The present tutorial analyses the Tennis Grand Slam tournaments main results from the statistical point of view. Specifically, I try to answer the following questions:

– How to fit the distribution of the Grand Slam tournaments number of victories across players?  
– How to compute the probability of having player’s victories greater than a specific number?  
– How the number of Grand Slam tournaments winners increases along with time?  
– How can we assign a metric to the tennis players based on the number of Grand Slam tournaments they won?

Our dataset is provided within ref. [1], whose content is based on what reported by the ESP site at: [ESPN site tennis history table](http://www.espn.com/tennis/history).

**Packages**

suppressPackageStartupMessages(library(fitdistrplus))

suppressPackageStartupMessages(library(extremevalues))

suppressPackageStartupMessages(library(dplyr))

suppressPackageStartupMessages(library(knitr))

suppressPackageStartupMessages(library(ggplot2))

**Analysis**

The analysis within the present tutorial is organized as follows:

* a basic data exploration is run
* log-normal distribution is fit against the Grand Slam tournaments victories data
* regression analysis is presented in order to evaluate the increase of tennis Grand Slam champions along with time
* population statistical dispersion is evaluated to determine a Tennis Quotient assigned to tournaments’ winners

**Basic Data Exploration**

Loading the Tennis Grand Slam tournaments dataset and running a basic data exploration.

url\_file <- "https://datascienceplus.com/wp-content/uploads/2017/12/tennis-grand-slam-winners\_end2017.txt"

slam\_win <- read.delim(url(url\_file), sep="\t", stringsAsFactors = FALSE)

dim(slam\_win)

*[1] 489 4*

kable(head(slam\_win), row.names=FALSE)

*| YEAR|TOURNAMENT |WINNER |RUNNER.UP |*

*|----:|:---------------|:-------------|:--------------|*

*| 2017|U.S. Open |Rafael Nadal |Kevin Anderson |*

*| 2017|Wimbledon |Roger Federer |Marin Cilic |*

*| 2017|French Open |Rafael Nadal |Stan Wavrinka |*

*| 2017|Australian Open |Roger Federer |Rafael Nadal |*

*| 2016|U.S. Open |Stan Wawrinka |Novak Djokovic |*

*| 2016|Wimbledon |Andy Murray |Milos Raonic |*

kable(tail(slam\_win), row.names=FALSE)

*| YEAR|TOURNAMENT |WINNER |RUNNER.UP |*

*|----:|:----------|:----------------|:------------------|*

*| 1881|U.S. Open |Richard D. Sears |William E. Glyn |*

*| 1881|Wimbledon |William Renshaw |John Hartley |*

*| 1880|Wimbledon |John Hartley |Herbert Lawford |*

*| 1879|Wimbledon |John Hartley |V. St. Leger Gould |*

*| 1878|Wimbledon |Frank Hadow |Spencer Gore |*

*| 1877|Wimbledon |Spencer Gore |William Marshall |*

nr <- nrow(slam\_win)

start\_year <- slam\_win[nr, "YEAR"]

end\_year <- slam\_win[1, "YEAR"]

(years\_span <- end\_year - start\_year + 1)

*[1] 141*

(total\_slam\_winners <- length(unique(slam\_win[,"WINNER"])))

*[1] 166*

So we have 166 winners spanning over 141 years of Tennis Grand Slam tournaments. We observe that during first and second World Wars, a reduced number of Grand Slam tournaments were played for obvious reasons.

slam\_win\_df <- as.data.frame(table(slam\_win[,"WINNER"]))

slam\_win\_df = slam\_win\_df %>% arrange(desc(Freq))

# computing champions' leaderboard position

pos <- rep(0, nrow(slam\_win\_df))

pos[1] <- 1

for(i in 2:nrow(slam\_win\_df)) {

pos[i] <- ifelse(slam\_win\_df$Freq[i] != slam\_win\_df$Freq[i-1], i, pos[i-1])

}

last\_win\_year = sapply(slam\_win\_df$Var1, function(x) {slam\_win %>% filter(WINNER == x) %>% dplyr::select(YEAR) %>% max()})

# creating and showing leaderboard dataframe

slam\_winners <- data.frame(RANK = pos,

PLAYER = slam\_win\_df$Var1,

WINS = slam\_win\_df$Freq,

LAST\_WIN\_YEAR = last\_win\_year)

kable(slam\_winners)

*| RANK|PLAYER | WINS| LAST\_WIN\_YEAR|*

*|----:|:---------------------------|----:|-------------:|*

*| 1|Roger Federer | 19| 2017|*

*| 2|Rafael Nadal | 16| 2017|*

*| 3|Pete Sampras | 14| 2002|*

*| 4|Novak Djokovic | 12| 2016|*

*| 4|Roy Emerson | 12| 1967|*

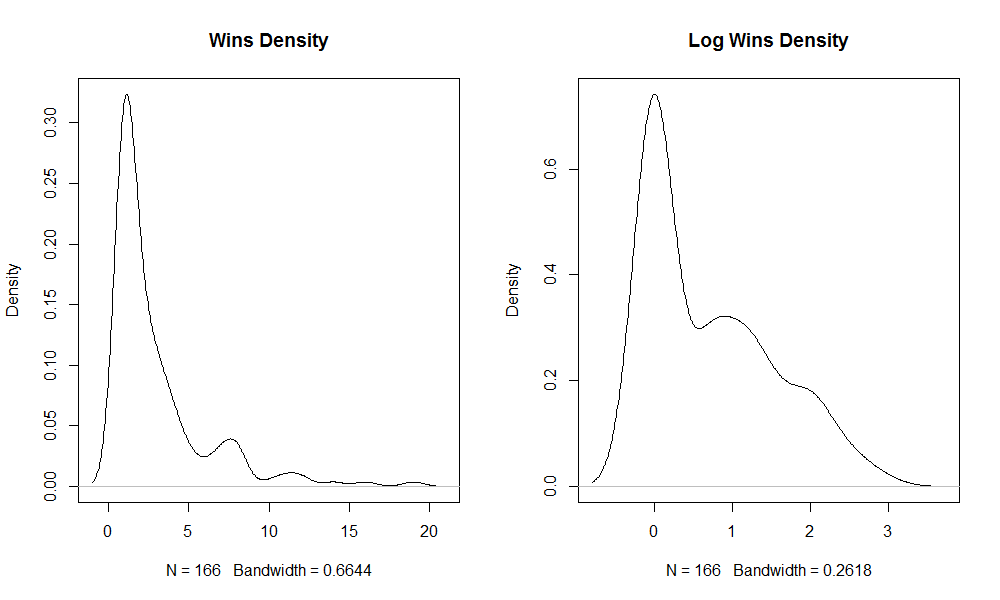
*...*

The WINS and log(WINS) distribution density are shown.

par(mfrow=c(1,2))

plot(density(slam\_winners$WINS), main = "Wins Density")

plot(density(log(slam\_winners$WINS)), main = "Log Wins Density")

Gives this plot:  
[](https://datascienceplus.com/wp-content/uploads/2017/12/wins_density.png)

You may want to arrange the same dataframe ordering by the champions’ last win year.

par(mfrow=c(1,1))

slam\_winners\_last\_win\_year = slam\_winners %>% arrange(LAST\_WIN\_YEAR)

kable(slam\_winners %>% arrange(desc(LAST\_WIN\_YEAR)))

*| RANK|PLAYER | WINS| LAST\_WIN\_YEAR|*

*|----:|:---------------------------|----:|-------------:|*

*| 1|Roger Federer | 19| 2017|*

*| 2|Rafael Nadal | 16| 2017|*

*| 4|Novak Djokovic | 12| 2016|*

*| 43|Andy Murray | 3| 2016|*

*...*

We may want to visualize the timeline of the number of Tennis Grand Slam champions.

df\_nwin = data.frame()

for (year in start\_year : end\_year) {

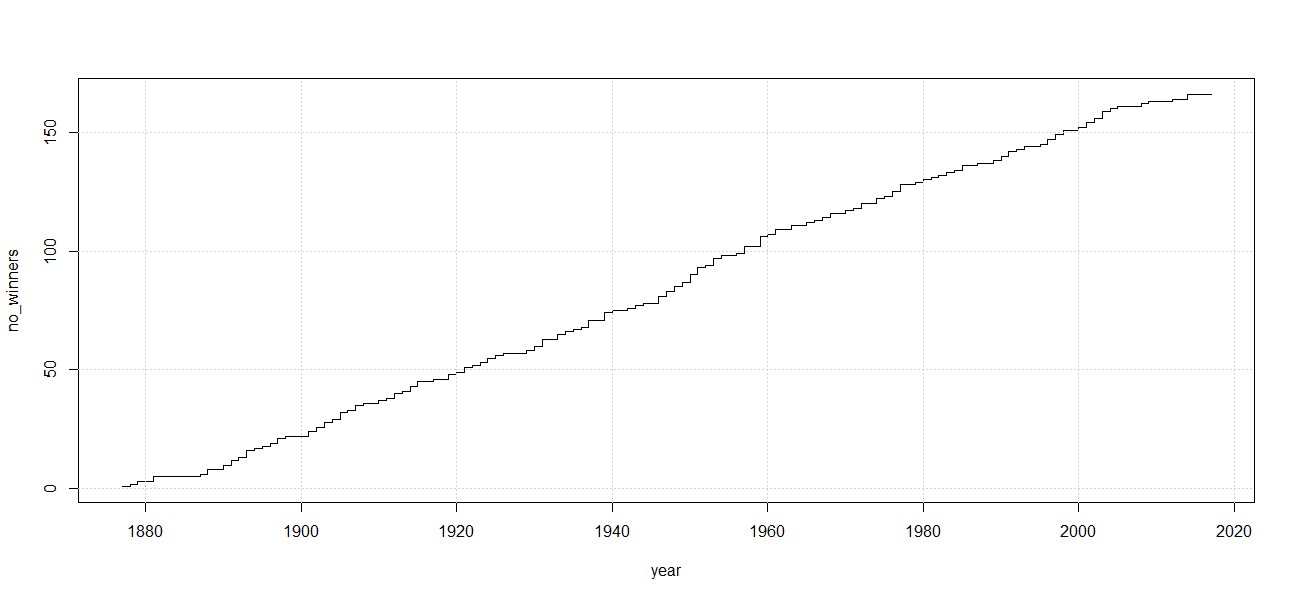
n\_slam\_winners = slam\_win %>% filter(YEAR <= year) %>% dplyr::select(WINNER) %>% unique %>% nrow

df\_nwin = rbind(df\_nwin, data.frame(YEAR = year, N\_WINNERS = n\_slam\_winners))

}

plot(x = df\_nwin$YEAR, y = df\_nwin$N\_WINNERS, type ='s', xlab = "year", ylab = "no\_winners")

grid()

Gives this plot:  
[](https://datascienceplus.com/wp-content/uploads/2017/12/no_winners_by_year.png)

We may want to visualize the timeline of the Grand Slam tournaments wins record.

df2\_nwin = data.frame()

for (year in start\_year : end\_year) {

slam\_win\_years = slam\_win %>% filter(YEAR <= year)

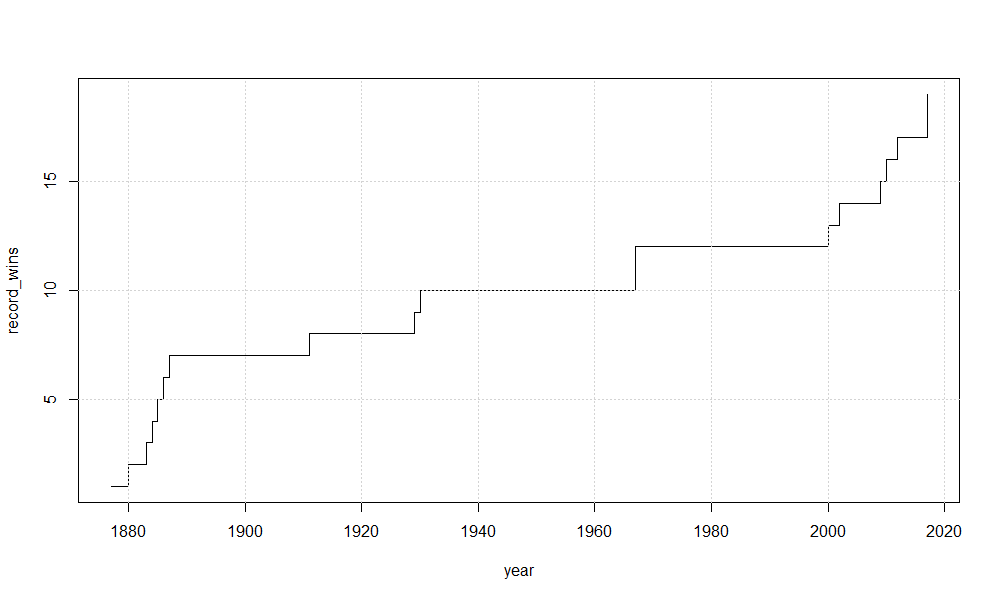
slam\_win\_record = as.data.frame(table(slam\_win\_years[,"WINNER"]))

df2\_nwin = rbind(df2\_nwin, data.frame(YEAR = year, RECORD\_WINS = max(slam\_win\_record$Freq)))

}

plot(x = df2\_nwin$YEAR, y = df2\_nwin$RECORD\_WINS, type ='s', xlab = "year", ylab = "record\_wins")

grid()

Gives this plot:  
[](https://datascienceplus.com/wp-content/uploads/2017/12/record_wins_fix.png)

It is interesting to have a look at the number of wins frequency.

wins\_frequency <- as.data.frame(table(slam\_winners[,"WINS"]))

colnames(wins\_frequency) <- c("WINS", "FREQUENCY")

kable(wins\_frequency)

*|WINS | FREQUENCY|*

*|:----|---------:|*

*|1 | 80|*

*|2 | 25|*

*|3 | 19|*

*|4 | 12|*

*|5 | 5|*

*|6 | 3|*

*|7 | 6|*

*|8 | 8|*

*|10 | 1|*

*|11 | 2|*

*|12 | 2|*

*|14 | 1|*

*|16 | 1|*

*|19 | 1|*

summary(slam\_winners[,"WINS"])

*Min. 1st Qu. Median Mean 3rd Qu. Max.*

*1.000 1.000 2.000 2.946 3.750 19.000*

**Probabilistic Distribution Fit**

We now take advantage of the fitdist() function within the fitdistr package to fit a lognormal distribution for our Grand Slam wins data.

fw <- fitdist(slam\_winners$WINS, "lnorm")

summary(fw)

*Fitting of the distribution ' lnorm ' by maximum likelihood*

*Parameters :*

*estimate Std. Error*

*meanlog 0.7047927 0.06257959*

*sdlog 0.8062817 0.04425015*

*Loglikelihood: -316.7959 AIC: 637.5918 BIC: 643.8158*

*Correlation matrix:*

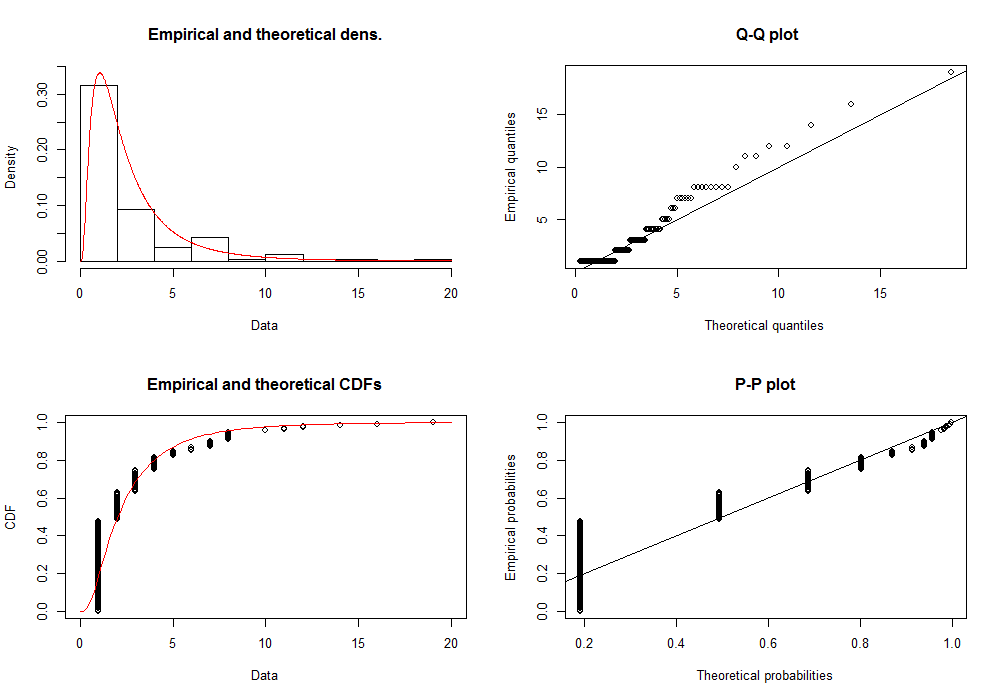
*meanlog sdlog*

*meanlog 1 0*

*sdlog 0 1*

Then we can plot the distribution fit results.

plot(fw)

Gives this plot:  
[](https://datascienceplus.com/wp-content/uploads/2017/12/fitdistr.png)

# left outliers quantile

left\_thresh <- 0.05

# right outliers quantile

right\_thresh <- 0.95

# determining the outliers

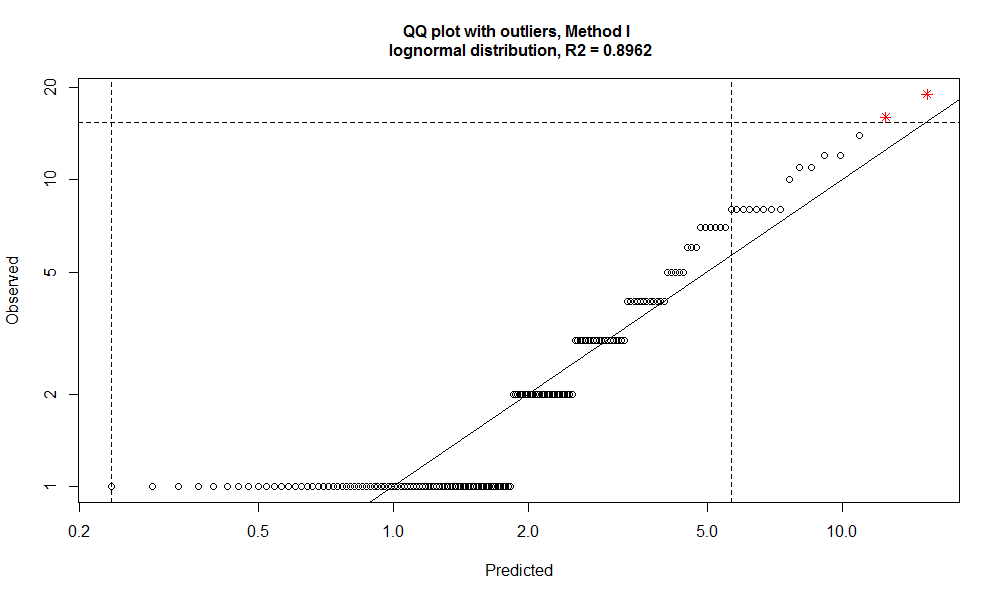
slam\_outlier <- getOutliersI(as.vector(slam\_winners$WINS),

FLim = c(left\_thresh, right\_thresh),

distribution = "lognormal")

# outliers are plotted in red color

outlierPlot(slam\_winners$WINS, slam\_outlier, mode="qq")

Gives this plot:  
[](https://datascienceplus.com/wp-content/uploads/2017/12/outlierplot.png)

The outliers are:

slam\_winners[slam\_outlier$iRight,]

*RANK PLAYER WINS LAST\_WIN\_YEAR*

*1 1 Roger Federer 19 2017*

*2 2 Rafael Nadal 16 2017*

The mean and standard deviation associated to the log-normal fit are:

(mean\_log <- fw$estimate["meanlog"])

*meanlog*

*0.7047927*

(sd\_log <- fw$estimate["sdlog"])

*sdlog*

*0.8062817*

Now we compute the probability associated with 19 and 16 wins.

# clearing names

names(mean\_log) <- NULL

names(sd\_log) <- NULL

# probability associated to 19 wins performance or more

(lnorm\_19 <- plnorm(19, mean\_log, sd\_log, lower.tail=FALSE))

*[1] 0.002736863*

# probability associated to 16 wins performance or more

(lnorm\_16 <- plnorm(16, mean\_log, sd\_log, lower.tail=FALSE))

*[1] 0.005164628*

However, if a random variable follows a log-normal distribution, its logarithm follows a normal distribution. Hence we fit the logarithm of the variable under analysis using a normal distribution and compare the results with above log-normal fit.

fw\_norm <- fitdist(log(slam\_winners$WINS), "norm")

summary(fw\_norm)

*Fitting of the distribution ' norm ' by maximum likelihood*

*Parameters :*

*estimate Std. Error*

*mean 0.7047927 0.06257959*

*sd 0.8062817 0.04425015*

*Loglikelihood: -199.8003 AIC: 403.6006 BIC: 409.8246*

*Correlation matrix:*

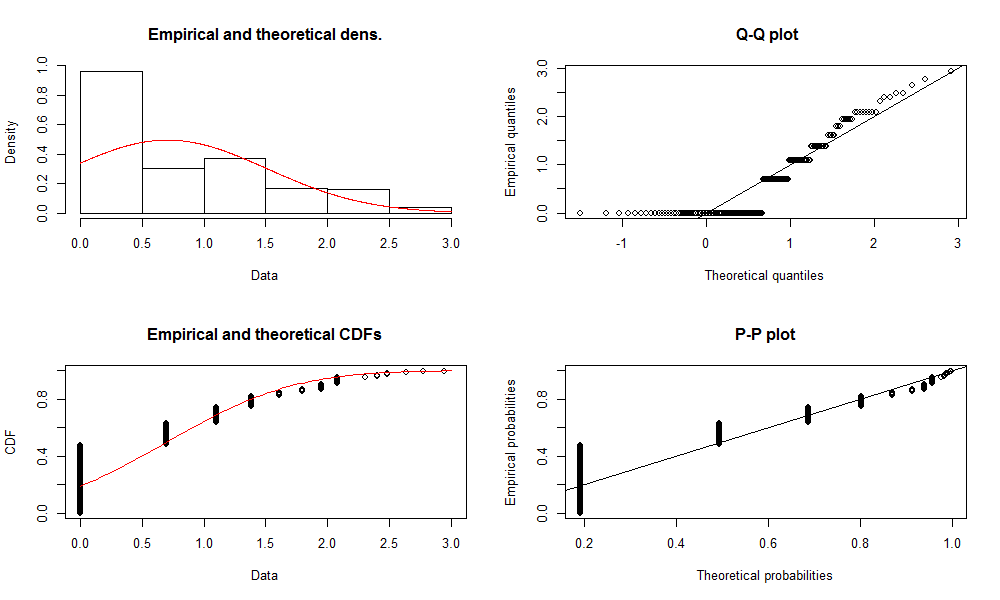
*mean sd*

*mean 1 0*

*sd 0 1*

Similarly, we plot the fit results.

plot(fw\_norm)

Gives this plot:  
[](https://datascienceplus.com/wp-content/uploads/2017/12/fitdistr_norm.png)

# left outliers quantile

left\_thresh <- 0.05

# right outliers quantile

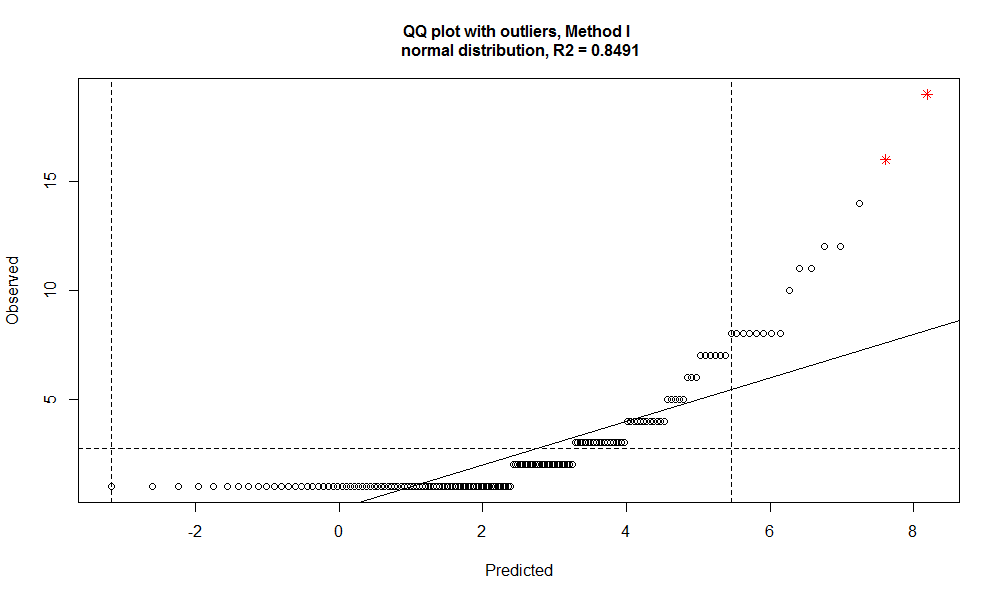
right\_thresh <- 0.95

slam\_outlier <- getOutliersI(log(as.vector(slam\_winners$WINS)),

FLim = c(left\_thresh, right\_thresh),

distribution = "normal")

outlierPlot(slam\_winners$WINS, slam\_outlier, mode="qq")

Gives this plot:  
[](https://datascienceplus.com/wp-content/uploads/2017/12/outlierplot_norm.png)

The outliers are:

slam\_winners[slam\_outlier$iRight,]

*RANK PLAYER WINS LAST\_WIN\_YEAR*

*1 1 Roger Federer 19 2017*

*2 2 Rafael Nadal 16 2017*

The mean and standard deviation values are:

# mean and standard deviation of the fitted lognormal distribution

(mean\_norm <- fw\_norm$estimate["mean"])

*mean*

*0.7047927*

(sd\_norm <- fw\_norm$estimate["sd"])

*sd*

*0.8062817*

As we can see above, same fitting parameters result from the two approaches, even though with different log-likelihood, AIC and BIC metrics. Now we compute the probability associated to 19 and 16 wins together with their distance from the mean in terms of multiples of the standard deviation.

# clearing names

names(mean\_norm) <- NULL

names(sd\_norm) <- NULL

# probability associated to the 19 wins performance or better

(norm\_19 <- pnorm(log(19), mean\_norm, sd\_norm, lower.tail=FALSE))

*[1] 0.002736863*

# standard deviation times from the mean associated to 19 wins

(deviation\_19 <- (log(19) - mean\_norm)/sd\_norm)

*[1] 2.777747*

# probability associated to the 16 wins performance or better

(norm\_16 <- pnorm(log(16), mean\_norm, sd\_norm, lower.tail=FALSE))

*[1] 0.005164628*

# standard deviation times from the mean associated to 16 wins

(deviation\_16 <- (log(16) - mean\_norm)/sd\_norm)

*[1] 2.564608*

As we can see above, we also obtained the same probability value as resulting from the log-normal distribution fit. In the following, we consider the second fitting approach (the one which takes the log of the original variable) for easing the computation of the distance from the mean in terms of multiples of the standard deviation.

**Regression Analysis**

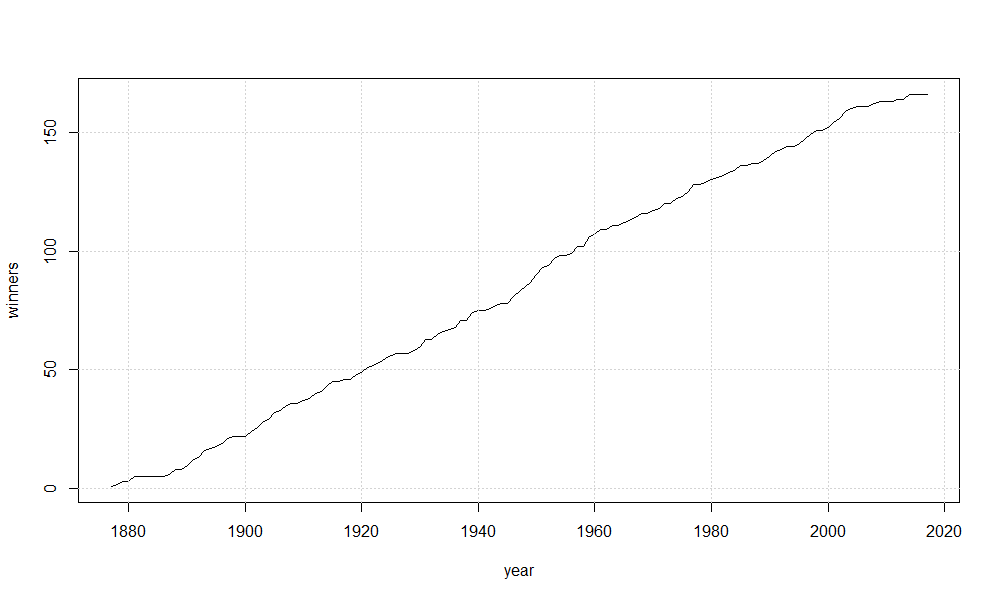
Let us see again the plot of the number of tennis Grand Slam winners against their timeline.

year <- df\_nwin $YEAR

winners <- df\_nwin$N\_WINNERS

plot(x = year, y = winners, type ='l')

grid()

Gives this plot:  
[](https://datascienceplus.com/wp-content/uploads/2017/12/winners_per_year.png)

It is visually evident the linear relationship between the variables. Hence, a linear regression would help in understanding how many newbie Grand Slam winners we may have each year.

year\_lm <- lm(winners ~ year)

summary(year\_lm)

*Call:*

*lm(formula = winners ~ year)*

*Residuals:*

*Min 1Q Median 3Q Max*

*-9.8506 -1.9810 -0.4683 2.6102 6.2866*

*Coefficients:*

*Estimate Std. Error t value Pr(>|t|)*

*(Intercept) -2.388e+03 1.220e+01 -195.8 <2e-16 \*\*\**

*year 1.270e+00 6.264e-03 202.8 <2e-16 \*\*\**

*---*

*Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1*

*Residual standard error: 3.027 on 139 degrees of freedom*

*Multiple R-squared: 0.9966, Adjusted R-squared: 0.9966*

*F-statistic: 4.113e+04 on 1 and 139 DF, p-value: < 2.2e-16*

Coefficients are reported as significant and the R-squared value is very high. On average each year, 1.27 Grand Slam tournaments newbie winners show up. Residuals analysis has not been reported for brevity. Similarly, we can regress the year against the number of winners.

n\_win\_lm <- lm(year ~ winners)

summary(n\_win\_lm)

*Call:*

*lm(formula = year ~ winners)*

*Residuals:*

*Min 1Q Median 3Q Max*

*-4.8851 -1.9461 0.3268 1.4327 7.9641*

*Coefficients:*

*Estimate Std. Error t value Pr(>|t|)*

*(Intercept) 1.880e+03 3.848e-01 4886.4 <2e-16 \*\*\**

*winners 7.846e-01 3.868e-03 202.8 <2e-16 \*\*\**

*---*

*Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1*

*Residual standard error: 2.379 on 139 degrees of freedom*

*Multiple R-squared: 0.9966, Adjusted R-squared: 0.9966*

*F-statistic: 4.113e+04 on 1 and 139 DF, p-value: < 2.2e-16*

Coefficients are reported as significant and the R-squared value is very high. On average, each new Grand Slam tournaments winner appears every 0.7846 fraction of year. Residuals analysis has not been reported for brevity. Such model can be used to predict the year when a given number of Grand Slam winners may show up. For example, considering Federer, Nadal and Sampras wins, we obtain:

# probability associated to the 14 wins performance or better

(norm\_14 <- pnorm(log(14), mean\_norm, sd\_norm, lower.tail=FALSE))

*[1] 0.008220098*

# standard deviation times from the mean associated to 14 wins

(deviation\_14 <- (log(14) - mean\_norm)/sd\_norm)

*[1] 2.398994*

# average number of Grand Slam winners to expect for a 19 Grand Slam wins champion

(x\_19 <- round(1/norm\_19))

*[1] 365*

# average number of Grand Slam winners to expect for a 16 Grand Slam wins champion

(x\_16 <- round(1/norm\_16))

*[1] 194*

# average number of Grand Slam winners to expect for a 14 Grand Slam wins champion

(x\_14 <- round(1/norm\_14))

*[1] 122*

The x\_19, x\_16 and x\_14 values can be interpreted as the average size of Grand Slam tournaments winners population to therein find a 19, 16, 14 times winner respectively. As a consequence, the prediction of the calendar years to see players capable to win 19, 16, 14 times is:

predict(n\_win\_lm, newdata = data.frame(winners = c(x\_19, x\_16, x\_14)), interval = "p")

*fit lwr upr*

*1 2166.732 2161.549 2171.916*

*2 2032.573 2027.779 2037.366*

*3 1976.084 1971.355 1980.813*

The table above shows the earliest year when, on average, to expect a Grand Slam tournament winner capable to win 19, 16, 14 times (fit column), together with lower (lwr) and upper (upr) bound predicted values. In the real world, 14 wins champion showed up a little bit later than expected by our linear regression model, whilst 16 and 19 win champions did much earlier than expected by the same model.

**Population Statistical Dispersion Analysis**

In our previous analysis, we computed the distance from the mean for 19, 16 and 14 Grand Slam tournaments win probabilities, distance expressed in terms of multiples of the standard deviation.

deviation\_19

*[1] 2.777747*

deviation\_16

*[1] 2.564608*

deviation\_14

*[1] 2.398994*

Based on above values, we can compute the probability to have a 19, 16 and 14 times winner. As we saw before, we resume up such result using the pnorm() function.

(prob\_19 <- pnorm(mean\_norm+deviation\_19\*sd\_norm, mean\_norm, sd\_norm, lower.tail = FALSE))

*[1] 0.002736863*

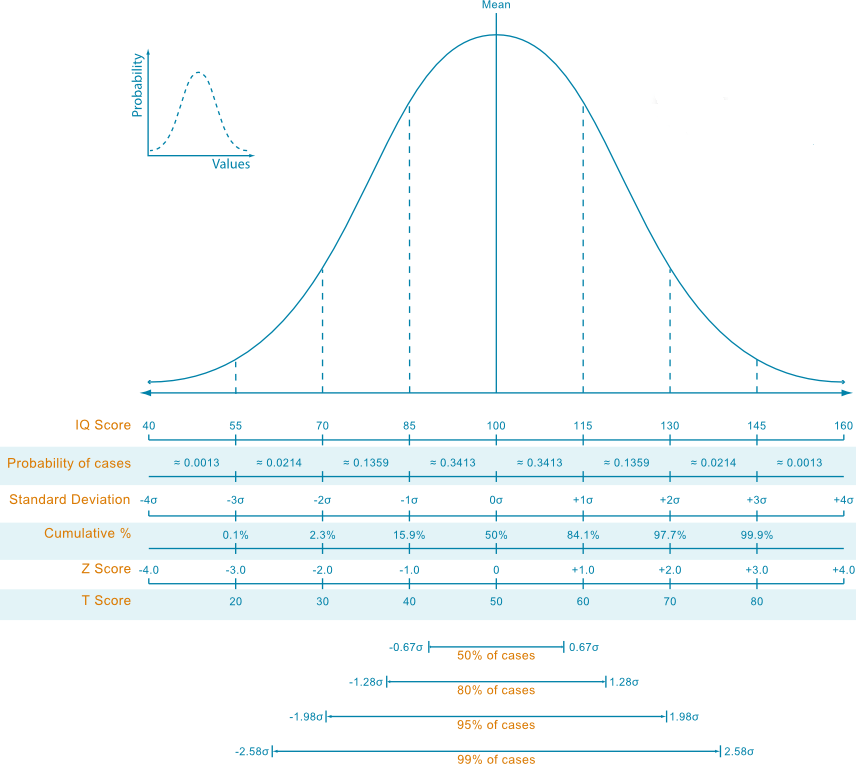
(prob\_16 <- pnorm(mean\_norm+deviation\_16\*sd\_norm, mean\_norm, sd\_norm, lower.tail = FALSE))

*[1] 0.005164628*

(prob\_14 <- pnorm(mean\_norm+deviation\_14\*sd\_norm, mean\_norm, sd\_norm, lower.tail = FALSE))

*[1] 0.008220098*

Similarly to the Intellectual Quotient (IQ) assigning a value equal to 100 at the mean and +/- 15 points for each standard deviation of distance from the mean itself, we can figure out a Tennis Quotient (TQ).

Here in below, we show a plot to remember how the IQ is computed:  
[](https://datascienceplus.com/wp-content/uploads/2017/12/iq-score-and-normal-distribution.png)

We notice that the median of our player’s population scores a TQ equal to 100.

(median\_value <- median(slam\_winners$WINS))

*[1] 2*

(deviation\_median <- (log(median\_value) - mean\_norm)/sd\_norm)

*[1] -0.01444343*

round(100 + 15\*deviation\_median)

*[1] 100*

We now compute the Tennis Quotients (TQ) for leading champions.

(Federer\_TQ <- round(100 + 15\*deviation\_19))

*[1] 142*

(Nadal\_TQ <- round(100 + 15\*deviation\_16))

*[1] 138*

(Sampras\_TQ <- round(100 + 15\*deviation\_14))

*[1] 136*

And what about for example 7 times Grand Slam tournament winner?

(deviation\_7 <- (log(7) - mean\_norm)/sd\_norm)

*[1] 1.53931*

TQ\_7wins <- round(100 + 15\*deviation\_7)

*[1] 123*

Let us then compute the Tennis Quotients (TQ) for all our tennis Grand Slam tournaments winners.

tq\_compute <- function(x) {

deviation\_x <- (log(x) - mean\_norm)/sd\_norm

round(100 + 15\*deviation\_x)

}

slam\_winners = slam\_winners %>% mutate(TQ = tq\_compute(WINS))

kable(slam\_winners)

*| RANK|PLAYER | WINS| LAST\_WIN\_YEAR| TQ|*

*|----:|:---------------------------|----:|-------------:|---:|*

*| 1|Roger Federer | 19| 2017| 142|*

*| 2|Rafael Nadal | 16| 2017| 138|*

*| 3|Pete Sampras | 14| 2002| 136|*

*| 4|Novak Djokovic | 12| 2016| 133|*

*....*

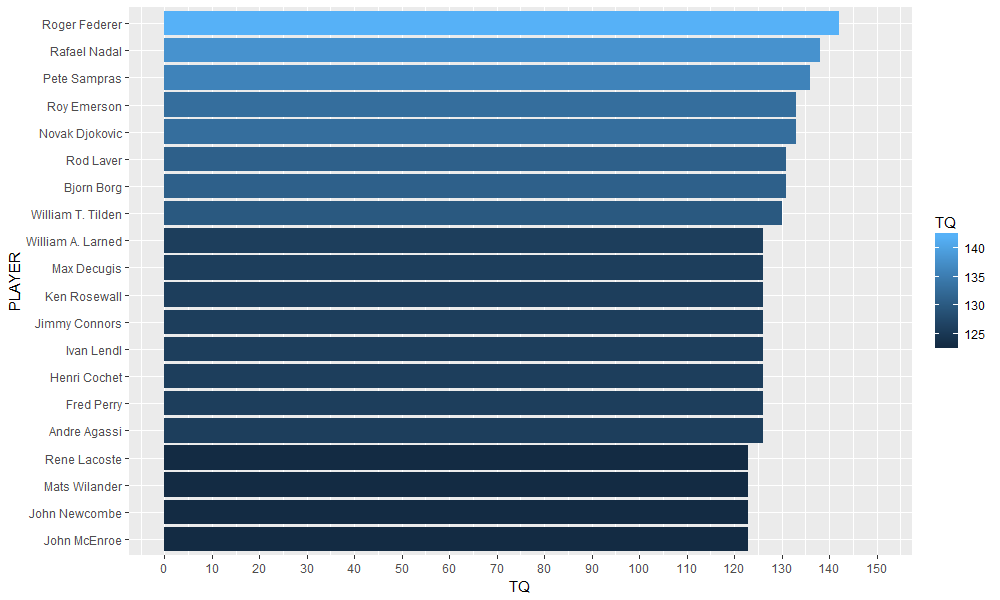
We then visualize the top twenty Grand Slam tournaments champions.

ggplot(data=slam\_winners[1:20,], aes(x=reorder(PLAYER, TQ), y=TQ, fill = TQ)) +

geom\_bar(stat ="identity") + coord\_flip() +

scale\_y\_continuous(breaks = seq(0,150,10), limits = c(0,150)) +

xlab("PLAYER")

Gives this plot.  
[](https://datascienceplus.com/wp-content/uploads/2017/12/barplot.png)

**Conclusions**

The answers to our initial questions:

– The log-normal distribution  
– Based upon the fitted distribution and taking advantage of plnorm() or pnorm() stats package functions, probabilities have been computed  
– A linear increase is a very good fit for that, resulting in significative regression coefficients and high R-squared values  
– Yes, we defined the Tennis Quotient similarly to the Intellectual Quotient and show the resulting leaderboard. Federer is confirmed “genius” in that respect, however, a few other very talented players are not that far from him.