Extreme Value Analysis on Facebook Stock Return

Khanh Truong - Paul Dublanche

December 2018

1 Introduction

1.1 Framework

Extreme Value Theory (EVT) is a field of mathematical statistics which analyses the tail of distributions. Two widely used approaches in practice are "Block Maxima" and "Peak Over Threshold".

There are various application of EVT such as risk management in finance and insurance sector, extreme floods prediction for disaster management, ecological populations forecast for ecosystem improvement, etc.

The aims of this project is to apply EVT into practical problem with real data set and to compare the results of the two methods. Since the authors have background of banking and tend work in banking-insurance area, a financial data is going to be used for demonstration in this analysis.

1.2 Motivation

After the first launch in 2004, Facebook expeditiously becomes the biggest social network online with 2.3 million users in September 2018 (1). It is not until May 17, 2012 that shares of stock of Facebook was publicly traded with initially price at US$38. It raised more than 3.5 times to US$135 on December 27, 2018, equivalent to 22.5% of Annualized Total Return - far higher than interest rate of every developed economy in the world.

Though investing in Facebook seems to be tremendously long-term-profitable, we would like to take a closer look if there is any daily return that is notably low which might cause a serious shock in the market. We extracted Facebook stock price (ticker "FB") from 01/06/2012 (first complete trading month) to 30/11/2018 (last complete trading month) for the analysis.

Instead of using "Closing Price", "Adjusted Closing Price" is the key metric employed in this project. "Adjusted Closing Price" is a modified version of "Closing Price" by taking more dividends, stock splits and new stock offerings into account. This quantity, therefore, reflects more accurate the value of the company than the "Closing Price" does.

Instead of working on standard return, there are several reasons of preferring log-return. First, log-return reduces the variation of the time series making it easier to fit the model. Second, for $\frac{price_{t+1}}{price_t}$ close to 1, log-return will be approximately equal to standard return (result of Taylor expansion). Thus the common interpretation (by percentage) does not change.

$$log(\frac{price_{t+1}}{price_t}) \approx \frac{price_{t+1}}{price_t} - 1$$

Third, log-return is more convenient for time-additivity. It implies much faster calculation of return between day $t+i$ and day t

$$log(\frac{price_{t+i}}{price_t}) = \log(price_{t+i}) - \log(price_t)$$

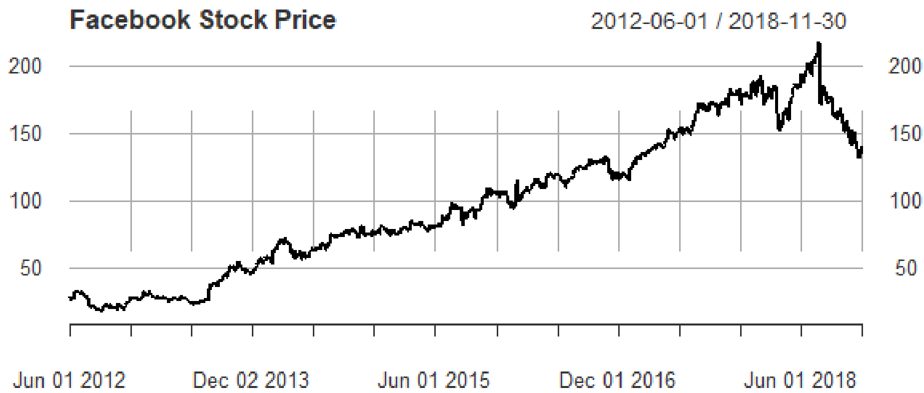

Figure 1: Facebook Stock Price

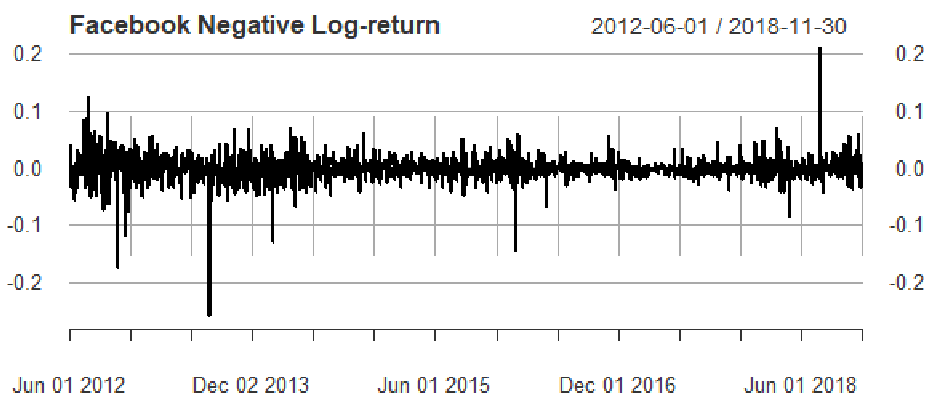

Figure 2: Facebook Negative Log-return

Given the prepared inputs, two main methods of EVT Block Maxima and Peak Over Threshold are both going to be employed in this project.

2 Block Maxima

2.1 GEV fit

Under Block Maxima method, the maximum negative log-return every month in the data is extracted. There are 78 observations, with respect to 78 months from June 2012 to November 2018. Even though we are aware this is not a sizable data set, we do not change to analyze another company (Microsoft for instance, stock price has been available from 1986) because we assume the project is a real case in practice. It is sometimes impossible to change to an “easier-data” in reality but adapt methods and be aware of what we are doing given small sample.

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.007973	0.023343	0.033528	0.039102	0.048815	0.210239

Table 1: Facebook Monthly Maxima Negative Log-return

According to EVT, the block maxima would follow Generalized extreme value (GEV) distribution with three parameters location $\mu \in R,\ scale\ \sigma > 0$ and shape $\zeta \in R$.

```
> (gev <- fgev(month.max$nlogreturn)) # GEV

Estimates
     loc     scale     shape
0.02716   0.01525   0.17448
Standard Errors
      loc      scale      shape
0.001956   0.001494   0.089259
```

Function *fgev* in package *evd* in *R* allows to estimate the parameters of the distribution. Let's have a quick comparison of the fitness to the data between GEV distribution and Gaussian distribution.

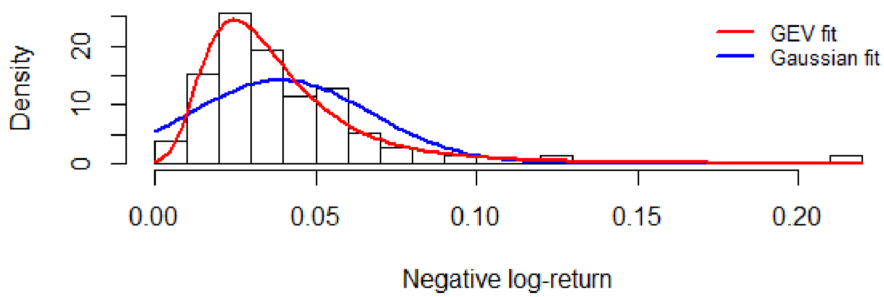

Figure 3: GEV fit and Gaussian fit

It is obvious that Gaussian distribution, whose characteristic is symmetric, is not able to capture long and heavy right tail of the data. On the other hand, GEV distribution seems to catch appropriately the shape of the histogram.

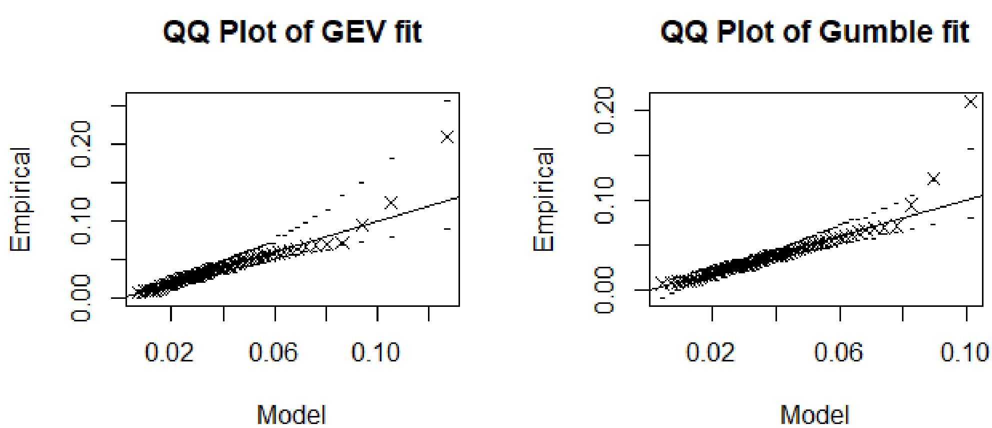

Figure 4: QQ Plot GEV fit and Gumbel fit

An usual check of the model is that if parameter *shape* is significantly different from 0. If we can not reject the null hypothesis, it is recommended to apply Gumbel distribution fit (a specific case of GEV in which $\zeta = 0$) since it will help reduce the range of confidence interval (CI) of prediction later (as seen in Figure 4).

Apart from the fact that point-wise CI of Gumbel fit is smaller, Figure 4 also shows that Gumbel under-estimates the large values in the data set: the last two empirical points 0.21 and 0.12 are out of the CI of the model. As risk analysts, we should prefer GEV for this reason since we should be very risk-adverse that choose model fits more to the worst cases (large values).

```
> (p_value <- 2*pnorm(abs(gev$estimate['shape'])/gev$std.err['shape'],lower.tail=FALSE))
     shape
0.05061229
> # likelihood ratio test, H0: 2 models are the same
> gumble <- fgev(month.max$nlogreturn, shape=0)
> anova(gev,gumble)
Analysis of Deviance Table

       M.Df Deviance Df  Chisq Pr(>chisq)
gev       3  -392.05
gumble    2  -385.55  1 6.5059    0.01075 *
```

Furthermore, based on the Wald test and likelihood ratio test, we again reject the null hypothesis (despite the fact that the statistic of Wald test is slightly greater than 0.05). Besides, GEV graphically seems to fit more to histogram of the data, especially at the peak (Figure 5). As a result, we keep fitting GEV, rather than Gumbel distribution.

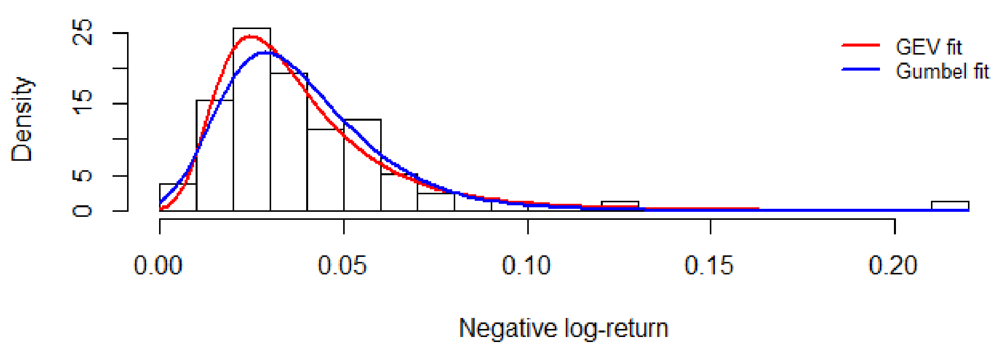

Figure 5: GEV fit and Gumbel fit

2.2 Return period - Return level

Function $fgev$ allows to re-parameterize the model with different parameters of GEV distribution: quantile, scale and shape. Choosing option *prop* $= \frac{1}{T}$ will generate the Value at Risk (VaR - Quantile) given the time T. In finance, we are not interested in VaR 100 years $VaR_{1/100}$ but quite short-term such as VaR 1 year $VaR_{1/1}$ or VaR 3 years $VaR_{1/3}$. Assume this project serves as a supportive document for an 1-year investment proposal of an investment fund. The following illustration takes 1-year period to calculate VaR.

```
> (gev.1year <- fgev(month.max$nlogreturn, prob = 1/12))

Estimates
quantile     scale     shape
 0.07241   0.01586   0.11204
```

It is expected that the maximum daily loss of of Facebook would be higher than 7.24% once every year (Equivalently: there is probability $\frac{1}{12}$ that the maximum daily loss of Facebook every month is higher than 7.24%).

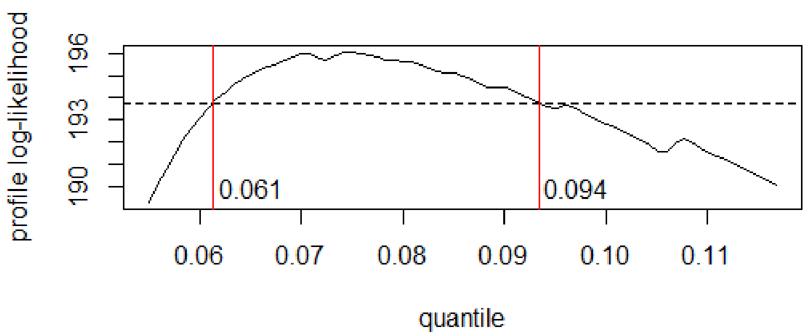

Figure 6: Log-likelihood of 1-Year Return Level GEV

CI of VaR can be produced by profiling the log-likelihood of the parameter “quantile”. Correspondingly, we can state that 95% VaR would be in (6.1%; 9.4%). This CI’s range is quite larger than usual symmetric CI’s (based on Central Limit Theorem). This might be partly because the profile log-likelihood is not smooth due to small sample size (see more on Appendix 5.3).

```
> confint(gev.1year)['quantile',] # compare to symmetric C.I
     2.5 %     97.5 %
0.05996662 0.08485348
```

	Left bound	Right bound
Profile log-likelohood CI	0.061	0.094
Symmetric CI	0.060	0.085

Table 2: Confidence Interval of Monthly Maxima Negative Log-return

We can obtain different VaR according to given time period by replacing values in option $prob = \frac{1}{T}$. The following return level plot is a visual summary of different tries on value *prob*. Given any time period T (in the x-axis), the projection on y-axis from the return level curve would give prediction VaR in period T. More detail discussion on return level plot is presented in Appendix 5.2.

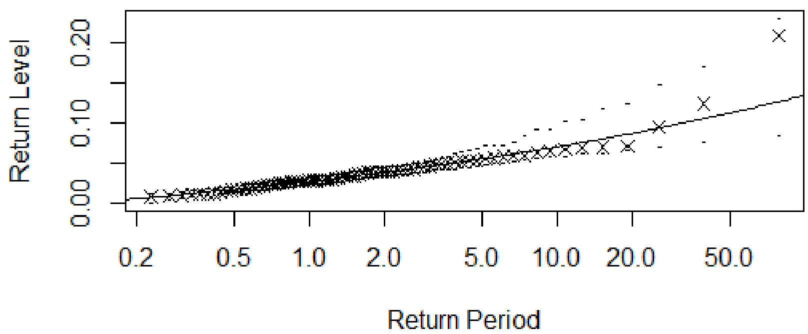

Figure 7: Return Level Plot GEV

3 Peak Over Threshold

Unlike block maxima, peak over threshold is another approach of ETV in which we study the distribution of values which are higher than some threshold. Given an appropriate threshold, the distribution of the exceedances would theoretically follow Generalized Pareto distribution (GPD). In this section, we apply peak over threshold to GPD version with two parameters: scale and shape. Detail comparison between GPD two parameters and GPD three parameters is presented in Appendix 5.4.

3.1 Threshold choice

An plainly important (and argumentative) step in peak over threshold is selecting the threshold. Exceedances of an unsuitable threshold will not shape into a GPD. Moreover, if the chosen threshold is excessively high, the sample size for analysis would be relatively small. Consequently, the result of the analysis is uninteresting (the CI of the parameters are vastly large, for instance). As a rule of thumb in practice, we pick a minimum threshold from which the distribution of exceedances would reasonably stable. In our case, $threshold = 0.0467$ seems to be proper.

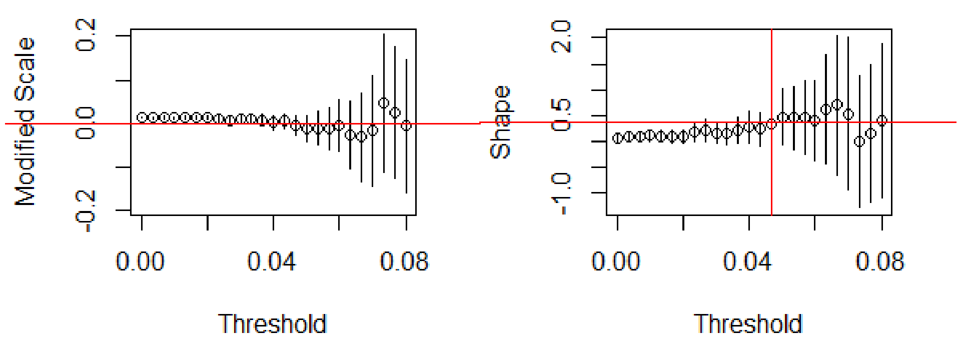

Figure 8: Variation of parameters of GPD given threshold

3.2 GPD fit

Recall that the negative log-return of Facebook stock is fairly stationary (Figure 2). Therefore we can apply peak over threshold method for the time series without the needs of including seasonal pattern into the parameters.

Function *fpot* with option $npp = 252$ (there are 252 trading days in a year) will estimate the two parameters of fitting GPD.

```
> (gpd <- fpot(FB$nlogreturn, threshold, npp = 252)) # Generalized Pareto

Estimates
   scale     shape
0.01195   0.35800
Standard Errors
   scale      shape
0.003113  0.211541
```

We can conduct similar tests to Block Maxima's part to examine the significance and CI of the parameters. In general, the model seems to fit very well to the data, based on the Q-Q plot - the points are located randomly around the diagonal line.

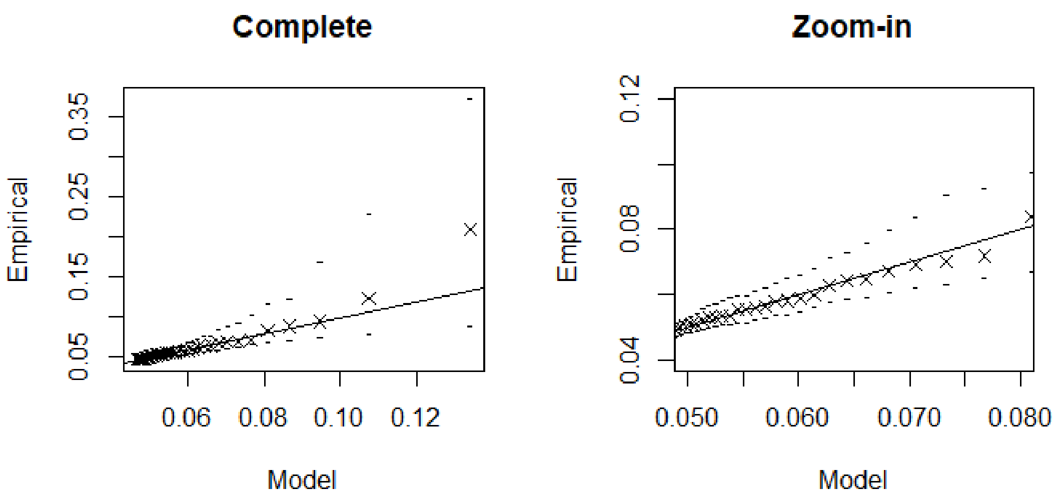

Figure 9: QQ Plot GPD fit

3.3 Return period - Return level

We can re-parameterize GPD similarly as we did for GEV to obtain VaR given time period, by add option *mper* to function *fpot*:

```
> (gpd.1year <- fpot(FB$nlogreturn, threshold, npp = 252, mper = 1))

Estimates
```

```
 rlevel    shape
0.07425  0.36037
```

The parameter *rlevel* is literally the *quantile* of the distribution (or VaR). In that manner, it is expected that the maximum daily loss of of Facebook would be higher than 7.43% once every year.

CI of *rlevel* relying on profiling log-likelihood is easily constructed by function $plot(profile)$. As a result, we are sure 95% that the maximum daily loss of of Facebook would be higher than somewhere between 6.5% and 9.3%.

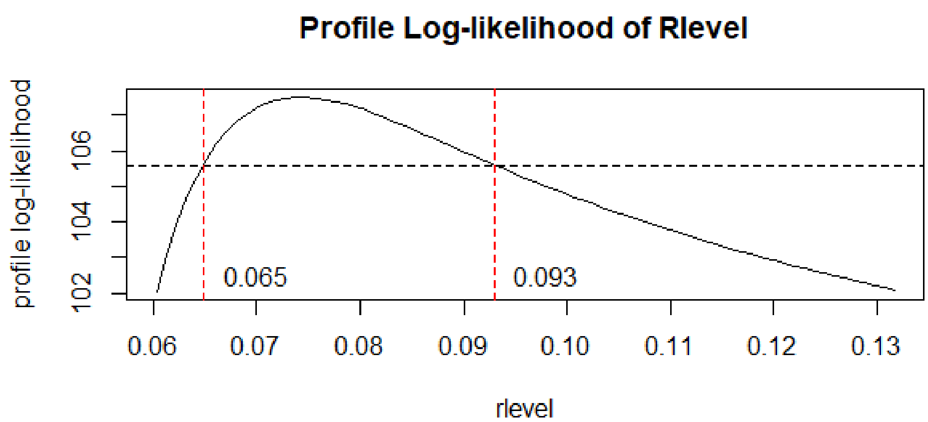

Figure 10: Log-likelihood of 1-Year Return Level GPD

We can obtain different VaR according to time period by replacing values in option *mper*. The following return level plot is a recap of our results.

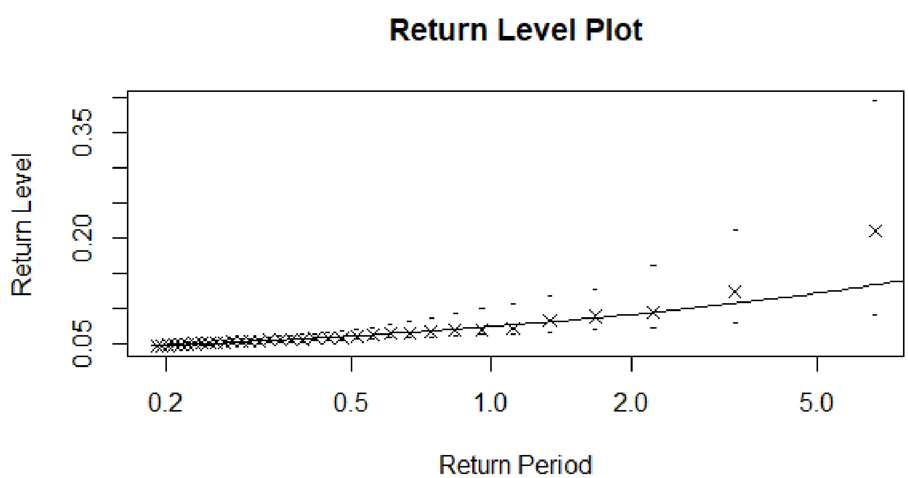

Figure 11: Return Level GPD

4 Compare Block Maxima and Peak Over Threshold

In this session we summarize the results of the two approaches. Overall, results from two methods are alike. Block maxima seems to estimate lower VaR in short-term (0.5 and 1 year), while peak over threshold technique estimates lower in long-term (3 years). Additionally, range of CI of VaR in block maxima seems to be wider in all the cases. Note that CI of block maxima for 3-year and longer period are not available since standard error is not able to be obtained from function $fgev$ (*"observed information matrix is singular"*, see more on full code appendix from Page 13).

Method	Block Maxima	Peak Over Threshold
0.5-Year	0.058	0.061
1-Year	0.072	0.074
3-Year	0.121	0.104

Table 3: VaR

	Block Maxima	Block Maxima	Peak Over Threshold	Peak Over Threshold
	Left Bound	Right Bound	Left Bound	Right Bound
0.5-Year	0.049	0.068	0.055	0.067
1-Year	0.060	0.085	0.062	0.086
3-Year	N.A	N.A	0.071	0.137

Table 4: Confidence Interval of VaR

To sum up, taking the average values of the two techniques, it is expected that the maximum daily loss of Facebook shareholder would be higher than 7.3% once every year. Besides, we can be sure 95% that this metric would be in range (6.1%; 8.6%). Remind that loosing 7.3% in one single day is extremely serious event in stock market. It might be a start of a crisis that takes long time afterwards to recover. Be aware of this event and its frequency will help the investors more resistant to market shocks and, therefore, make wiser decisions.

5 Appendix

5.1 Re-parameterized GEV

The parameter *quantile* in re-parameterized GEV is indeed a function of the three parameters in original GEV. The following code is to compare the parameter *quantile* in re-parameterized GEV and the one obtained by manual calculation in 1-year, 0.5 year and 3-year return levels, respectively. The results turn out that they are fairly similar together, which supports to the mentioned theory.

```
> gev <- fgev(month.max$nlogreturn) # recall GEV to compare
> gev.1year <- fgev(month.max$nlogreturn, prob = 1/12)
> gev.0.5year <- fgev(month.max$nlogreturn, prob = 1/6)
> gev.3year <- fgev(month.max$nlogreturn, prob = 1/36, std.err = FALSE)
> # standard error can't be estimated because information matrix is singular

> loc <- gev$estimate['loc']
> scale <- gev$estimate['scale']
> shape <- gev$estimate['shape']

> # check re-parameterization in three models
> loc - (scale/shape)*(1-(-log(1-1/12))^(-shape)) # 1 year
```

```
       loc
0.07358201
> gev.1year$estimate['quantile']
  quantile
0.07241005

> loc - (scale/shape)*(1-(-log(1-1/6))^(-shape)) # 0.5 year
       loc
0.05737984
> gev.0.5year$estimate['quantile'
  quantile
0.05844216

> loc - (scale/shape)*(1-(-log(1-1/36))^(-shape)) # 3 years
      loc
0.1026801
> gev.3year$estimate['quantile']
 quantile
0.1213169
```

5.2 Return period - Return level plot

About the Return level curve: The x-axis is not exactly the duration time T but a function of T, which denoted as *period*(T) in the code. Nevertheless, the output of *period*(T) is closed to T especially when T is large. The y-axis is the result of function *qgev* with option $p = \frac{1}{T}$

About the empirical points: The x-axis is again the *period*(T_i) with $T_i = \frac{n+1}{i}$ for n is the sample size and $i = 1, 2, ..., n$. (1 corresponds to the largest value, 2 corresponds to the second-largest value, ...). The y-axis is the value of $x_{(i)}$.

```
# Return period - Return level Plot ========================================
rl(gev) # return level plot

# Check return level curve ------------------------------------------------
rl.1year <- qgev(1/12, gev$estimate["loc"], gev$estimate["scale"],
  gev$estimate["shape"], lower.tail=FALSE) # estimate 1-year return level
period <- function(T){
  return(-1/log(1- 1/(T)))} # x-axis of rl(gev) is not T but period(T)
abline(v=period(12), h=rl.1year, col='red') # Check return level curve

# Check empirical points --------------------------------------------------
largest <- max(month.max$nlogreturn)
second.largest <- sort(month.max$nlogreturn)[nrow(month.max)-1]

abline(v=period(nrow(month.max)+1), h=largest, col='blue') # Check point max
# Check point second max
abline(v=period((nrow(month.max)+1)/2), h=second.largest, col='blue')
```

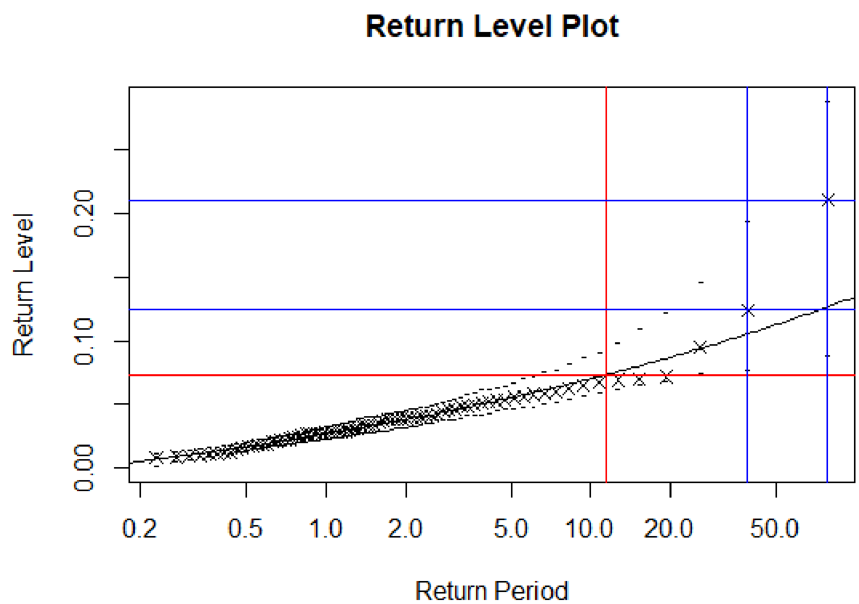

Figure 12: Return level plot

5.3 Smoothing log-likelihood function

If we lessen the time period T, into 6 months for instance, the profile log-likelihood would be smoother than the one in Figure 6.

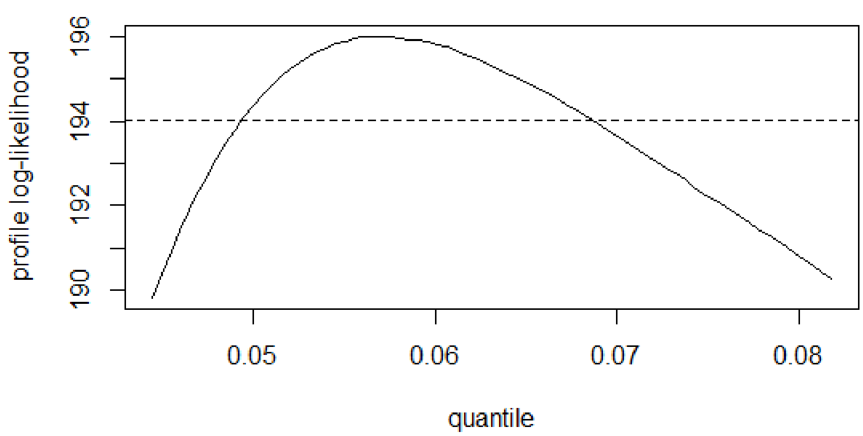

Figure 13: Log-likelihood of 0.5-Year Return Level GEV

5.4 Two versions of GPD

There are three parameters in original GPD: location, scale and shape. Point Process approach in EVT will estimate the three parameters. In our project, however, we have used the other form GPD with two parameters: scale and shape.

The parameter *scale* in two-parameter-GPD is actually a function of the three original parameters in point process and the threshold. The following code compares *scale* in this form and the one obtained by manual calculation from point process' parameters. The similarity of outcomes is supportive to the theory that they are equivalent.

```
> gpd <- fpot(FB$nlogreturn, threshold, npp = 252) # Generalized Pareto
> pp <- fpot(FB$nlogreturn, threshold, npp = 252, model = "pp") # model pp
> # Check relationship of parameters between 2 models
> pp$estimate['scale']+pp$estimate['shape']*(threshold-pp$estimate['loc'])
     scale
0.01193875
> gpd$estimate['scale'] # very similar to method Generalized Pareto
     scale
0.01195393
```

References

[1] Facebook Company Info. (n.d.). Retrieved December 29, 2018, from https://newsroom.fb.com/company-info/

```r
# ===============================================================================
#                              Khanh Truong | Paul Dublanche
#                              Extreme Value Theory - Project
#                                        12/2018
# ===============================================================================

# Import library and data -------------------------------------------------------
rm(list=ls())
library('evd')
library('quantmod')

getSymbols("FB",from='2012-06-01',to='2018-12-01')
head(FB)
tail(FB)
(plot.price <- plot.xts(FB$FB.Adjusted,main='Facebook Stock Price'))

# Prepare data ------------------------------------------------------------------
FB$nlogreturn <- -diff(log(FB$FB.Adjusted)) # calculate negative log-return
(plot.return <- plot.xts(FB$nlogreturn,main='Facebook Negative Log-return'))

FB <- FB[!is.na(FB$nlogreturn),] # remove first row
FB <- data.frame(date=index(FB), coredata(FB)) # transform to dataframe

FB$Year <- as.numeric(substr(FB$date,1,4)) # extract year
FB$Month <- as.numeric(substr(FB$date,6,7)) # extract month
FB$Weekday <- weekdays(FB$date) # extract week-day

month.max <- aggregate(nlogreturn ~ Year+Month, data=FB, FUN=max) # monthly max
summary(month.max$nlogreturn)


### =============================================================================
### BLOCK MAXIMA ================================================================
### =============================================================================

# Fit models ====================================================================
library(MASS) # library for function 'fitdistr'
(normal <- fitdistr(month.max$nlogreturn, densfun="normal")) # Gaussian
(gev <- fgev(month.max$nlogreturn)) # GEV

# Gaussian versus GEV
hist(month.max$nlogreturn, pch=20, breaks=25, prob=TRUE, main=NULL,
     xlab='Negative log-return')
curve(dnorm(x, normal$estimate['mean'], normal$estimate['sd']),col="blue",lwd=2,add=T)
curve(dgev(x, gev$estimate['loc'], gev$estimate['scale'], gev$estimate['shape']),
      col="red", lwd=2, add=T)
legend("topright", legend=c("GEV fit","Gaussian fit"), col=c("red","blue"), lwd=2,
       cex=0.8, box.lty = 0)

# Model checking ================================================================

# Significance of parameters ----------------------------------------------------
```

```
confint(gev) # Symmetric confidence intervel quick way

# Symmetric confidence intervel manual way
c(gev$estimate['loc']-qnorm(0.975)*gev$std.err['loc'],
  gev$estimate['loc']+qnorm(0.975)*gev$std.err['loc'])
c(gev$estimate['scale']-qnorm(0.975)*gev$std.err['scale'],
  gev$estimate['scale']+qnorm(0.975)*gev$std.err['scale'])
c(gev$estimate['shape']-qnorm(0.975)*gev$std.err['shape'],
  gev$estimate['shape']+qnorm(0.975)*gev$std.err['shape']) # contain 0

# Parameter 'shape' -----------------------------------------------------------------
# Wald test
(p_value <- 2*pnorm(abs(gev$estimate['shape'])/gev$std.err['shape'],lower.tail=FALSE))
# likelihood ratio test, H0: 2 models are the same
gumble <- fgev(month.max$nlogreturn, shape=0)
anova(gev,gumble)
# likelihood ratio test by hand
pchisq((abs(gev$deviance-gumble$deviance)), df=1, lower.tail=FALSE)
# => 'shape' is significantly different from 0 => keep gev

# Plot GEV and Gumbel
hist(month.max$nlogreturn, breaks=25, prob=TRUE, main=NULL,
     xlab = 'Negative log-return')
curve(dgev(x, gev$estimate['loc'], gev$estimate['scale'],
           gev$estimate['shape']), col="red", lwd=2, add=T)
curve(dgev(x, gumble$estimate['loc'], gumble$estimate['scale'], 0),
      col="blue", lwd=2, add=T)
legend("topright", legend=c("GEV fit","Gumbel fit"), col=c("red","blue"),
       lwd=2, cex=0.8, box.lty=0)
# GEV fits more, especially at the peak

par(mfrow=c(1,2))
qq(gev, main="QQ Plot of GEV fit")
qq(gumble, main="QQ Plot of Gumbel fit")
# Understandingly pointwise CI of Gumbel is smaller than GEV
# It is because one parameter is fixed
# But Gumble is not fit with large values (underestimate the large values)

qq(gev, xlim=c(0.01,0.03), ylim=c(0,0.06), main="QQ Plot of GEV fit")
qq(gumble, xlim=c(0.01,0.03), ylim=c(0,0.06), main="QQ Plot of Gumble fit")
# zoom in. very alike

# Choose GEV since:
# 1.'shape' is significant different from 0 (wald 0.05, likelihood 0.01)
# 2. As a risk analyst, we are more risk-adverse,
# we choose model which fits more to the worst case (large values)


# Return period - Return level =========================================================
gev <- fgev(month.max$nlogreturn) # recall GEV to compare

gev.1year <- fgev(month.max$nlogreturn, prob = 1/12) # 1-year return level
gev.0.5year <- fgev(month.max$nlogreturn, prob = 1/6) # 0.5-year return level
gev.3year <- fgev(month.max$nlogreturn, prob = 1/36, std.err = FALSE) # 3-year rl
```

```r
# standard error 3-year rl can't be estimated because information matrix is singular

# Check re-parameterization in model gev ------------------------------------------------
loc <- gev$estimate['loc']
scale <- gev$estimate['scale']
shape <- gev$estimate['shape']

# check re-parameterization in three models
loc - (scale/shape)*(1-(-log(1-1/12))^(-shape)) # 1 year
gev.1year$estimate['quantile'] # similar to 'quantile' of re-parameterized

loc - (scale/shape)*(1-(-log(1-1/6))^(-shape))  # 0.5 year
gev.0.5year$estimate['quantile'] # similar to 'quantile' of re-parameterized

loc - (scale/shape)*(1-(-log(1-1/36))^(-shape)) # 3 years
gev.3year$estimate['quantile'] # similar to 'quantile' of re-parameterized

rm(loc,scale,shape)

# Return period - Return level Plot ==================================================
rl(gev,main = NULL) # return level plot

# Check return level curve --------------------------------------------------------------
rl.1year <- qgev(1/12, gev$estimate["loc"], gev$estimate["scale"],
  gev$estimate["shape"], lower.tail=FALSE) # estimate 1-year return level
period <- function(T){
  return(-1/log(1- 1/(T)))} # x-axis of rl(gev) is not T but period(T)
abline(v=period(12), h=rl.1year, col='red') # Check return level curve

# Check empirical points ----------------------------------------------------------------
largest <- max(month.max$nlogreturn)
second.largest <- sort(month.max$nlogreturn)[nrow(month.max)-1]

abline(v=period(nrow(month.max)+1), h=largest, col='blue') # check max
abline(v=period((nrow(month.max)+1)/2), h=second.largest, col='blue') # second max

# Intepretation: 0.07358201  is expected to be exceeded every 1 years.
# Equivalently: There is probability 1/12 (0.083) that
# 0.07358201 is exceeded each month


# Confidence intervel of Return level ================================================
plot(profile(gev.1year, "quantile"), main='Log-likelihood of 1-Year Return Level')
# locator(2)
abline(v=c(0.06125737,0.09350722),col='red') # get C.I from locator(2)
text(0.065,190,'0.061') # add text on the plot
text(0.097,190,'0.094') # add text on the plot
confint(gev.1year)['quantile',] # compare to symmetric C.I

# For appendix: gev.0.5year models obtains more data
# So log-likelihood function would be smoothier.
plot(profile(gev.0.5year, "quantile"), main='Log-likelihood of 6-Month Return Level')
```

```r
### ===============================================================================
### PEAK OVER THRESHOLD ===========================================================
### ===============================================================================
plot.return # stationary

# Choose threshold ================================================================
sum(FB$nlogreturn>0.1,na.rm = TRUE)
sum(FB$nlogreturn>0.1,na.rm = TRUE)/nrow(FB) # very small sample

sum(FB$nlogreturn>0.08,na.rm = TRUE)
sum(FB$nlogreturn>0.08,na.rm = TRUE)/nrow(FB) # small sample

# Plot to select threshold
# All the fixed numbers below are just for nice plot purpose
par(mfrow=c(1,2))
tcplot(FB$nlogreturn,tlim=c(0,0.08),ylims = c(-0.195,0.2),which = 1)
abline(h=0,col='red',xpd=TRUE)

tcplot(FB$nlogreturn,tlim=c(0,0.08),which = 2)
abline(h=0.36,col='red',xpd=TRUE)
abline(v=0.0467,col='red')

threshold <- 0.0467 # select threshold


# Fit GPD distribution ============================================================
(gpd <- fpot(FB$nlogreturn, threshold, npp = 252)) # Generalized Pareto
(pp <- fpot(FB$nlogreturn, threshold, npp = 252, model = "pp")) # model point process

# Check relationship of parameters between 2 models
pp$estimate['scale']+pp$estimate['shape']*(threshold-pp$estimate['loc'])
gpd$estimate['scale'] # very similar to method Generalized Pareto

par(mfrow=c(1,2)) # modeling checking
qq(gpd, main="Complete")
qq(gpd, xlim=c(0.05,0.08), ylim=c(0.04,0.12), main="Zoom-in")



par(mfrow=c(1,1)) # plot histogram and the estimator
exceedances <- FB[FB$nlogreturn>=threshold,'nlogreturn']
hist(exceedances, breaks=10, prob=TRUE,
     main=NULL,xlab = 'Negative log-return',ylim = c(0,40))
curve(dgpd(x,pp$estimate['loc'],pp$estimate['scale'],pp$estimate['shape']),
      col="blue", lwd=2, add=T)
legend("topright", legend=c("GPD fit"), col=c("blue"),
       lwd=2, cex=0.8, box.lty = 0)

# Generalized Pareto 1-year
(gpd.1year <- fpot(FB$nlogreturn, threshold, npp = 252, mper = 1))

# Profile log-likelihood
plot(profile(gpd),'shape')
# locator(2)
```

```r
abline(v=c(0.05151051,0.94064803),col='red') # get C.I from locator(2)
text(0.15,100,'0.052') # add text on the plot
text(1.05,100,'0.941') # add text on the plot

confint(gpd)['shape',]

(p_value <- 2*(1-pnorm(abs(gpd$estimate['shape'])/gpd$std.err['shape']))) # Wald test


# Return period ==================================================================
par(mfrow=c(1,1))

(gpd <- fpot(FB$nlogreturn, threshold, npp = 252))
(gpd.0.5year <- fpot(FB$nlogreturn, threshold, mper = 0.5, npp = 252))
(gpd.1year <- fpot(FB$nlogreturn, threshold, mper = 1, npp = 252))
(gpd.3year <- fpot(FB$nlogreturn, threshold, mper = 3, npp = 252))

plot(profile(gpd.1year),"rlevel")
# locator(2)
abline(v=c(0.06475264,0.09296453),col='red',lty=2) # get C.I from locator(2)
text(0.07,102.5,'0.065') # add text on the plot
text(0.098,102.5,'0.093') # add text on the plot

confint(gpd.1year)['rlevel',]

rl(gpd)


### ============================================================================
### COMPARE GEV AND GPD ========================================================
### ============================================================================
# Compare estimates
gev.0.5year$estimate['quantile']
gpd.0.5year$estimate['rlevel']

gev.1year$estimate['quantile']
gpd.1year$estimate['rlevel']

gev.3year$estimate['quantile']
gpd.3year$estimate['rlevel']

# Compare CI
confint(gev.0.5year)['quantile',]
confint(gpd.0.5year)['rlevel',]

confint(gev.1year)['quantile',]
confint(gpd.1year)['rlevel',]

confint(gev.3year)['quantile',]
confint(gpd.3year)['rlevel',]

### ============================================================================
```