

TWD - hw 2

Michał Matuszyk

November 2023

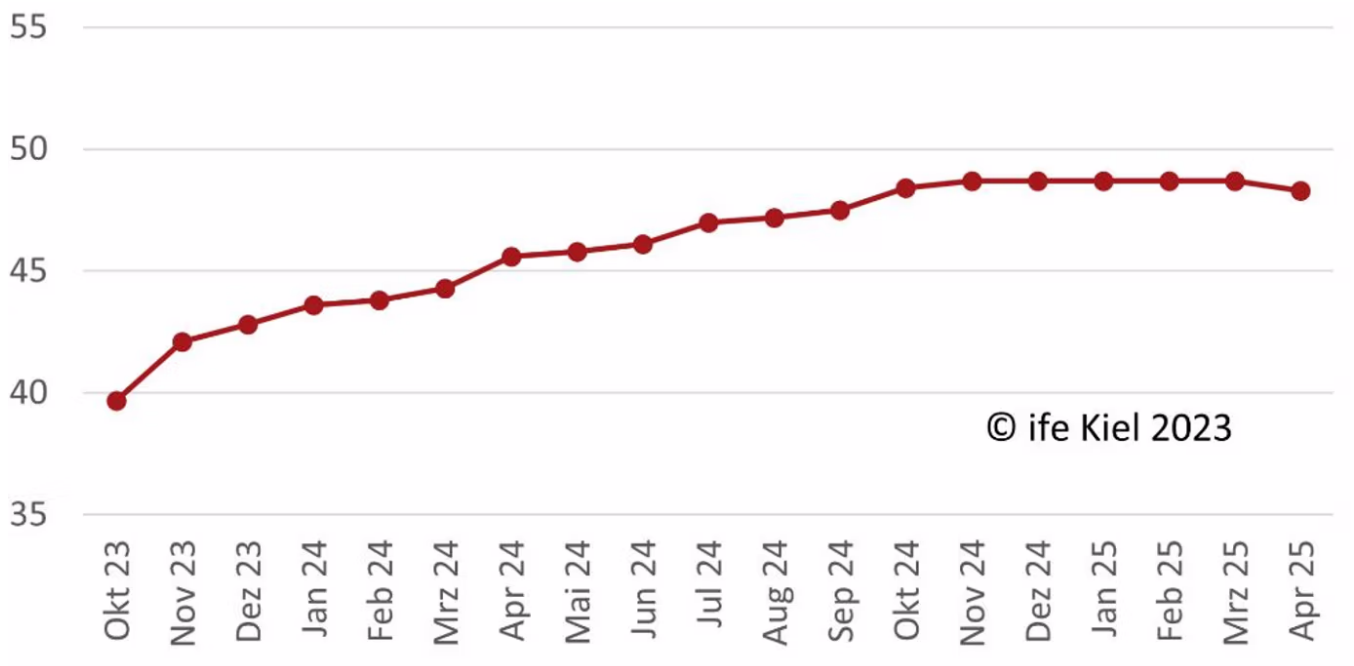
1 Wizualizacja źródłowa

Artykuł z dnia 3 listopada 2023 businessinsider.com.pl

Źródło wykresu: businessinsider.com.pl (data dostępu 7/11/2023)

Wykres (data dostępu 7/11/2023)

1.1 Wykres



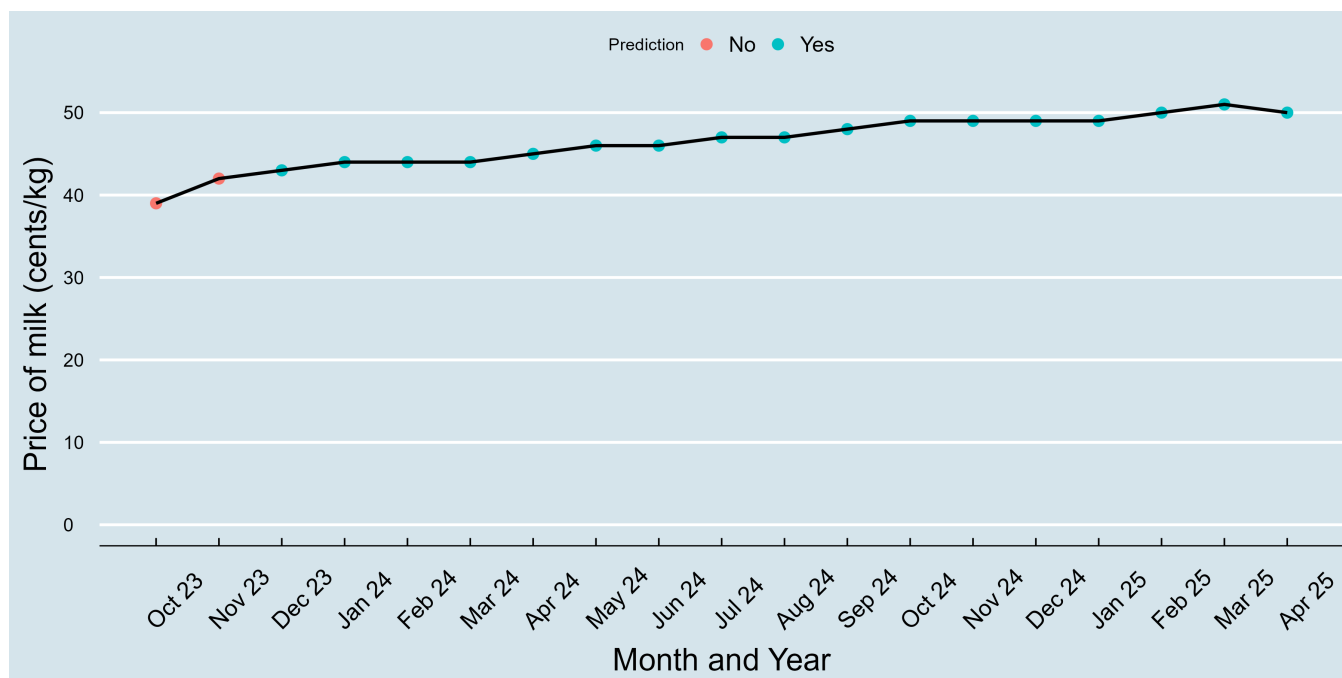
Rysunek 1: Oryginalny wykres

1.2 Poprawki

Należy poprawić:

1. Skala y - zaczyna się od 35, przez co wzrost jest zaburzony - łatwo odnieść wrażenie, że cena gwałtownie rośnie
2. Brak opisów osi Ox oraz Oy
3. Brak zaznaczenia, że dane są używane do Listopada, natomiast od Grudnia są to predykcje

1.3 Poprawiony wykres



Rysunek 2: Poprawiony wykres

1.4 Dlaczego lepszy

Ulepszenia wykresu:

1. Skala y zaczyna się od 0, przez co lepiej widać, że wzrost jest zdecydowanie niższy niż w oryginale
2. Dodanie opisów osi, przez co lepiej można zrozumieć dane
3. Użycie różnych kolorów punktów pozwala zaznaczyć, że większość danych prezentowanych jest predykcją

1.5 Kod użyty do wygenerowania wykresu

```
# Wczytanie potrzebnych modułów
library(dplyr)
library(ggplot2)
library(ggthemes)
# Używany do theme_economist()

# Stworzenie ramki danych
df <- data.frame(
  Value = c(39, 42, 43, 44, 44, 44, 45, 46, 46, 47, 47, 48, 49, 49, 49, 49, 50, 51, 50),
  Month_Year = c("Oct_23", "Nov_23", "Dec_23", "Jan_24", "Feb_24", "Mar_24", "Apr_24", "May_24", "Jun_24", "Jul_24", "Aug_24", "Sep_24", "Oct_24", "Nov_24", "Dec_24", "Jan_25", "Feb_25", "Mar_25", "Apr_25")
)
# Wartości odczytane z oryginalnego wykresu

# Modyfikacja ramki danych
df$Month_Year <- factor(df$Month_Year, levels = unique(df$Month_Year))
# Aby kolejność się nie zmieniła
```

```

df$future <- ifelse(as.Date(paste("01", df$Month_Year, sep = "_"),
                             format = "%d_%b_%y") > as.Date("01_Nov_23", format = "%d_%b_%y"))
# Dodaje kolumnę, aby stwierdzić czy są dane, czy przewidywania

# Tworzenie wykresu
wykres_poprawiony <- ggplot(data = df) +
# Tworze pusty wykres z danymi z df
  geom_point(aes(x = as.numeric(factor(Month_Year)), y = Value, color = factor(future)), size = 1) +
# Dodaje punkty na wykres
  geom_line(aes(x = as.numeric(factor(Month_Year)), y = Value, ), size = 1) +
# Dodaje linie, aby lepiej było widac zmiany
  scale_x_continuous(breaks = as.numeric(factor(df$Month_Year)), labels = df$Month_Year) +
# Dodaje os x
  scale_y_continuous(limits = c(0, 51)) +
# Dodaje os y, która zaczyna się od 0 a nie od 35 jak w oryginalnej POPRAWA
  xlab("Month_and_Year") +
# Dodanie opisu osi x – brakowało go w oryginalnej POPRAWA
  ylab("Price_of_milk_(cents/kg)") +
# Dodanie opisu osi y – brakowało go w oryginalnej POPRAWA
  guides(color = guide_legend(title = "Prediction")) +
# Dodanie legendy
  theme_economist() +
# Według mnie jeden z najładniejszych motywów
  theme(axis.text.y = element_text(size = 12),
# Zwiększenie rozmiaru elementów osi y
        axis.title.x = element_text(size = 20,
# Zwiększenie rozmiaru tytułu osi x
                                     margin = margin(t = 0.5, unit = "line")),
# Dodanie marginesu, według mnie lepiej tak wygląda
        axis.title.y = element_text(size = 20, margin = margin(r = 0.5, unit = "line")),
# Zwiększenie rozmiaru czcionki tytułu na osi y
        axis.text.x = element_text(angle = 45, hjust = 0, size = 15),
# Obrócenie (według mnie lepiej wygląda) i zwiększenie tekstu na osi x
        plot.margin = margin(b = 0.5, l = 0.5, r = 0.5, t = 0.5, unit = "line"))
# Dodanie marginesu do całego wykresu
)

# Zapisanie wykresu
ggsave("wykres_poprawiony.png", plot = wykres_poprawiony, width = 12, height = 6, dpi = 800)
# Ustawiam proporcje jak w oryginalnej, zwiększyłem dpi (wg mnie lepiej wygląda z większą), za

```