R Notebook

Cited by other indicates those cited in a protest, by an attorney or agent not acting in a representative capacity but on behalf of a single inventor, and by the applicant

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(rmongodb)
library(poweRlaw)
```

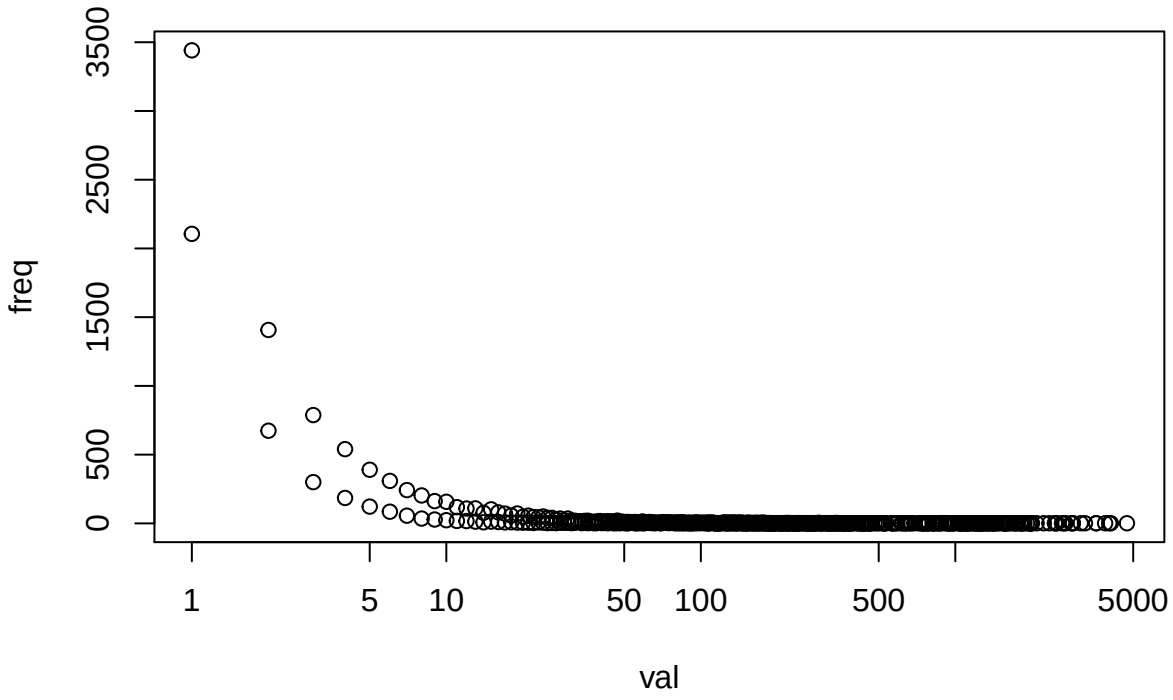
```
## Warning: package 'poweRlaw' was built under R version 3.2.5
```

```
if (file.exists("Dat/orderFrequencies.rds")) {
    orderFrequencies <- readRDS("Dat/orderFrequencies.rds")
} else {
    mongo <- mongo.create()
    mongo.is.connected(mongo)
    db <- "sotonproject"
    orderFrequencies <- mongo.find.all(mongo, "sotonproject.orderFrequencies")
    mongo.destroy(mongo)
    orderFrequencies <- lapply(orderFrequencies, unlist)
    orderFrequencies <- as.data.frame(orderFrequencies)
    orderFrequencies <- t(orderFrequencies)
    rownames(orderFrequencies) <- NULL
    colnames(orderFrequencies) <- c("Year", "Order", "Examiner", "Other", "Total", "Total2")
    orderFrequencies <- as.data.frame(apply(orderFrequencies, 2, as.numeric))
}
```

There are some differences between the order calculated through the citations using map reduce and the order calculated while processing the data. These differences are by and large negligible by there are rare large scale differences spread throughout the data. The larger errors are caused by a fault while uploading the data to mongodb, causing duplicate uploads of a section of 2013 and will be resolved soon.

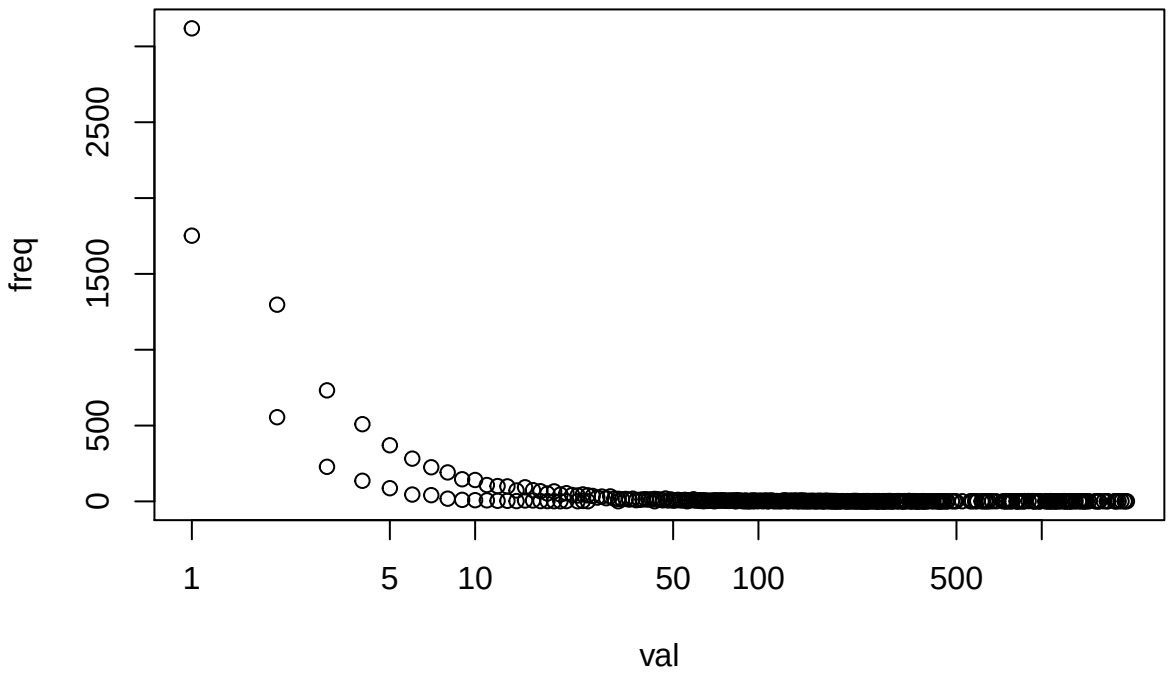
```
orderFrequencies$processingDifferences <- orderFrequencies$Total - orderFrequencies$Total2
dat <- orderFrequencies %>% filter(Year != 2019) %>% select(processingDifferences) %>% table %>% as.data
names(dat) <- c("val", "freq")
dat$val <- abs(as.numeric(as.character(dat$val)))
plot(dat, log = "x")
```

```
## Warning in xy.coords(x, y, xlabel, ylabel, log): 1 x value <= 0 omitted
## from logarithmic plot
```




```
dat <- orderFrequencies %>% filter(Year != 2013) %>% select(processingDifferences) %>% table %>% as.data
names(dat) <- c("val", "freq")
dat$val <- abs(as.numeric(as.character(dat$val)))
plot(dat, log = "x")
```

```
## Warning in xy.coords(x, y, xlabel, ylabel, log): 1 x value <= 0 omitted
## from logarithmic plot
```




```
summary(orderFrequencies %>% filter(Year != 2019) %>% select(processingDifferences))
```

```
##  processingDifferences
##  Min.   :-3232
##  1st Qu.:    0
##  Median :    1
##  Mean   :    0
##  3rd Qu.:    3
##  Max.   : 4725
```

```
lm_func <- function(x = NA, y = "count", var = "Total2", year = 2002, data = orderFrequencies, xmin = 2
    dat <- data %>% filter(Year == year, Order > xmin, Order < xmax)
    dat$Order[dat$Order == 0] <- NA
    dat[,var][dat[,var] == 0] <- NA
    model <- switch(y,
                    "count" = lm(data = dat, formula = log10(get(var)) ~ log10(Order)),
                    "freq" = lm(data = dat, formula = log10(freq) ~ log10(Order)))
    list(func = model$coefficients[2] * log10(x) + model$coefficients[1], coef = model$coefficients)
}
lm_func()
```

```
## $func
## log10(Order)
##           NA
##
## $coef
##  (Intercept) log10(Order)
```

```
##     6.925476    -2.677550
```

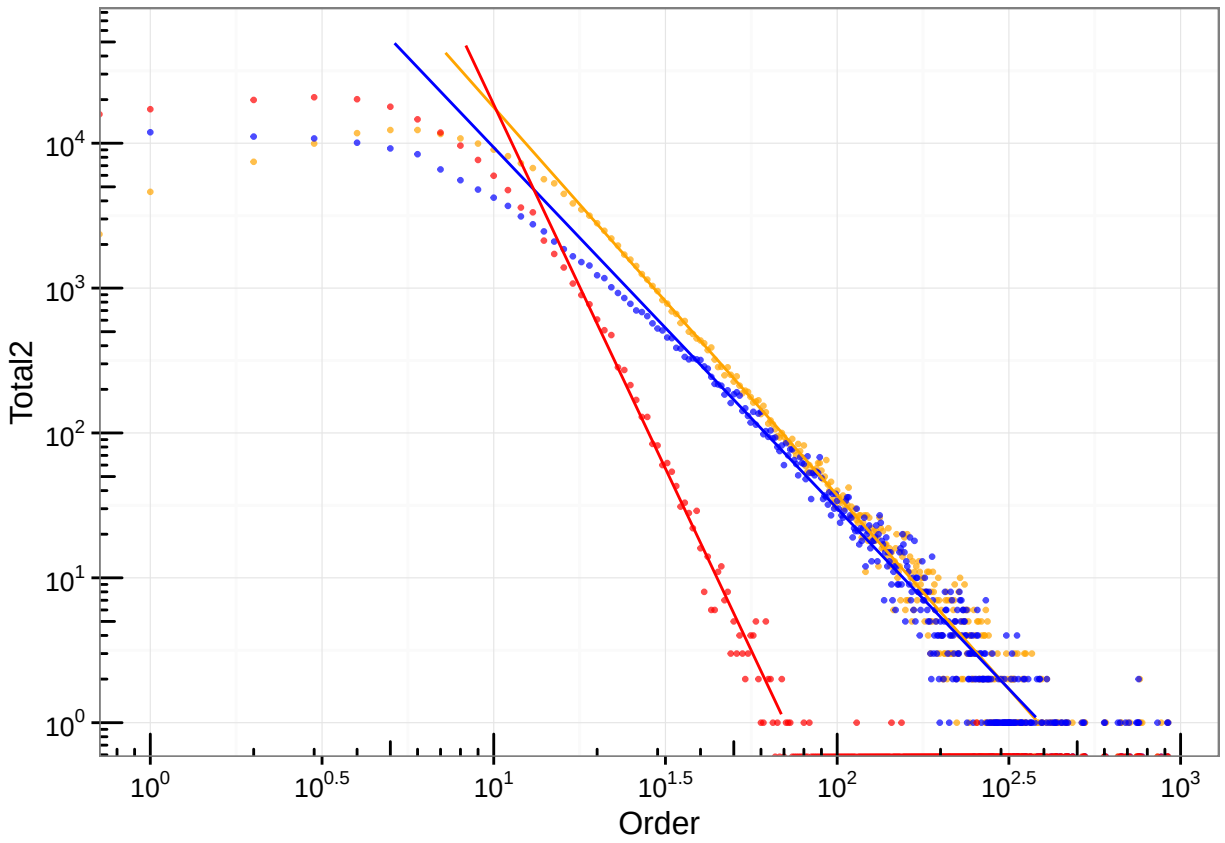
```
dat <- orderFrequencies %>% filter(Year == 2002)
ggplot(data = dat, aes(x = Order, y = Total2, colour = as.factor(Year))) +
    geom_point(size = 0.5, alpha = .7, colour = "orange") +
    geom_point(size = 0.5, aes(y = Other), colour = "blue", alpha = .7) +
    geom_point(size = 0.5, aes(y = Examiner), colour = "red", alpha = .7) +
    scale_x_log10(
        breaks = scales::trans_breaks("log10", function(x) 10^x),
        labels = scales::trans_format("log10", scales::math_format(10^.x))
    ) +
    scale_y_log10(
        breaks = scales::trans_breaks("log10", function(x) 10^x),
        labels = scales::trans_format("log10", scales::math_format(10^.x)),
        limits = c(1, 5e4)
    ) +
    stat_function(fun = function(x) lm_func(x)$func,
                  geom = 'line', colour = "orange") +
    stat_function(fun = function(x) lm_func(x, var = "Other")$func,
                  geom = 'line', colour = "blue") +
    stat_function(fun = function(x) lm_func(x, var = "Examiner", xmax = 90)$func,
                  geom = 'line', colour = "red") +
    theme_bw() +
    annotation_logticks()
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```

```
## Warning: Removed 42 rows containing missing values (geom_path).
```

```
## Warning: Removed 37 rows containing missing values (geom_path).
```

```
## Warning: Removed 69 rows containing missing values (geom_path).
```



We can see that the examiner in red has a smaller range of values (steeper gradient) and therefore the overall distribution tends to that of the "other". For low orders however the examiner has higher averages.

Investigating Changing Power Law

```
fit_pl <- function(year, var) {
    time <- Sys.time()
    print(paste("Processing", year))

    # Libraries
    library(poweRlaw)

    ## INITIALISE DATA ############################
    # Load data
    dat <- readRDS("Dat/orderFrequencies.rds")
    # Filter year
    ddyear <- dat %>% filter(Year == year) %>% group_by() %>% select(Order, get(var))
    # Produce tall vector (as poweRlaw interacts with)
    vec <- NULL
    for(row in seq_len(nrow(ddyear))) {
        vec <- c(vec, rep(as.numeric(ddyear[row,"Order"]), ddyear[row,var]))
    }
    # Remove zeros because log scale can't handle it.
    vec <- vec[vec != 0]
```

```
    ## Fit power law ########################
    m_pl <- displ$new(vec)
    est = estimate_pars(m_pl)
    est_pl <- estimate_xmin(m_pl)
    m_pl$setXmin(est_pl)

    print(Sys.time() - time)
    return(m_pl)
}
fit_pl(year = 2002, var = "Total2")
```

```
## [1] "Processing 2002"
## Time difference of 34.44727 secs
```

```
## Reference class object of class "displ"
```

```
## Field "xmin":
```

```
## [1] 19
```

```
## Field "pars":
```

```
## [1] 2.74325
```

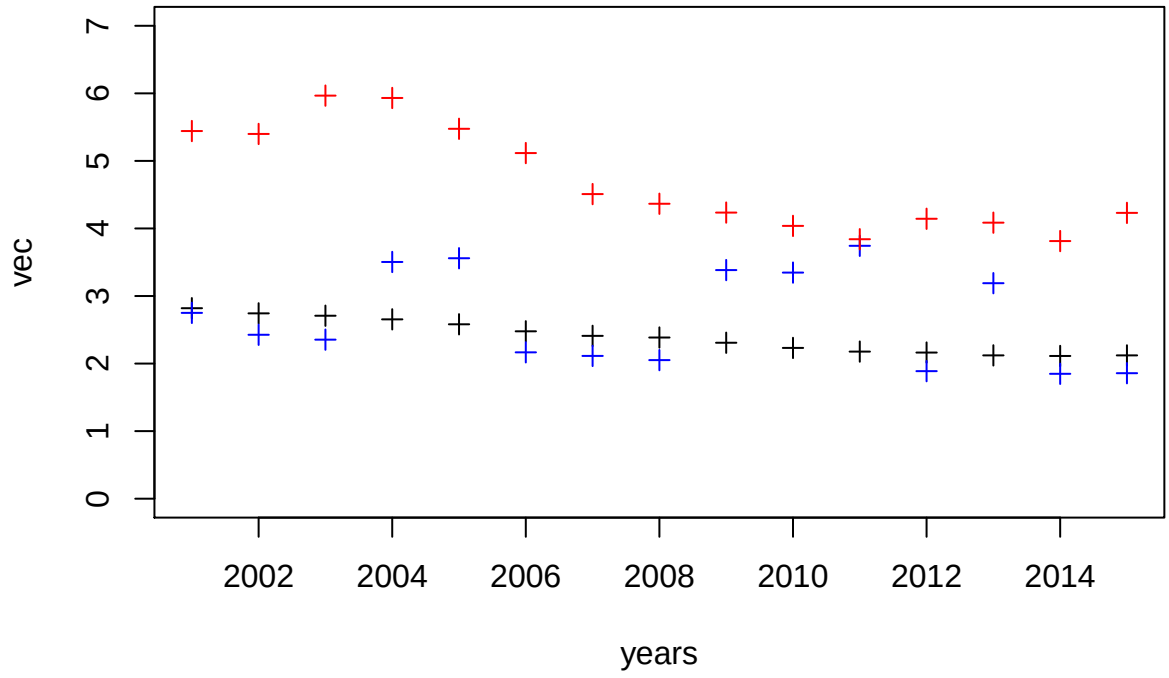
```
## Field "no_pars":
```

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

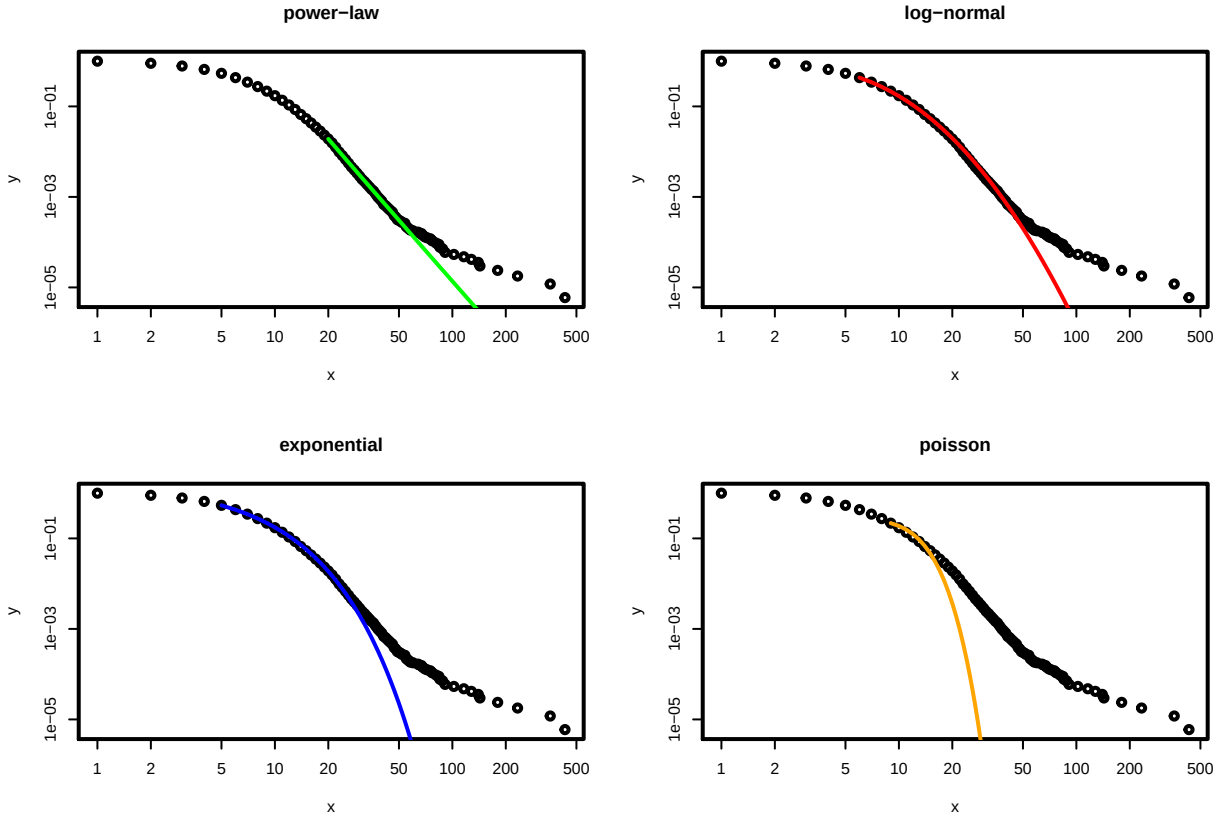
```
years <- 2001:2015
if (file.exists("Dat/power_law_fits_Orders.rdata")) {
    load("Dat/power_law_fits_Orders.rdata")
} else {
    pl <- list()
    for (yr in years) {
        pl[[yr]] <- fit_pl(yr, "Total2")
    }

    plExaminer <- list()
    years <- 2001:2015
    for (yr in years) {
        plExaminer[[yr]] <- fit_pl(yr, "Examiner")
    }

    plOther <- list()
    years <- 2001:2015
    for (yr in years) {
        plOther[[yr]] <- fit_pl(yr, "Other")
    }
    save(pl, plExaminer, plOther, file = "Dat/power_law_fits_Orders.rdata")
}
```

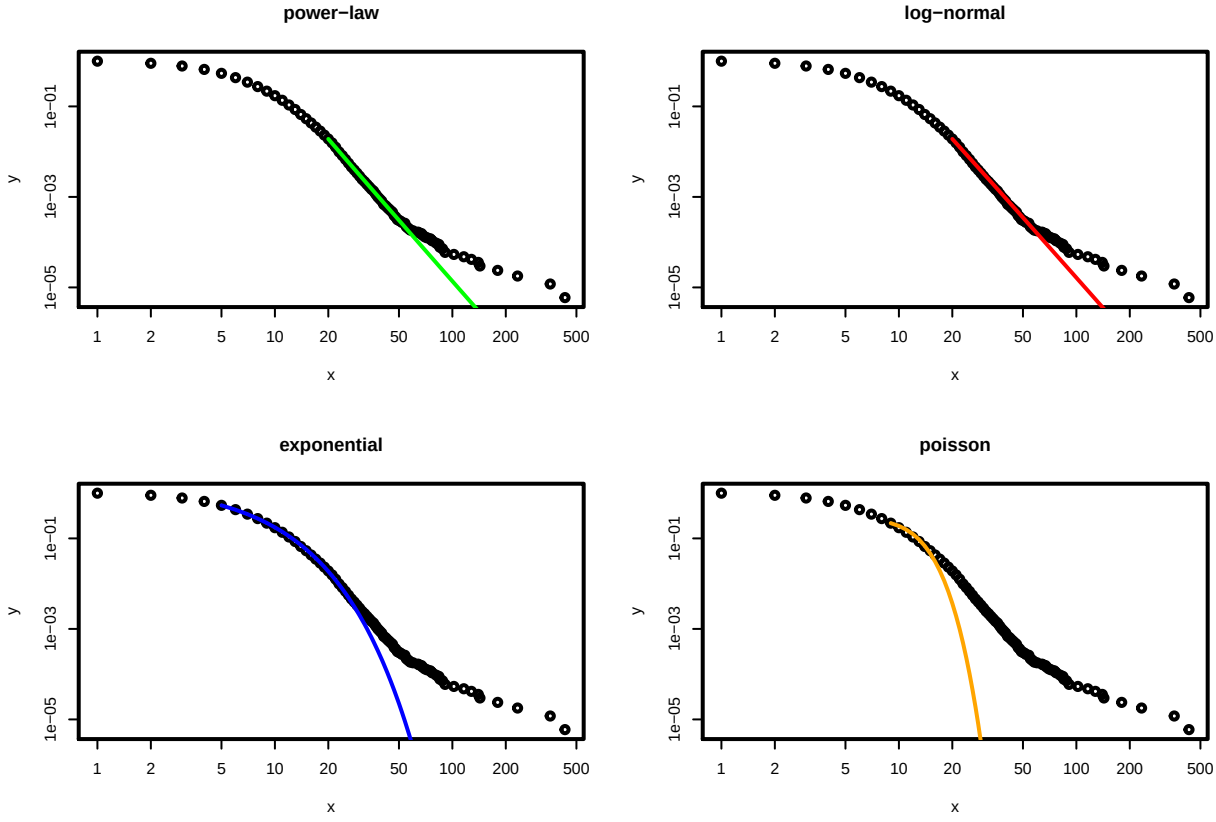
```
vec <- NULL
for (yr in years) {
    vec <- c(vec, pl[[yr]]$getPars())
}
vecExaminer <- NULL
for (yr in years) {
    vecExaminer <- c(vecExaminer, plExaminer[[yr]]$getPars())
```

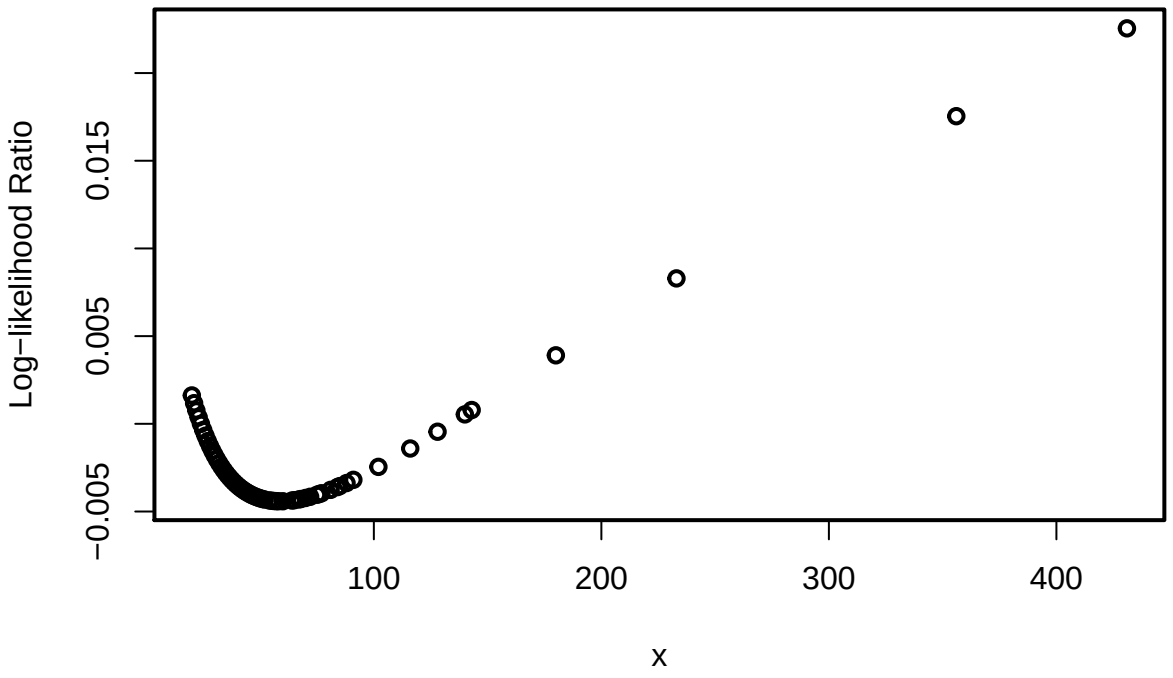
```
}
vecOther <- NULL
for (yr in years) {
    vecOther <- c(vecOther, plOther[[yr]]$getPars())
}


plot(years, vec, ylim = c(0,7), pch = 3)
points(years, vecOther, col = "blue", pch = 3)
points(years, vecExaminer, col = "red", pch = 3)
```



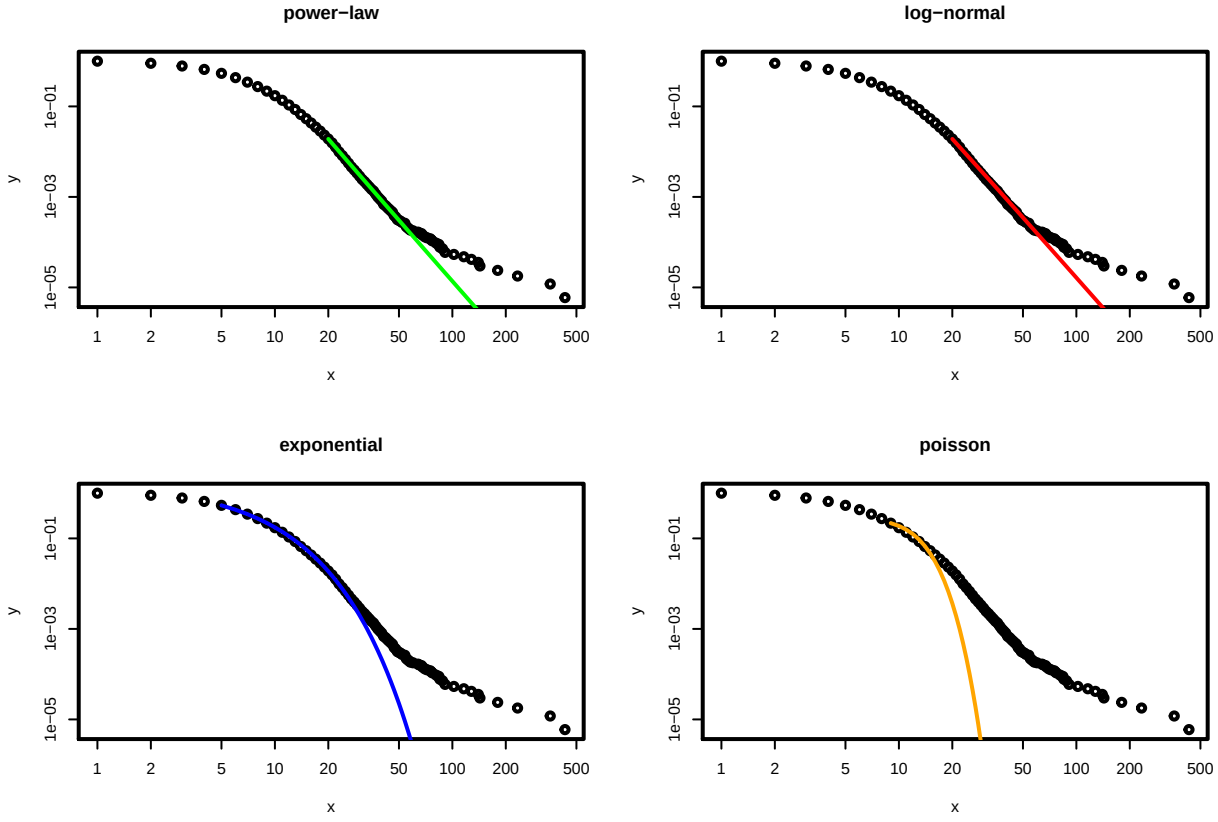

```
source("order_fit_distributions.R")
ExaminerDistributions <- fit_distributions(2001, out_lab = "test", degree_distribution_all = orderFrequ
```


power−law
log−normal
exponential
poisson
y
x
1e−01
1e−03
1e−05
1
2
5
10
20
50
100
200
500


power−law
log−normal
exponential
poisson
y
x
1e−01
1e−03
1e−05
1
2
5
10
20
50
100
200
500




```
par(mfrow = c(2,2))
    par(cex = .5)
    par(lwd = 2)
    plot(ExaminerDistributions$distributions[[3]], main = "power-law");
    lines(ExaminerDistributions$distributions[[3]], col="green")
    plot(ExaminerDistributions$distributions[[2]], main = "log-normal");
    lines(ExaminerDistributions$distributions[[2]], col = "red")
    plot(ExaminerDistributions$distributions[[1]], main = "exponential");
    lines(ExaminerDistributions$distributions[[1]], col="blue")
    plot(ExaminerDistributions$distributions[[4]], main = "poisson");
    lines(ExaminerDistributions$distributions[[4]], col="orange")
```



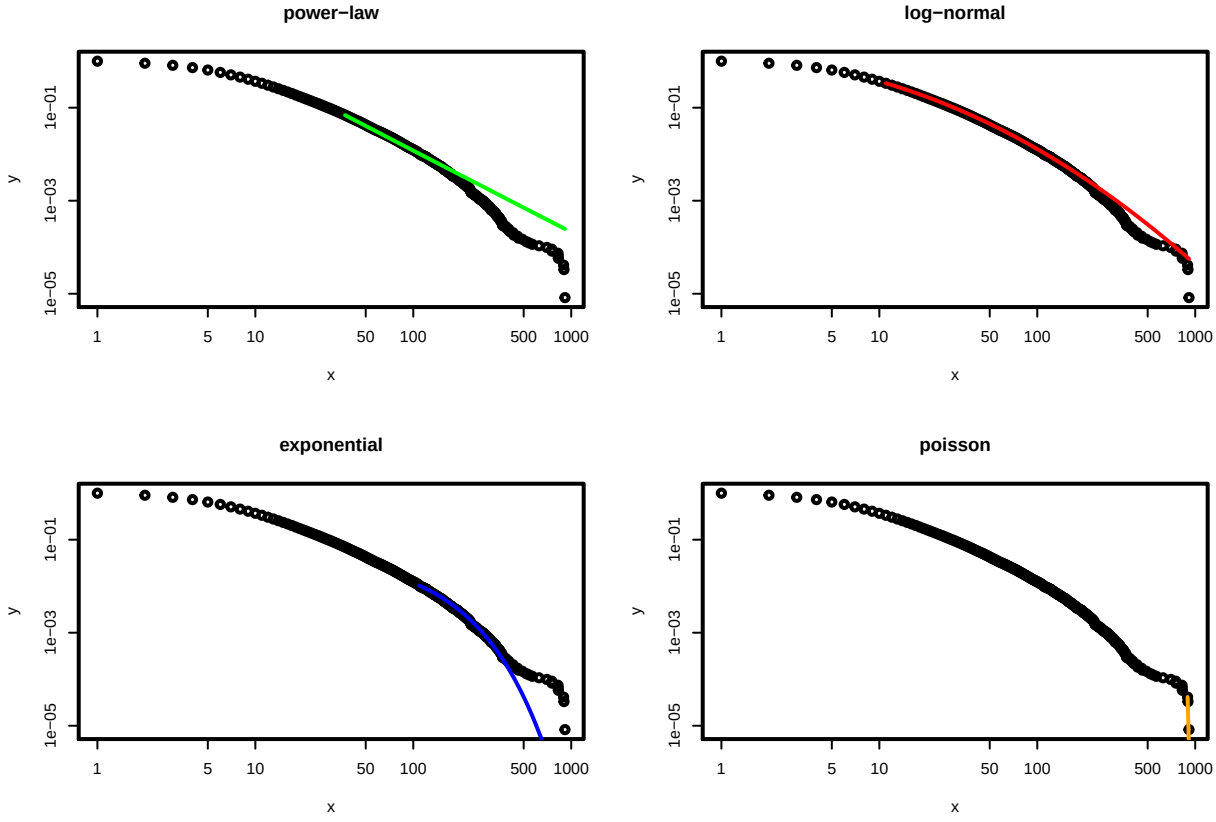

```
ExaminerDistributions$comparisons[[1]][1:3] # upper limit on probability given powerlaw is true
```

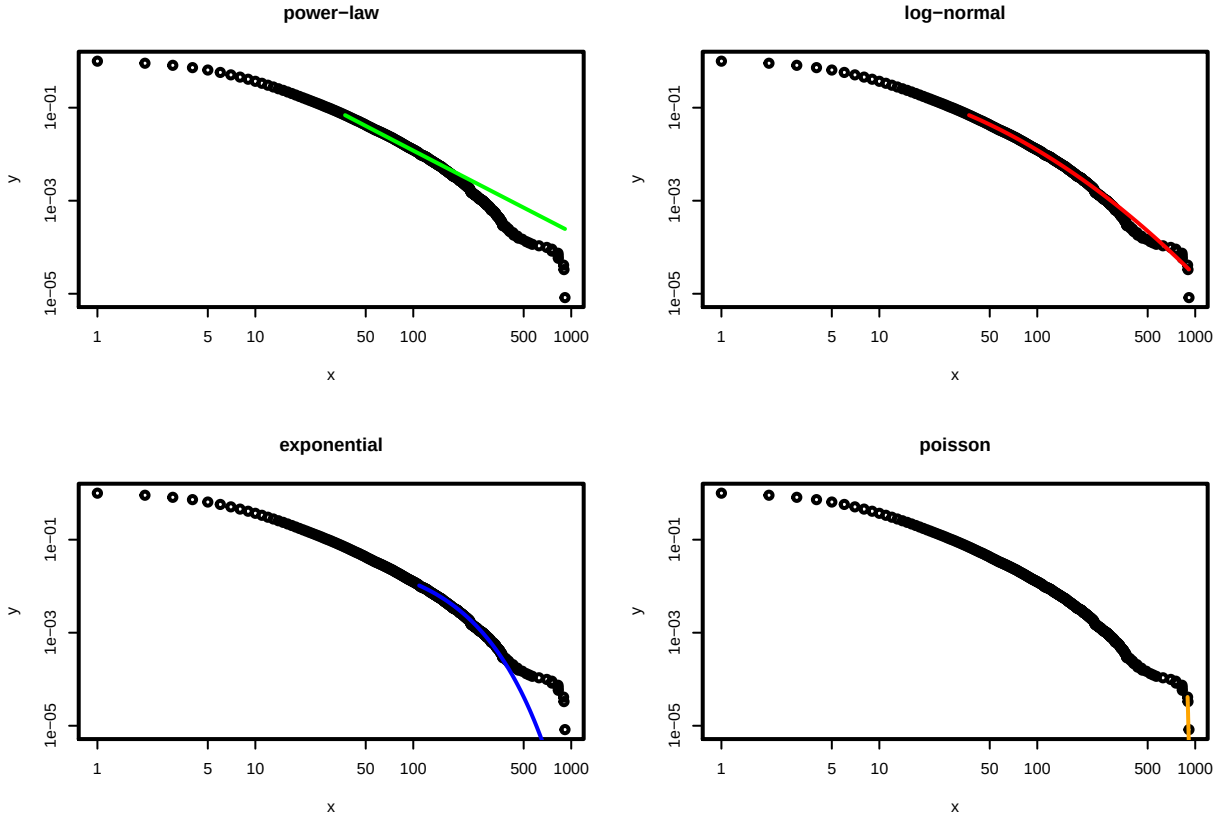
```
## $test_statistic
## [1] 1.766592
##
## $p_one_sided
## [1] 0.9613517
##
## $p_two_sided
## [1] 0.0772965
```

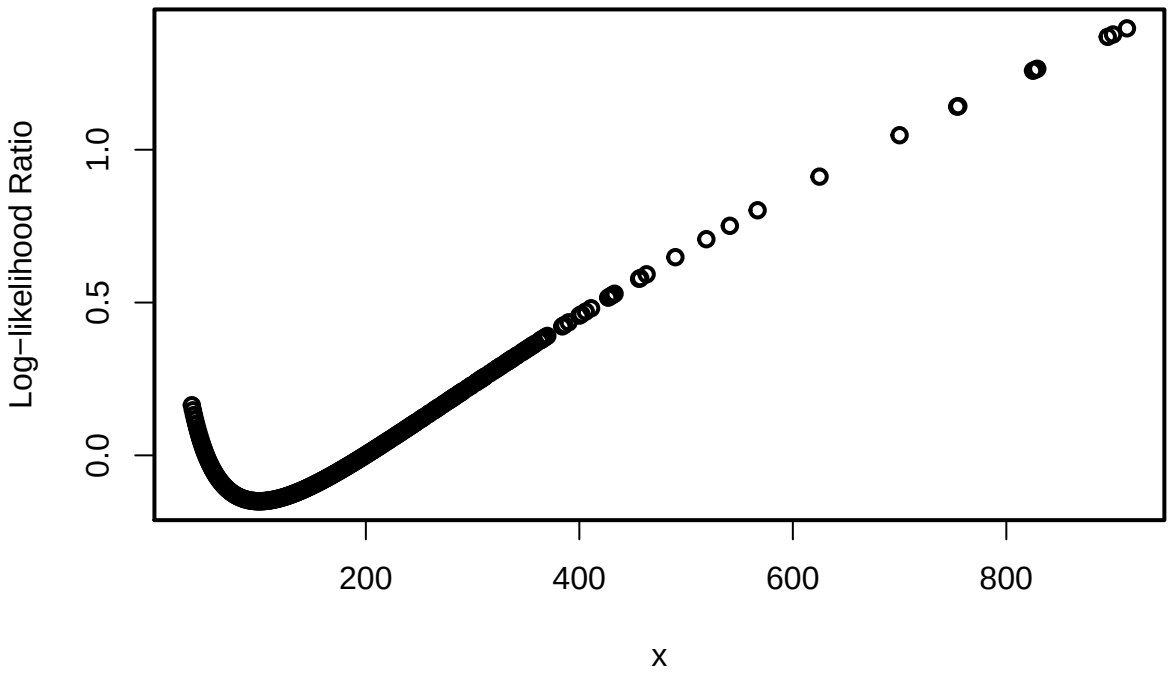
```
ExaminerDistributions$comparisons[[2]][1:3] # upper limit on probability given lognormal is true
```

```
## $test_statistic
## [1] -1.766592
##
## $p_one_sided
## [1] 0.03864825
##
## $p_two_sided
## [1] 0.0772965
```

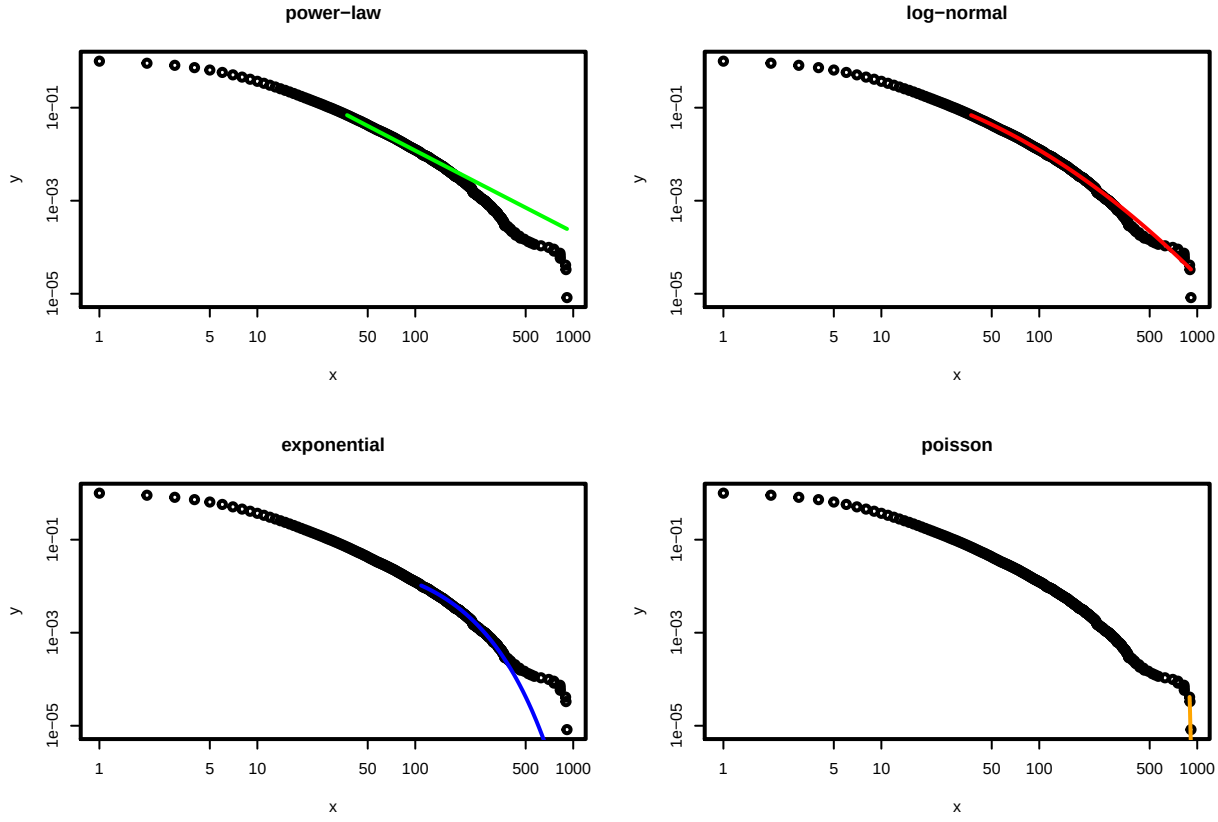
```
OtherDistributions <- fit_distributions(2001, out_lab = "test", degree_distribution_all = orderFrequenc
```


power−law
log−normal
exponential
poisson
y
x
1e−01
1e−03
1e−05
1
5
10
50
100
500
1000


power−law
log−normal
exponential
poisson
y
x
1e−01
1e−03
1e−05
1
5
10
50
100
500
1000




```
par(mfrow = c(2,2))
    par(cex = .5)
    par(lwd = 2)
    plot(OtherDistributions$distributions[[3]], main = "power-law");
    lines(OtherDistributions$distributions[[3]], col="green")
    plot(OtherDistributions$distributions[[2]], main = "log-normal");
    lines(OtherDistributions$distributions[[2]], col = "red")
    plot(OtherDistributions$distributions[[1]], main = "exponential");
    lines(OtherDistributions$distributions[[1]], col="blue")
    plot(OtherDistributions$distributions[[4]], main = "poisson");
    lines(OtherDistributions$distributions[[4]], col="orange")
```




```
OtherDistributions$comparisons[[1]][1:3]
```

```
## $test_statistic
## [1] -7.706742
##
## $p_one_sided
## [1] 6.453516e-15
##
## $p_two_sided
## [1] 1.290703e-14
```

```
OtherDistributions$comparisons[[2]][1:3]
```

```
## $test_statistic
## [1] 7.706742
##
## $p_one_sided
## [1] 1
##
## $p_two_sided
## [1] 1.287859e-14
```

‘‘‘