, which is used to assess inflation in the UK economy, looking at this in relation to the national minimum/living wage.

```
# Load libraries
library(plyr) # For data manipulation
library(tidyverse) # For data preparation
library(here) # Set working directory
library(readxl) # Read xlsx files
library(plotly) # To make interactive graphs
```

Data Origins

Two datasets were created for this visualisation, one for Freddo price information and one for national mininum/living wage information, both for the time period from 2020 to 2023.

The Freddo price data for 2020 - 2018 was found through an webpage about the Freddo Index, whereas the price data for 2019 - 2023 was found using news articles from these years.

The Minimum Wage data was found through two UK government datasets and one UK government webpage.

All links used for creating the two datasets are listed below.

Freddo prices information sources:

2020 - 2018 prices: https://www.vouchercloud.com/resources/the-freddo-index

2019 price: https://www.mirror.co.uk/money/shopping-deals/freddos-10p-again-tesco-one-13875080

2020 price: https://www.bigissue.com/news/fact-fiction-do-freddos-show-that-the-minimum-wage-should-be-18-per-hour/

2021 price: https://metro.co.uk/2021/08/26/joe-lycett-wades-into-freddo-war-as-investigates-rising-prices-15157463/

2022 price: https://www.walesonline.co.uk/news/cost-of-living/now-how-cost-freddo-changed-23633251

2023 price: https://www.thecourier.co.uk/fp/education/schools/4049089/freddos-teachers-wages-fife-video/

National Minimum and National Living Wage sources:

2000 - 2019 wages: https://www.gov.uk/government/publications/20-years-of-the-national-minimum-wage

2020 wage: https://www.gov.uk/national-minimum-wage-rates

2021 - 2023 wages: https://www.gov.uk/government/publications/the-national-minimum-wage-in-2023

The data

In the Freddo Prices dataset, there is a column for year and a column for the price of a Freddo in pence.

```
# Load Freddo prices data
freddo <- read_xlsx(here("data", "Freddo Prices.xlsx"))

# Show first 6 rows of the data
head(freddo)
```

```
## # A tibble: 6 x 2
##     Year FreddoCostPence
##    <dbl>          <dbl>
## 1  2000             10
## 2  2001             10
## 3  2002             10
## 4  2003             10
## 5  2004             10
## 6  2005             10
```

For looking at the national minimum/living wage, as there is varying degrees of this for different age groups, I decided to approach this looking at the wages targeted at 25 year olds, this being my current age. In 2016, the national minimum wage become the national living wage for UK workers aged 23 and over. This information should be considered when observing my visualisation - in the visualisations, we shall refer to the national minimum/living wage as the minimum wage. In the original datasets, some years provide two wage amounts, one for April and one for October of the same year. In these cases, I have used the April figure, so each minimum wage value is from the April of the year it belongs to.

In the National Minimum Wage dataset, there is a column for year, a column identifying whether the national minimum wage or national living wage was in use and a column for the wage of the particular year, in pounds

```
# Load National Minimum Wage data
N_M_Wage <- read_xlsx(here("data", "National_Minimum_wage.xlsx"))

# Show first 6 rows of the data
head(N_M_Wage)
```

```
## # A tibble: 6 x 3
##     Year NMW_or_NLW WagePounds
##    <dbl> <chr>           <dbl>
## 1  2000 NMW               3.7
## 2  2001 NMW               4.1
## 3  2002 NMW               4.2
## 4  2003 NMW               4.5
## 5  2004 NMW               4.85
## 6  2005 NMW               5.05
```

Research Question

How has the number of Freddos an individual can afford, with the national minimum/living wage, changed over the past 23 years?

Data Preparation

To prepare the data for visualisation, I joined the two data sets together, creating two new columns: one to show Freddo prices in decimals and one to show number of Freddos afforded with an hours minimum wage pay. For the latter column, I then rounded down the values, some including decimals, to a whole number. Some columns were renamed too, to make them easier to work with going forward.

```
# Join Freddo and National minimum wage tables together

Freddo_and_N_M_Wage <- inner_join(freddo, N_M_Wage, by = "Year")

# Making new columns

# Use FreddoCostWhole column to make a FreddoCostDec column, to show Freddo
# cost in decimals (to work out Freddos afforded)

Freddo_and_N_M_Wage <- mutate(Freddo_and_N_M_Wage, FreddoCostPence_Dec
                                 = FreddoCostPence/100)

# Make new column to show Freddos afforded with minimum wage

Freddo_and_N_M_Wage <- mutate(Freddo_and_N_M_Wage, FreddosAfforded =
                                 WagePounds/FreddoCostPence_Dec)

# Update FreddosAfforded column with rounded down values of

Freddo_and_N_M_Wage <- Freddo_and_N_M_Wage %>%
  mutate_at(vars(FreddosAfforded), ~ floor(.))
```

Now the data is prepared, we will make a new dataframe, keeping columns needed for the data visualisation only.

```
# Make new dataframe

FA_Data <- Freddo_and_N_M_Wage %>%
  select(Year, WagePounds, FreddosAfforded)

# Rename FreddosAfforded column

FA_Data <- FA_Data %>%
  rename(FreddosAfford = FreddosAfforded)

# Show first 6 rows of final dataframe

head(FA_Data)
```

```
## # A tibble: 6 x 3
```

```
##      Year WagePounds FreddosAfford
##     <dbl>      <dbl>         <dbl>
## 1   2000        3.7            37
## 2   2001        4.1            40
## 3   2002        4.2            42
## 4   2003        4.5            45
## 5   2004       4.85            48
## 6   2005       5.05            50
```

Now we are ready to start on the data visualisations!

Visualisation 1

For the first visualisation, I wanted to plot the individual data points as well as a line to reflect the changes in them. I decided to make a scatterplot, with a general additive model (gam) to provide a moving average line.

```r
p <- ggplot(data = FA_Data,
            mapping = aes(x = Year,
                          y = FreddosAfford))
# allocate "Year" column to x axis and column "FreddosAfford" to y axis.

p + geom_point(color = "springgreen4") +
  # adds points to the scatter plot with colour set as a shade of dark green.
  geom_smooth(method = "gam", color = "purple4") +
  # adds gam line to the scatter plot, with colour set as a shade of dark purple.
  labs(x = "Year", y = "Number of Freddos afforded with
       Minimum Wage",
       title = "Freddos Afforded With The Minimum Wage:
       From 2000 to 2023") +
  # set labels for x axis, y axis and for the title of the graph
  theme(plot.title = element_text(color = "purple4", size = 15, hjust = 0.5, face =
                                    "bold")) +
  # formatting title with colour, size, position and bold setting
  theme(panel.background = element_blank()) +
  # set graph background as blank
  theme(axis.line = element_line(colour = "purple4")) +
  # set colour of axis line as a shade of dark purple
  theme(panel.border = element_rect(color = NA, fill = NA)) +
  # set graph border as blank
  theme(text = element_text(color = "purple4"),
        axis.text = element_text(color = "purple4")) +
  # set text colour as a shade of dark purple
 theme(text = element_text(family = "Palatino")) # set font of graph
```

Freddos Afforded With The Minimum Wage:
From 2000 to 2023
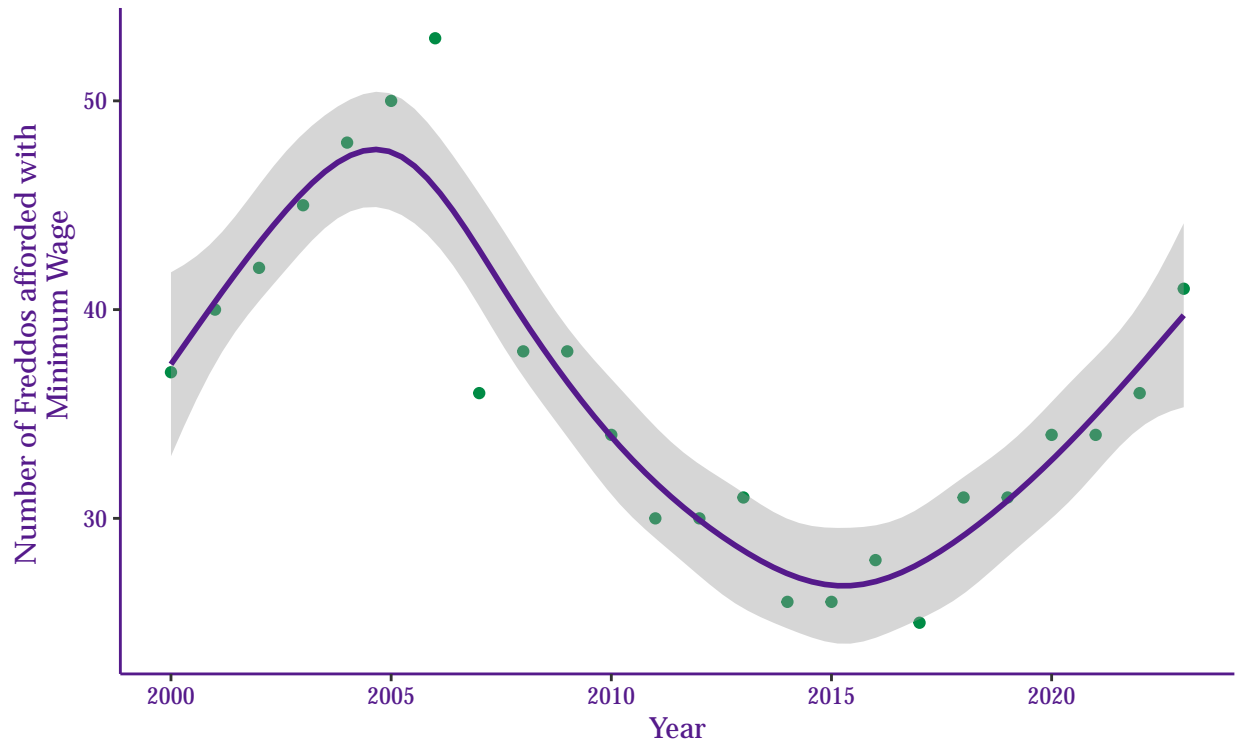
This visualisation presents the data reasonably well - but it felt a bit basic to me. I wanted to appropriately reflect how the number of Freddos afforded has changed dramatically over the years. A smooth curve didn't seem appropriate - I felt that a line which represented the steep changes would be better.

Visualisation 2

For my second attempt, I decided to make a graph with a line that connected up all the data points, to represent the steep changes in a better way. I also decided to introduce another element to the graph: formatting the size of the data points to represent the minimum wage amount increase throughout the years. This is known as a bubbleplot.

```
p <- ggplot(data = FA_Data,
            mapping = aes(x = Year,
                          y = FreddosAfford))
# allocate "Year" column to x axis and column "FreddosAfford" to y axis.

p + geom_line(size = 1, color = "springgreen4") +
  # add lines to connect points, setting size and colour as a shade of dark green
  geom_point(color = "purple4", aes(size = WagePounds)) +
  # adds points to the scatter plot with colour set as a shade of dark purple
  # and point size dependent on WagePounds variable
  scale_size_continuous(range = c(1, 5.5)) +
  # set size range of the points
  labs(x = "Year", y = "Number of Freddos afforded with
       Minimum Wage",
```

```
         title = "Freddos Afforded With The Minimum Wage:
         From 2000 to 2023", size = "Minimum
Wage
(Pounds)") +
  # set labels for x axis, y axis and for the title of the graph
  theme(plot.title = element_text(color = "purple4", size = 15, hjust = 0.5, face =
                                  "bold")) +
  # formatting title with colour, size, position and bold setting
  theme(panel.background = element_blank()) +
  # set graph background as blank
  theme(axis.line = element_line(colour = "purple4")) +
  # set colour of axis line as a shade of dark purple
  theme(panel.border = element_rect(colour = "purple4", fill = NA, size = 3)) +
  # add a purple panel border, set size and no fill
  theme(text = element_text(color = "purple4"),
        axis.text = element_text(color = "purple4")) +
  # set colour of graph and axis label text
   theme(text = element_text(family = "Palatino")) # set font of graph
```


Freddos Afforded With The Minimum Wage: From 2000 to 2023
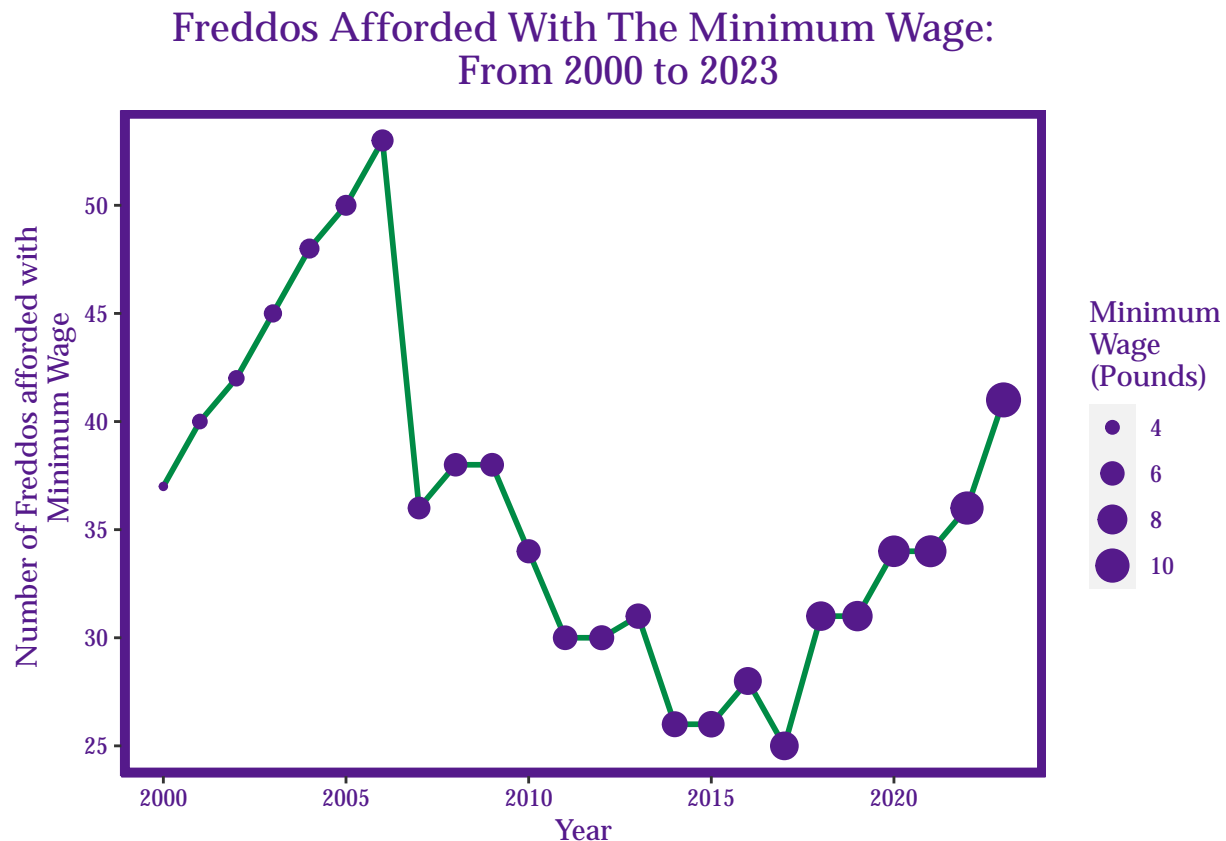
This graph gives the kind of line I had in mind, showing the sharp drops between years. I added in formatting to make the data point circles vary in size according to the minimum wage of each year. However, this idea did not translate as well as I thought it would onto the plot itself - it looked a bit clunky.

Visualisation 3

With my final visualisation, I decided to keep the data point formatting to represent the minimum wage, but include a line of best fit instead of lines connecting the data points. I also wanted to bring an element of interactivity to the visualisation, so included the use of ggplotly.

```
p <- ggplot(data = FA_Data,
            mapping = aes(x = Year,
                          y = FreddosAfford)) +
  # allocate "Year" column to x axis and column "FreddosAfford" to y axis.
  geom_point(aes(size = WagePounds), color = "purple4") +
  # adds points to the scatter plot with colour set as a shade of dark purple
  # and point size dependent on WagePounds variable
  scale_size_continuous(range = c(1, 5.5)) +
  # set size range of the points
  geom_smooth(method = "lm", color = "springgreen4", se = F) +
  # add linear model line, setting colour as a shade of dark green
  labs(x = "Year", y = "Number of Freddos afforded with hours worth of
       Minimum Wage",
       title = "Freddos Afforded With The Minimum Wage:
       From 2000 to 2023",
       size = "Minimum
Wage
(Pounds)") +
  # set labels for x axis, y axis and for the title of the graph
  theme(plot.title = element_text(color = "purple4", size = 15, hjust = 0.5, face =
                                  "bold")) +
  # formatting title with colour, size, position and bold setting
  theme(panel.background = element_blank()) +
  # set graph background as blank
  theme(axis.line = element_line(colour = "purple4", size = 3)) +
  #set colour of axis line, as a shade of dark purple, set size
  theme(panel.border = element_rect(colour = NA, fill = NA, size = 3)) +
  # add a purple panel border, set size and no fill
  theme(text = element_text(color = "purple4"),
        axis.text = element_text(color = "purple4")) +
  # set colour of graph and axis label text
  theme(text = element_text(family = "Palatino")) # set font of graph

ggplotly(p)
```

Save final visualisation

```
ggsave(here("plots", "freddograph.png"))
```

Summary

The visualisations present an interesting history of the 'Freddo index' and the minimum wage since 2000. The minimum wage has risen steadily throughout, but the number of Freddos an individual can afford with the minimum wage has not. From 2000 to 2005 there was a continuous increase, before this gradually

declined (albeit with some slight increases) until 2017, when the increases started again, but not the same level as there was in the first several years since 2000.

I was pleased with final visualisation I made. In particular, I am glad I made a interactive graph, with hovering over the data points providing the specific year, minimum wage and Freddos afforded amount. Including a linear model line felt appropriate, to show an overall trend line, which is interesting when this is judged against the data points.

If I were to run this project again, I would wish to explore more options with formatting, using animation and finding the best way to include the cost of the Freddo itself in the graph. I decided not to include this information in the graph so as to avoid information overload, though addition of this could help to improve context.