NICEI - Northern Ireland Composite Economic Index

2023-04-19

Contents:

This document will outline the steps taken, including the code, to achieve the plots in the “Growth Rates” panel of the NICEI Q4 2022 factsheet which can be found here (https://www.nisra.gov.uk/system/files/statistics/NICEI%20Q4%202022%20factsheet.PDF). No other panels have

been completed but can be ameneded to this document if so desired.

Panel 1: Growth Rates

1. NICEI Trend to Q4 2022

Step 1: Installing relevant libraries

```
library(readxl)
library(janitor)
library(ggplot2)
library(dplyr)
library(reshape2)
library(tidyverse)
```

Step 2: Data Import and Preparation

Loading data

Import from Excel spreadsheet in local directory

```
df <- read_excel("NICEI-Tables-Q4-2022.xlsx", sheet = "Table 1", skip = 1)
```

Data Preparation

Selection

Dropping last 3 columns that are unused and merging the Year and Quarter column

```
df1 <- df %>%
  select(c(0:5)) %>%
  mutate(date = paste(Year, Quarter, sep="\nQ"), .before = 1, .keep = "unused")
```

Restructuring

Now for restructuring. This code takes the data from the 4 columns and restructures it to 3 columns which contain the date, the variable (previous column headers i.e the variables of interest) and their respective values. This is useful for colouring plots based on category as is needed in this case.

```
df2<-melt(df1, id.vars="date")
```

Step 3. Plotting

Now to plot the data to find out what it looks like and what more needs done

Initial plot

```
ggnicei<-ggplot() +
  geom_line(data = df2,
            aes(x=date, y=value,
                group = variable,
                color = variable)
            )
ggnicei
```

The plot lines look correct but work needs done to make the legend legible and prettify the plot so it resembles the original plot published. To alter the X-axis values so they're readable, a function will be created to return a function as the breaks argument; this, in turn, will return the appropriate x-axis spacing i.e the nth value specified when the function is called.

Plot formatting/aesthetics

```
#Create function
every_nth = function(n) {
  return(
    function(x){
      x[c(TRUE, rep(FALSE, n - 1))]
      }
    )
}
```

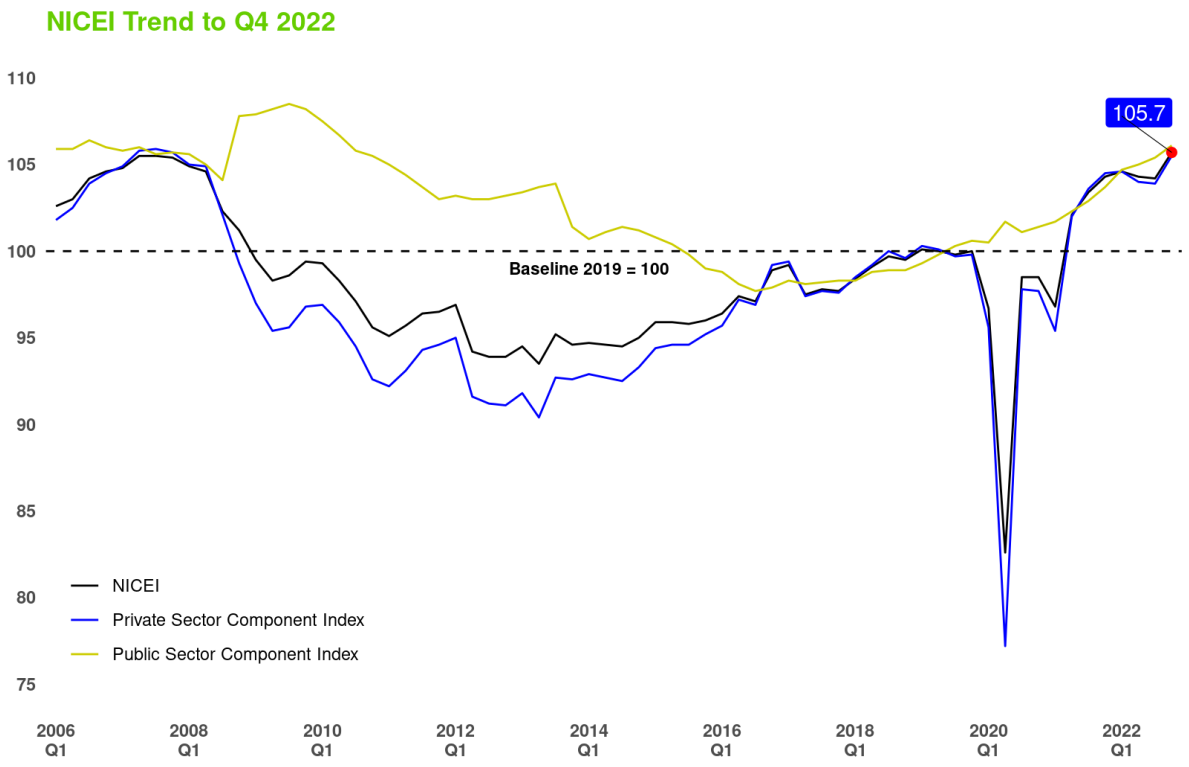
The plot can now be redone implementing some formatting such as:

- scale_colour_manual: this formats the plot in relation to colour
- geom_hline: this inserts a horizontal line
- annotate: this permits annotation on the plot
- theme: this has many possible components that all will affect the aesthetic of the plot

```
final_plot<-ggnicei +
  scale_color_manual(values= c("black", "blue", "#cccc00"),             # selecting colo
urs for the lines
                     labels=c("NICEI",                                  # creating the l
ine labels for the legend
                              "Private Sector Component Index",
                              "Public Sector Component Index")) +
  geom_hline(yintercept=100, linetype = 'dashed') +                    # generating the
dashed horizontal line (baseline)
  annotate("text",                                                     # Forming the tex
t for the baseline
           x="2014\nQ1",
           y= 99,
           label = "Baseline 2019 = 100",
           size =3,
           fontface = "bold") +
  theme(line = element_blank(),                                  # plot theme
        panel.grid.major = element_blank(),                      # no major gridlines
        panel.grid.minor = element_blank(),                      # no minor gridlines
        panel.background = element_blank(),                      # blank background
        plot.title = element_text(colour = "chartreuse3",        # title formatting: colo
ur and bold
                                  face = "bold"),
        axis.text = element_text(face = "bold"),                 # formatting axis text:
bold
        axis.line = element_line(colour = "NA"),                 # no axis lines
        legend.title=element_blank(),                            # remove legend title
        legend.key=element_blank(),                              # remove background colo
ur from legend key
        legend.position = c(0.15,0.15),                          # legend position
        plot.margin = unit(c(2,2,2,2), "cm")) +                  # adjusting plot padding
for exported image
  scale_x_discrete(breaks = every_nth(n = 8)) +                  # implementing the funct
ion for axis labels every 8th (Q1 every other year)
  scale_y_continuous(breaks = seq(75, 110, by=5),                # formatting Y-axis: 75-
115 in increments of 5
                     limits=c(75,110))+
  annotate(geom = "point",      # inserting red point at Y-value 105.7
           x = "2022\nQ4",
           y = 105.7,
           colour = "red",
           size = 2) +
  annotate(geom = "label",      # inserting blue label with white font stating 105.7
           x = "2021\nQ4",
           y = 108,
           label = "105.7",
           hjust = "left",
           fill="blue",
           color = "white") +
  annotate("segment",           # connecting line between point and label
           x = "2022\nQ1",
           xend = "2022\nQ4",
```

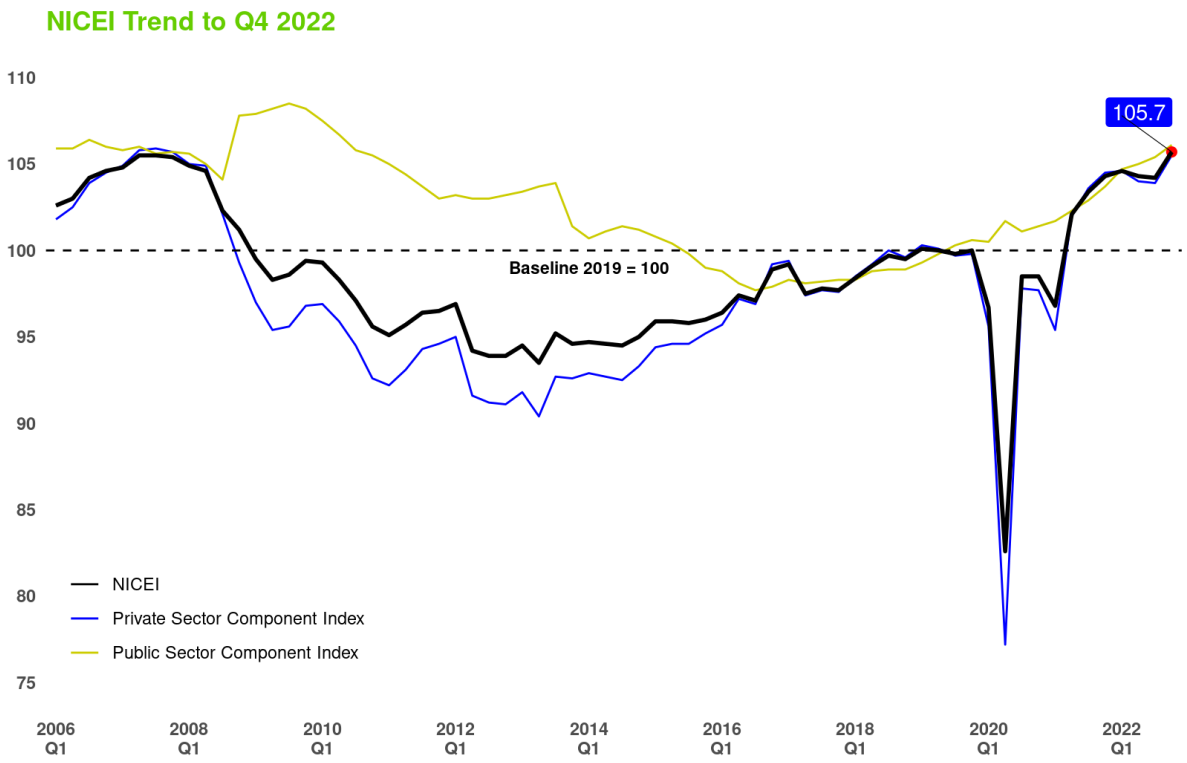
```
           y = 107.8, yend = 105.7,
           linewidth = 0.2) +
  labs(x = "",                                    # no X-axis title
       y="",                                      # no Y-axis title
       title = "NICEI Trend to Q4 2022")          # plot title

final_plot
```



This plot seems much more comparable to the original in the publication; however, the plot in the original pdf seems to have a thicker NICEI line. If this is desired, all that is required is to overlay the final_plot with the same plot line but to the required thickness.

```
df4<-filter(df2, variable == "NICEI")   # extract the values associated with the require
d plot line
final_plot2<- final_plot +              # add these extracted values to the previous plo
t, formatting appropriately
  geom_line(data=df4,
            aes(x=date,
                y=value,
                group = 1),
            linewidth = 1)
final_plot2
```



Step 4. Export plot image

Instead of exporting the plot as an image manually within R Studio, a scripted export of plot image can be done; however, it must be noted that there is a decrease in text readability using this method and so will need increased. An example of the manually exported plot is in the local directory named, “NICEI_r_plot.png”. Scripted export can be done as follows:

```
png("NICEI-plot.png",
    width = 950,
    height = 750)
final_plot2
dev.off()
```

2. Growth Rates by Index

Step 1: Installing relevant libraries

```
library(readxl)
library(xts)
library(formattable)
library(flextable)
library(tidyverse)
library(gridExtra)
library(grid)
library(officer)
```

Step 2: Data Import, Cleaning and Preparation

Loading data

Read in data (excel format) from local directory

```
df <- read_excel("NICEI-Tables-Q4-2022.xlsx", sheet = "Table 1", skip = 1)
formattable(head(df))
```

Year	Quarter	NICEI	Private Sector	Public Sector	Services	Production	Construction
2006	1	102.6	101.8	105.9	99.3	101.7	134.9
2006	2	103.0	102.5	105.9	99.9	103.3	134.8
2006	3	104.2	103.9	106.4	101.0	103.2	143.1
2006	4	104.6	104.5	106.0	102.0	103.1	142.9
2007	1	104.8	104.9	105.8	102.5	104.1	137.9
2007	2	105.5	105.8	106.0	103.2	104.7	143.3

Calculations

This table contains data not present in the raw data table, instead, it must be calculated. In this section each row of the final growth table will calculated before being appended to each other, creating the final table.

Calulating value differences

This code chunk below is creating the first 3 rows of the table - calculating the differences in the required data

```
Q_Q <- (df[nrow(df),] - df[nrow(df)-1,])/df[nrow(df)-1,]    # Calculate Q/Q row
Y_Y <- (df[nrow(df),] - df[nrow(df)-4,])/df[nrow(df)-4,]    # Calculate Y/Y row
Tri <- (df[nrow(df),] - df[nrow(df)-12,])/df[nrow(df)-12,]  # Calculate Triennial row
```

Rolling 4Q Average

The last row of the final table is the rolling 4Q average of the data. This requires more detailed work. Firstly the required data from the raw form must be selected

```
# Create new column with year and Quarter combined, separated with a space, and delete t
he two individual columns
df1<- df %>% mutate(date = paste(Year, Quarter, sep="\nQ"), .before = 1, .keep = "unuse
d")
```

Once selected, this must be converted to a time series object in order to apply the time series functions

```
# Convert from data frame to a time series object to enable use of moving/rolling averag
e calculations
TS <- df1 %>% ts(frequency = 4, start = c(2006,1), end = c(2022, 4))    # frequency = 4
means data consists of quarterly samples.
```

Now the required rolling 4Q averages can be returned

```
# Calculate the 4Q rolling average of dataset
r4Q <-TS %>%                    # all data (window() would be used if calculations were t
o be on data subsets)
  as.xts() %>%                  # coerces the timeseries object to an eXtensible format r
equired for time-based analysis
  apply.yearly(mean) %>%        # apply mean function to each distinct year of data
  as.data.frame()               # pipe data to data frame for additional work

# calculation of fractional difference for rolling 4Q annual average
r4q<-(r4Q[nrow(r4Q),] - r4Q[nrow(r4Q)-1,])/r4Q[nrow(r4Q)-1,]
```

Step 3. Tabulation and aesthetics

Now all required rows have been obtained, it is time to combine and format accordingly as per orginial publication

Tabulation of individual rows

The following code chunk combines the rows and appends different column names which fit the formatting required for the arrow inclusion code (compose package)

```
dt<-rbind(Q_Q[,-c(1:2)],
          Y_Y[,-c(1:2)],
          Tri[,-c(1:2)],
          r4q[,-1]) %>%
  round(digits = 3)

# create the column of names
growth <- data.frame("Growth Rates" = c("Q/Q",
                                        "Y/Y",
                                        "Triennial \n(3Yr) Change",
                                        "Rolling 4Q \nAnnual Ave")
                     )

DT<-cbind(growth, dt)   # combine base table with names
colnames(DT)<-c("growth", "NICEI", "private", "public", "services", "production", "const
ruction")  #convert column names to single value names (needed for the compose later)
formattable(DT)
```

	growth	NICEI	private	public	services	production	construction
1	Q/Q	0.014	0.015	0.007	0.010	-0.006	0.088
2	Y/Y	0.013	0.010	0.023	0.018	-0.011	-0.007
3	Triennial (3Yr) Change	0.057	0.057	0.055	0.062	0.046	0.046
2022 Q4	Rolling 4Q Annual Ave	0.030	0.031	0.026	0.044	0.025	-0.036

The values in the table are almost correct; the original publication has the values as percentages.

```
pDT<- DT %>% mutate(across(-c(1),percent))    #convert values to percentage using the fo
rmattable package
formattable(pDT)
```

	growth	NICEI	private	public	services	production	construction
1	Q/Q	1.40%	1.50%	0.70%	1.00%	-0.60%	8.80%
2	Y/Y	1.30%	1.00%	2.30%	1.80%	-1.10%	-0.70%
3	Triennial (3Yr) Change	5.70%	5.70%	5.50%	6.20%	4.60%	4.60%
2022 Q4	Rolling 4Q Annual Ave	3.00%	3.10%	2.60%	4.40%	2.50%	-3.60%

Aesthetics

Now the table is complete in its basic form, the colouring and arrows need to be added The formattable package has been used to this point but moving forward the flextable package will be used instead as this permits more advanced formatting such as the addition of icons (the arrows in this case).

```
dft<-flextable(pDT) %>% theme_box()   # pretty print of table using flextable (easier to
customise than formattable)

#prettify table
dft<-bold(dft, part = "header")                    # bold header
dft<-bg(dft, bg = "olivedrab3", part = "header")  # fill the header with colour
dfxt <- dft                                        # rename for next step of adding arrow
s
dfxt
```

growth	NICEI	private	public	services	production	construction
Q/Q	1.40%	1.50%	0.70%	1.00%	-0.60%	8.80%
Y/Y	1.30%	1.00%	2.30%	1.80%	-1.10%	-0.70%
Triennial (3Yr) Change	5.70%	5.70%	5.50%	6.20%	4.60%	4.60%
Rolling 4Q	3.00%	3.10%	2.60%	4.40%	2.50%	-3.60%

growth	NICEI	private	public	services	production	construction
Annual Ave						

In the complex process of adding coloured arrows corresponding to values, each column must be addressed individually and the green arrows will be added first as this is the predominant colour. Then each column with red arrows must will be amended to reflect this. This is a manual process because there is not a consistent rule in the values upon which to allocate the arrows. In the final table, each value corresponds to its previous value in a different year/quarter, which is dependent on the category it is describing. There are R emoticon/graphics packages available but it was easier to find .png files and save them to the local directory and upload

```
dfxt2 <- flextable::compose(dfxt,   #data
                 j = 2,   #2nd column
                 value = as_paragraph(     # allows concatenation of text chunks and ima
ges etc
                   as_image(src = "arrow_up.png", width = .15, height = .15),   # arrow
first as per the publication being replicated
                   " ",    # space follows the arrow
                   as_chunk(NICEI, props = fp_text(color = "black")    # The text to be
added, which is the value in the column NICEI.
                            )                                           # This line is w
hy the column names needed changed earlier (line 49)
                   ),
                 part = "body")   # noting that the compose function is applied to the t
able body
dfxt2    #table check
```

growth	NICEI	private	public	services	production	construction
Q/Q	⇧ 1.40%	1.50%	0.70%	1.00%	-0.60%	8.80%
Y/Y	⇧ 1.30%	1.00%	2.30%	1.80%	-1.10%	-0.70%
Triennial (3Yr) Change	⇧ 5.70%	5.70%	5.50%	6.20%	4.60%	4.60%
Rolling 4Q Annual Ave	⇧ 3.00%	3.10%	2.60%	4.40%	2.50%	-3.60%

```
dfxt2 <- flextable::compose(dfxt2,
                 j = 3,
                 value = as_paragraph(
                   as_image(src = "arrow_up.png", width = .15, height = .15), " ",
                   as_chunk(private, props = fp_text(color = "black"))
                 ),
                 part = "body")
dfxt2
```

growth	NICEI	private	public	services	production	construction
Q/Q	1.40%	1.50%	0.70%	1.00%	-0.60%	8.80%
Y/Y	1.30%	1.00%	2.30%	1.80%	-1.10%	-0.70%
Triennial (3Yr) Change	5.70%	5.70%	5.50%	6.20%	4.60%	4.60%
Rolling 4Q Annual Ave	3.00%	3.10%	2.60%	4.40%	2.50%	-3.60%

```
dfxt2 <- flextable::compose(dfxt2,
                 j = 4,
                 value = as_paragraph(
                   as_image(src = "arrow_up.png", width = .15, height = .15), " ",
                   as_chunk(public, props = fp_text(color = "black"))
                 ),
                 part = "body")
dfxt2
```

growth	NICEI	private	public	services	production	construction
Q/Q	1.40%	1.50%	0.70%	1.00%	-0.60%	8.80%
Y/Y	1.30%	1.00%	2.30%	1.80%	-1.10%	-0.70%
Triennial (3Yr) Change	5.70%	5.70%	5.50%	6.20%	4.60%	4.60%
Rolling 4Q Annual Ave	3.00%	3.10%	2.60%	4.40%	2.50%	-3.60%

```
dfxt2 <- flextable::compose(dfxt2,
                 j = 5,
                 value = as_paragraph(
                   as_image(src = "arrow_up.png", width = .15, height = .15), " ",
                   as_chunk(services, props = fp_text(color = "black"))
                 ),
                 part = "body")
dfxt2
```

growth	NICEI	private	public	services	production	construction
Q/Q	1.40%	1.50%	0.70%	1.00%	-0.60%	8.80%
Y/Y	1.30%	1.00%	2.30%	1.80%	-1.10%	-0.70%

growth	NICEI	private	public	services	production	construction
Triennial (3Yr) Change	5.70%	5.70%	5.50%	6.20%	4.60%	4.60%
Rolling 4Q Annual Ave	3.00%	3.10%	2.60%	4.40%	2.50%	-3.60%

```
dfxt2 <- flextable::compose(dfxt2,
                  j = 6,
                  value = as_paragraph(
                    as_image(src = "arrow_up.png", width = .15, height = .15), " ",
                    as_chunk(production, props = fp_text(color = "black"))
                  ),
                  part = "body")
dfxt2
```

growth	NICEI	private	public	services	production	construction
Q/Q	1.40%	1.50%	0.70%	1.00%	-0.60%	8.80%
Y/Y	1.30%	1.00%	2.30%	1.80%	-1.10%	-0.70%
Triennial (3Yr) Change	5.70%	5.70%	5.50%	6.20%	4.60%	4.60%
Rolling 4Q Annual Ave	3.00%	3.10%	2.60%	4.40%	2.50%	-3.60%

```
dfxt2 <- flextable::compose(dfxt2,
                  j = 7,
                  value = as_paragraph(
                    as_image(src = "arrow_up.png", width = .15, height = .15), " ",
                    as_chunk(construction, props = fp_text(color = "black"))
                  ),
                  part = "body")
dfxt2
```

growth	NICEI	private	public	services	production	construction
Q/Q	1.40%	1.50%	0.70%	1.00%	-0.60%	8.80%
Y/Y	1.30%	1.00%	2.30%	1.80%	-1.10%	-0.70%
Triennial (3Yr) Change	5.70%	5.70%	5.50%	6.20%	4.60%	4.60%

growth	NICEI	private	public	services	production	construction
Rolling 4Q Annual Ave	3.00%	3.10%	2.60%	4.40%	2.50%	-3.60%

```
dfxt2<-  flextable::compose(dfxt2,
                 j = 6, i = ~production < 0.011,
                 value = as_paragraph(
                   as_image(src = "arrow_down.png", width = .15, height = .15), " ",
                   as_chunk(production, props = fp_text(color = "black"))
                 ),
                 part = "body")
dfxt2
```

growth	NICEI	private	public	services	production	construction
Q/Q	1.40%	1.50%	0.70%	1.00%	-0.60%	8.80%
Y/Y	1.30%	1.00%	2.30%	1.80%	-1.10%	-0.70%
Triennial (3Yr) Change	5.70%	5.70%	5.50%	6.20%	4.60%	4.60%
Rolling 4Q Annual Ave	3.00%	3.10%	2.60%	4.40%	2.50%	-3.60%

```
dfxt2 <- flextable::compose(dfxt2,
                 j = 7, i = ~construction < 0.04,
                 value = as_paragraph(
                   as_image(src = "arrow_down.png", width = .15, height = .15), " ",
                   as_chunk(construction, props = fp_text(color = "black"))
                 ),
                 part = "body")

dfxt2   #this should be the completed body
```

growth	NICEI	private	public	services	production	construction
Q/Q	1.40%	1.50%	0.70%	1.00%	-0.60%	8.80%
Y/Y	1.30%	1.00%	2.30%	1.80%	-1.10%	-0.70%
Triennial (3Yr) Change	5.70%	5.70%	5.50%	6.20%	4.60%	4.60%
Rolling 4Q	3.00%	3.10%	2.60%	4.40%	2.50%	-3.60%

growth	NICEI	private	public	services	production	construction
Annual Ave						

Step 4. Final prettification

The arrows are all the correct colour and attributed to the correct values. Now the column names can be changed back to the correct format.

```
dfxt3 <- set_header_labels(dfxt2,
                           values = list(
                             growth = "Growth Rates",
                             NICEI = "NICEI",
                             private = "Private Sector",
                             public = "Public Sector",
                             services = "Services",
                             production = "Production",
                             construction = "Construction"
                           )
)

dfxt3
```

Growth Rates	NICEI	Private Sector	Public Sector	Services	Production	Construction
Q/Q	↑ 1.40%	↑ 1.50%	↑ 0.70%	↑ 1.00%	↓ -0.60%	↑ 8.80%
Y/Y	↑ 1.30%	↑ 1.00%	↑ 2.30%	↑ 1.80%	↓ -1.10%	↓ -0.70%
Triennial (3Yr) Change	↑ 5.70%	↑ 5.70%	↑ 5.50%	↑ 6.20%	↑ 4.60%	↑ 4.60%
Rolling 4Q Annual Ave	↑ 3.00%	↑ 3.10%	↑ 2.60%	↑ 4.40%	↑ 2.50%	↓ -3.60%

Finally, the last act is to amend the column widths and text alignment to make the table more aesthetically pleasing and readable

```
#prettifying the table via formatting to reflect the publication be replicated
dfxt4 <- width(dfxt3, j = 1, width = 1.3086) %>%    # Adjusting the width of column 1
  align(align = "center", part = "all") %>%         # Center aligning all of table conte
nts
  align(j=1, align = "left", part = "body")         # Left aligning column 1

growth_table <-dfxt4    # Final table
growth_table
```

Growth Rates	NICEI	Private Sector	Public Sector	Services	Production	Construction
Q/Q	⬆ 1.40%	⬆ 1.50%	⬆ 0.70%	⬆ 1.00%	⬇ -0.60%	⬆ 8.80%
Y/Y	⬆ 1.30%	⬆ 1.00%	⬆ 2.30%	⬆ 1.80%	⬇ -1.10%	⬇ -0.70%
Triennial (3Yr) Change	⬆ 5.70%	⬆ 5.70%	⬆ 5.50%	⬆ 6.20%	⬆ 4.60%	⬆ 4.60%
Rolling 4Q Annual Ave	⬆ 3.00%	⬆ 3.10%	⬆ 2.60%	⬆ 4.40%	⬆ 2.50%	⬇ -3.60%

3. Contribution to Quarterly Change in NICEI (Q4 2022)

Step 1: Installing relevant libraries

```
library(readxl)
library(tidyverse)
library(formattable)
library(ggplot2)
```

Step 2: Data Import and Preparation

Loading data

Import from Excel spreadsheet in local directory

```
df <- read_excel("NICEI-Tables-Q4-2022.xlsx",
                                  sheet = "Table 6",
                                  range = "A2:B8")
```

Data preparation

In the publication, the plot has an evident gap between the NICEI bar and the rest. To replicate this in R, an empty row will be created plotted. Row 5 does not contain any values, only NAs, which for the characters column (1) will need placed with an empty value.

```
# Replace row 5 column 1 NA with empty space.
df[5,1]<-""

formattable(df)    #table check
```

Sector	Quarterly contribution (pps)*
Services	0.5
Production	-0.1
Construction	0.6

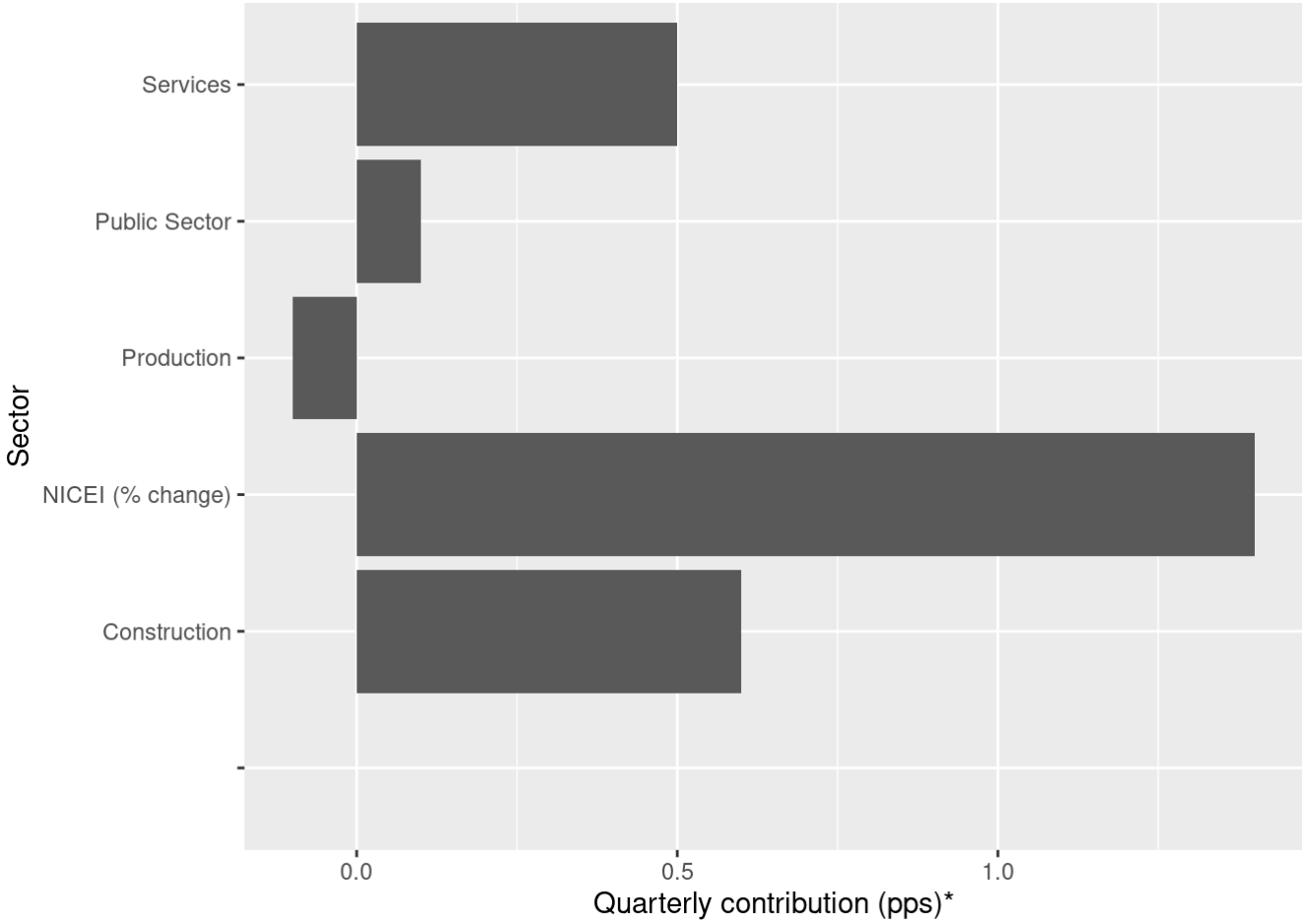
Sector	Quarterly contribution (pps)*
Public Sector	0.1
	NA
NICEI (% change)	1.4

Step 3. Plotting

Initial plot

Initial plot of histogram to check layout and comparison to original

```
ggplot(df,                                          # data
       aes(Sector,                                  # X values
           `Quarterly contribution (pps)*`))  +     # Y values
  geom_col() +                                      # Plot type
  coord_flip()                                      # Horizontal layout
```



The plot needs quite some work. The order is incorrect, the bars are too thick, there is no colour, no data labels and there are both axis titles and background colour.

Plot preparation

This code chunk attempts to ensure the correct order of sequence and generate the appropriate colour scheme. Note there are only 3 colours but 5 variables. A group will need to be created and according to colour, attributed the variables.

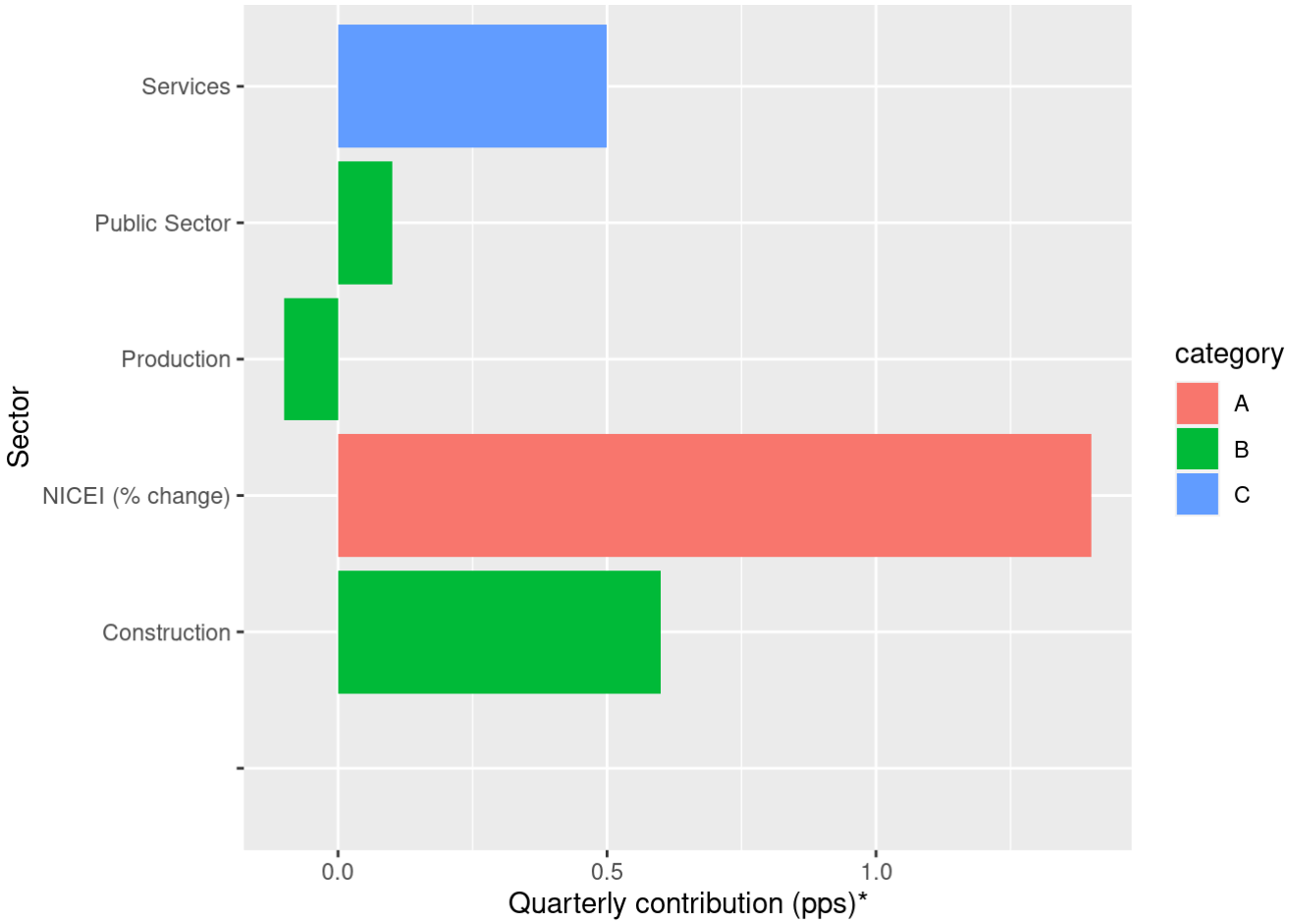
```
df1<-map_df(df, rev)      # reverse the order of the columns in df
df1$category<- c("A", "B", "B", "B", "B", "C")      # creating a new column to assist wi
th colour scheme in original
formattable(df1)    # table check
```

Sector	Quarterly contribution (pps)*	category
NICEI (% change)	1.4	A
	NA	B
Public Sector	0.1	B
Construction	0.6	B
Production	-0.1	B
Services	0.5	C

The table is now in the order required for plotting and the categories can be used to allocated colour to the bars. Time to plot check.

```
hbplt<-ggplot(df1,
              aes(Sector,                                  # X-axis values
                  `Quarterly contribution (pps)*`,         # Y-axis
                  fill = category)) +                      # colours based on column "cate
gory"
  geom_col() +                                             # plot type
  coord_flip()                                             # flip plot to horizontal

hbplt
```

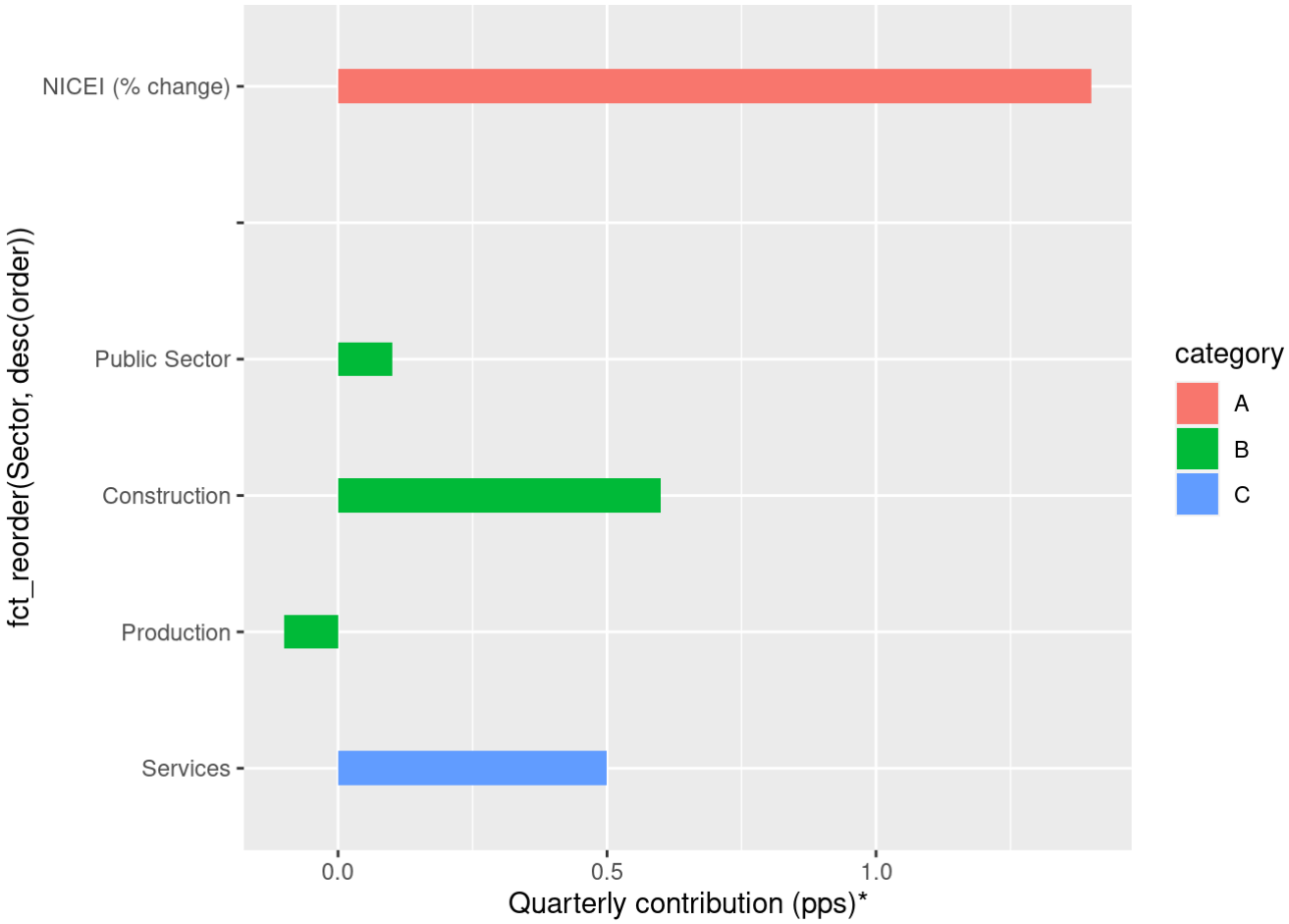


The colours are allocated correctly but the sequence is still wrong; it isn't plotting in same order as table. To ensure fixed sequencing another column will be created called "order" which will number the rows 1:5 based on the sequence they should be plotted.

```
df2<-df1        # creating new data frame based on previous

df2$order<-c(1:nrow(df2))     # adding row attributing an order for the plot

hbplt2 <- ggplot(df2,
                 aes(fct_reorder(Sector, desc(order)),    # allocates Sector plotting se
quence based on 'order'
                `Quarterly contribution (pps)*`,
                fill = category)) +
          geom_col(width = 0.25) +       # changes column width so they're thinner
          coord_flip()
hbplt2
```

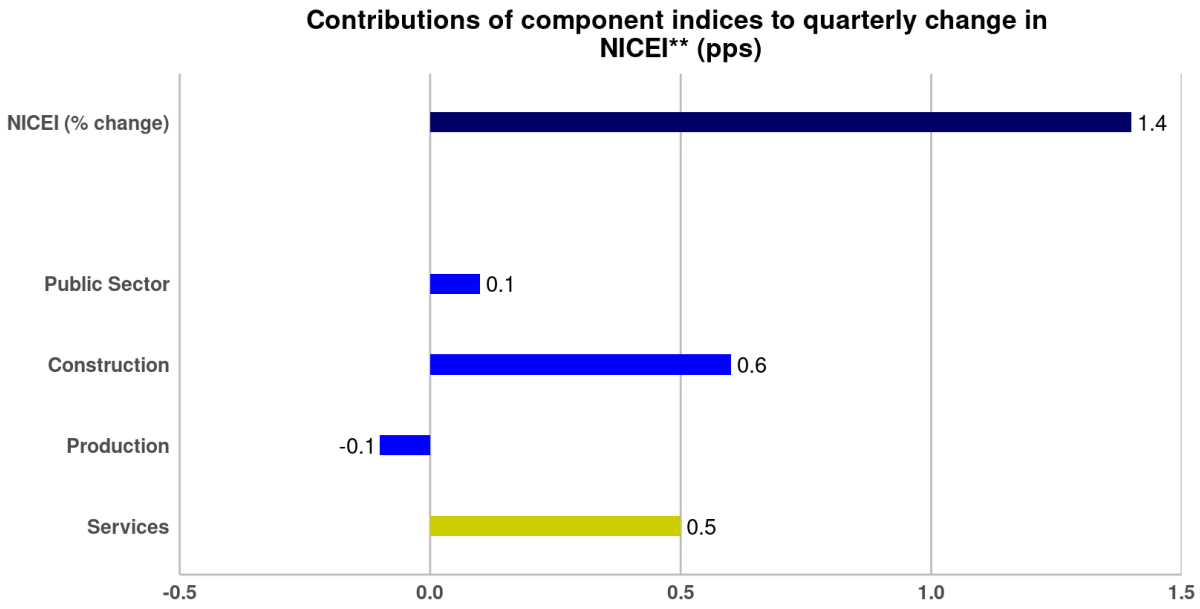


The order column works, now to finish the plotting aesthetics

Final Plotting

```
final_plot <- hbplt2 +
  theme_classic() +           # blank background as per original image
  geom_text(aes(label =         # adding data labels to the plot
                  ifelse(`Quarterly contribution (pps)*` < 0, `Quarterly contribution (p
ps)*`, "")),   # if value is negative
            hjust = 1.1) +     # position label on left
  geom_text(aes(label =
                  ifelse( `Quarterly contribution (pps)*` > 0, `Quarterly contribution
(pps)*`, "")),  # if value is negative
            hjust = -0.2) +    # position label on left
  scale_fill_manual(values= c("#000066",         # colour first category
                              "blue",            # colour second category
                              "#cccc00")) +      # colour third category
  theme(aspect.ratio = 1/2,                                          # change aspect rati
o of plot
        plot.margin = unit(c(2,2,2,2), "cm"),                        # pad area around pl
ot for image capture
        panel.grid.major.x = element_line(color = "grey"),           # colour major gridl
ines grey
        plot.title = element_text(hjust = 0.5,                       # centre the plot ti
tle and bolden
                                  face="bold"),
        axis.text = element_text(size=10, face =                     # make axis text bol
d and size 10
                                   "bold"),
        axis.line = element_line(colour = "grey"),                   # colour axis lines
grey
        axis.ticks = element_blank(),                                # remove axis ticks
        legend.position="none") +                                    # no legend
  scale_y_continuous(limits=c(-0.5,1.5),         # alter y axis to start at -0.5 and end
at 1.5
                     expand = c(0, 0)) +         # y axis limits to start at plot origin
  labs(x = "",                                   # plot to have no X- or Y- titles but a
main plot title
       y="",
       title = "Contributions of component indices to quarterly change in \nNICEI** (pp
s)"
       )

final_plot    #sanity check
```



Step 4. (Optional) Script to save plot as image

As before, the plot image can be manually exported or a code chunk can be inserted to automtically do this; however, when this is executed the text sizes are much smaller and can be difficult to read. to fix this the code chunk from above is repeated but with the text sizes need amended.

Repeat code chunk but amend text sizes

```
final_plot_sized <- ggplot(df2,
                     aes(fct_reorder(Sector, desc(order)),   # attributes a sequence of
plotting Sector based on 'order'
                         `Quarterly contribution (pps)*`,
                         fill = category)) +
  geom_col(width = 0.25) +          # adjusts the column widths so they are thinner
  theme_classic() +           # blank background as per original image
  geom_text(aes(label =         # adding data labels to the plot
                  ifelse(`Quarterly contribution (pps)*` < 0, `Quarterly contribution (p
ps)*`, "")),   # if value is negative
            hjust = 1.1) +     # position label on left
  geom_text(aes(label =
                  ifelse( `Quarterly contribution (pps)*` > 0, `Quarterly contribution
(pps)*`, "")),  # if value is negative
            hjust = -0.2) +    # position label on left
  scale_fill_manual(values= c("#000066",         # colour first category
                              "blue",            # colour second category
                              "#cccc00")) +      # colour third category
  theme(aspect.ratio = 1/2,                                          # change aspect rati
o of plot
        plot.margin = unit(c(2,2,2,2), "cm"),                        # pad area around pl
ot for image capture
        panel.grid.major.x = element_line(color = "grey"),           # colour major gridl
ines grey
        plot.title = element_text(size = 17,                         # centre the plot ti
tle, size = 17 and bolden
                                  hjust = 0.5,
                                  face="bold"),
        axis.text = element_text(size=13, face =                     # make axis text bol
d and size 13
                                   "bold"),
        axis.line = element_line(colour = "grey"),                   # colour axis lines
grey
        axis.ticks = element_blank(),                                # remove axis ticks
        legend.position="none") +                                    # no legend
  coord_flip()+                                    # flip plot to horizontal
  scale_y_continuous(limits=c(-0.5,1.5),           # alter y axis to start at -0.5 and end
at 1.5
                     expand = c(0, 0)) +           # y axis limits to start at plot origin
  labs(x = "",                                     # plot to have no X- or Y- titles but a
main plot title
       y="",
       title = "Contributions of component indices to quarterly change in \nNICEI** (pp
s)"
  )
```

Automatic plot image export

```
png("contributions_resized.png", width = 950, height = 750)
final_plot_sized
dev.off()
```