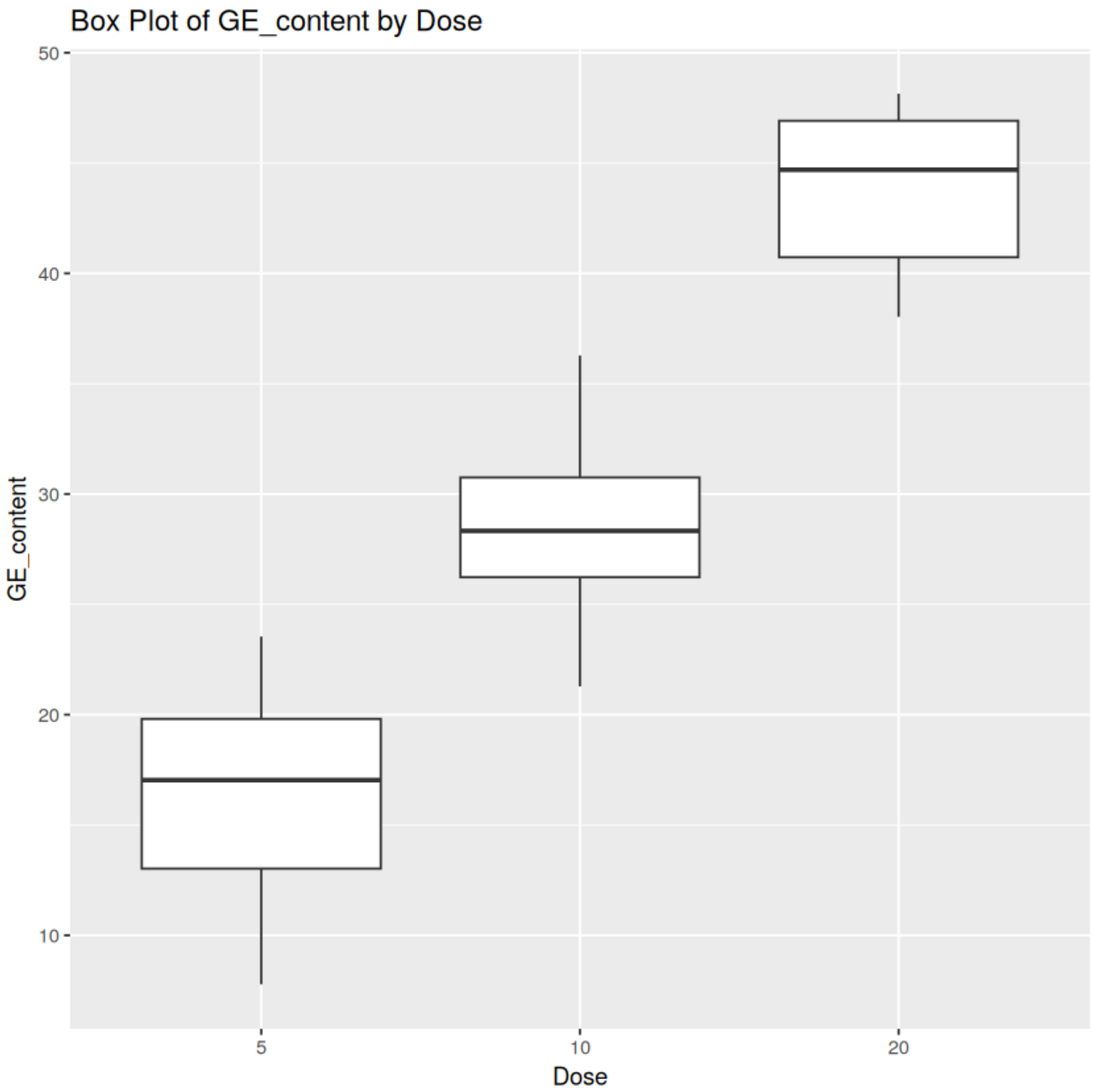
Task_3_a

October 8, 2025

[107]:

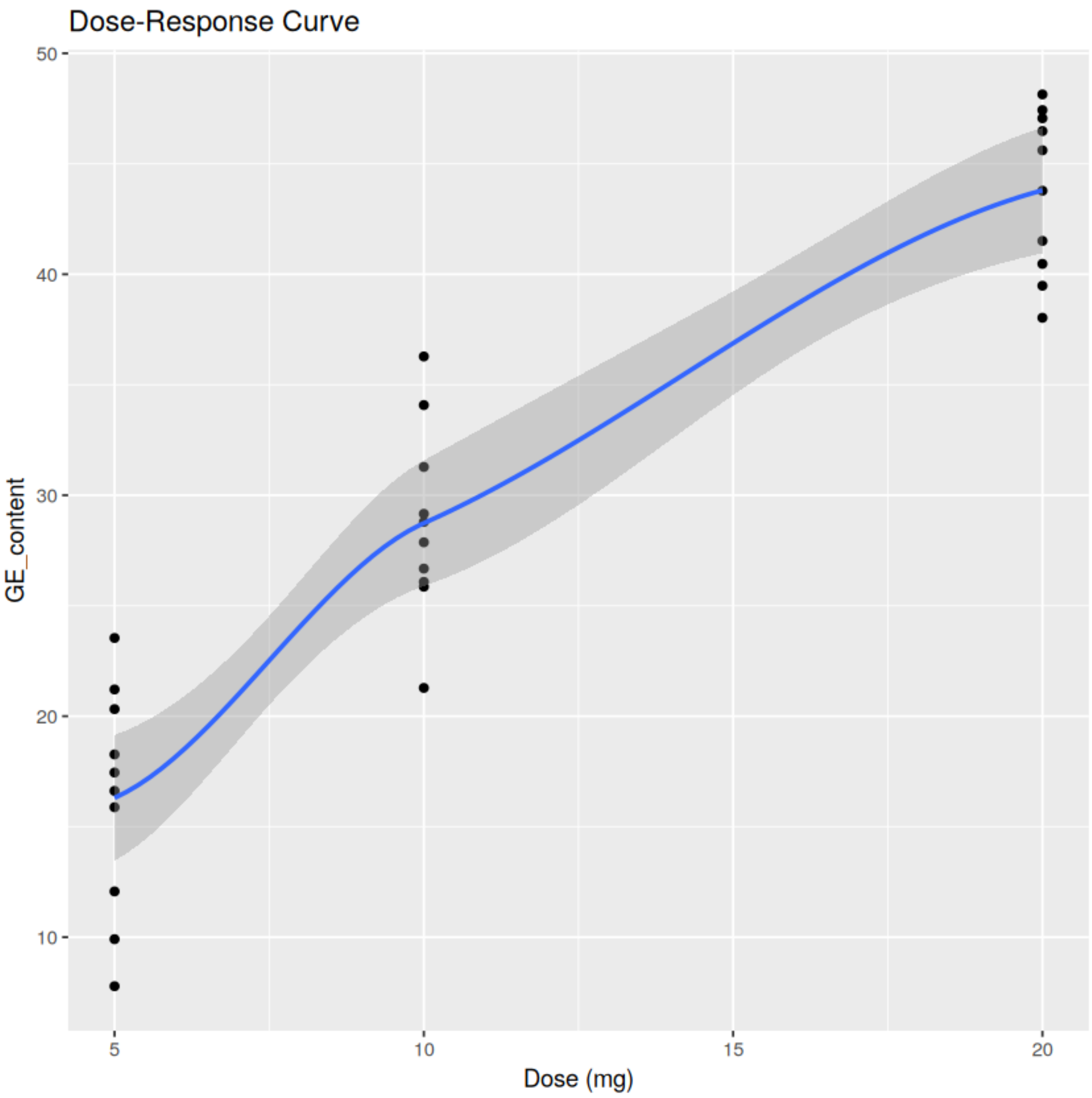
```r
ggplot(data_low, aes(x = factor(Dose), y = GE_content)) +
  geom_boxplot() +
  labs(title = "Box Plot of GE_content by Dose", x = "Dose", y = "GE_content")
```



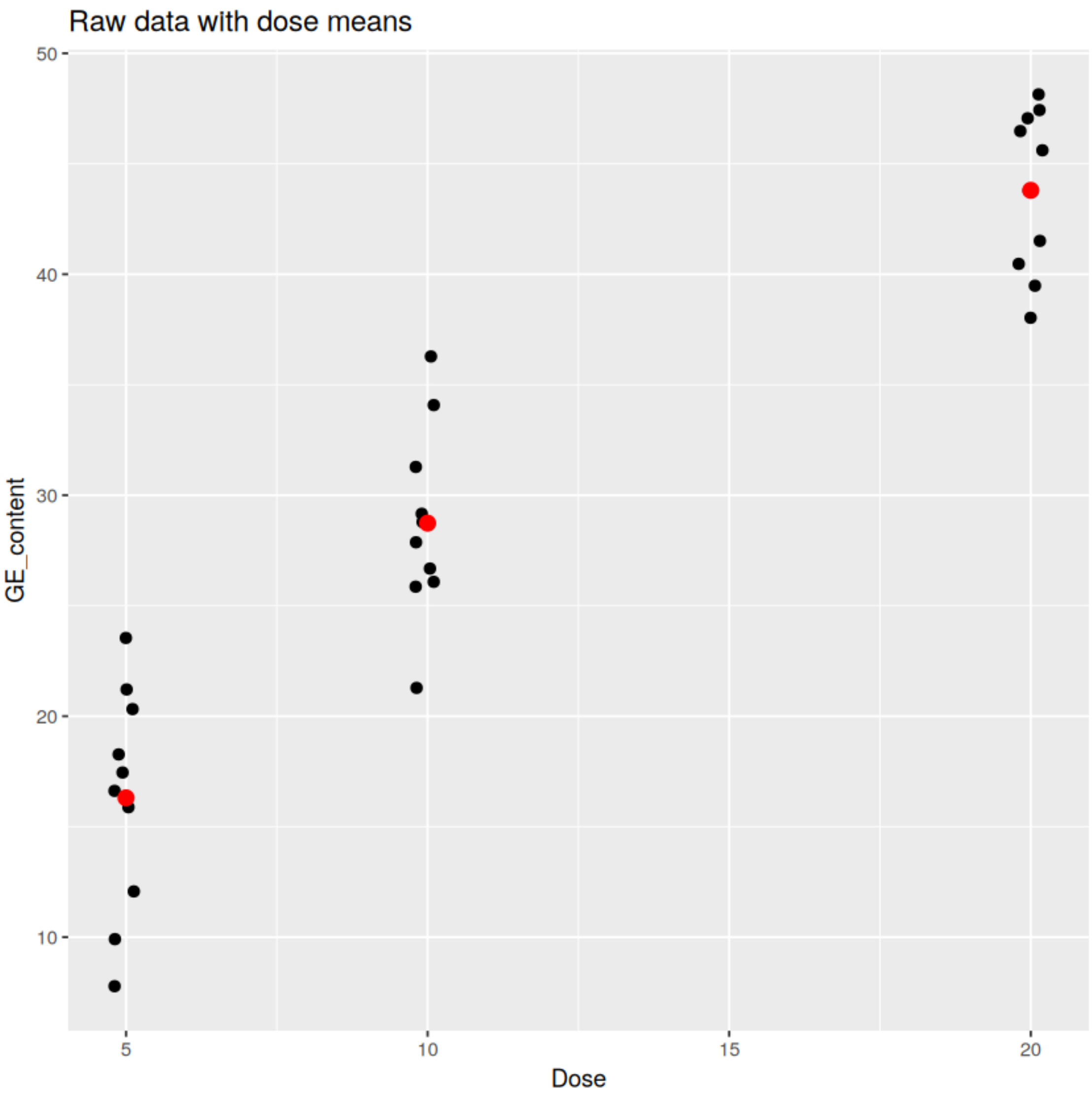
[108]:

```r
# Plot the data
ggplot(data_low, aes(x = Dose, y = GE_content)) +
  geom_point() +
  geom_smooth(method = "loess") +
  labs(title = "Dose-Response Curve", x = "Dose (mg)", y = "GE_content")
```

```
`geom_smooth()` using formula = 'y ~ x'
Warning message in simpleLoess(y, x, w, span, degree = degree, parametric =
parametric, :
"pseudoinverse used at 4.925"
Warning message in simpleLoess(y, x, w, span, degree = degree, parametric =
parametric, :
"neighborhood radius 15.075"
Warning message in simpleLoess(y, x, w, span, degree = degree, parametric =
parametric, :
"reciprocal condition number  1.2461e-16"
Warning message in simpleLoess(y, x, w, span, degree = degree, parametric =
parametric, :
"There are other near singularities as well. 227.26"
Warning message in predLoess(object$y, object$x, newx = if (is.null(newdata))
object$x else if (is.data.frame(newdata))
as.matrix(model.frame(delete.response(terms(object)), :
"pseudoinverse used at 4.925"
Warning message in predLoess(object$y, object$x, newx = if (is.null(newdata))
object$x else if (is.data.frame(newdata))
as.matrix(model.frame(delete.response(terms(object)), :
"neighborhood radius 15.075"
Warning message in predLoess(object$y, object$x, newx = if (is.null(newdata))
object$x else if (is.data.frame(newdata))
as.matrix(model.frame(delete.response(terms(object)), :
"reciprocal condition number  1.2461e-16"
Warning message in predLoess(object$y, object$x, newx = if (is.null(newdata))
object$x else if (is.data.frame(newdata))
as.matrix(model.frame(delete.response(terms(object)), :
"There are other near singularities as well. 227.26"
```



[112]:
```
ggplot(data_low, aes(Dose, GE_content)) +
  geom_jitter(width = 0.2, height = 0, size = 2) +
  stat_summary(fun = mean, geom = "point", colour = "red", size = 3) +
  labs(title = "Raw data with dose means")
```



[113]:
```
library(drc)

ml1 <- drm(GE_content ~ Dose, data = data_low,
fct = LL.3(names = c("Hill slope", "Min", "Max")))
```

[114]:
```
summary(ml1)
```

```
Model fitted: Log-logistic (ED50 as parameter) with lower limit at 0 (3 parms)

Parameter estimates:

                       Estimate Std. Error t-value  p-value
```

```
Hill slope:(Intercept) -1.15247    0.41204 -2.7970 0.009393 **
Min:(Intercept)        76.56995   37.85265  2.0228 0.053095 .
Max:(Intercept)        15.55276   13.87981  1.1205 0.272356
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error:

 4.38386 (27 degrees of freedom)
```

[115]:

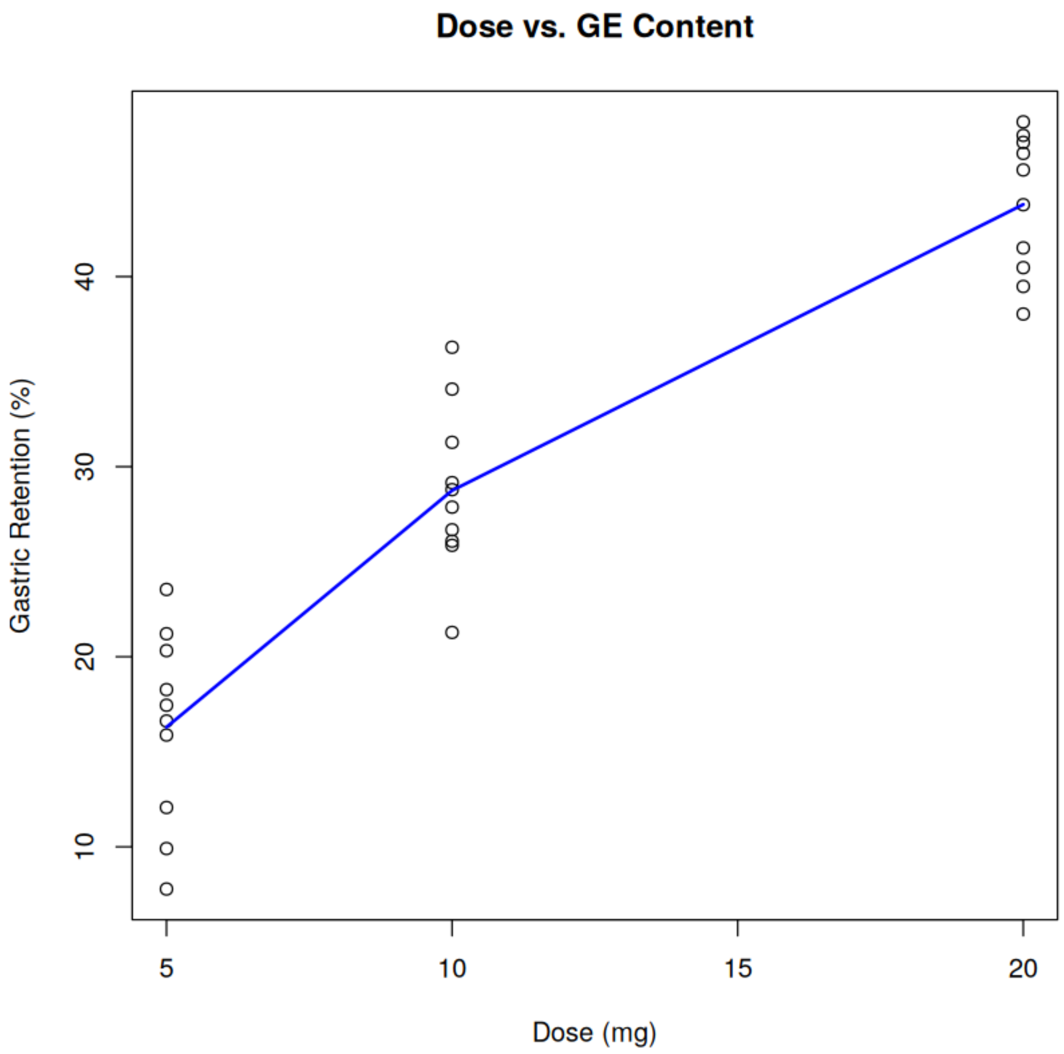
```
summary(ml1)[[3]]
```

A matrix: 3 × 4 of type dbl

	Estimate	Std. Error	t-value	p-value
Hill slope:(Intercept)	-1.152473	0.4120441	-2.796964	0.009392691
Min:(Intercept)	76.569954	37.8526532	2.022842	0.053095391
Max:(Intercept)	15.552757	13.8798116	1.120531	0.272355939

[116]:

```
# Plot the data and the fitted polynomial regression line
plot(data_low$Dose, data_low$GE_content, main = "Dose vs. GE Content", xlab = "Dose (mg)", ylab = "Gastric Retention (%)")
lines(sort(data_low$Dose), predict(ml1, newdata = data[order(data_low$Dose), ]), col = "blue", lwd = 2)
```



```
[117]: ED(ml1, 50)
```

```
Estimated effective doses

       Estimate Std. Error
e:1:50   15.553     13.880
```

15.553 dose to produce a retention of 50% of gastric content at 30 minutes.

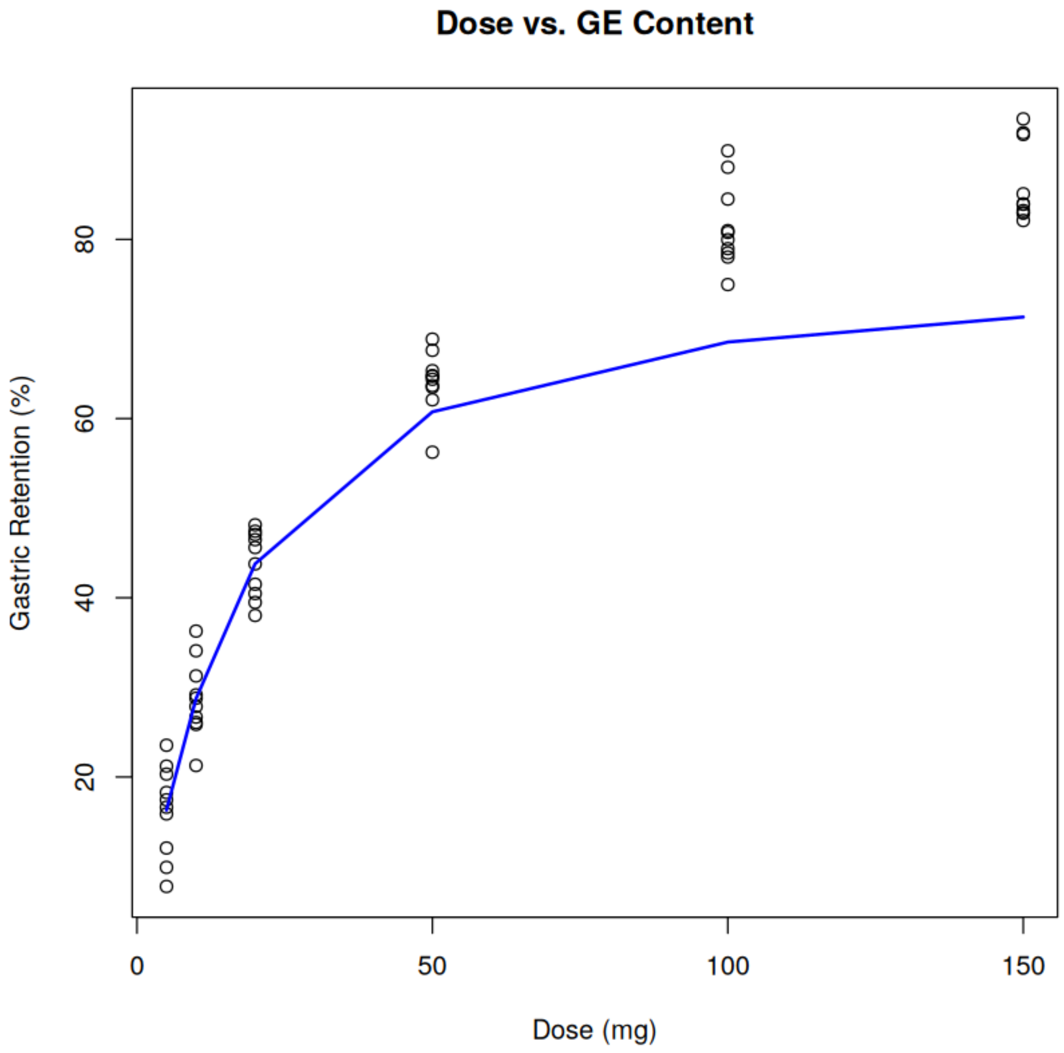
```
[118]: ED(ml1, 80)
```

```
Estimated effective doses
```

```
        Estimate Std. Error
e:1:80    51.786     68.174
```

51.786 the dose that produces 80% gastric retention at 30 minutes.

[119]:
```
# Plot the data and the fitted LL.3 regression line
plot(data$Dose, data$GE_content, main = "Dose vs. GE Content", xlab = "Dose␣
 ↪(mg)", ylab = "Gastric Retention (%)")
lines(sort(data$Dose), predict(ml1, newdata = data[order(data$Dose), ]), col =␣
 ↪"blue", lwd = 2)
```



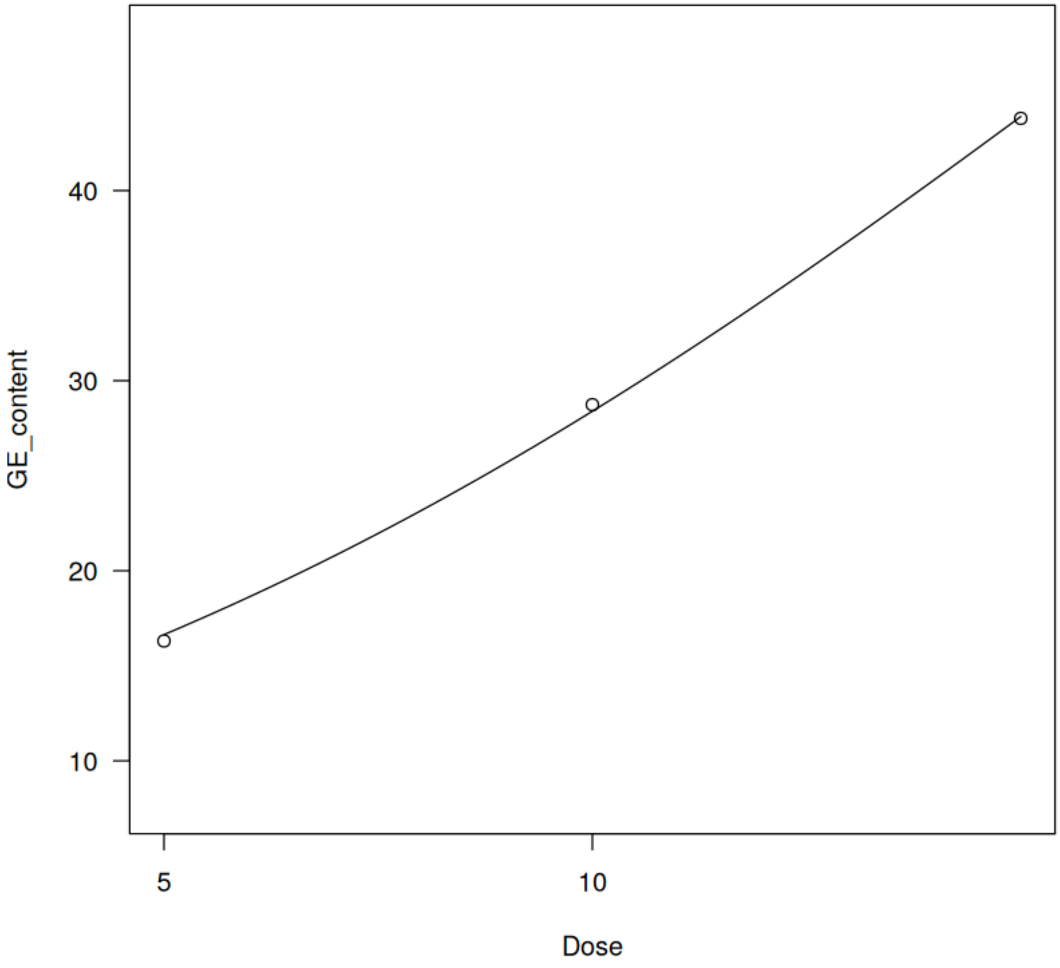
[120]:

```
k1 <- drm(GE_content ~ Dose, data = data_low,
fct = MM.2())
summary(k1)
plot(k1)
AIC(ml1, k1)
```

```
Model fitted: Michaelis-Menten (2 parms)

Parameter estimates:

              Estimate Std. Error t-value   p-value
d:(Intercept)  96.6920    15.3002  6.3197 7.777e-07 ***
e:(Intercept)  24.0561     6.1135  3.9349 0.0005002 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error:

 4.314761 (28 degrees of freedom)
```

A data.frame: 2 × 2

	df	AIC
	<dbl>	<dbl>
ml1	4	178.6513
k1	3	176.7890



Log-logistic (ED50 as parameter) with lower limit at 0, used the fct for Dose-Response Model. Michaelis-Menten fct can be used as a replacement for fct as it improves it by 1%.

[121]:
```
# getMeanFunctions()
```

[122]:
```
# # Fit a polynomial regression model
# model_poly <- lm(GE_content ~ poly(Dose, 2), data = data)

# # Check the summary of the model
# summary(model_poly)
```

[123]:
```
ml2 <- drm(GE_content ~ Dose, data = data,
fct = MM.2())
```

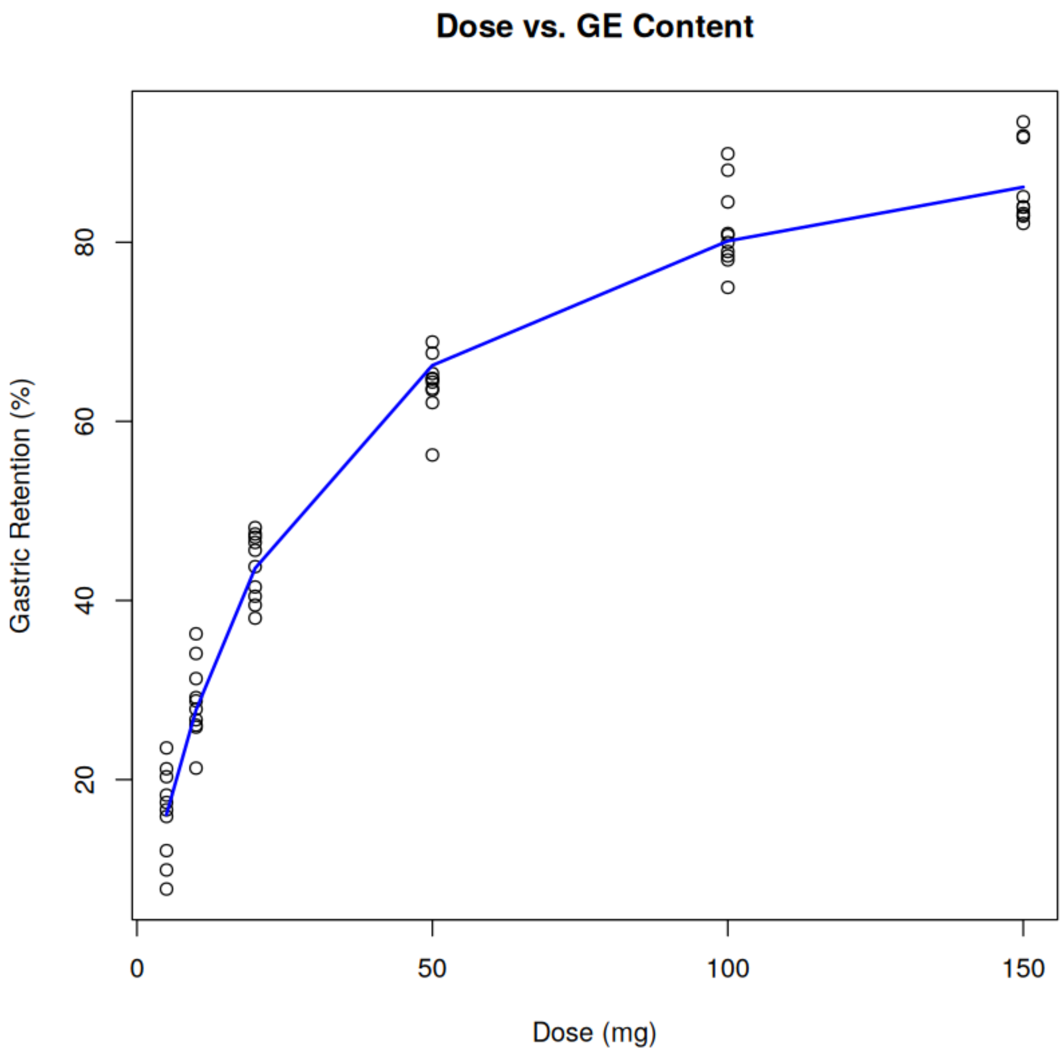
[124]:

```
summary(ml2)
```

```
Model fitted: Michaelis-Menten (2 parms)

Parameter estimates:

              Estimate Std. Error t-value   p-value    
d:(Intercept) 101.3913     1.8639  54.398 < 2.2e-16 ***
e:(Intercept)  26.5098     1.5221  17.417 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error:

 4.284886 (58 degrees of freedom)
```

[125]:

```
# Plot the data and the fitted polynomial regression line
plot(data$Dose, data$GE_content, main = "Dose vs. GE Content", xlab = "Dose␣
 ↪(mg)", ylab = "Gastric Retention (%)")
lines(sort(data$Dose), predict(ml2, newdata = data[order(data$Dose), ]), col =␣
 ↪"blue", lwd = 2)
```



[126]:

```
# Create a sequence of effects
effects <- seq(10, 90, by = 10)
# Estimate doses for each effect
doses <- numeric(length(effects))
for (i in 1:length(effects)) {
result <- ED(ml2, effects[i])
doses[i] <- result
}
# Create a data frame for plotting
df <- data.frame(Effect = effects, Dose = doses)
```

```
Estimated effective doses
```

```
       Estimate Std. Error
e:1:10  2.94554    0.16912

Warning message in doses[i] <- result:
"number of items to replace is not a multiple of replacement length"


Estimated effective doses

       Estimate Std. Error
e:1:20  6.62746    0.38052

Warning message in doses[i] <- result:
"number of items to replace is not a multiple of replacement length"


Estimated effective doses

       Estimate Std. Error
e:1:30 11.36136    0.65232

Warning message in doses[i] <- result:
"number of items to replace is not a multiple of replacement length"


Estimated effective doses

       Estimate Std. Error
e:1:40  17.6732     1.0147

Warning message in doses[i] <- result:
"number of items to replace is not a multiple of replacement length"


Estimated effective doses

       Estimate Std. Error
e:1:50  26.5098     1.5221

Warning message in doses[i] <- result:
"number of items to replace is not a multiple of replacement length"


Estimated effective doses

       Estimate Std. Error
e:1:60  39.7648     2.2831

Warning message in doses[i] <- result:
"number of items to replace is not a multiple of replacement length"


Estimated effective doses
```

```
       Estimate Std. Error
e:1:70  61.8563     3.5515
```

```
Warning message in doses[i] <- result:
"number of items to replace is not a multiple of replacement length"
```

```

Estimated effective doses

       Estimate Std. Error
e:1:80 106.0394     6.0883
```

```
Warning message in doses[i] <- result:
"number of items to replace is not a multiple of replacement length"
```

```

Estimated effective doses

       Estimate Std. Error
e:1:90  238.589     13.699
```

```
Warning message in doses[i] <- result:
"number of items to replace is not a multiple of replacement length"
```

[127]:

```r
# First, install plotly if you haven't already
# install.packages("plotly")

library(plotly)

# Assuming df is your data frame with Dose and Effect columns
# Create the interactive plot
p <- ggplot(df, aes(x = Dose, y = Effect)) +
  geom_line() +
  geom_point() +
  labs(x = "Dose", y = "Effect (%)")

# Convert to interactive plotly object
ggplotly(p, tooltip = c("x", "y"))
```

```
HTML widgets cannot be represented in plain text (need html)
```

Dose:106.0393 - Effect:80

Dosage level suggested for the follow-up experiment