

# Wojskowa Akademia Techniczna



## Analiza i wizualizacja danych

Wizualizacja danych wieloczynnikowych

**Prowadzący:** xxxx

**Wykonał:** Konrad Prusaczyk

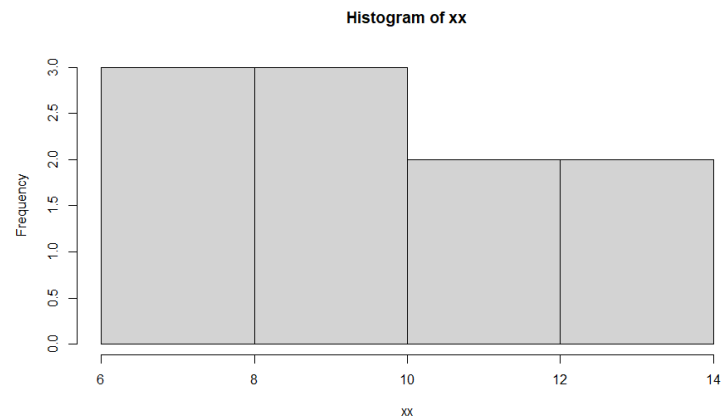
**Grupa:** WCY18IJ3S1

**Data wykonania:** 22/04/2021

## Zadanie 1

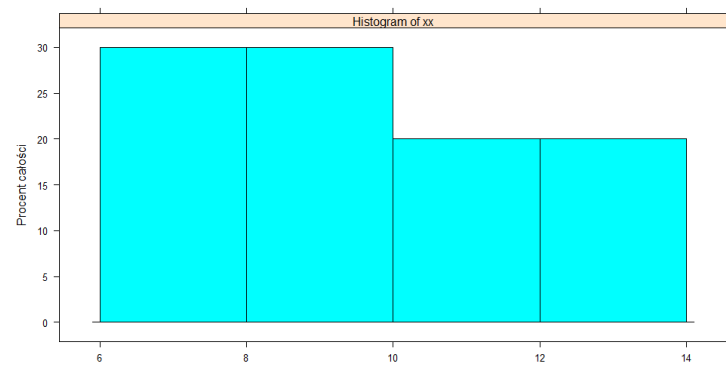
### Program 1

```
par(mai = c(1, 1, 1, 1), omi = c(0, 0, 0, 0))
xx <- c(9.20, 6.00, 6.00, 11.25, 11.00, 7.25, 9.7, 13.25, 14.00, 8.00)
hist(xx, breaks = c(6, 8, 10, 12, 14), right = F)
```



### Wykres generowany pakietem lattice

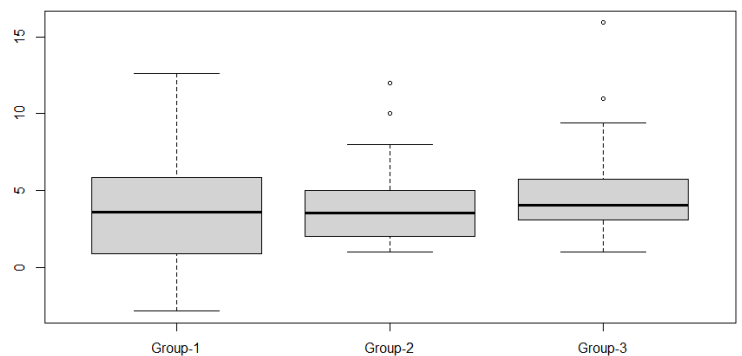
```
library(lattice)
par(mai = c(1, 1, 1, 1), omi = c(0, 0, 0, 0))
xx <- c(9.20, 6.00, 6.00, 11.25, 11.00, 7.25, 9.7, 13.25, 14.00, 8.00)
histogram(~ xx | "Histogram of xx",
breaks = c(6, 8, 10, 12, 14),
right = F)
```



## Zadanie 2

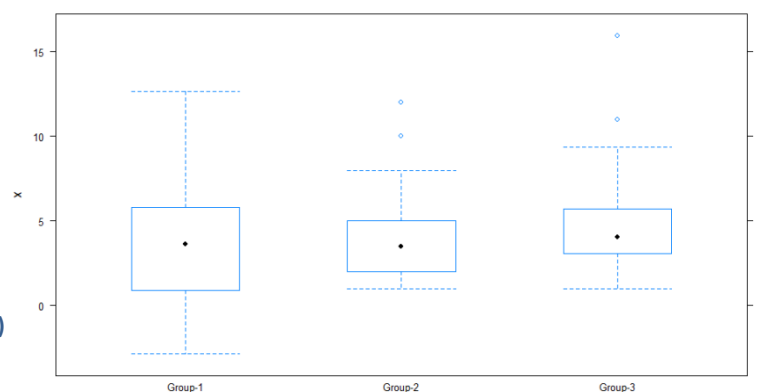
### Program 2

```
par(mai = c(1, 1, 1, 1), omi = c(0, 0, 0, 0))
set.seed(591)
xx1 <- rnorm(20, mean = 3, sd = 3.6)
xx2 <- rpois(40, lambda = 3.5)
xx3 <- rchisq(31, df = 5, ncp = 0)
box1 <- boxplot(xx1, xx2, xx3, names = c("Group-1",
"Group-2", "Group-3"), cex = 0.7)
```

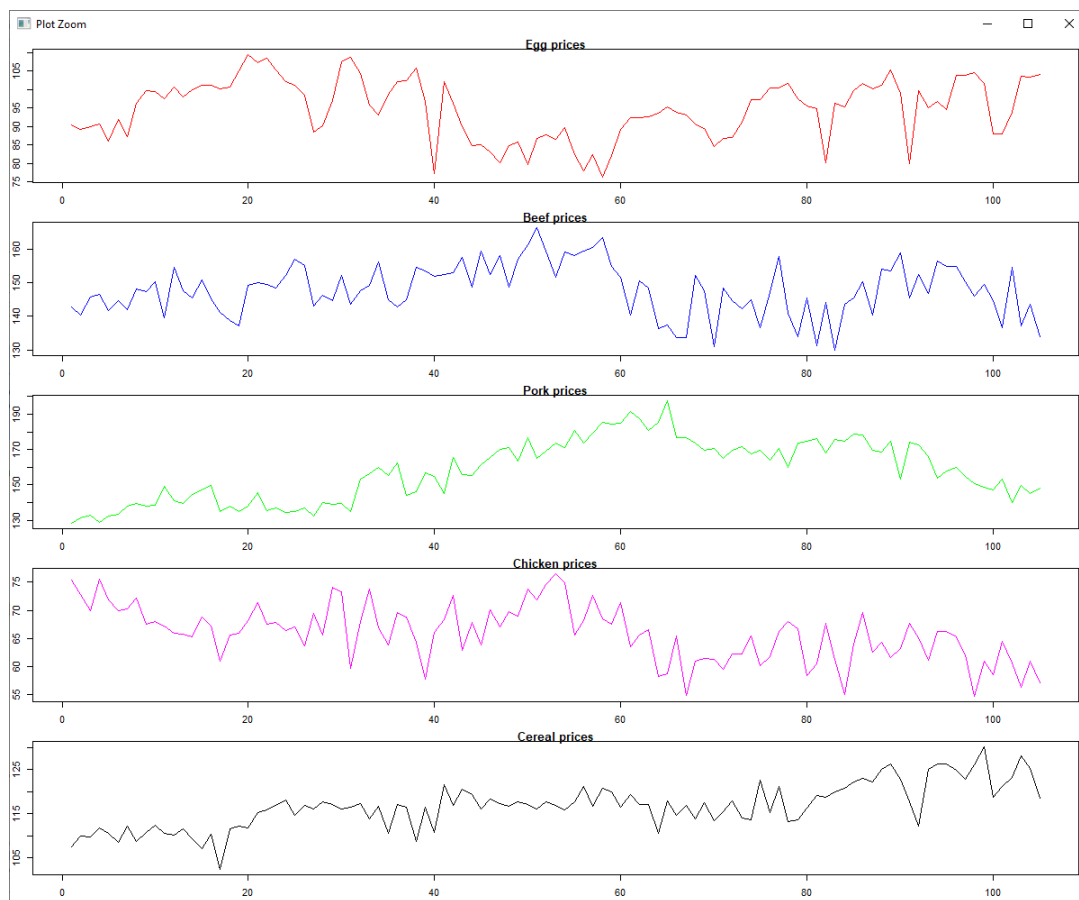


### Wykres generowany pakietem lattice

```
par(mai = c(1, 1, 1, 1), omi = c(0, 0, 0, 0))
set.seed(591)
xx1 <- rnorm(20, mean = 3, sd = 3.6)
xx2 <- rpois(40, lambda = 3.5)
xx3 <- rchisq(31, df = 5, ncp = 0)
library(lattice)
df <- data.frame(x=c(xx1, xx2, xx3),
group=factor(c(rep("Group-1", length(xx1)),
rep("Group-2", length(xx2)), rep("Group-3", length(xx3)))))
bwplot(x~group, df, cex = 0.7)
```



Poniżej przedstawiłem zmiany zmiennych odpowiadających cenom produktów w czasie.

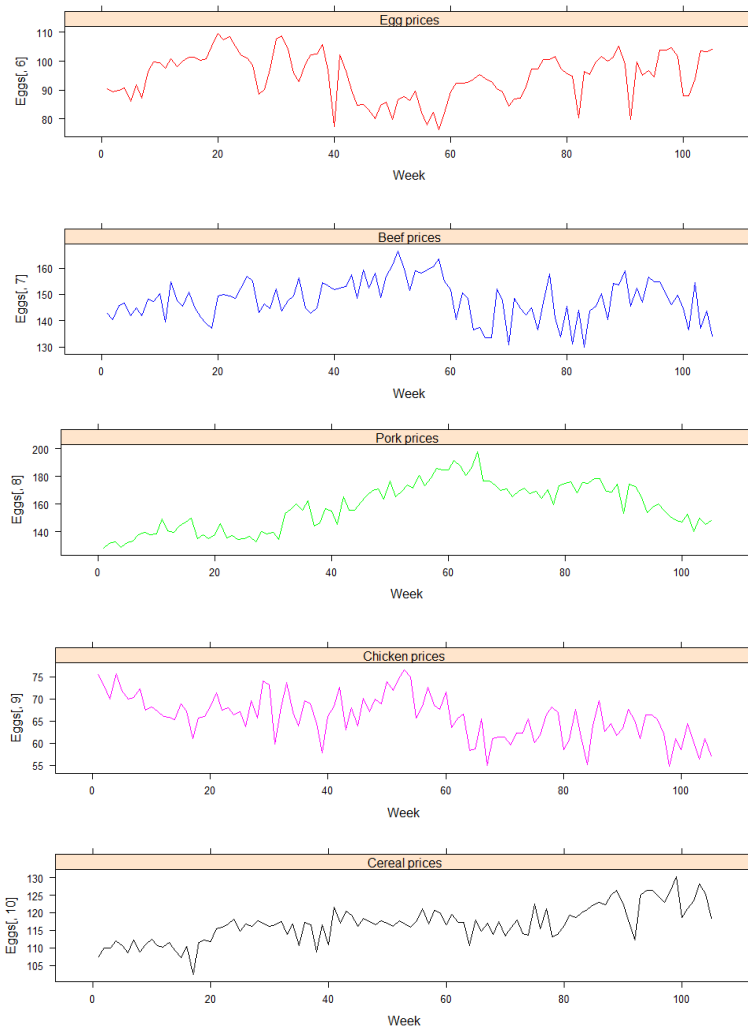


### Kod zad 3

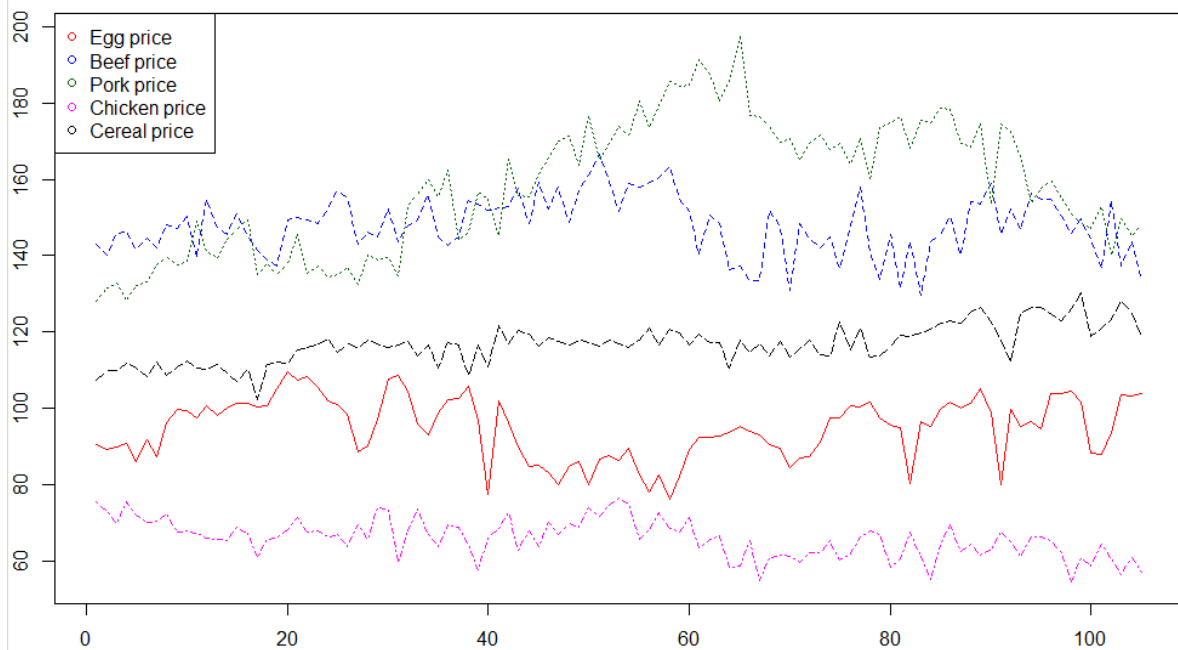
```
attach(mtcars)
par(mfrow=c(5,1))
ts.plot(Eggs[,6], main="Egg prices", col="red")
ts.plot(Eggs[,7], main="Beef prices", col="blue")
ts.plot(Eggs[,8], main="Pork prices", col="green")
ts.plot(Eggs[,9], main="Chicken prices", col="magenta")
ts.plot(Eggs[,10], main="Cereal prices", col="black")
```

### Zamiana wykresów na pakiet lattice:

```
names = c("Egg prices", "Beef prices", "Pork prices", "Chicken prices", "Cereal prices")
p1 <- xyplot(Eggs[,6] ~ Week | "Egg prices", data=df2, type = "l", col.line = "red")
p2 <- xyplot(Eggs[,7] ~ Week | "Beef prices", data=df2, type = "l", col.line = "blue")
p3 <- xyplot(Eggs[,8] ~ Week | "Pork prices", data=df2, type = "l", col.line = "green")
p4 <- xyplot(Eggs[,9] ~ Week | "Chicken prices", data=df2, type = "l", col.line = "magenta")
p5 <- xyplot(Eggs[,10] ~ Week | "Cereal prices", data=df2, type = "l", col.line = "black")
print(p1, split = c(1, 1, 1, 2), more = TRUE)
print(p2, split = c(1, 2, 1, 2), more = TRUE)
print(p3, split = c(1, 3, 1, 2), more = TRUE)
print(p4, split = c(1, 4, 1, 2), more = TRUE)
print(p5, split = c(1, 5, 1, 2), more = FALSE) # more = FALSE is redundant
```



Wykres przedstawiający zestawienie zmian cen każdego z produktów w czasie podobnie jak wyżej.

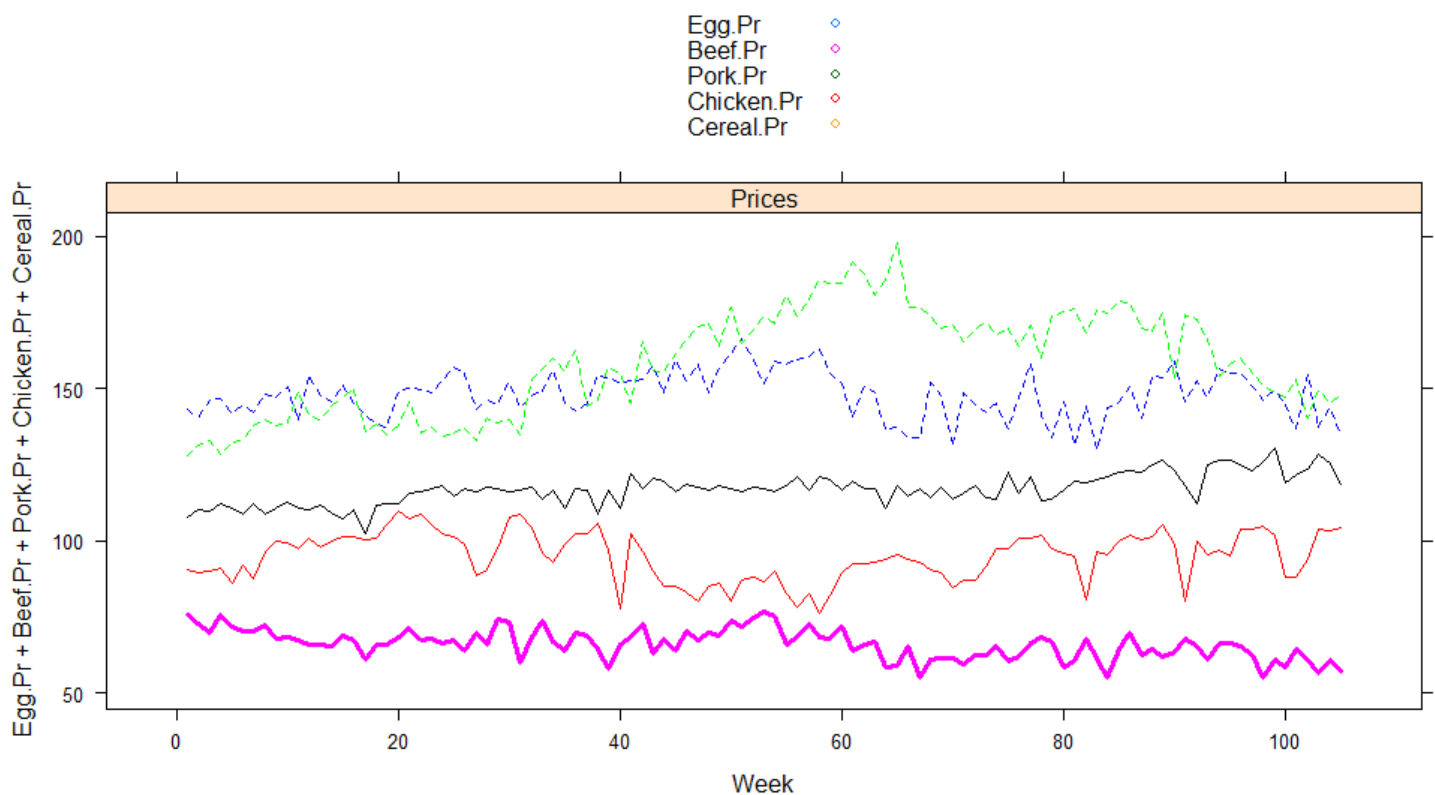


### Kod zad 3

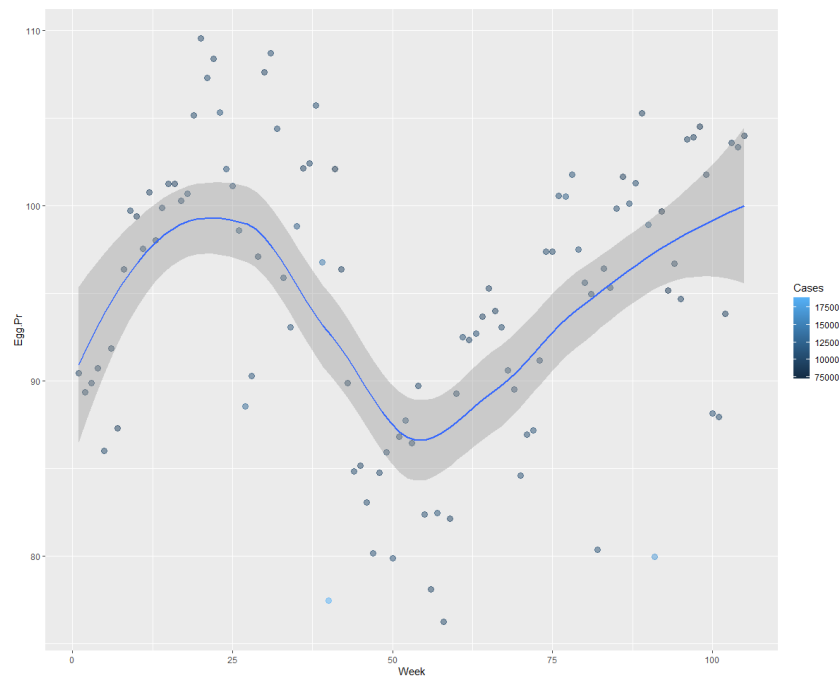
```
par(mfrow=c(1,1))
par(bg="white")
matplot(df[,6:10], type = "l", pch=1, col= c("red", "blue", "darkgreen", "magenta", "black")) #plot
legend("topleft", legend = c("Egg price","Beef price", "Pork price", "Chicken price", "Cereal price"), col=c("red",
"blue", "darkgreen", "magenta", "black"), pch=1)
```

Zamiana wykresów na pakiet lattice:

```
xyplot(Egg.Pr + Beef.Pr + Pork.Pr + Chicken.Pr + Cereal.Pr ~ Week | "Prices",
  data = df2,
  auto.key = TRUE,
  type = "l",
  lty = c(1, 2, 2, 1),
  lwd = c(1, 1, 1, 3),
  col.line = c("red", "blue", "green", "magenta", "black"),
  layout = c(1,1))
```



Wykres przedstawiający zmiany ceny jajek przy jednoczesnym uwzględnieniu sprzedaży detalicznej jaj w skrzynkach. Kolory punktów określają ilość sprzedanych skrzyń jajek przy obecnej cenie jaj w danym tygodniu.

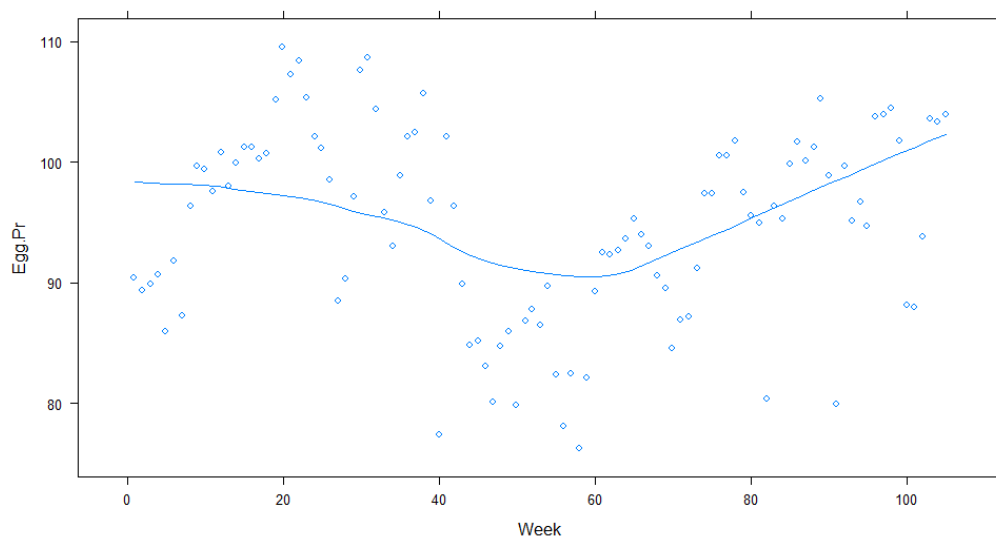


### Kod zad 3

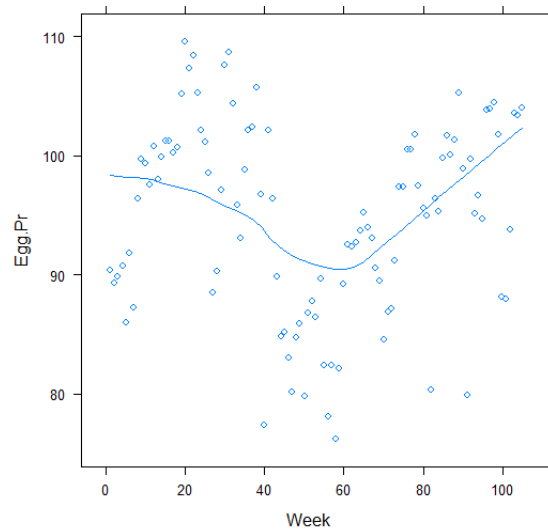
```
df %>%
  ggplot(aes(Week, Egg.Pr, colour=Cases)) +
  geom_point(size = 3, alpha=0.5)+
  geom_smooth()
```

Zamiana wykresów na pakiet lattice:

```
xyplot(Egg.Pr ~ Week, df, jitter.x = TRUE,
  type = c("p", "smooth"),
  xlab = "Week",
  ylab = "Egg.Pr",
  layout = c(1,1))
```

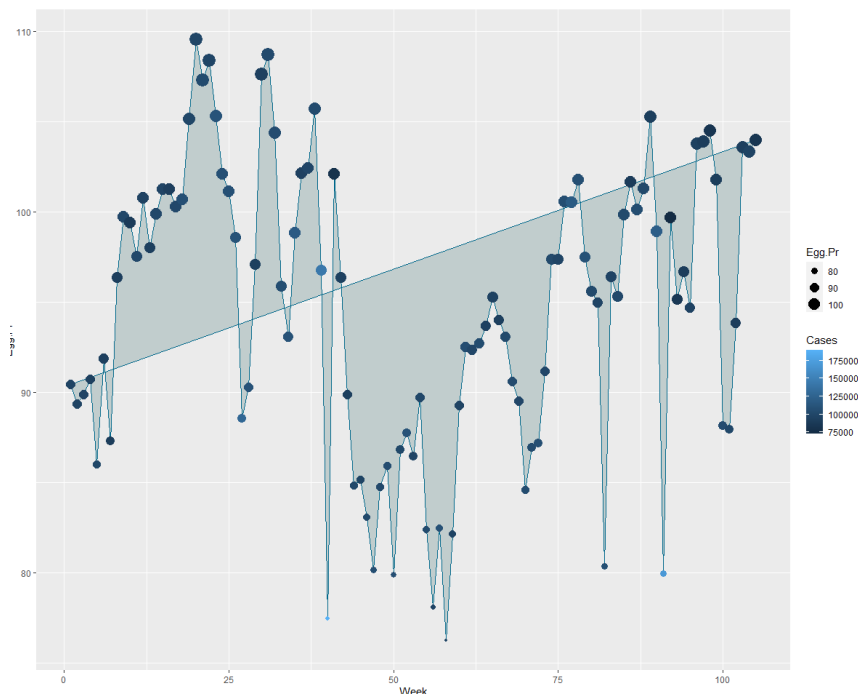


Argument "smooth" wylicza funkcję straty w inny sposób w pakiecie lattice, stąd wynika różnica w ekstremach między jedną a drugą funkcją. Dodatkowo, między obydwoma wykresami jest różnica w layout'cie przez co również funkcja może w małym stopniu się różnić. Po zmianie argumentu layout = c(2,1) otrzymujemy wykres:



W pakiecie lattice nie ma możliwości nadania reprezentacji dodatkowej zmiennej poprzez nasilenie koloru wypełnienia punktów. (Albo ja się tego nie doszukałem w dokumentacji). Stąd też brak legendy dla brakującej zmiennej cases.

**Wykres przedstawiający zmiany ceny jajek przy jednoczesnym uwzględnieniu ilości sprzedanych jaj w skrzyniach. Widać, że przy wysokich cenach jajek sprzedaż detaliczna jest niska i na odwrót.**

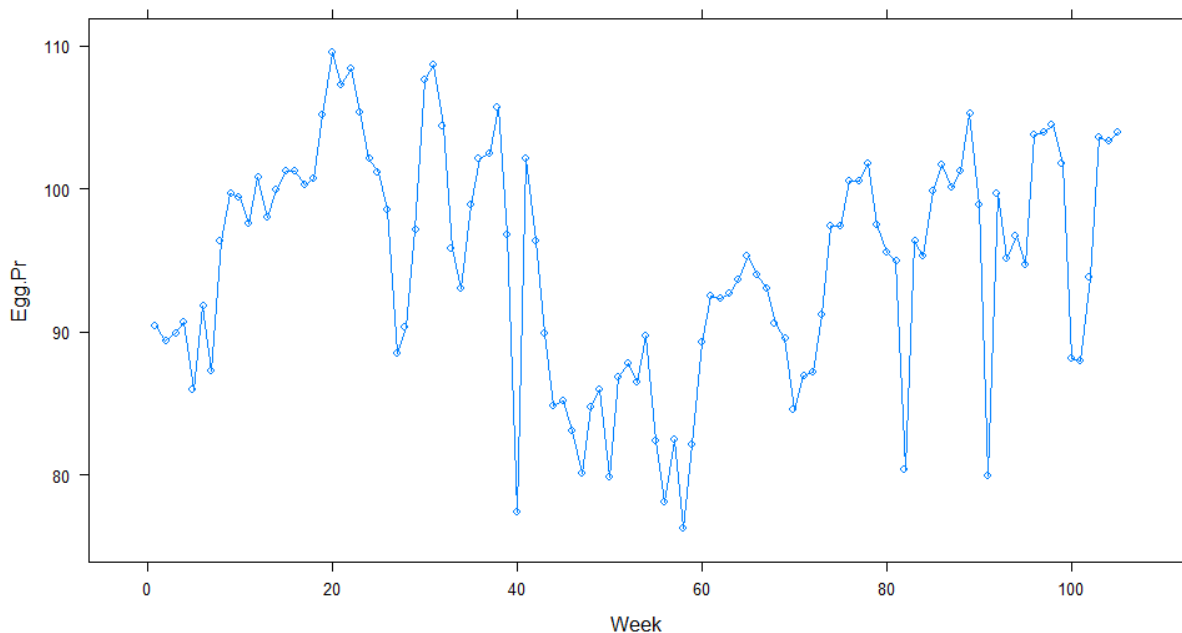


## Kod zad 3

```
df %>%  
  ggplot(aes(Week, Egg.Pr)) +  
  geom_polygon(colour="deepskyblue4", fill="azure3")+  
  geom_point(aes(size=Egg.Pr,  
    colour=Cases))
```

Zamiana wykresów na pakiet lattice:

```
xyplot(df2$Egg.Pr ~ df2$Week, df, jitter.x = TRUE,  
  type = c("o"),  
  xlab = "Week",  
  ylab = "Egg.Pr",  
  layout = c(1,1))
```



W pakiecie `lattice` nie ma możliwości nadania reprezentacji dodatkowej zmiennej poprzez nasilenie koloru wypełnienia punktów lub zwiększenia ich objętości. (Albo ja się tego nie doszukałem w dokumentacji). Stąd też brak legendy dla brakującej zmiennej `cases` oraz pogrupowania `Egg.Pr`. Również nie udało mi się zamalować obszaru „utworzonych figur” z zakreślonych punktów. W pakiecie `lattice` brakuję podobnej funkcji do `geom_point()` dostępnej w pakiecie `ggplot`.