View counts of '100 Days of Code - Learn Python' course

Data comes from Replit's YouTube playlist 100 Days of Code - Learn Python. View counts were collected using `yt-dlp` tool:

```
yt-dlp --print "\"%(title)s\",\"%(view_count)n\"" --skip-download \
https://www.youtube.com/playlist?list=PLto9KpJAqHMQNY3XP0JqLs7NyeU_dnNj0 > view_counts.csv
```

Data cleaning

The playlist has 182 videos, for some days there are separate videos with solution which are viewed less. I omit these videos, leaving only the main ones with most views on a given day.

```
df <- read.csv("view_counts.csv")
colnames(df) <- c("title", "view_count")

df <- df %>% mutate(title_short = str_extract(df$title, "^[\\w]+\\s[\\d]+"))
df <- df %>% filter(str_extract(df$title_short, "[\\d]+") %>% str_sub(1, 1) != "0")
df <- df %>% mutate(day = str_extract(df$title_short, "[\\d]+") %>% as.integer())
df <- df %>% group_by(day) %>% summarize(max_count = max(view_count))
```

Fitting of power law, exponential and log-normal distibutions

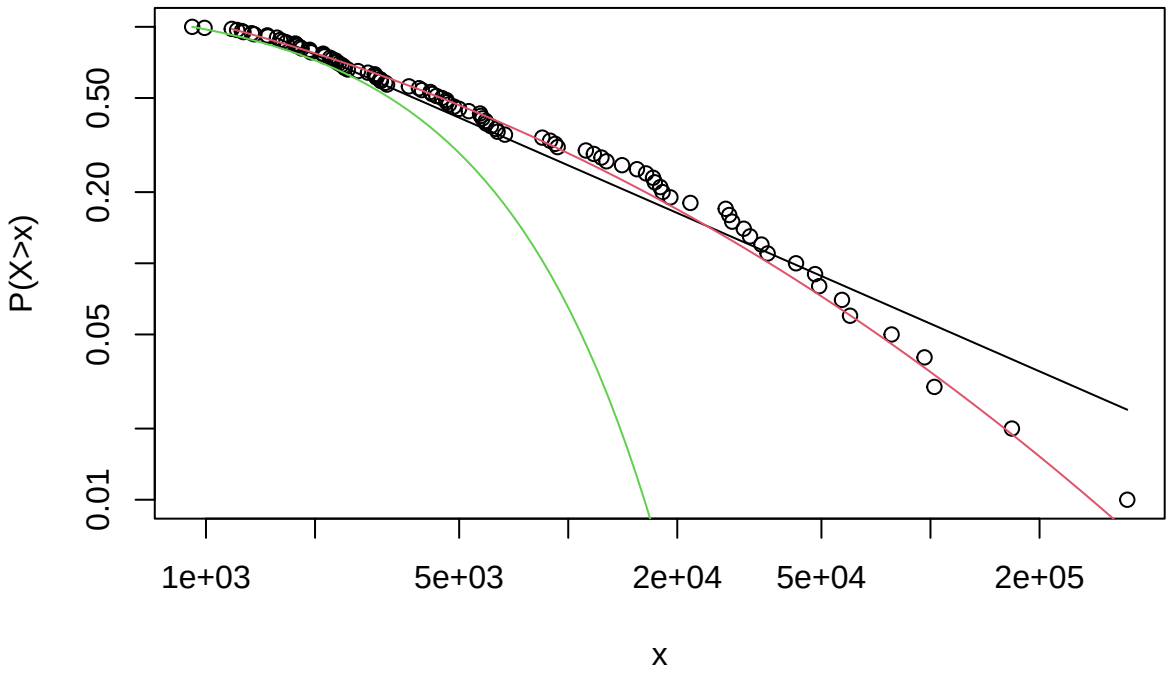
First x_{min} has to be estimated for each distribution, it is assumed that the distribution doesn't hold for smaller values (here: view counts) and the value of x_{min} is chosen so that the maximum difference between observed and expected values is the smallest.

```
pl_fit <- displ$new(df$max_count)
max_value <- max(df$max_count)
pl_xmin <- estimate_xmin(pl_fit, xmax = max_value)
pl_fit$setXmin(pl_xmin)
pl_pars <- estimate_pars(pl_fit)
pl_fit$setPars(pl_pars)

exp_fit <- disexp$new(df$max_count)
exp_xmin <- estimate_xmin(exp_fit, xmax = max_value)
exp_fit$setXmin(exp_xmin)
exp_pars <- estimate_pars(exp_fit)
exp_fit$setPars(exp_pars)

lnorm_fit <- dislnorm$new(df$max_count)
lnorm_xmin <- estimate_xmin(lnorm_fit, xmax = max_value)
lnorm_fit$setXmin(lnorm_xmin)
lnorm_pars <- estimate_pars(lnorm_fit)
lnorm_fit$setPars(lnorm_pars)
```

Plots of the fitted distributions

Black - power law, green - exponential, red - log-normal. Y axis describes complementary cumulative distribution function, $CCDF = 1 - CDF$.

```
plot(lnorm_fit, ylab="P(X>x)")
lines(pl_fit, col=1)
lines(lnorm_fit, col=2)
lines(exp_fit, col=3)
```



Testing if observed data is generated from a given distribution

Alternative hypothesis is that it's not, a p-value > 0.05 means that the data set could be generated with the tested distribution, although it could also be generated with a different one.

```
pl_p = bootstrap_p(pl_fit, threads = 8, xmax = max_value)
## Expected total run time for 100 sims, using 8 threads is 43.1 seconds.
exp_p <- bootstrap_p(exp_fit, threads = 8, xmax = max_value)
## Expected total run time for 100 sims, using 8 threads is 24.8 seconds.
lnorm_p <- bootstrap_p(lnorm_fit, threads = 8, xmax = max_value)
## Expected total run time for 100 sims, using 8 threads is 52.5 seconds.

sprintf("p-value for power law: %f", pl_p$p)
## [1] "p-value for power law: 0.600000"
sprintf("p-value for exponential: %f", exp_p$p)
## [1] "p-value for exponential: 0.000000"
sprintf("p-value for log-normal: %f", lnorm_p$p)
## [1] "p-value for log-normal: 0.670000"
```

Comparing distributions

Both the power law and the log-normal distributions are plausible. To test which one is a better fit, both must use the same x_min value. A p-value > 0.05 means that it cannot be said that one of the distributions fits better.

```
lnorm_fit$setXmin(1570)
lnorm_pars <- estimate_pars(lnorm_fit)
lnorm_fit$setPars(lnorm_pars)
```

```
comp <-compare_distributions(lnorm_fit, pl_fit)
print(comp$p_two_sided)
```

```
## [1] 0.2346014
```