

Kinetyka

Bartosz Pluciński

19 01 2021

1. Wstęp

Miejsce na wstęp teoretyczny.

2. Wykonanie eksperymentu

```
# Wczytywanie pakietu do robienia wykresów
library(ggplot2)
library(gridExtra)
# Molowy współczynnik ekstynkcji utlenionego pirogalolu
mwe = 2640
# Odstępy czasu po których dokonywano pomiaru absorbancji;
# od 0 do 180 s, co 10 s
time = seq(0, 180, 10)
```

3. Wyniki

3.1 Pierwsze stężenie substratu - 62,5 ppm

```
# wartości zmierzonej absorbancji uszeregowane w czasie;
# miejsca dziesiętne oddzielone kropką!
abs_s2a = c(0, 0.03, 0.044,
            0.061, 0.074, 0.087,
            0.099, 0.11, 0.121,
            0.129, 0.137, 0.144,
            0.151, 0.157, 0.163,
            0.168, 0.174, 0.177,
            0.182)
abs_s2b = c(0, 0.12, 0.15,
            0.18, 0.20, 0.21,
            0.21, 0.21, 0.21,
            0.21, 0.21, 0.21,
            0.21, 0.21, 0.21,
            0.21, 0.21, 0.21,
            0.21)
abs_s2c = c(0, 0.12, 0.15,
            0.18, 0.20, 0.21,
```

```

0.21, 0.21, 0.21,
0.21, 0.21, 0.21,
0.21, 0.21, 0.21,
0.21, 0.21, 0.21,
0.21)
# Obliczamy stężenie produktu C = abs * mwe
C_s2a = abs_s2a / mwe
C_s2b = abs_s2b / mwe
C_s2c = abs_s2c / mwe
# Obliczamy szybkość początkową reakcji
V0_st2a = (C_s2a[6] - C_s2a[1]) / (time[6] - time[1])
V0_st2a

```

```
## [1] 6.590909e-07
```

```

V0_st2b = (C_s2b[6] - C_s2b[1]) / (time[6] - time[1])
V0_st2b

```

```
## [1] 1.590909e-06
```

```

V0_st2c = (C_s2c[6] - C_s2c[1]) / (time[6] - time[1])
V0_st2c

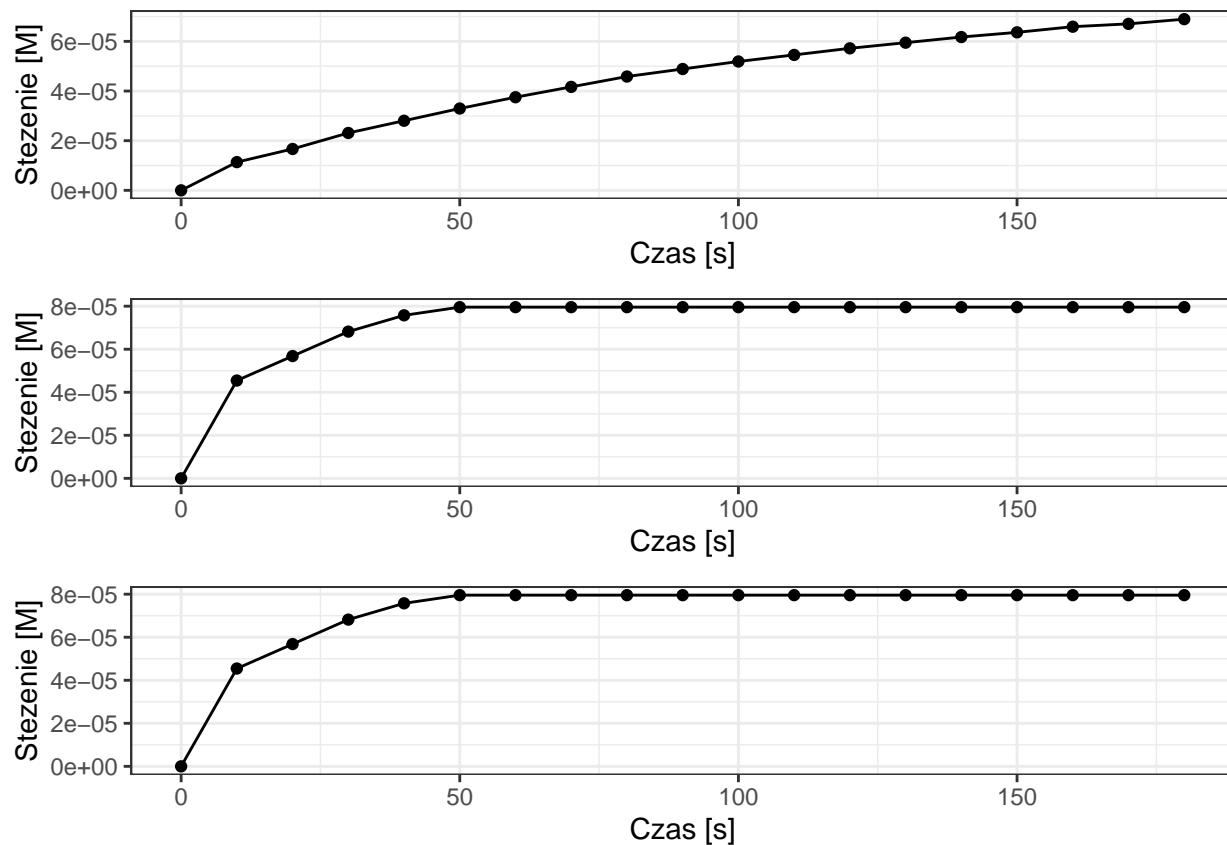
```

```
## [1] 1.590909e-06
```

```

# wykres pokazujący zmiany stężenia utlenionego pirogalolu w czasie
stezenie_2a = data.frame(time, C_s2a)
vplot_c2a = ggplot(data = stezenie_2a, aes(time, C_s2a)) +
  geom_point() +
  geom_line() +
  xlab("Czas [s]") +
  ylab("Stężenie [M]") +
  theme_bw()
stezenie_2b = data.frame(time, C_s2b)
vplot_c2b = ggplot(data = stezenie_2b, aes(time, C_s2b)) +
  geom_point() +
  geom_line() +
  xlab("Czas [s]") +
  ylab("Stężenie [M]") +
  theme_bw()
stezenie_2c = data.frame(time, C_s2c)
vplot_c2c = ggplot(data = stezenie_2c, aes(time, C_s2c)) +
  geom_point() +
  geom_line() +
  xlab("Czas [s]") +
  ylab("Stężenie [M]") +
  theme_bw()
gridExtra::grid.arrange(vplot_c2a, vplot_c2b, vplot_c2c, nrow = 3)

```



3.2 Drugie stężenie substratu - 125 ppm

```
# wartości zmierzonej absorbancji uszeregowane w czasie;
# miejsca dziesiętne oddzielone kropką!
abs_s3a = c(0, 0.014, 0.033,
            0.052, 0.074, 0.094,
            0.112, 0.129, 0.146,
            0.161, 0.174, 0.186,
            0.199, 0.212, 0.221,
            0.232, 0.241, 0.249,
            0.258)
abs_s3b = c(0, 0.12, 0.15,
            0.18, 0.20, 0.21,
            0.21, 0.21, 0.21,
            0.21, 0.21, 0.21,
            0.21, 0.21, 0.21,
            0.21, 0.21, 0.21,
            0.21)
abs_s3c = c(0, 0.12, 0.15,
            0.18, 0.20, 0.21,
            0.21, 0.21, 0.21,
            0.21, 0.21, 0.21,
            0.21, 0.21, 0.21,
            0.21, 0.21, 0.21,
```

```

0.21)
# Obliczamy stężenie produktu C = abs * mwe
C_s3a = abs_s3a / mwe
C_s3b = abs_s3b / mwe
C_s3c = abs_s3c / mwe
# Obliczamy szybkość początkową reakcji
V0_st3a = (C_s3a[6] - C_s3a[1]) / (time[6] - time[1])
V0_st3a

```

```
## [1] 7.121212e-07
```

```

V0_st3b = (C_s3b[6] - C_s3b[1]) / (time[6] - time[1])
V0_st3b

```

```
## [1] 1.590909e-06
```

```

V0_st3c = (C_s3c[6] - C_s3c[1]) / (time[6] - time[1])
V0_st3c

```

```
## [1] 1.590909e-06
```

```

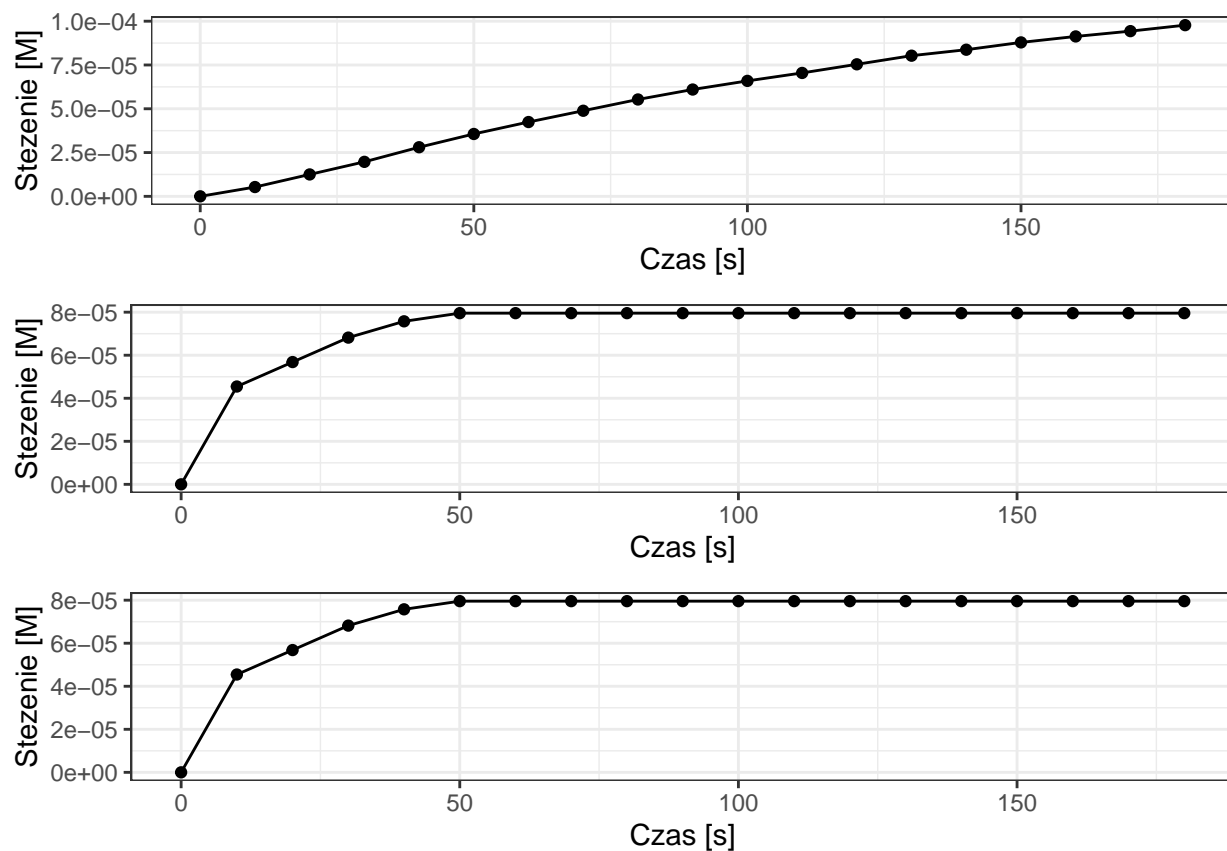
# wykres pokazujący zmiany stężenia utlenionego pirogalolu w czasie
stezenie_3a = data.frame(time, C_s3a)
vplot_c3a = ggplot(data = stezenie_3a, aes(time, C_s3a)) +
  geom_point() +
  geom_line() +
  xlab("Czas [s]") +
  ylab("Stężenie [M]") +
  theme_bw()

stezenie_3b = data.frame(time, C_s3b)
vplot_c3b = ggplot(data = stezenie_3b, aes(time, C_s3b)) +
  geom_point() +
  geom_line() +
  xlab("Czas [s]") +
  ylab("Stężenie [M]") +
  theme_bw()

stezenie_3c = data.frame(time, C_s3c)
vplot_c3c = ggplot(data = stezenie_3c, aes(time, C_s3c)) +
  geom_point() +
  geom_line() +
  xlab("Czas [s]") +
  ylab("Stężenie [M]") +
  theme_bw()

gridExtra::grid.arrange(vplot_c3a, vplot_c3b, vplot_c3c, nrow = 3)

```



3.3 Trzecie stężenie substratu - 250 ppm

```
# wartości zmierzonej absorbancji uszeregowane w czasie;
# miejsca dziesiętne oddzielone kropką!
abs_s4a = c(0, 0.03, 0.053,
            0.086, 0.116, 0.155,
            0.189, 0.22, 0.252,
            0.283, 0.308, 0.333,
            0.357, 0.379, 0.399,
            0.419, 0.437, 0.454,
            0.469)
abs_s4b = c(0, 0.12, 0.15,
            0.18, 0.20, 0.21,
            0.21, 0.21, 0.21,
            0.21, 0.21, 0.21,
            0.21, 0.21, 0.21,
            0.21, 0.21, 0.21,
            0.21)
abs_s4c = c(0, 0.12, 0.15,
            0.18, 0.20, 0.21,
            0.21, 0.21, 0.21,
            0.21, 0.21, 0.21,
            0.21, 0.21, 0.21,
            0.21, 0.21, 0.21,
```

```

0.21)
# Obliczamy stężenie produktu C = abs * mwe
C_s4a = abs_s4a / mwe
C_s4b = abs_s4b / mwe
C_s4c = abs_s4c / mwe
# Obliczamy szybkość początkową reakcji
V0_st4a = (C_s4a[6] - C_s4a[1]) / (time[6] - time[1])
V0_st4a

```

```
## [1] 1.174242e-06
```

```

V0_st4b = (C_s4b[6] - C_s4b[1]) / (time[6] - time[1])
V0_st4b

```

```
## [1] 1.590909e-06
```

```

V0_st4c = (C_s4c[6] - C_s4c[1]) / (time[6] - time[1])
V0_st4c

```

```
## [1] 1.590909e-06
```

```

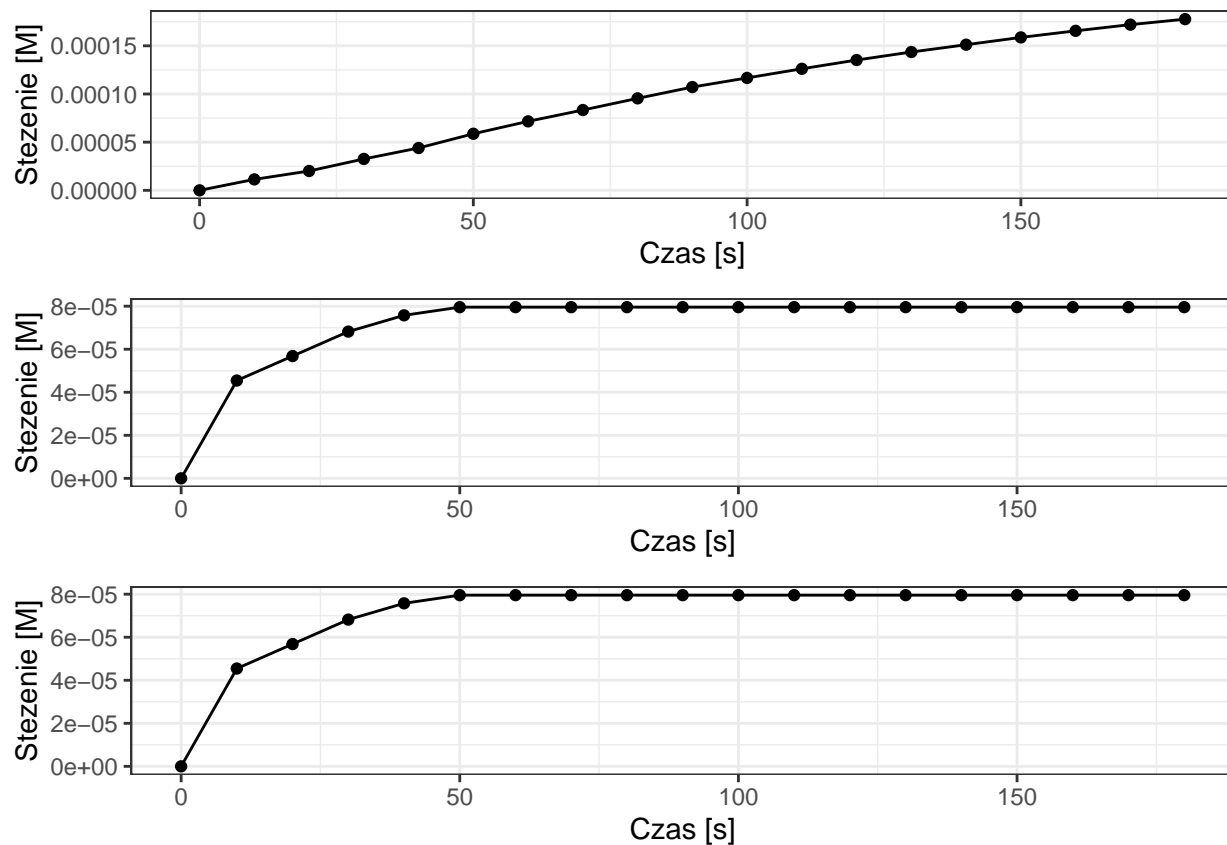
# wykres pokazujący zmiany stężenia utlenionego pirogalolu w czasie
stezenie_4a = data.frame(time, C_s4a)
vplot_c4a = ggplot(data = stezenie_4a, aes(time, C_s4a)) +
  geom_point() +
  geom_line() +
  xlab("Czas [s]") +
  ylab("Stężenie [M]") +
  theme_bw()

stezenie_4b = data.frame(time, C_s4b)
vplot_c4b = ggplot(data = stezenie_4b, aes(time, C_s4b)) +
  geom_point() +
  geom_line() +
  xlab("Czas [s]") +
  ylab("Stężenie [M]") +
  theme_bw()

stezenie_4c = data.frame(time, C_s4c)
vplot_c4c = ggplot(data = stezenie_4c, aes(time, C_s4c)) +
  geom_point() +
  geom_line() +
  xlab("Czas [s]") +
  ylab("Stężenie [M]") +
  theme_bw()

gridExtra::grid.arrange(vplot_c4a, vplot_c4b, vplot_c4c, nrow = 3)

```



3.4 Czwarte stężenie substratu - 500 ppm

```
# wartości zmierzonej absorbancji uszeregowane w czasie;
# miejsca dziesiętne oddzielone kropką!
abs_s5a = c(0, 0.018, 0.043,
            0.084, 0.135, 0.19,
            0.246, 0.298, 0.354,
            0.397, 0.444, 0.486,
            0.526, 0.564, 0.598,
            0.632, 0.66, 0.688,
            0.712)
abs_s5b = c(0, 0.12, 0.15,
            0.18, 0.20, 0.21,
            0.21, 0.21, 0.21,
            0.21, 0.21, 0.21,
            0.21, 0.21, 0.21,
            0.21, 0.21, 0.21,
            0.21)
abs_s5c = c(0, 0.12, 0.15,
            0.18, 0.20, 0.21,
            0.21, 0.21, 0.21,
            0.21, 0.21, 0.21,
            0.21, 0.21, 0.21,
            0.21, 0.21, 0.21,
```

```

0.21)
# Obliczamy stężenie produktu C = abs * mwe
C_s5a = abs_s5a / mwe
C_s5b = abs_s5b / mwe
C_s5c = abs_s5c / mwe
# Obliczamy szybkość początkową reakcji
V0_st5a = (C_s5a[6] - C_s5a[1]) / (time[6] - time[1])
V0_st5a

```

```
## [1] 1.439394e-06
```

```

V0_st5b = (C_s5b[6] - C_s5b[1]) / (time[6] - time[1])
V0_st5b

```

```
## [1] 1.590909e-06
```

```

V0_st5c = (C_s5c[6] - C_s5c[1]) / (time[6] - time[1])
V0_st5c

```

```
## [1] 1.590909e-06
```

```

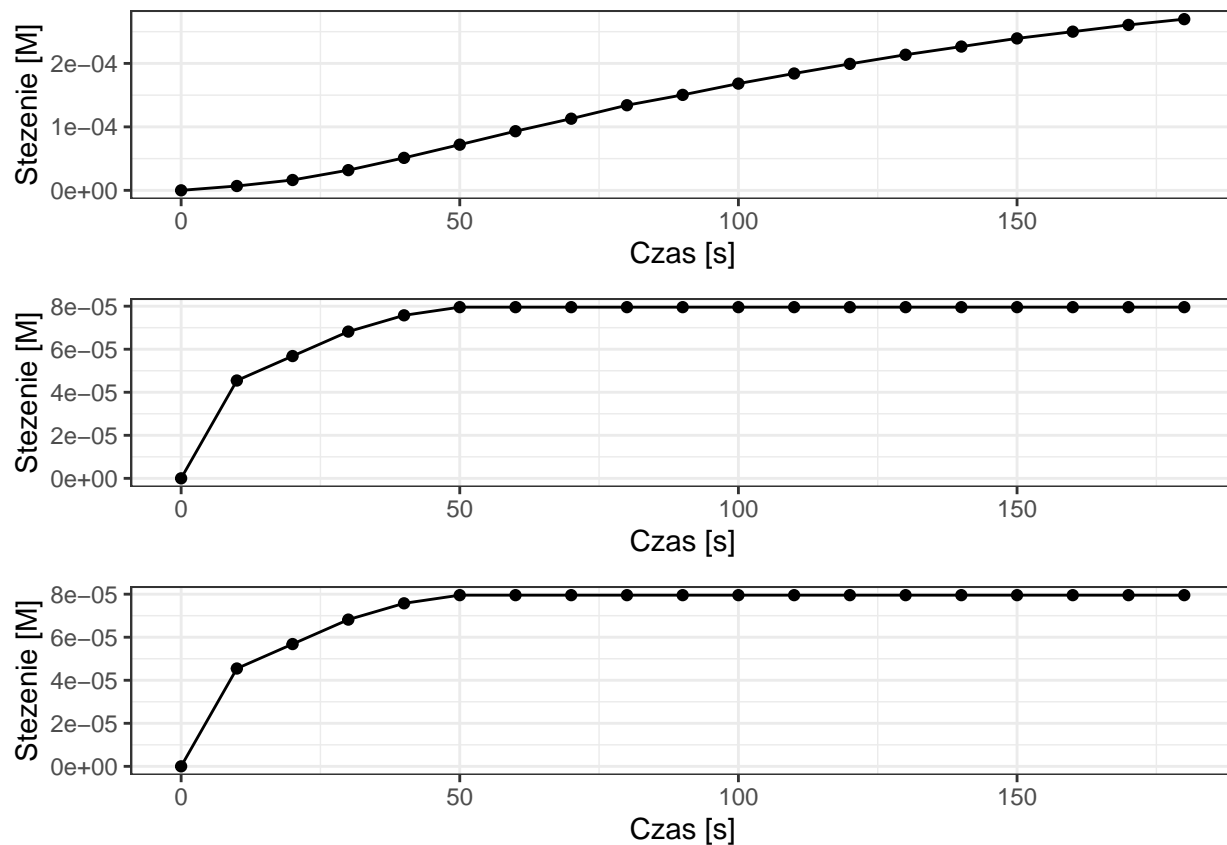
# wykres pokazujący zmiany stężenia utlenionego pirogalolu w czasie
stezenie_5a = data.frame(time, C_s5a)
vplot_c5a = ggplot(data = stezenie_5a, aes(time, C_s5a)) +
  geom_point() +
  geom_line() +
  xlab("Czas [s]") +
  ylab("Stężenie [M]") +
  theme_bw()

stezenie_5b = data.frame(time, C_s5b)
vplot_c5b = ggplot(data = stezenie_5b, aes(time, C_s5b)) +
  geom_point() +
  geom_line() +
  xlab("Czas [s]") +
  ylab("Stężenie [M]") +
  theme_bw()

stezenie_5c = data.frame(time, C_s5c)
vplot_c5c = ggplot(data = stezenie_5c, aes(time, C_s5c)) +
  geom_point() +
  geom_line() +
  xlab("Czas [s]") +
  ylab("Stężenie [M]") +
  theme_bw()

gridExtra::grid.arrange(vplot_c5a, vplot_c5b, vplot_c5c, nrow = 3)

```

4. Obliczanie K_m i V_{max} - wykres

4.1 Powtórzenie "a"

```
st = c(0.0029, 0.0054, 0.0112, 0.0224)
rev_st = 1/st
v0_a = c(V0_st2a, V0_st3a, V0_st4a, V0_st5a)
rev_v0_a = 1/v0_a
# rev_v0_a = c(1.48, 2.18, 2.53, 2.89, 3.48)
parameters_a = data.frame(rev_st, rev_v0_a)
LB_plot_a = ggplot(data = parameters_a, aes(rev_st, rev_v0_a)) +
  geom_point() +
  geom_smooth(method = "lm", formula = y~x) +
  xlab("Odwrotność stężenia substratu [1/C]") +
  ylab("1/V0") +
  #xlim(-3, 6) +
  #ylim(-2, 5) +
  theme_bw()
```

4.2 Powtórzenie "b"

```

#rev_v0_b = c(1.48, 2.18, 2.53, 2.89, 3.48)
v0_b = c(V0_st2b, V0_st3b, V0_st4b, V0_st5b)
rev_v0_b = 1/v0_b
parameters_b = data.frame(rev_st, rev_v0_b)
LB_plot_b = ggplot(data = parameters_b, aes(rev_st, rev_v0_b)) +
  geom_point() +
  geom_smooth(method = "lm", formula = y~x) +
  xlab("Odwrotność stężenia substratu [1/C]") +
  ylab("1/V0") +
  #xlim(-3, 6) +
  #ylim(-2, 5) +
  theme_bw()

```

4.3 Powtórzenie “c”

```

v0_c = c(V0_st2c, V0_st3c, V0_st4c, V0_st5c)
rev_v0_c = 1/v0_c
#rev_v0_c = c(1.48, 2.18, 2.53, 2.89, 3.48)
parameters_c = data.frame(rev_st, rev_v0_c)
LB_plot_c = ggplot(data = parameters_c, aes(rev_st, rev_v0_c)) +
  geom_point() +
  geom_smooth(method = "lm", formula = y~x) +
  xlab("Odwrotność stężenia substratu [1/C]") +
  ylab("1/V0") +
  #xlim(-3, 6) +
  #ylim(-2, 5) +
  theme_bw()

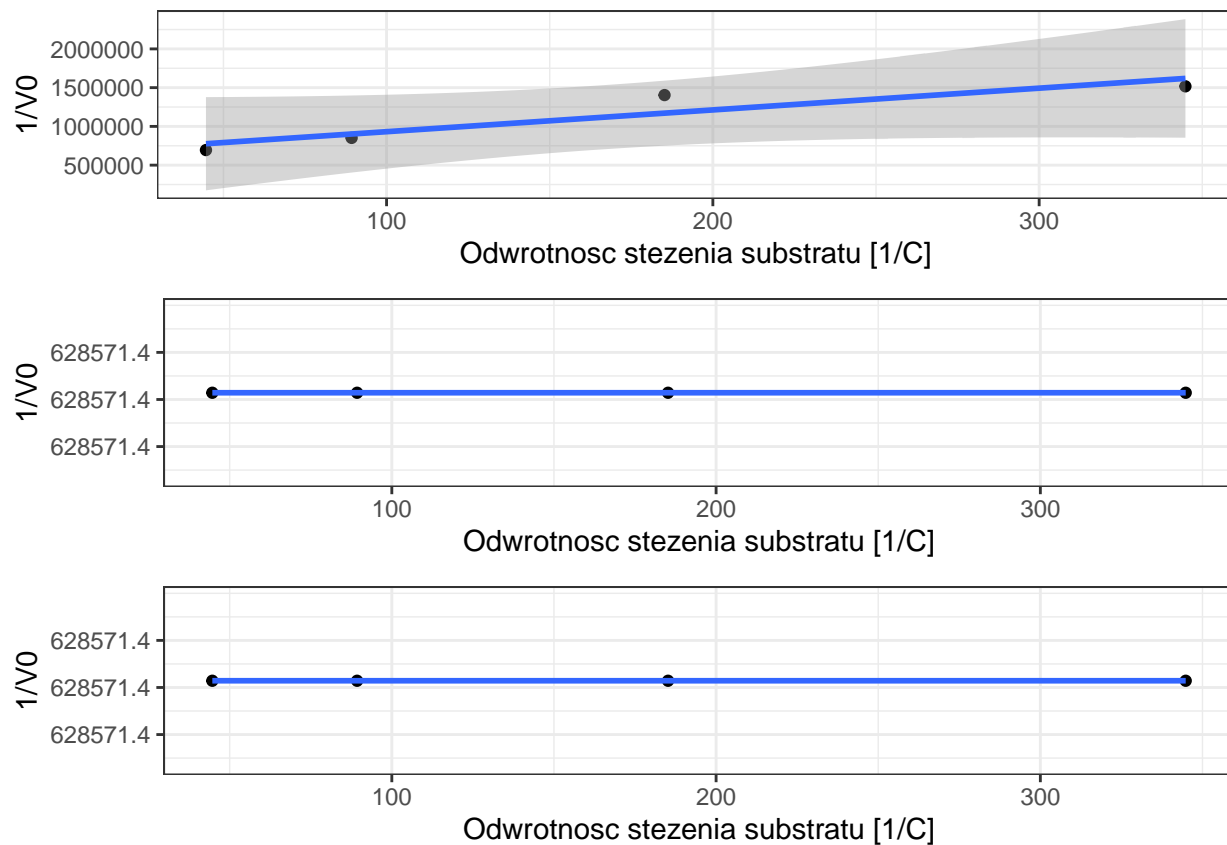
```

4.4 Wykresy Lineweavera–Burka

```

gridExtra::grid.arrange(LB_plot_a, LB_plot_b, LB_plot_c, nrow = 3)

```



5. Obliczanie K_m i V_{max} - obliczenia

Obliczamy parametry równania

$$y = ax + b$$

. Odwrotność szybkości maksymalnej enzymu odczytujemy z miejsca przecięcia prostej z osią Y , odwrotność stałej Michaelisa z miejsca przecięcia z osią X . Współczynnik 'a' odpowiada stosunkowi wartości

$$\frac{K_m}{V_{max}}$$

5.1 Równanie prostej dla powtórzenia "a"

```
model = lm(rev_v0_a ~ rev_st)
# Współczynnik 'a'
a_a = model$coefficients[2]
a_a
```

```
## rev_st
## 2807.569
```

```
# Współczynnik 'b'
b_a = model$coefficients[1]
b_a
```

```
## (Intercept)
##      650946.4
```

```
# Szybkość maksymalna wynosi:
```

```
Vmax_a = 1/b_a
Vmax_a
```

```
## (Intercept)
## 1.536225e-06
```

```
# Stała Michaelisa wynosi:
```

```
Km_a = a_a/b_a
Km_a
```

```
##      rev_st
## 0.004313057
```

5.2 Równanie prostej dla powtórzenia “b”

```
model = lm(rev_v0_b ~ rev_st)
# Współczynnik 'a'
a_b = model$coefficients[2]
a_b
```

```
## rev_st
##      0
```

```
# Współczynnik 'b'
```

```
b_b = model$coefficients[1]
b_b
```

```
## (Intercept)
##      628571.4
```

```
# Szybkość maksymalna wynosi:
```

```
Vmax_b = 1/b_b
Vmax_b
```

```
## (Intercept)
## 1.590909e-06
```

```
# Stała Michaelisa wynosi:
```

```
Km_b = a_b/b_b
Km_b
```

```
## rev_st
##      0
```

5.3 Równanie prostej dla powtórzenia “c”

```
model = lm(rev_v0_c ~ rev_st)
# Współczynnik 'a'
a_c = model$coefficients[2]
a_c
```

```
## rev_st
##      0
```

```
# Współczynnik 'b'
b_c = model$coefficients[1]
b_c
```

```
## (Intercept)
##    628571.4
```

```
# Szybkość maksymalna wynosi:
Vmax_c = 1/b_c
Vmax_c
```

```
## (Intercept)
## 1.590909e-06
```

```
# Stała Michaelisa wynosi:
Km_c = a_c/b_c
Km_c
```

```
## rev_st
##      0
```

6. Dyskusja