Case study 3

212STG18 예지혜

2021년 3월 20일

Part 1) Kospi 일별수익률의 VAR 계산

```
setwd("C:/Users/JIHYE/Desktop/대학원/이론통계/과제3")
kospi <- read.csv("kospi.csv", skip = 3, nrows = 2472, col.names = c("date", "P"))
kospi$P <- as.numeric(gsub(",","",kospi$P))
kospi$date <- as.Date(kospi$date)
kospi <- na.omit(kospi)
head(kospi)
```

```
##         date       P
## 1 2011-01-03 2070.08
## 2 2011-01-04 2085.14
## 3 2011-01-05 2082.55
## 4 2011-01-06 2077.61
## 5 2011-01-07 2086.20
## 6 2011-01-10 2080.81
```

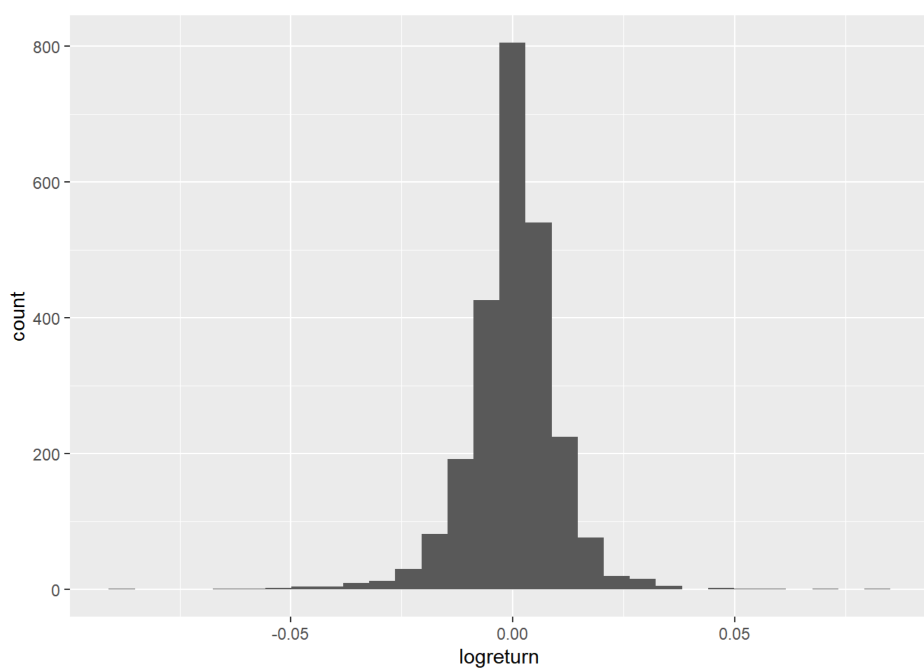
```
summary(kospi)
```

```
##       date                  P
##  Min.   :2011-01-03   Min.   :1458
##  1st Qu.:2013-06-25   1st Qu.:1958
##  Median :2015-12-23   Median :2029
##  Mean   :2015-12-29   Mean   :2081
##  3rd Qu.:2018-07-02   3rd Qu.:2160
##  Max.   :2020-12-30   Max.   :2873
```

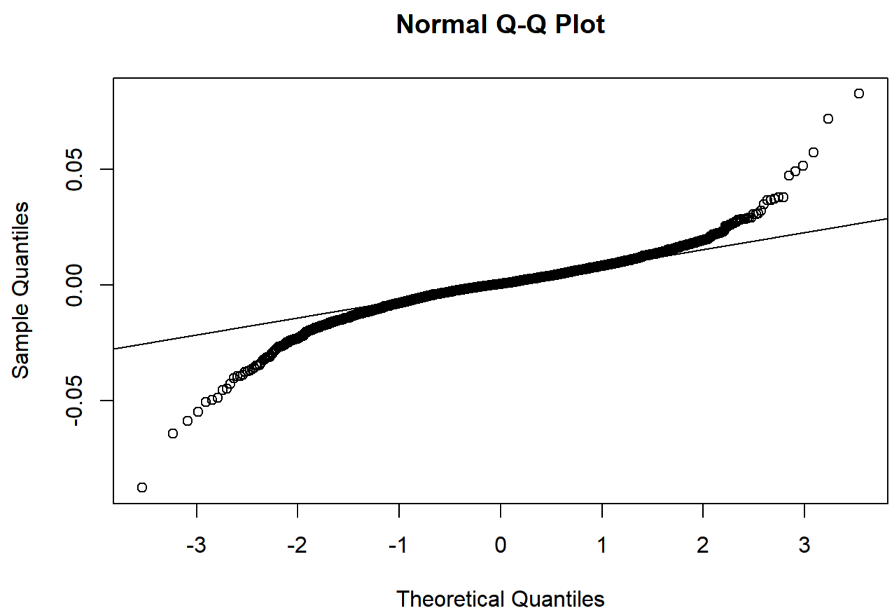
a) log return rt = ln(Pt/Pt-1)의 histogram 및 normal QQplot

```
kospi$return[1] <- NA
for (i in c(2:nrow(kospi))){
  kospi$return[i] <- kospi$P[i]/kospi$P[i-1]
}
kospi$logreturn <- log(kospi$return)
library(ggplot2)
ggplot(kospi[-1,], aes(x=logreturn)) + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
qqnorm(kospi[-1,]$logreturn); qqline(kospi[-1,]$logreturn)
```



꼬리 부분에서 정규성을 이야기하기 어렵다.

b) Logistic, Laplace, T distribution Q-Q plot

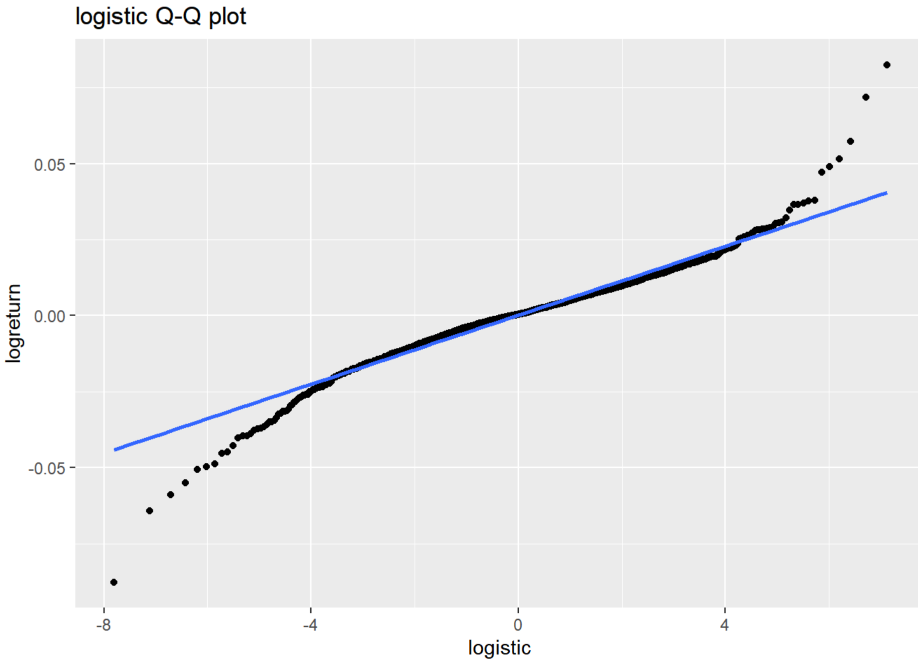
```
kospi$rank <- rank(kospi$logreturn)
kospi$pr <- kospi$rank/(nrow(kospi)+1)
kospi$logistic <- log(kospi$pr/(1-kospi$pr))
library(jmuOutlier)
```

```
## Warning: package 'jmuOutlier' was built under R version 3.6.3
```

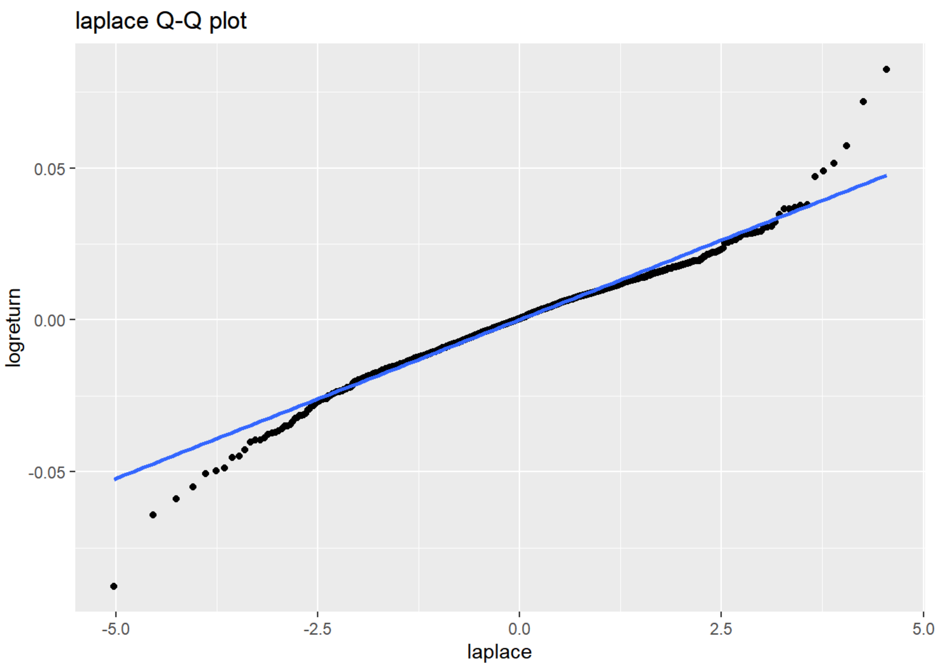
```
kospi$laplace <- qlaplace(kospi$pr)
kospi$t1 <- qt(kospi$pr, 1)
kospi$t2 <- qt(kospi$pr, 2)
kospi$t3 <- qt(kospi$pr, 3)
kospi$t4 <- qt(kospi$pr, 4)

library(gridExtra)
```

```
## Warning: package 'gridExtra' was built under R version 3.6.3
```

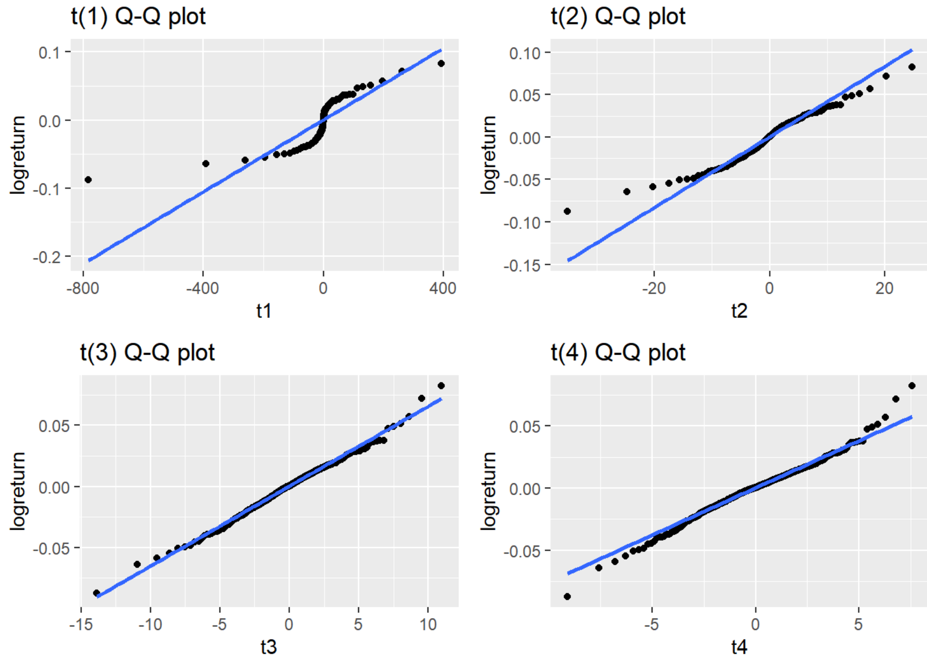
```
ggplot(kospi[-1,], aes(logistic, logreturn)) + geom_point() +
  geom_smooth(method="lm", se=F) + labs(title = "logistic Q-Q plot")
```




```
ggplot(kospi[-1,], aes(laplace, logreturn)) + geom_point() +
  geom_smooth(method="lm", se=F) + labs(title = "laplace Q-Q plot")
```




```
t1 <- ggplot(kospi[-1,], aes(t1, logreturn)) + geom_point() +
  geom_smooth(method="lm", se=F) + labs(title = "t(1) Q-Q plot")
t2 <- ggplot(kospi[-1,], aes(t2, logreturn)) + geom_point() +
  geom_smooth(method="lm", se=F) + labs(title = "t(2) Q-Q plot")
t3 <- ggplot(kospi[-1,], aes(t3, logreturn)) + geom_point() +
  geom_smooth(method="lm", se=F) + labs(title = "t(3) Q-Q plot")
t4 <- ggplot(kospi[-1,], aes(t4, logreturn)) + geom_point() +
  geom_smooth(method="lm", se=F) + labs(title = "t(4) Q-Q plot")
grid.arrange(t1,t2,t3,t4, ncol = 2)
```



t(3) 분포가 가장 적합해보인다.

VaR 계산 : VaR = - mu - sigma * qt

```
alpha <- c(0.01, 0.004, 0.001, 0.0004)
VaR <- -mean(kospi$logreturn, na.rm=T)-sd(kospi$logreturn, na.rm=T)*qt(alpha,3)
cbind(alpha, VaR)
```

```
##      alpha        VaR
## [1,] 1e-02 0.04735186
## [2,] 4e-03 0.06598109
## [3,] 1e-03 0.10668675
## [4,] 4e-04 0.14560161
```

c) GPD 적합

```
library(dplyr)
```

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

```
## The following object is masked from 'package:gridExtra':
##
##     combine
```

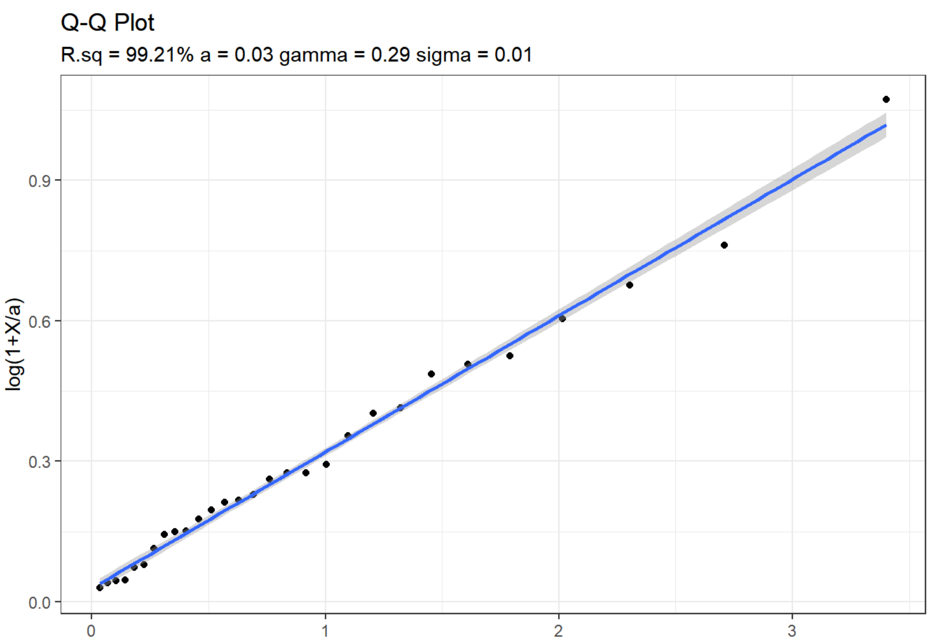
```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

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

```
kospi <- read.csv("kospi.csv", skip = 3, nrows = 2472, col.names = c("date", "P"))
kospi$P <- as.numeric(gsub(",","",kospi$P))
kospi$date <- as.Date(kospi$date)
kospi <- na.omit(kospi)
kospi <- kospi %>% mutate(rt=log(P/lag(P))) %>% mutate(Loss=-rt, X=Loss-0.03) %>% filter(X>0)
kospi <- kospi %>% mutate(rank=rank(X), pr=rank/(nrow(kospi)+1), exp=qexp(pr))
```

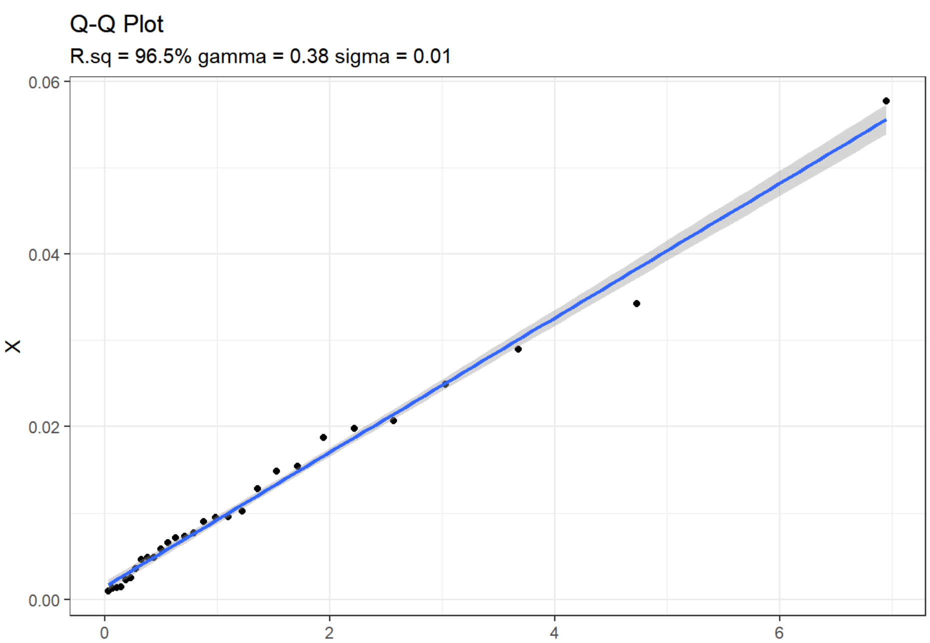
방법 1.

```
method1 <- function(a){
  method1.lm <- lm(log(1+X/a)~exp, data=kospi)
  gamma.hat <- method1.lm$coefficients[2]
  r2 <- summary(method1.lm)$r.squared
  return(c(gamma.hat, r2))
}

a.matrix <- data.frame("a"=seq(0.01, 0.5, by=0.01), "gamma"=NA, "r2"=NA)

for (i in 1:nrow(a.matrix)){
  a.matrix[i,2:3] <- method1(a.matrix[i,1])
}

final.a <- a.matrix[which.max(a.matrix$r2),1]
final.gamma <- a.matrix[which.max(a.matrix$r2),2]
final.sigma <- final.a*final.gamma

ggplot(kospi, aes(exp, log(1+X/final.a))) + geom_point() +
  geom_smooth(method="lm", se=T) +
  labs(title = "Q-Q Plot",
       subtitle = paste("R.sq = ", round(a.matrix[which.max(a.matrix$r2),3]*100,2), "%",
                   " a = ", final.a,
                   " gamma = ", round(final.gamma,2),
                   " sigma = ", round(final.sigma,2), sep=""),
       x="", y="log(1+X/a)") + theme_bw()
```



방법 2.

```
method2 <- function(gamma){
  kospi <- kospi %>% mutate(g.inverse=((1-pr)^(-gamma)-1)/gamma)
  method2.lm <- lm(X ~ g.inverse, data=kospi)
  gamma.hat <- method2.lm$coefficients[2]
  r2 <- summary(method2.lm)$r.squared
  return(c(gamma.hat, r2))
}

gamma.matrix <- data.frame("gamma"=seq(0.01, 1, by=0.01), "slope"=NA, "r2"=NA)

for (i in 1:nrow(gamma.matrix)){
  gamma.matrix[i,2:3] <- method2(gamma.matrix[i,1])
}
final.gamma <- gamma.matrix[which.max(gamma.matrix$r2),1]
final.sigma <- gamma.matrix[which.max(gamma.matrix$r2),2]

kospi <- kospi %>% mutate(g.inverse=((1-pr)^(-final.gamma)-1)/final.gamma)
ggplot(kospi, aes(g.inverse, X)) + geom_point() +
  geom_smooth(method="lm", se=T) +
  labs(title = "Q-Q Plot",
       subtitle = paste("R.sq = ", round(a.matrix[which.max(gamma.matrix$r2),3]*100,2), "%",
                        " gamma = ", round(final.gamma,2),
                        " sigma = ", round(final.sigma,2), sep=""),
       x="", y="X") + theme_bw()
```



방법 3.

```
library(optimx)
```

```
## Warning: package 'optimx' was built under R version 3.6.3
```

```
mle.fun <- function(para){
  sum(log((1/para[2])*((1+para[1]*kospi$X/para[2])^(-1/para[1]-1))), na.rm=T)
}
optim.result <-optim(par=c(final.gamma, final.sigma),mle.fun,method="CG",hessian=T,control=list(fnscale=-1))
data.frame(gamma.hat = optim.result$par[1], sigma.hat = optim.result$par[2])
```

```
##   gamma.hat   sigma.hat
## 1 0.3799621 0.009309995
```

Part 2) 인간의 수명분포

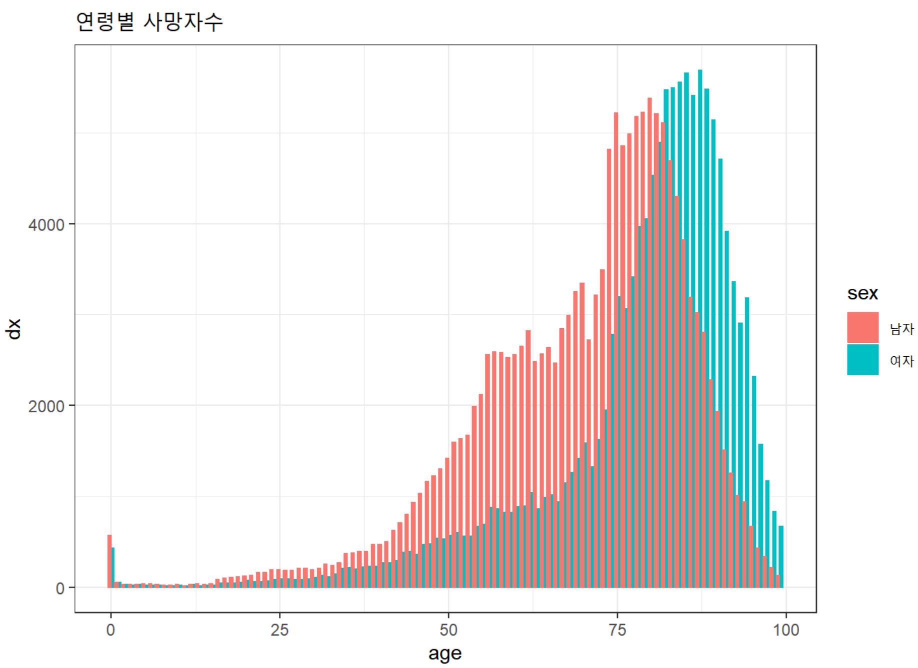
a) 연령별 성별 사망자수 그래프

```
age <- read.csv("age.csv")
age <- age[,-1]
names(age) <- c("age", "sex", "dx")
age$age <- as.numeric(gsub("세", "", age$age))
head(age)
```

```
##   age  sex  dx
## 1   0 남자 569
## 2   0 여자 431
## 3   1 남자  50
## 4   1 여자  47
## 5   2 남자  31
## 6   2 여자  26
```

```
ggplot(age, aes(age, dx, col = sex, fill = sex)) +
  geom_histogram(stat="identity", position="dodge") + theme_bw() +
  labs(title = "연령별 사망자수")
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



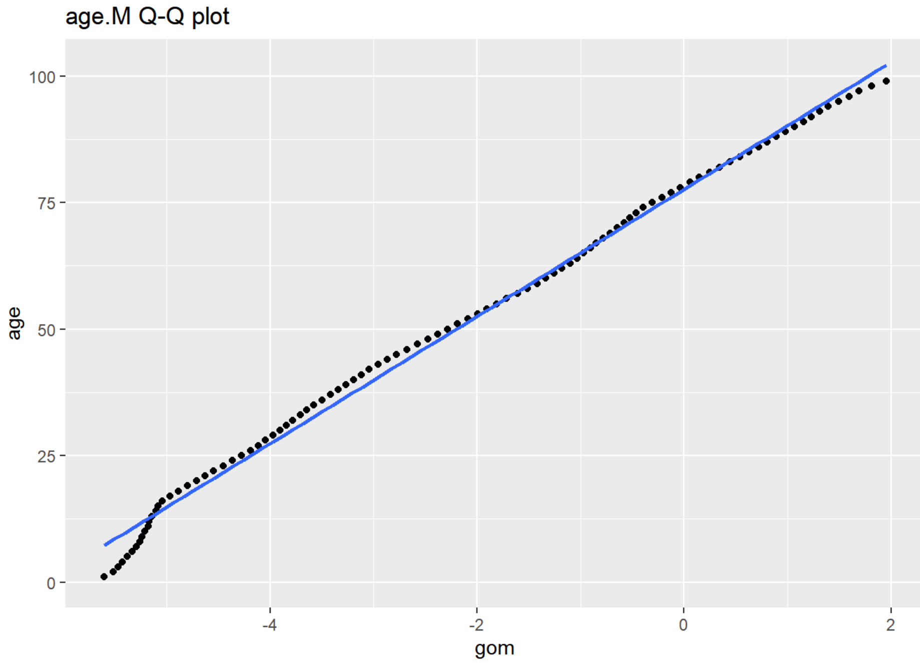
남성이 여성보다 이른 나이에 사망한다. 또한 태어난 직후를 제외하곤, 7~80대까지 사망자 수가 꾸준히 증가한다.

b) gompertz Q-Q plot

```
age.M <- age[age$sex=="남자",]
age.F <- age[age$sex=="여자",]
age.M <- age.M %>% mutate(new.dx=cumsum(dx), px=lag(new.dx)/sum(dx), gom=log(-log(1-px)))
age.F <- age.F %>% mutate(new.dx=cumsum(dx), px=lag(new.dx)/sum(dx), gom=log(-log(1-px)))
ggplot(age.M, aes(gom, age)) + geom_point() +
  geom_smooth(method="lm", se=F) + labs(title = "age.M Q-Q plot")
```

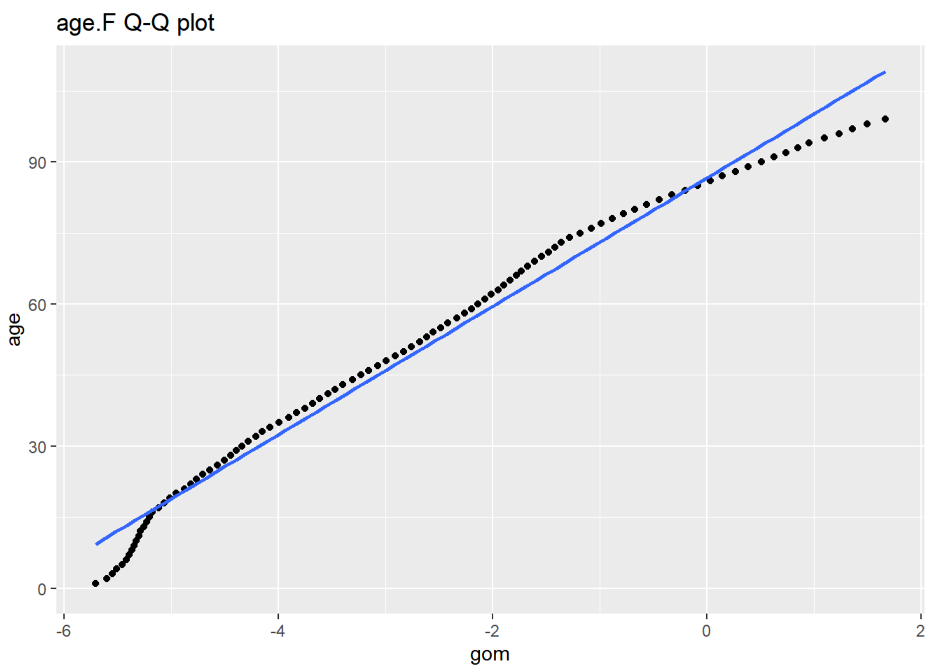
```
## Warning: Removed 1 rows containing non-finite values (stat_smooth).
```

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




```
ggplot(age.F, aes(gom, age)) + geom_point() +
  geom_smooth(method="lm", se=F) + labs(title = "age.F Q-Q plot")
```

```
## Warning: Removed 1 rows containing non-finite values (stat_smooth).

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




```
lm.M <- lm(age~gom, age.M)
(mu.M <- lm.M$coefficients[1])
```

```
## (Intercept)
##    77.65008
```

```
(sigma.M <- lm.M$coefficients[2])
```

```
##      gom
## 12.57082
```

```
lm.F <- lm(age~gom, age.F)
(mu.F <- lm.F$coefficients[1])
```

```
## (Intercept)
##    86.64412
```

```
(sigma.F <- lm.F$coefficients[2])
```

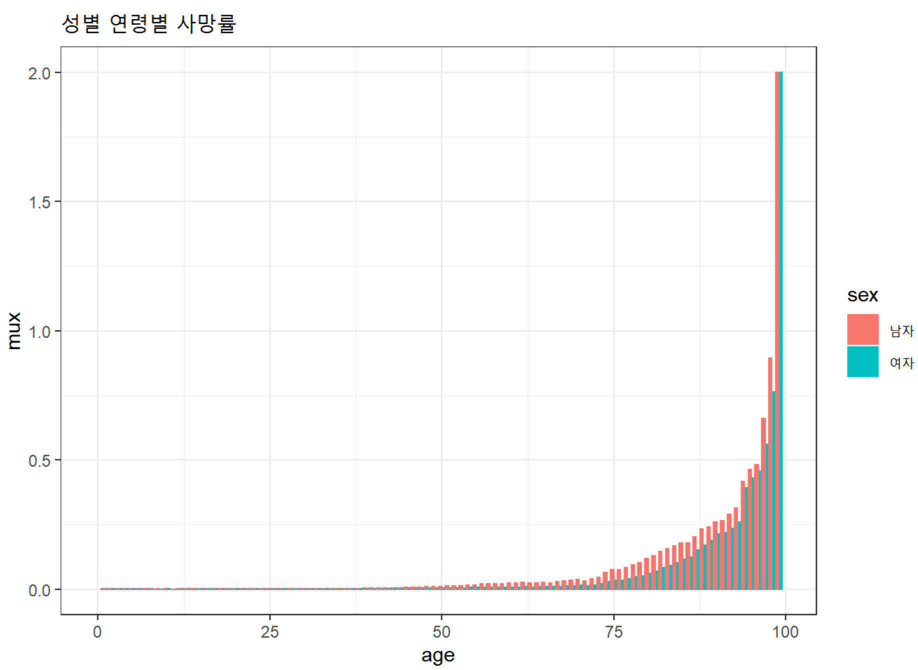
```
##      gom
## 13.55266
```

c) mux, ln(mux)

```
age.M <- age.M %>% mutate(lx = sum(dx) - lag(new.dx), qx = dx/lx, mux = dx/(lx-dx/2))
age.F <- age.F %>% mutate(lx = sum(dx) - lag(new.dx), qx = dx/lx, mux = dx/(lx-dx/2))

ggplot(rbind(age.M, age.F), aes(age, mux, col = sex, fill = sex)) +
  geom_histogram(stat="identity", position="dodge") + theme_bw() +
  labs(title = "성별 연령별 사망률")
```

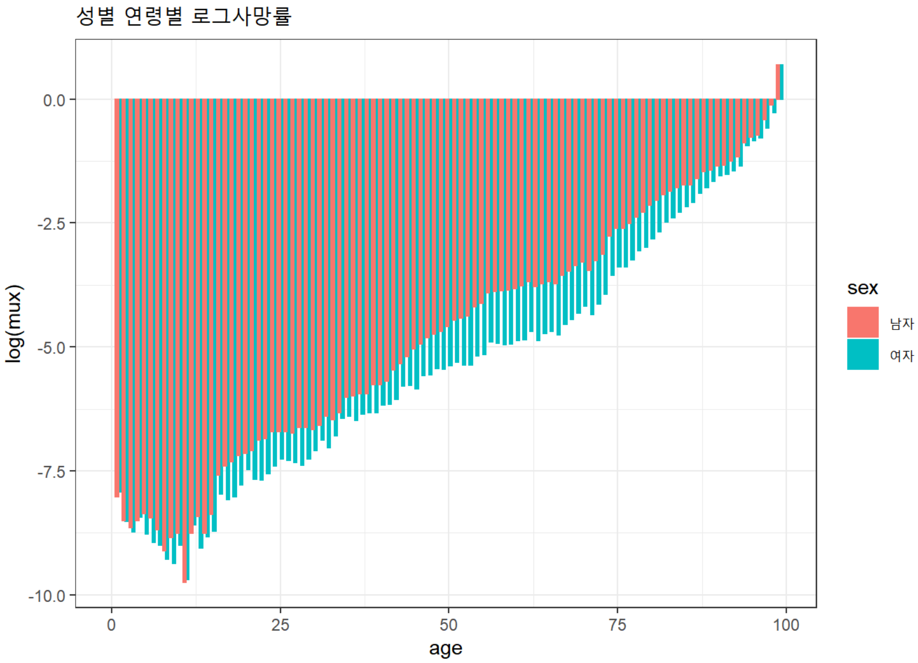
```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```




```
ggplot(rbind(age.M, age.F), aes(age, log(mux), col = sex, fill = sex)) +
  geom_histogram(stat="identity", position="dodge") + theme_bw() +
  labs(title = "성별 연령별 로그사망률")
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad

## Warning: Removed 2 rows containing missing values (geom_bar).
```




```
lm.M2 <- lm(log(mux) ~ age, age.M)
a.M2 <- lm.M2$coefficients[1]
b.M2 <- lm.M2$coefficients[2]

lm.F2 <- lm(log(mux) ~ age, age.F)
a.F2 <- lm.F2$coefficients[1]
b.F2 <- lm.F2$coefficients[2]

df <- data.frame(a=c(a.F2, a.M2), b=c(b.F2, b.M2), row.names = c("F", "M"))
df
```

```
##           a          b
## F -9.641270 0.08610494
## M -9.173373 0.08795627
```

Part 3) 기상변수 극값 분포 및 홍수피해액 분포

```
rain1 <- read.csv("rain1.csv", skip = 8)
rain2 <- read.csv("rain2.csv", skip = 8)
rain <- rbind(rain1, rain2)
rain <- rain[,c(5,6,7)]
names(rain) <- c("year", "date", "x")
rain$date <- as.Date(rain$date)
rain <- na.omit(rain)
head(rain)
```

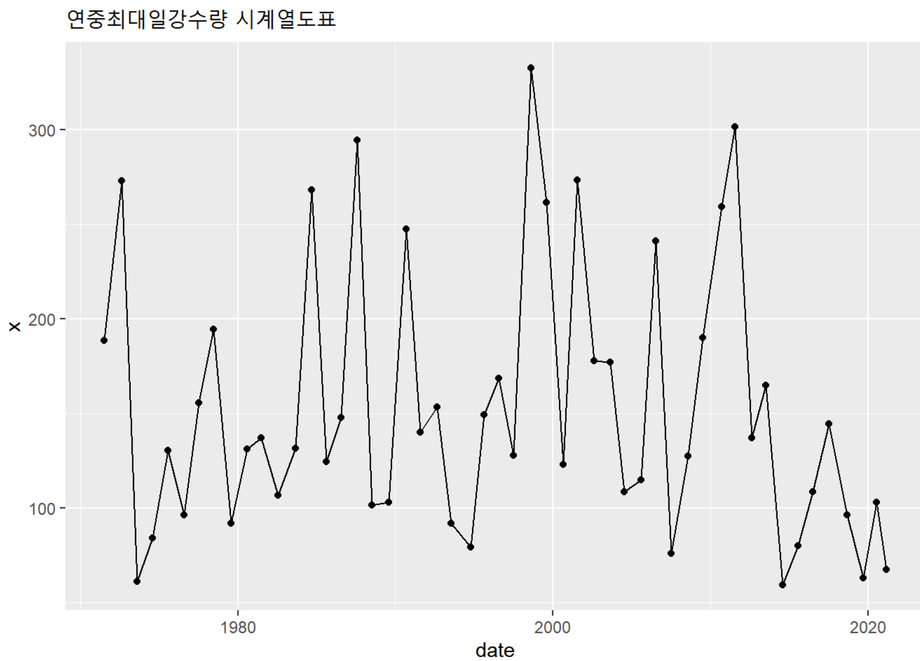
```
##   year       date     x
## 1 1998 1998-08-08 332.8
## 2 1987 1987-07-27 294.6
## 3 1972 1972-08-19 273.2
## 4 1984 1984-09-01 268.2
## 5 1999 1999-08-02 261.6
## 6 1990 1990-09-11 247.5
```

```
flood <- read.csv("flood.csv")
flood <- flood[,c(1,2)]
names(flood) <- c("year", "damage")
head(flood)
```

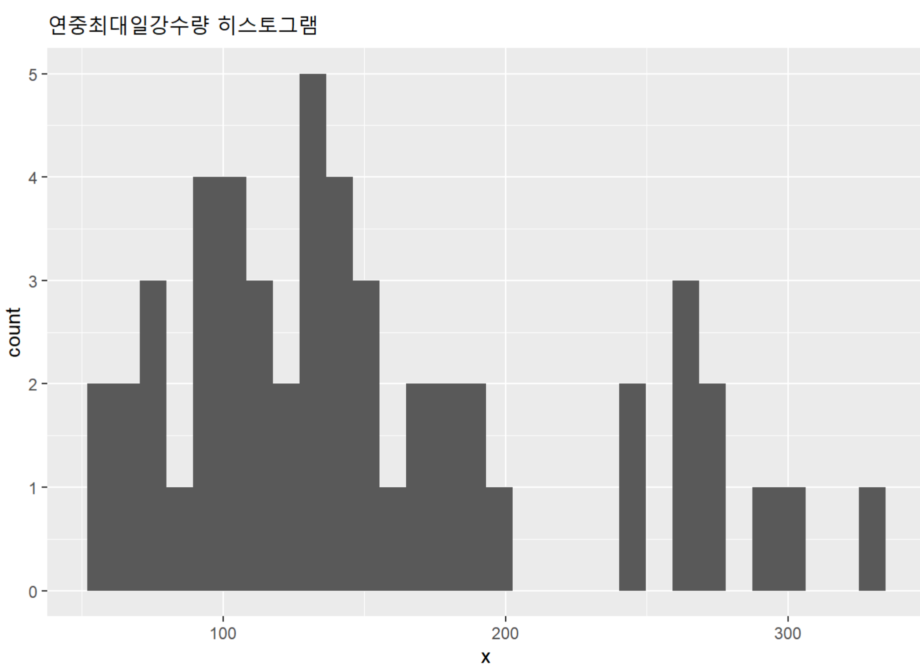
```
##   year  damage
## 1 1971  119191
## 2 1972 1236611
## 3 1973    8872
## 4 1974   10170
## 5 1975    3265
## 6 1976   52034
```

a) 연중최대일강수량{x} 자료의 시계열도표 및 histogram

```
ggplot(rain, aes(date, x)) + geom_point() + geom_line() + labs(title = "연중최대일강수량 시계열도표")
```

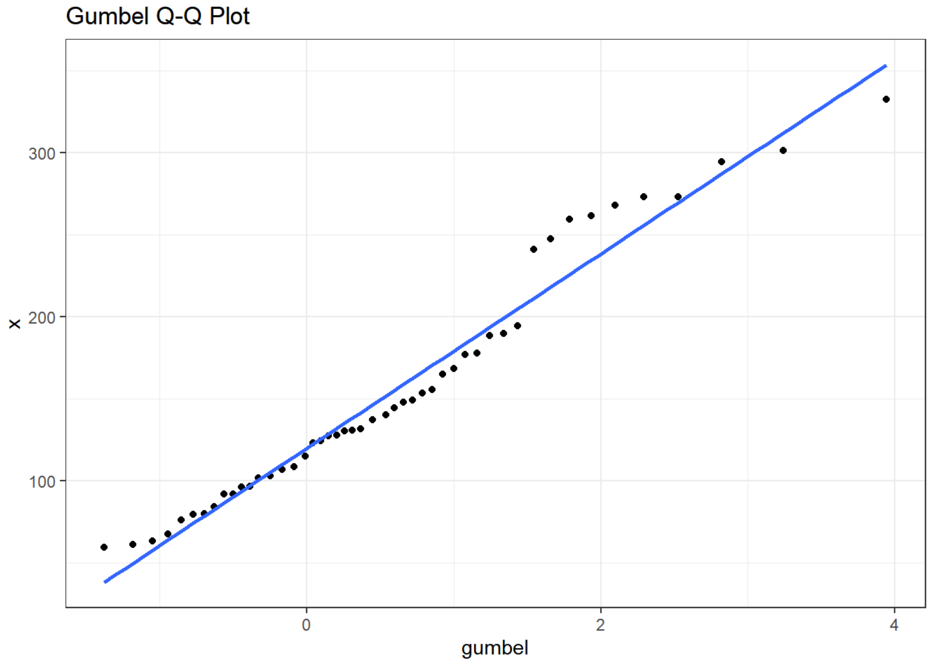



```
ggplot(rain, aes(x)) + geom_histogram(bins = 30) + labs(title ="연중최대일강수량 히스토그램")
```



b) Gumbel 최대극값분포

```
rain <- rain %>% mutate(rank = rank(x), pr=rank/(nrow(rain)+1), gumbel=-log(-log(pr)))
ggplot(rain, aes(gumbel, x)) + geom_point() +
  geom_smooth(method="lm", se=F) +
  labs(title = "Gumbel Q-Q Plot") + theme_bw()
```




```
lm.1 <- lm(x ~ gumbel, rain)
mu <- lm.1$coefficients[1]; mu
```

```
## (Intercept)
##    119.7599
```

```
sigma <- lm.1$coefficients[2]; sigma
```

```
##   gumbel
## 59.36668
```

c) T년 최대 강수량

```
t <- c(50,100,200)
recent <- data.frame(t, mut = mu + sigma*(-log(-log(1-1/t))))
recent
```

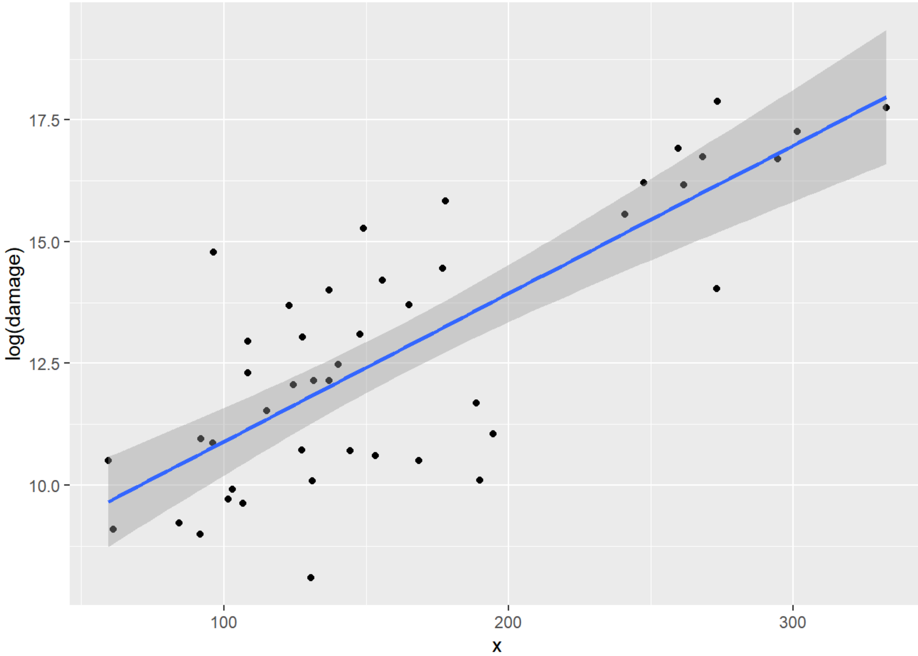
```
##     t      mut
## 1  50 351.4051
## 2 100 392.8555
## 3 200 434.1547
```

d) (lndamage, x)의 회귀 적합

```
df <- merge(rain, flood, by="year")
head(df)
```

```
##   year       date     x rank         pr     gumbel  damage
## 1 1971 1971-07-17 188.6   39 0.75000000  1.2458993  119191
## 2 1972 1972-08-19 273.2   47 0.90384615  2.2916836 1236611
## 3 1973 1973-08-17  61.2    2 0.03846154 -1.1811431    8872
## 4 1974 1974-08-03  84.3    8 0.15384615 -0.6269017   10170
## 5 1975 1975-07-25 130.5   24 0.46153846  0.2572306    3265
## 6 1976 1976-08-13  96.2   11 0.21153846 -0.4404129   52034
```

```
ggplot(df[df$damage>0,], aes(x, log(damage))) + geom_point() + geom_smooth(method="lm", se=T)
```




```
lm.df <- lm(log(damage) ~ x, df[df$damage>0,])
summary(lm.df)
```

```
##
## Call:
## lm(formula = log(damage) ~ x, data = df[df$damage > 0, ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7286 -1.2338  0.2919  1.1531  3.9943
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 7.849954   0.653262  12.017 2.46e-15 ***
## x           0.030419   0.003708   8.204 2.45e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.708 on 43 degrees of freedom
## Multiple R-squared:  0.6101, Adjusted R-squared:  0.6011
## F-statistic:  67.3 on 1 and 43 DF,  p-value: 2.445e-10
```

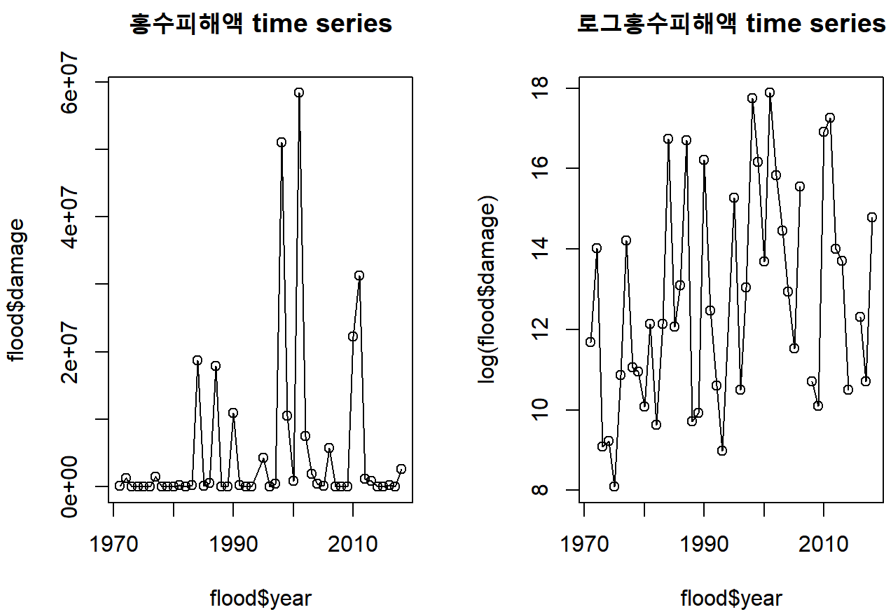
```
data.frame(a=lm.df$coefficients[1], b=lm.df$coefficients[2])
```

```
##                    a          b
## (Intercept) 7.849954 0.03041901
```

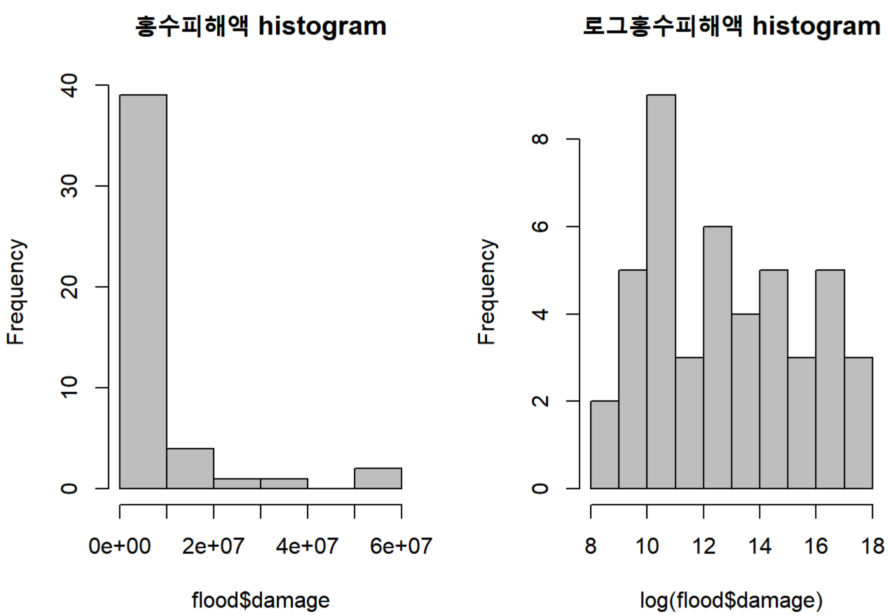
최다 강수량과 로그 홍수피해액이 선형관계에 있음을 의미한다.

e) 홍수피해액 y와 lny의 시계열 도표 및 histogram, frechet 분포 QQ plot

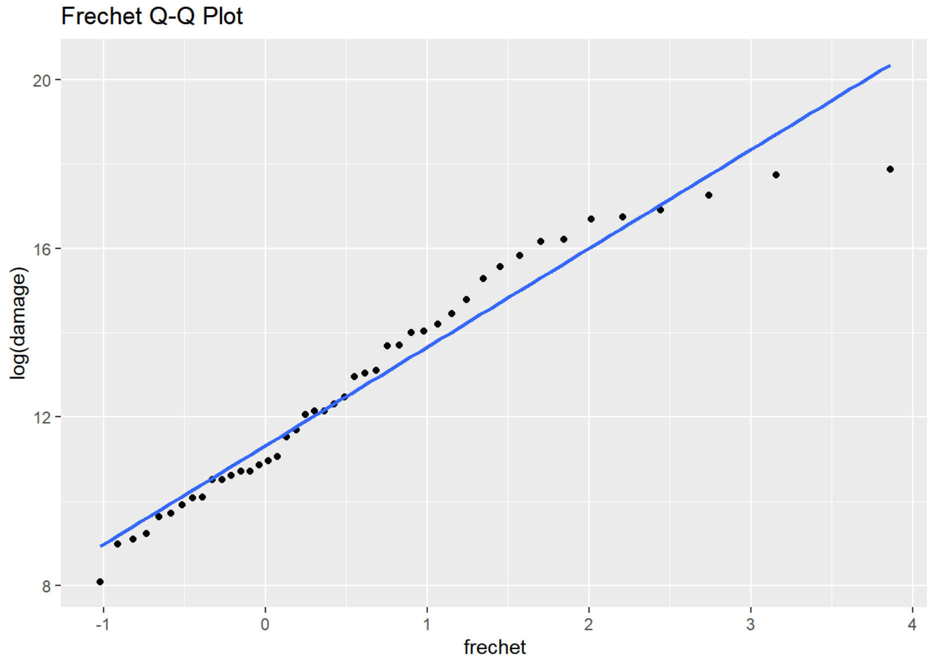
```
par(mfrow = c(1,2))
plot(flood$year, flood$damage, main = "홍수피해액 time series"); lines(flood$year, flood$damage)
plot(flood$year, log(flood$damage), main = "로그홍수피해액 time series"); lines(flood$year, log(flood$damage))
```




```
hist(flood$damage, main = "홍수피해액 histogram", col = "grey")
hist(log(flood$damage), main = "로그홍수피해액 histogram", col = "grey")
```




```
flood <- flood %>% mutate(rank = rank(log(damage)), pr = rank/(nrow(flood)+1), frechet = -log(-log(pr)))
ggplot(flood[flood$damage>0,], aes(frechet, log(damage)))+ geom_point() +
  geom_smooth(method="lm", se=F) + labs(title = "Frechet Q-Q Plot")
```




```
lm.f <- lm(log(damage)~frechet, flood[flood$damage>0,])
data.frame(mu = lm.f$coefficients[1], sigma = lm.f$coefficients[2])
```

```
##                   mu    sigma
## (Intercept) 11.31558 2.342298
```

Part 4) EVT에 근거한 추론의 타당성 검토

```
library(readxl)
```

```
## Warning: package 'readxl' was built under R version 3.6.3
```

```
seoul <- read_excel("서울일최대강우량-1777-2019.xls")
seoul <- data.frame(seoul)
names(seoul) <- c("years", "old_new", "rainfall")
seoul <- na.omit(seoul)
seoul$rainfall <- as.numeric(seoul$rainfall)
```

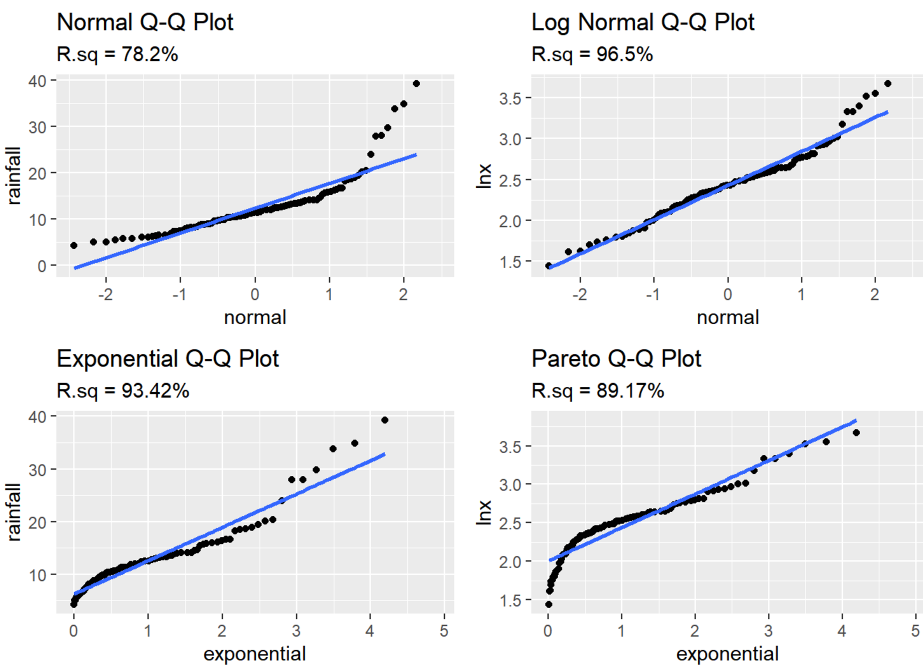
```
## Warning: 강제형변환에 의해 생성된 NA 입니다
```

```
head(seoul)
```

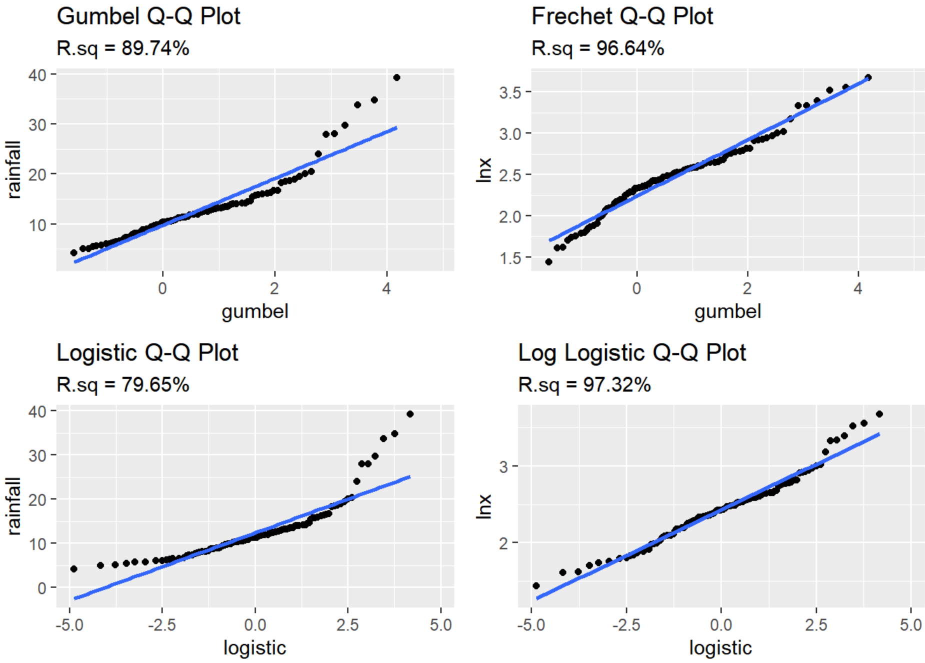
```
##   years old_new rainfall
## 1  1777       1     5.05
## 2  1778       1     6.32
## 3  1779       1    15.68
## 4  1780       1    12.79
## 5  1781       1     8.58
## 6  1782       1    18.68
```

a) 조선시대 자료의 분포 찾기

```
joseon <- seoul[seoul$old_new==1,]
joseon <- joseon %>% mutate(lnx=log(rainfall), rank=rank(rainfall), pr=rank/(nrow(joseon)+1), normal=qnorm(pr), exponential = -log(1-pr), logistic=log(pr/(1-pr)), gumbel=-log(-log(pr)))
normal <- ggplot(joseon, aes(normal, rainfall)) + geom_point() + geom_smooth(method="lm", se=F) +
  labs(title = "Normal Q-Q Plot", subtitle = paste("R.sq = ", round(summary(lm(rainfall ~ normal, joseon))$r.squared,4)*100, "%", sep = ""))
lognormal <- ggplot(joseon, aes(normal, lnx)) + geom_point() + geom_smooth(method="lm", se=F) +
  labs(title = "Log Normal Q-Q Plot", subtitle = paste("R.sq = ", round(summary(lm(lnx ~ normal, joseon))$r.squared,4)*100, "%", sep = ""))
exponential <- ggplot(joseon, aes(exponential, rainfall)) + geom_point() + geom_smooth(method="lm", se=F) +
  labs(title = "Exponential Q-Q Plot", subtitle = paste("R.sq = ", round(summary(lm(rainfall ~ exponential, joseon))$r.squared,4)*100, "%", sep = ""))
pareto <- ggplot(joseon, aes(exponential, lnx)) + geom_point() + geom_smooth(method="lm", se=F) +
  labs(title = "Pareto Q-Q Plot", subtitle = paste("R.sq = ", round(summary(lm(lnx ~ exponential, joseon))$r.squared,4)*100, "%", sep = ""))
gumbel <- ggplot(joseon, aes(gumbel, rainfall)) + geom_point() + geom_smooth(method="lm", se=F) +
  labs(title = "Gumbel Q-Q Plot", subtitle = paste("R.sq = ", round(summary(lm(rainfall ~ gumbel, joseon))$r.squared,4)*100, "%", sep = ""))
frechet <- ggplot(joseon, aes(gumbel, lnx)) + geom_point() + geom_smooth(method="lm", se=F) +
  labs(title = "Frechet Q-Q Plot", subtitle = paste("R.sq = ", round(summary(lm(lnx ~ gumbel, joseon))$r.squared,4)*100, "%", sep = ""))
logistic <- ggplot(joseon, aes(logistic, rainfall)) + geom_point() + geom_smooth(method="lm", se=F) +
  labs(title = "Logistic Q-Q Plot", subtitle = paste("R.sq = ", round(summary(lm(rainfall ~ logistic, joseon))$r.squared,4)*100, "%", sep = ""))
loglogistic <- ggplot(joseon, aes(logistic, lnx)) + geom_point() + geom_smooth(method="lm", se=F) +
  labs(title = "Log Logistic Q-Q Plot", subtitle = paste("R.sq = ", round(summary(lm(lnx ~ logistic, joseon))$r.squared,4)*100, "%", sep = ""))

grid.arrange(normal, lognormal, exponential, pareto, ncol=2)
```




```
grid.arrange(gumbel, frechet, logistic, loglogistic, ncol=2)
```



Log logistic 분포가 가장 적합해보인다.

```
lm.jo <- lm(lnx~ logistic, joseon)
mu <- lm.jo$coefficients[1]
sigma <- lm.jo$coefficients[2]

t <- c(50, 100, 200)
pr <- 1- 1/t
past <- data.frame(t, ut=exp(mu + sigma*(log(pr/(1-pr)))))
past
```

```
##     t       ut
## 1  50 28.56092
## 2 100 33.75021
## 3 200 39.83395
```

```
recent$mut_cm <- recent$mut/10
recent; past
```

```
##     t      mut   mut_cm
## 1  50 351.4051 35.14051
## 2 100 392.8555 39.28555
## 3 200 434.1547 43.41547
```

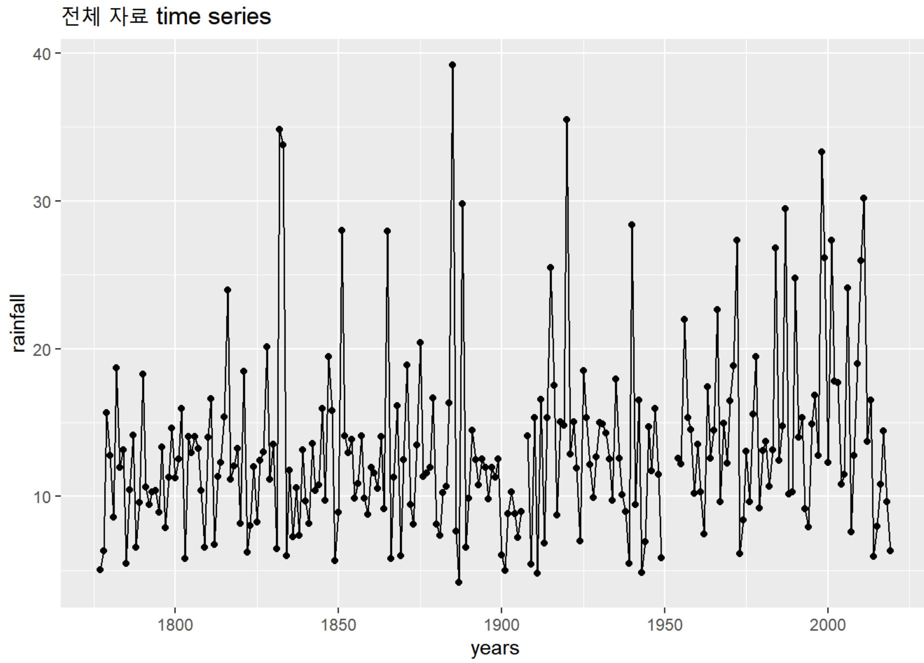
```
##     t       ut
## 1  50 28.56092
## 2 100 33.75021
## 3 200 39.83395
```

최근 자료의 경우 단위가 mm이므로 cm로 바꿔준 것과 비교해보면, 과거에 비해 최근의 강수량이 더 많았다고 해석할 수 있다. 이는 지구 온난화 등 기후 요인에 의해 강수량의 증가추세를 띠고 있다고 해석할 수 있다.

c) 전체 자료

```
ggplot(seoul, aes(years, rainfall)) + geom_point() + geom_line() + labs(title = "전체 자료 time series")
```

```
## Warning: Removed 5 rows containing missing values (geom_point).
```



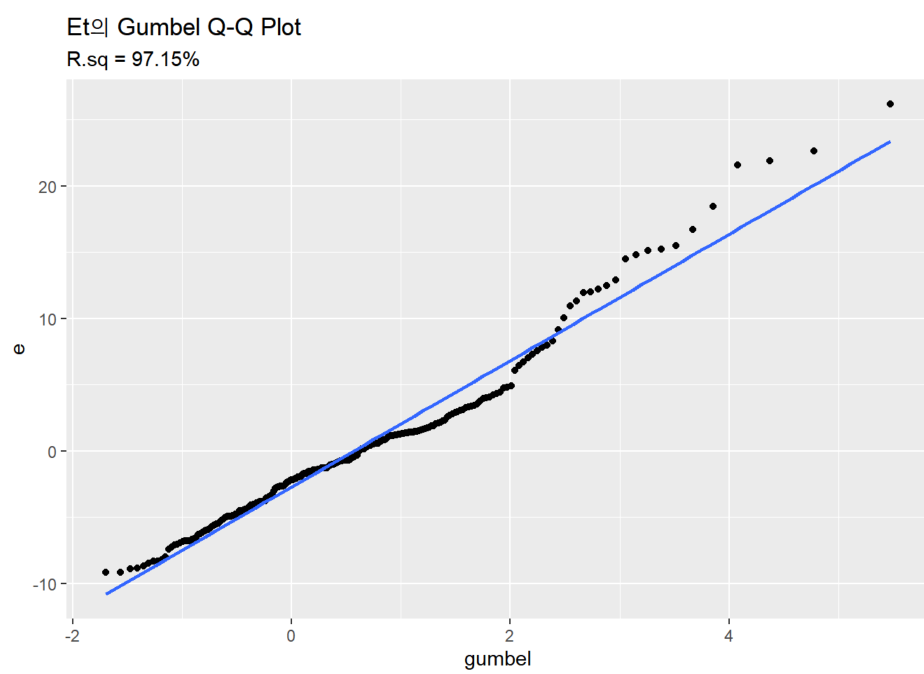

```
seoul$t <- seoul$years - seoul$years[1]
lm.seoul <- lm(rainfall ~ t, seoul)
alpha.hat <- lm.seoul$coefficients[1]
beta.hat <- lm.seoul$coefficients[2]
data.frame(alpha = alpha.hat, beta = beta.hat)
```

```
##                alpha       beta
## (Intercept) 11.32235 0.01591076
```

기울기가 0보다 크며 0.05 수준에서 유의하므로 미세하게 증가하고 있다고 해석할 수 있다.

d) 잔차의 Gumbel 분포 적합

```
et <- data.frame(e = lm.seoul$residuals)
et <- et %>% mutate(rank = rank(e), pr = rank/(nrow(et)+1), gumbel = -log(-log(pr)))
ggplot(et, aes(gumbel, e)) + geom_point() + geom_smooth(method="lm", se=F) +
  labs(title = "Et의 Gumbel Q-Q Plot", subtitle = paste("R.sq = ", round(summary(lm(e ~ gumbel, et))$r.squared,4)*100, "%", sep
= ""))
```



```
lm.et <- lm(e ~ gumbel, et)
mu.hat <- lm.et$coefficients[1]
sigma.hat <- lm.et$coefficients[2]
data.frame(mu = mu.hat, sigma = sigma.hat)
```

```
##                    mu    sigma
## (Intercept) -2.710168 4.767673
```

e) 2019년 일 최대 강우량이 30cm 이상일 확률

```
x.2019 <- alpha.hat + beta.hat*(2019-seoul$years[1]) + mu.hat
prob.hat <- 1 - exp(-exp(-(30-x.2019)/sigma.hat))
data.frame(Probability = paste(round(prob.hat*100,2), "%"), row.names = "")
```

```
##  Probability
##       2.49 %
```

d) mle

```
mle.function <- function(variable, t=(2019-seoul$years[1])) {
  alpha <- variable[1]
  beta <- variable[2]
  sigma <- variable[3]
  ut <- (seoul$rainfall - alpha - beta * t)/sigma
  return(sum(log(exp(-ut-exp(-ut)))- log(sigma), na.rm = T))
}
library(optimx)
final <- optim(par=c(alpha.hat, beta.hat, sigma.hat), mle.function, method="BFGS", hessian=T, control=list(fnscale=-1))
mle.alpha <- final$par[1]
mle.beta <- final$par[2]
mle.sigma <- final$par[3]

mle.2019 <- mle.alpha + mle.beta*(2019-seoul$years[1])
mle.prob <- 1 - exp(-exp(-(30-mle.2019)/mle.sigma))
data.frame(Probability = paste(round(mle.prob*100,2), "%"), row.names = "")
```

```
##  Probability
##        100 %
```

```
data.frame(alpha = c(alpha.hat, mle.alpha), beta = c(beta.hat, mle.beta),
           sigma = c(sigma.hat, mle.sigma), prob = c(prob.hat, mle.prob), row.names = c("method.e", "mle"))
```

```
##             alpha        beta      sigma       prob
## method.e 11.32235  0.01591076   4.767673 0.02494691
## mle      11.34296 -0.18173855 -24.233972 0.99999826
```

probability에는 차이가 조금 있지만, 대부분 비슷하게 추정되었다.