Mownit lab5

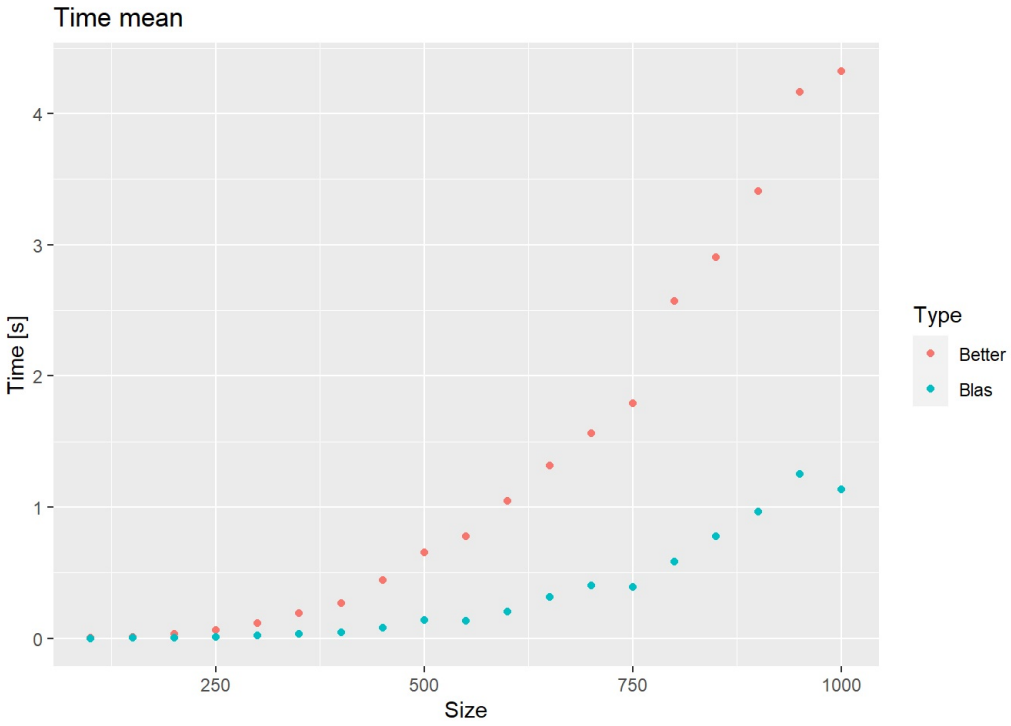
Zad 1

```
options(scipen = 999) # scientific notation off
require(tidyverse)
# C times "Size" , "better" , "blas"
results <- read_csv("c_times.csv")[c(1,3,4)]
results[c(2,3)] <- results[c(2,3)] / 1000.0 # [ms] to [s]
results
```

```
## # A tibble: 190 x 3
##     Size  Better     Blas
##    <dbl>   <dbl>    <dbl>
##  1   100 0.00354 0.00181
##  2   100 0.00381 0.000567
##  3   100 0.00499 0.000601
##  4   100 0.00368 0.00053
##  5   100 0.00487 0.000785
##  6   100 0.00348 0.000633
##  7   100 0.00510 0.000529
##  8   100 0.00360 0.000527
##  9   100 0.00359 0.000545
## 10   100 0.00368 0.00229
## # ... with 180 more rows
```

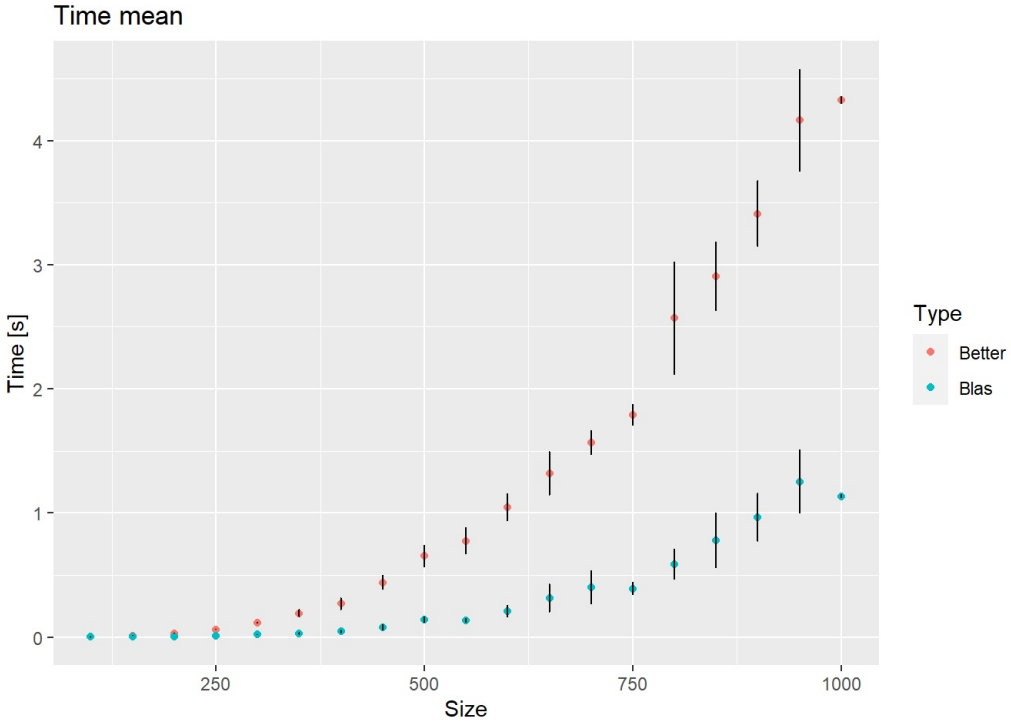
Zad 2

```
avg_dplyr <- results %>%
  group_by(Size) %>%
  summarise(
    better_mean = mean(Better),
    better_sd = sd(Better),
    blas_mean = mean(Blas),
    blas_sd = sd(Blas)
  )  %>%
  ungroup() %>%
  arrange(Size)


mean_plot <- ggplot(avg_dplyr, aes(x=Size)) +
  geom_point(aes(y=better_mean ,col="Better")) +
  geom_point(aes(y=blas_mean, col="Blas")) +
  ylab("Time [s]")+
  ggtitle("Time mean")+
  labs(color = "Type")

mean_plot
```



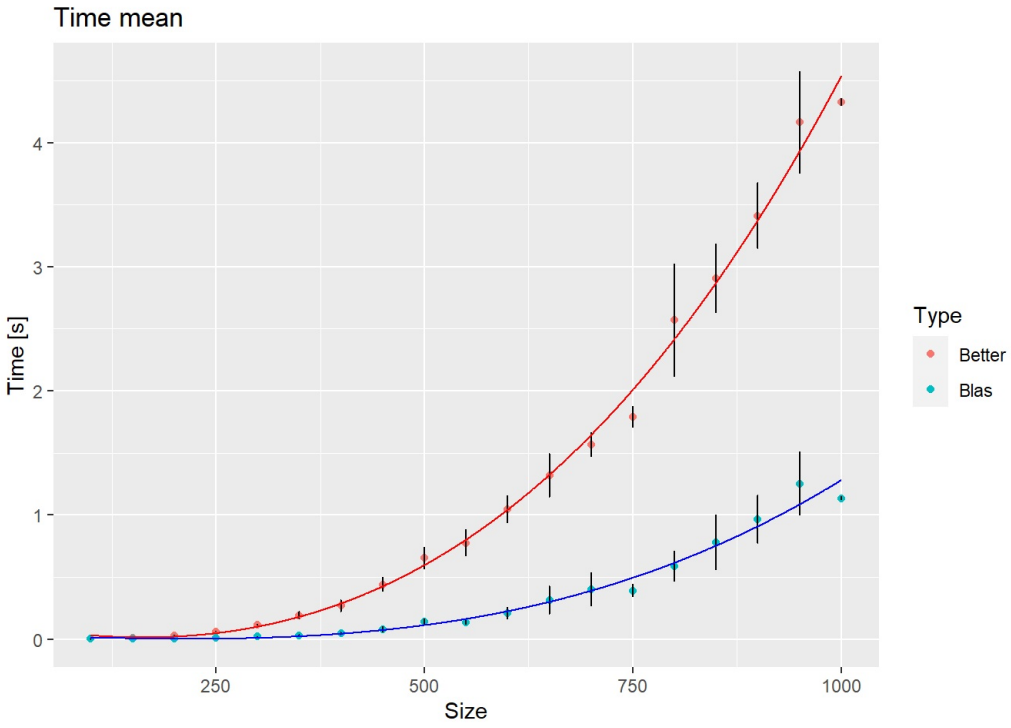
Zad 3

```
mean_sd_plot <- mean_plot +
  geom_errorbar(aes(ymin = better_mean-better_sd, ymax = better_mean+better_sd, width=.25))+
  geom_errorbar(aes(ymin = blas_mean-blas_sd, ymax = blas_mean+blas_sd, width=.25))

mean_sd_plot
```



Zad 4

```
# Fit poly
better_fit <- lm(better_mean ~ poly(Size, 3, raw=TRUE), data=avg_dplyr)
blas_fit <- lm(blas_mean ~ poly(Size, 3, raw=TRUE), data=avg_dplyr)

better_approx = data.frame(Size = seq(100, 1000, 0.01))
better_approx$better_mean = predict(better_fit, better_approx)

blas_approx = data.frame(Size = seq(100, 1000, 0.01))
blas_approx$blas_mean = predict(blas_fit, blas_approx)

#Plot
times_poly_fitted <- mean_sd_plot+
  geom_line(data=better_approx, aes(Size,better_mean), color="red")+
  geom_line(data=blas_approx, aes(Size,blas_mean), color="blue")

times_poly_fitted
```



Zad 5

```
coronavirus_data <- read_csv("coronavirus.csv")
coronavirus_data <- filter(coronavirus_data,location=="Poland")
coronavirus_data <- select(coronavirus_data, date, total_cases)

coronavirus_data
```

```
## # A tibble: 405 x 2
##    date       total_cases
##    <date>           <dbl>
##  1 2020-03-04           1
##  2 2020-03-05           1
##  3 2020-03-06           5
##  4 2020-03-07           5
##  5 2020-03-08          11
##  6 2020-03-09          16
##  7 2020-03-10          22
##  8 2020-03-11          31
##  9 2020-03-12          49
## 10 2020-03-13          68
## # ... with 395 more rows
```

First range

```
require(lubridate)
first_range <- filter(coronavirus_data,
                      date >= ymd("2020-08-01"),
                      date < ymd("2020-11-01"))
first_range
```

```
## # A tibble: 92 x 2
##    date       total_cases
##    <date>           <dbl>
##  1 2020-08-01       46346
##  2 2020-08-02       46894
##  3 2020-08-03       47469
##  4 2020-08-04       48149
##  5 2020-08-05       48789
##  6 2020-08-06       49515
##  7 2020-08-07       50324
##  8 2020-08-08       51167
##  9 2020-08-09       51791
## 10 2020-08-10       52410
## # ... with 82 more rows
```

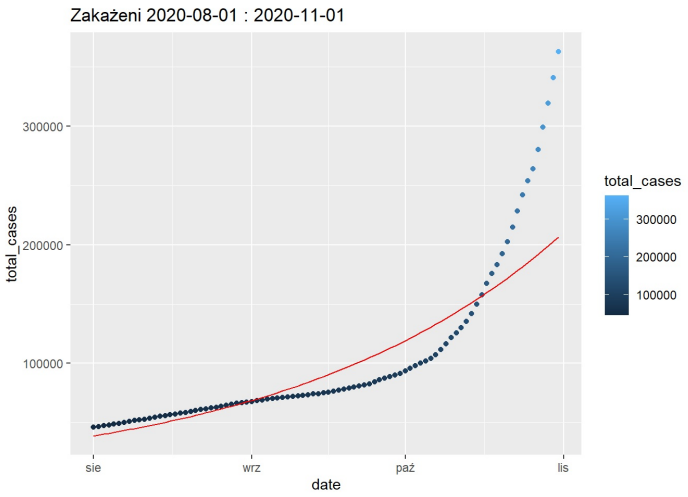
Second range

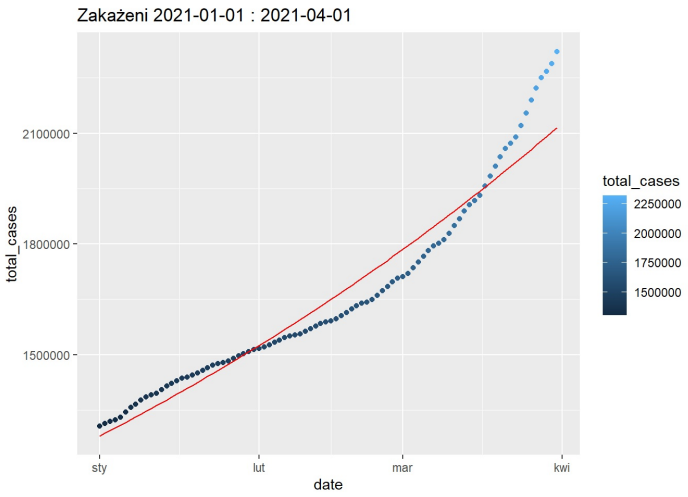
```
second_range <- filter(coronavirus_data,
                       date >= ymd("2021-01-01"),
                       date < ymd("2021-04-01"))
second_range
```

```
## # A tibble: 90 x 2
##    date       total_cases
##    <date>           <dbl>
##  1 2021-01-01     1305774
##  2 2021-01-02     1312780
##  3 2021-01-03     1318562
##  4 2021-01-04     1322947
##  5 2021-01-05     1330543
##  6 2021-01-06     1344763
##  7 2021-01-07     1356882
##  8 2021-01-08     1365645
##  9 2021-01-09     1376389
## 10 2021-01-10     1385522
## # ... with 80 more rows
```

Fitting

```
first_fit <- lm(log(total_cases) ~ date, data=first_range)
second_fit <- lm(log(total_cases) ~ date, data=second_range)


first_approx = data.frame(date=seq(min(first_range$date),max(first_range$date),"day"))
first_approx$total_cases <- exp(predict(first_fit, first_approx))

second_approx = data.frame(date=seq(min(second_range$date),max(second_range$date),"day"))
second_approx$total_cases <- exp(predict(second_fit, second_approx))

# First plot
p1 <- ggplot(first_range, aes(date,total_cases, color=total_cases))+
  geom_point()+
  geom_line(data=first_approx, aes(date,total_cases), color="red")+
  ggtitle("Zakażeni 2020-08-01 : 2020-11-01")

# Second plot
p2 <- ggplot(second_range, aes(date,total_cases, color=total_cases))+
  geom_point()+
  geom_line(data=second_approx, aes(date,total_cases), color="red")+
  ggtitle("Zakażeni 2021-01-01 : 2021-04-01")

require(gridExtra)
grid.arrange(p1, p2, ncol=2)
```


Zakażeni 2020-08-01 : 2020-11-01
300000
200000
100000
total_cases
sie
wrz
paź
lis
date
total_cases
300000
200000
100000


Zakażeni 2021-01-01 : 2021-04-01
2100000
1800000
1500000
total_cases
sty
lut
mar
kwi
date
total_cases
2250000
2000000
1750000
1500000