Finance

Barbara, Frederik, Sierra

26/11/2021

Stocks: Airlines:
Lufthansa (German); DLAKY
Southwest Airlines; LUV
Easy Jet (British); EZJ
RyanAir Holdings plc (British); RYAAY
United Airlines Holdings Inc (American); UAL
American Airlines Group Inc (American); AAL
Spirit Airlines Inc (American); SAVE

Beer Industry:
Carlsberg (Danish); CABGY Heineken (Netherlands); HEINY
Anheuser-Busch InBev (Belgium); BUD
Harboe (Danish); CPH: HARB-B
Guinness from Diageo; DEO
Molson Coors Brewing; TAP Tsingtao Brewery Group; OTCMKTS: TSGTF Asahi Group Holding Ltd; OTCMKTS: ASBRF

```
returns <- tq_get(tickers, get="stock.prices") %>%
  group_by(symbol) %>%
  tq_transmute(select=adjusted,
               mutate_fun=periodReturn,
               period="monthly",
               col_rename = "monthly_return")

returns <- returns %>%
  pivot_wider(names_from = symbol, values_from = monthly_return)

#monthly returns
returns <- returns[1:131,]
```

Part 1 1. Table of mean, sd, skewness, kurtosis and beta.

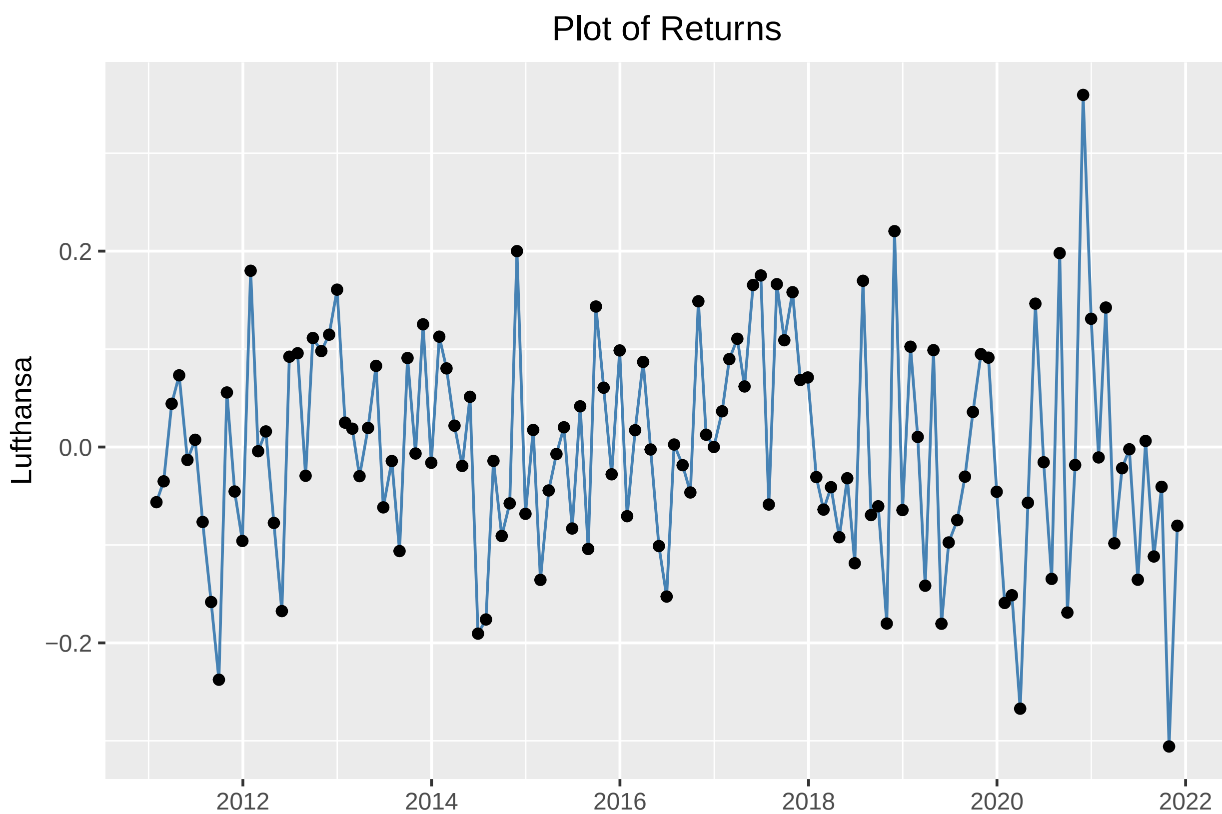
```
returns <- as.data.frame(returns)
returnsNA <- as.data.frame(na.omit(returns))
returnsIVV <- returns[1:16]

# columns of table
Stocks <- stock_names[1:15]
Means <- rep(NA, 15)
Sd <- rep(NA,15)
Skewness <- rep(NA,15)
Kurtosis <- rep(NA,15)
```

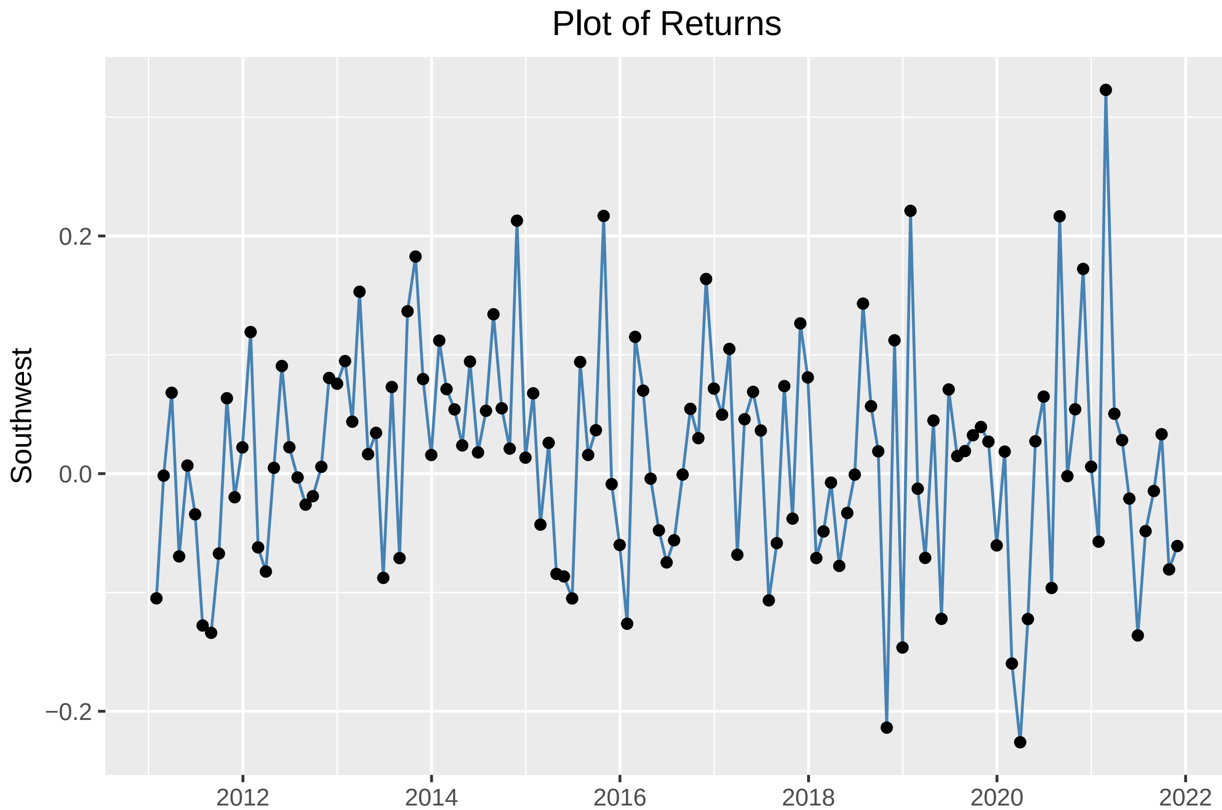
```r
table <- data.frame(Stocks,Means,Sd,Skewness,Kurtosis)


# means

for(i in 1:15)
  {
    table$Means[i] <- mean(returnsNA[,i+1])
  }

# sd

for(i in 1:15)
  {
    table$Sd[i] <- sd(returnsNA[,i+1])
  }

# skewness should be around 0 to be normally distributed

for(i in 1:15)
  {
    table$Skewness[i] <- Skew(returnsNA[,i+1])
  }

# kurtosis should be around 3 to be normally distributed

for(i in 1:15)
  {
    table$Kurtosis[i] <- Kurt(returnsNA[,i+1])
  }

# betas
#cAPM model, risk free and market portfolio IVV
table
```

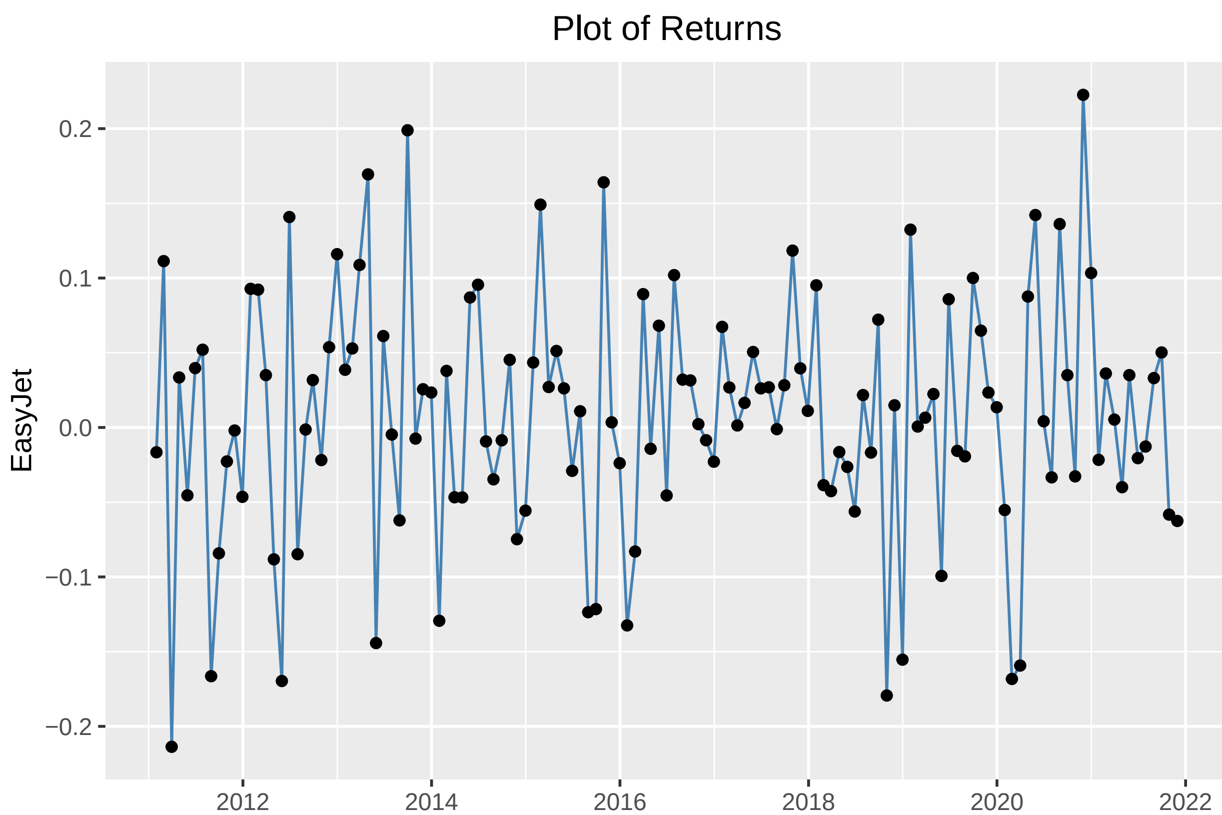
```
##       Stocks         Means         Sd    Skewness  Kurtosis
## 1  Lufthansa -0.0005930083 0.11241027  0.08667032 0.1488168
## 2  Southwest  0.0171885468 0.09277315  0.21544579 0.5677506
## 3    EasyJet  0.0079285412 0.07815002 -0.05386680 0.3734066
## 4    RyanAir  0.0134598915 0.08810162  0.03841435 0.7835927
## 5     United  0.0117156991 0.11468114 -0.40046581 2.6059592
## 6   American  0.0128577554 0.14004608  0.92824441 3.3713287
## 7     Spirit  0.0129167067 0.12816431 -0.27125023 3.0894567
## 8  Carlsberg  0.0048833354 0.07151443 -0.38252473 0.9578615
## 9   Heineken  0.0068923311 0.06196124 -0.11333085 0.3347802
## 10        AB  0.0025169668 0.07827691 -0.03417490 1.4085001
## 11    Harboe -0.0004547538 0.06049694  0.65593756 0.6983457
## 12  Guinness  0.0101843141 0.04608098  0.24639304 1.2094121
## 13    Molson  0.0015157556 0.07310860  0.77435046 2.3018081
## 14  Tsingtao  0.0073310755 0.09659968  0.29240859 1.0474625
## 15     Asahi  0.0083592940 0.07193289 -0.41777156 4.1653067
```

2. Plot each set of returns

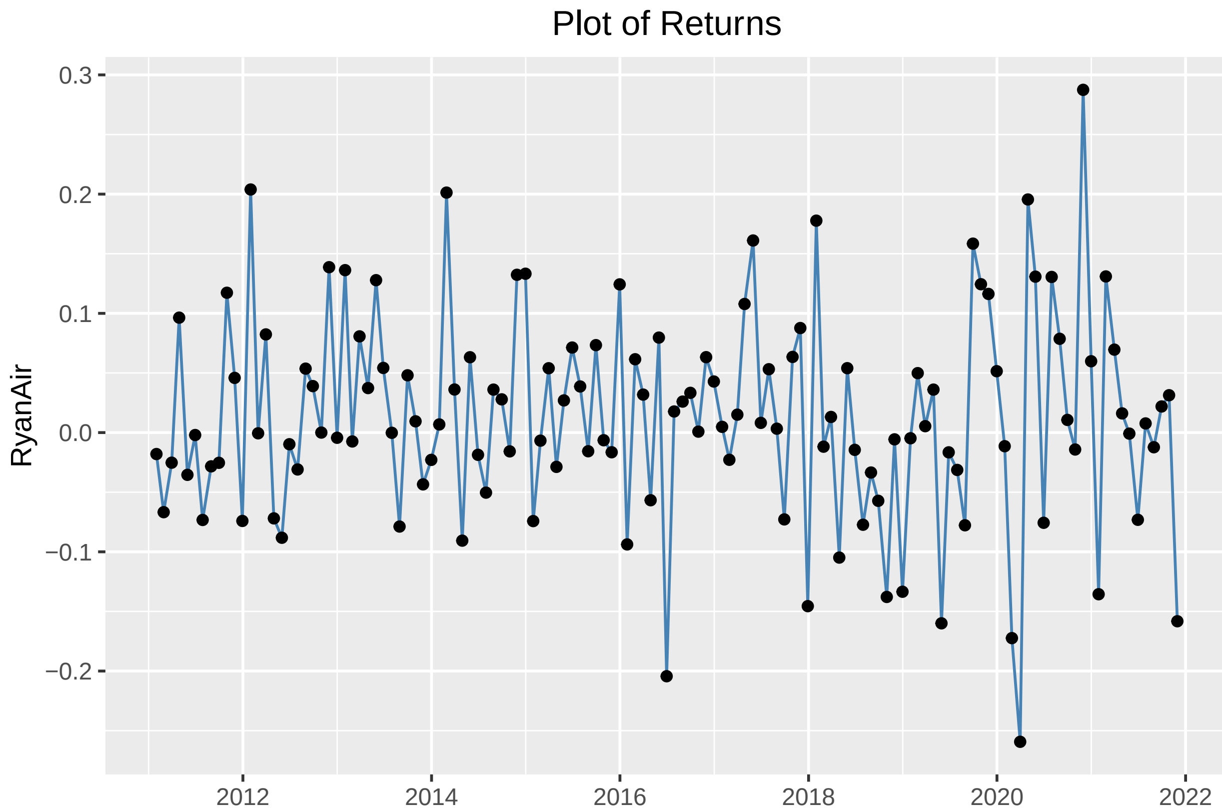
```
returns <- as.data.frame(returns)

for (i in 1:15)
{
  names <- list(stock_names)
  print(ggplot(returns, aes(x=date, y=returns[,i+1])) +
  geom_line(color="steelblue") +
  geom_point() +
  xlab("") +
  ylab(names[[1]][i])+
  ggtitle("Plot of Returns") +
  theme(plot.title = element_text(hjust = 0.5)))
}
```

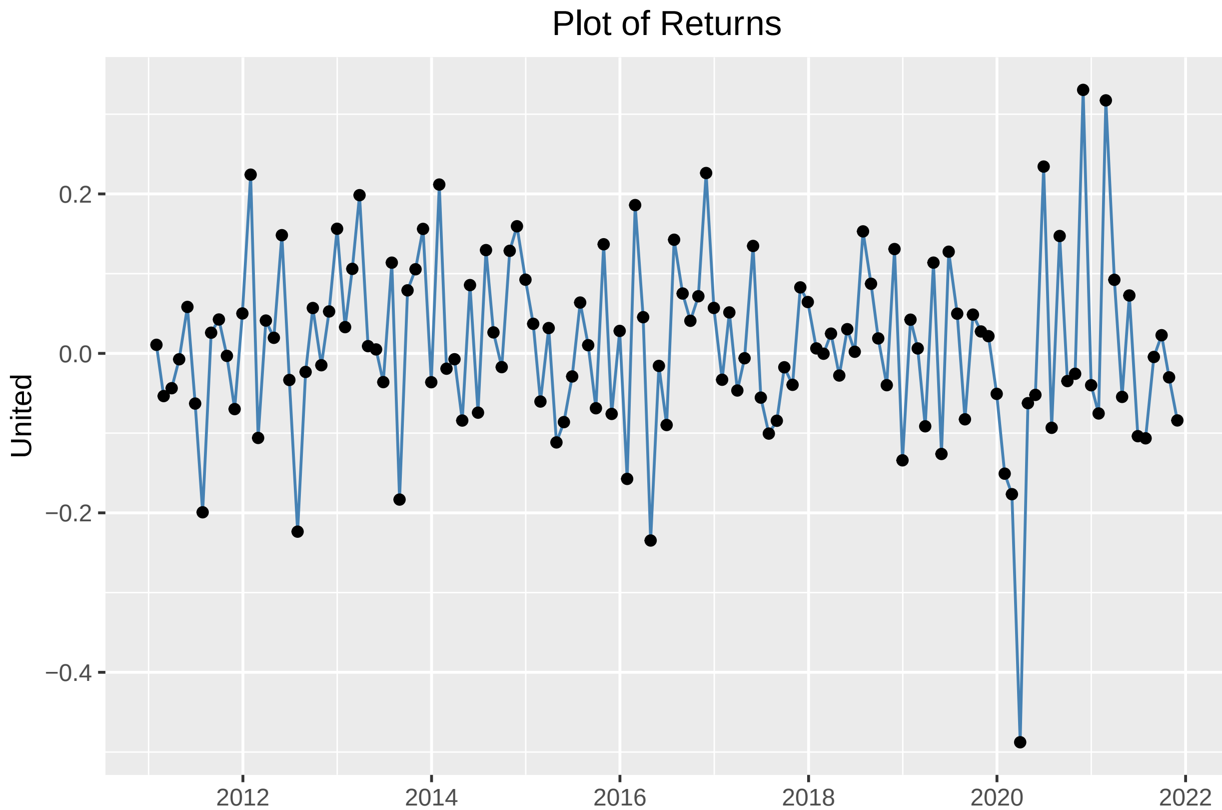


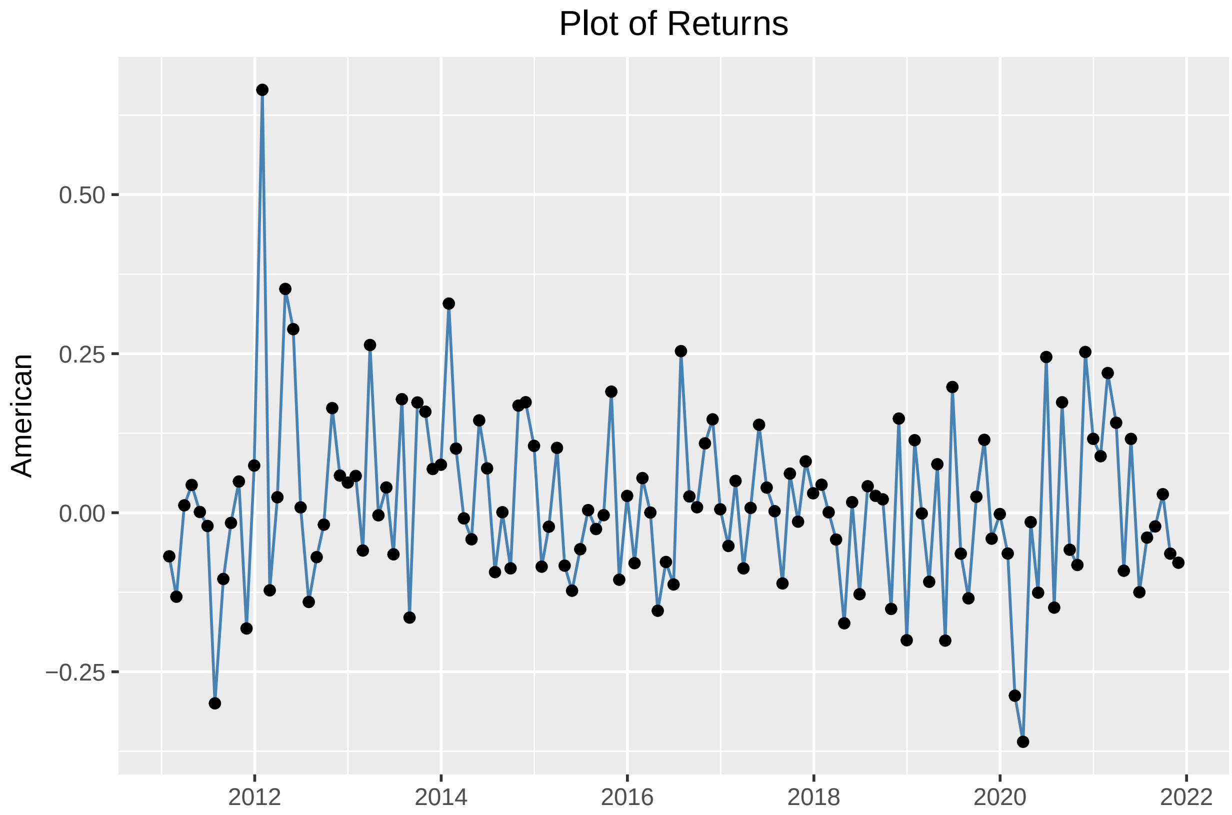

Plot of Returns
Southwest
0.2
0.0
−0.2
2012
2014
2016
2018
2020
2022



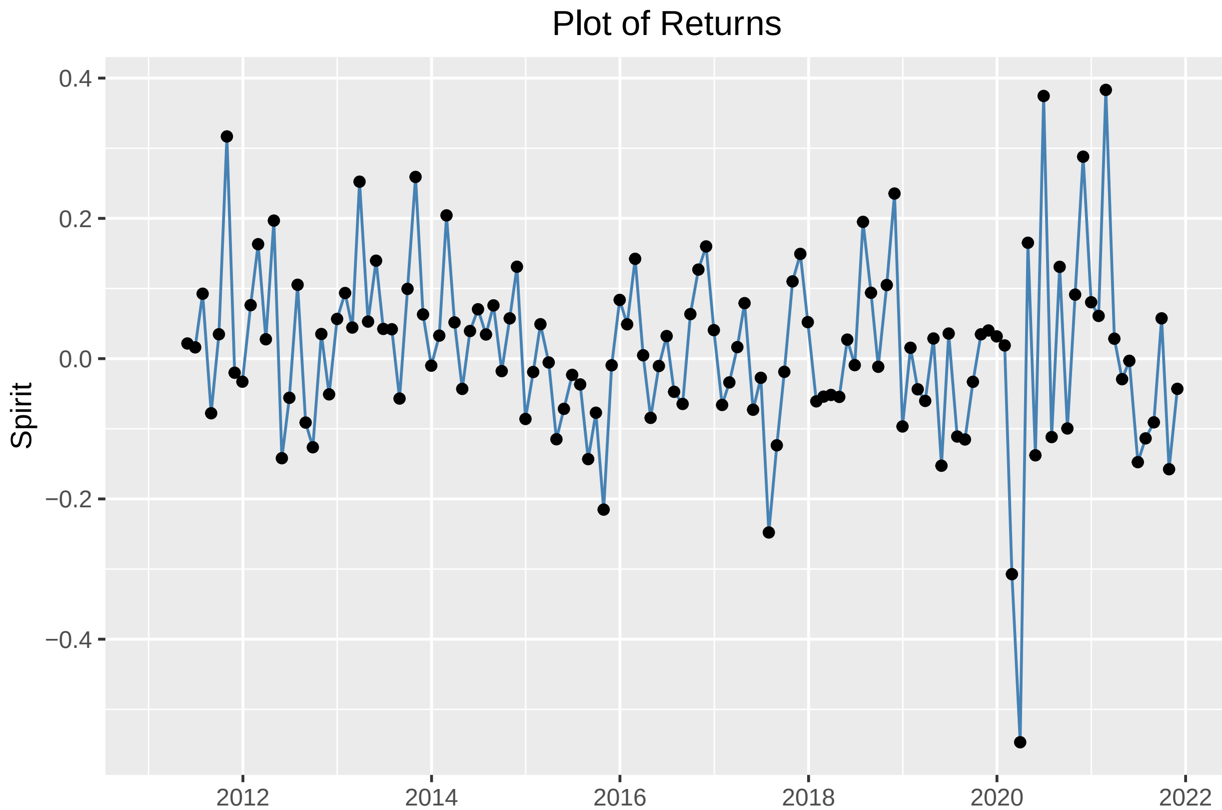
Plot of Returns
EasyJet
0.2
0.1
0.0
−0.1
−0.2
2012
2014
2016
2018
2020
2022

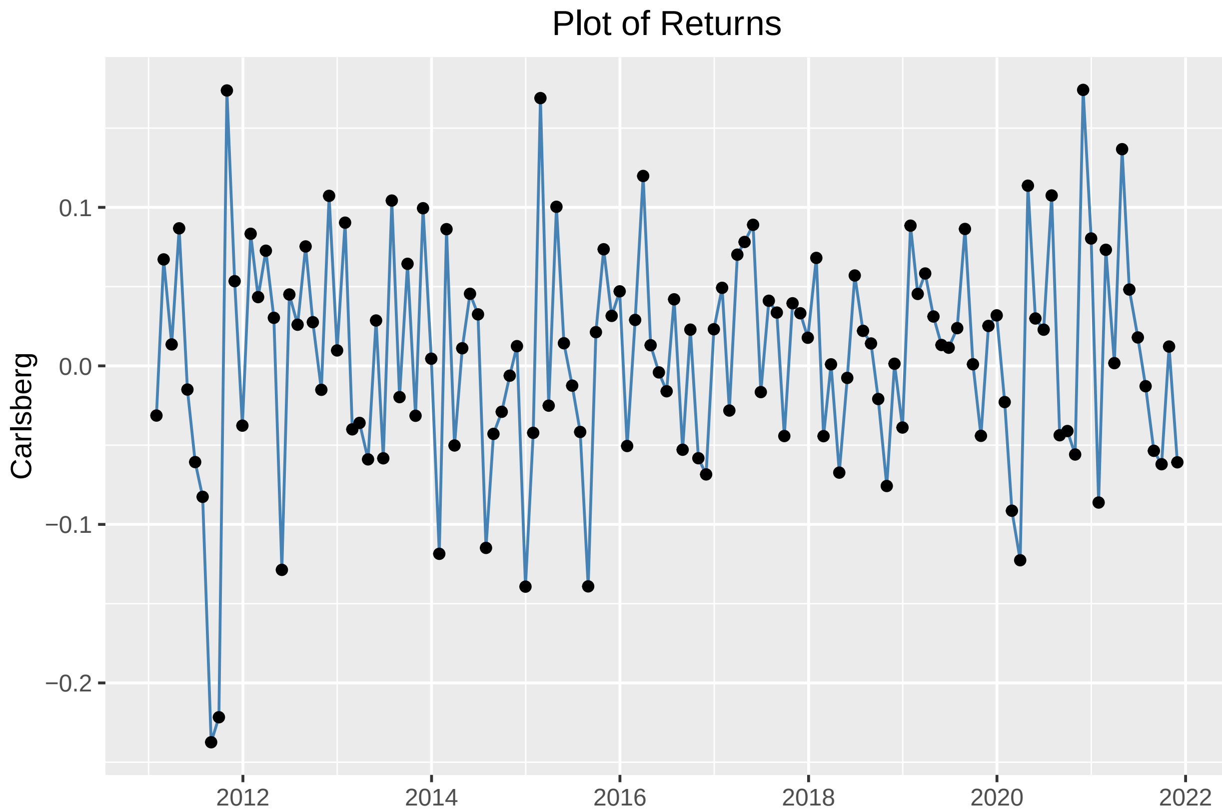

Plot of Returns
RyanAir
0.3
0.2
0.1
0.0
−0.1
−0.2
2012
2014
2016
2018
2020
2022

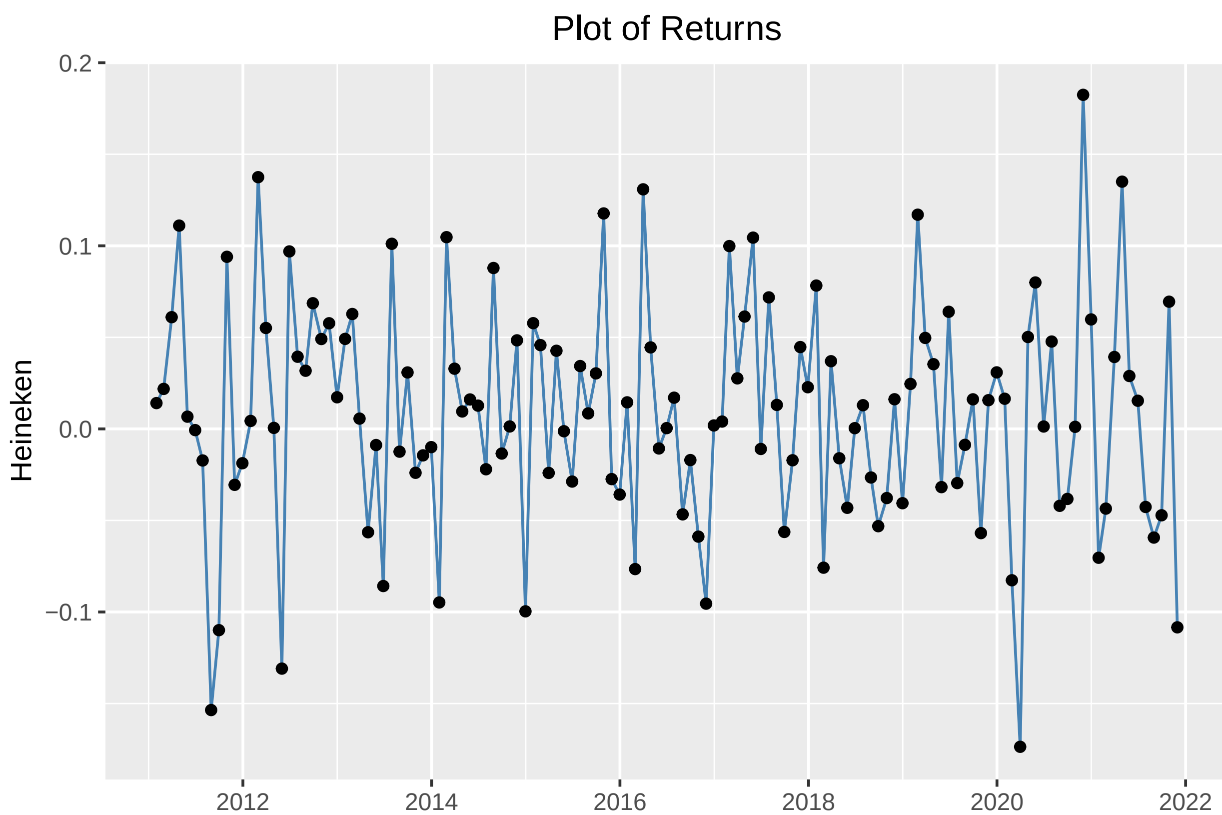

Plot of Returns
United
0.2
0.0
−0.2
−0.4
2012
2014
2016
2018
2020
2022

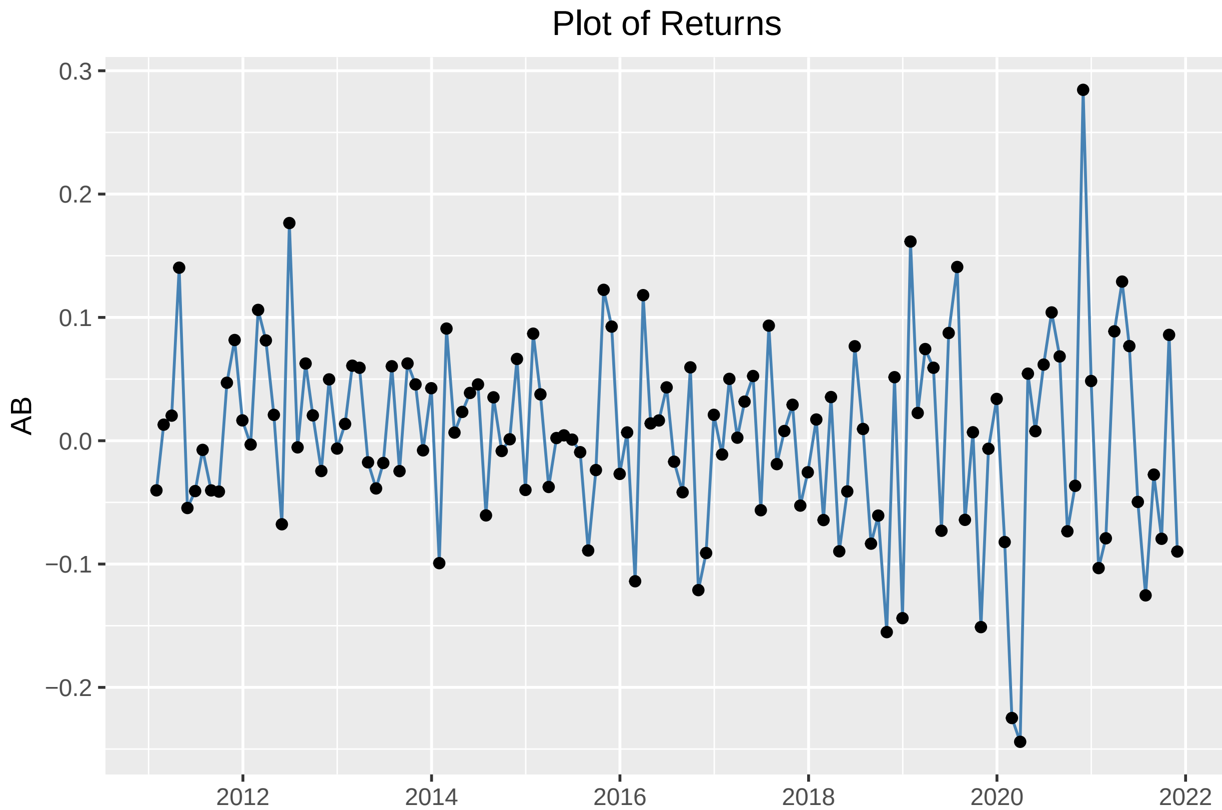



```
## Warning: Removed 4 row(s) containing missing values (geom_path).

## Warning: Removed 4 rows containing missing values (geom_point).
```

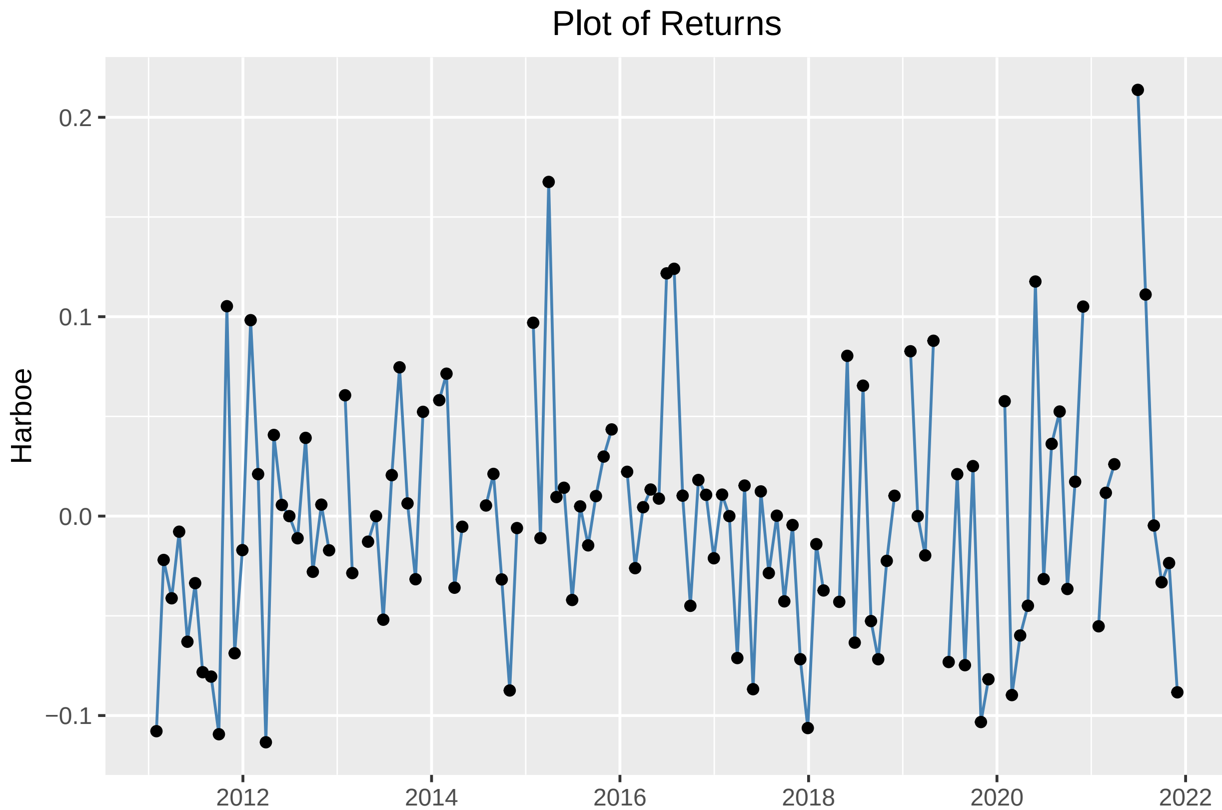

Plot of Returns
Spirit
0.4
0.2
0.0
−0.2
−0.4
2012
2014
2016
2018
2020
2022

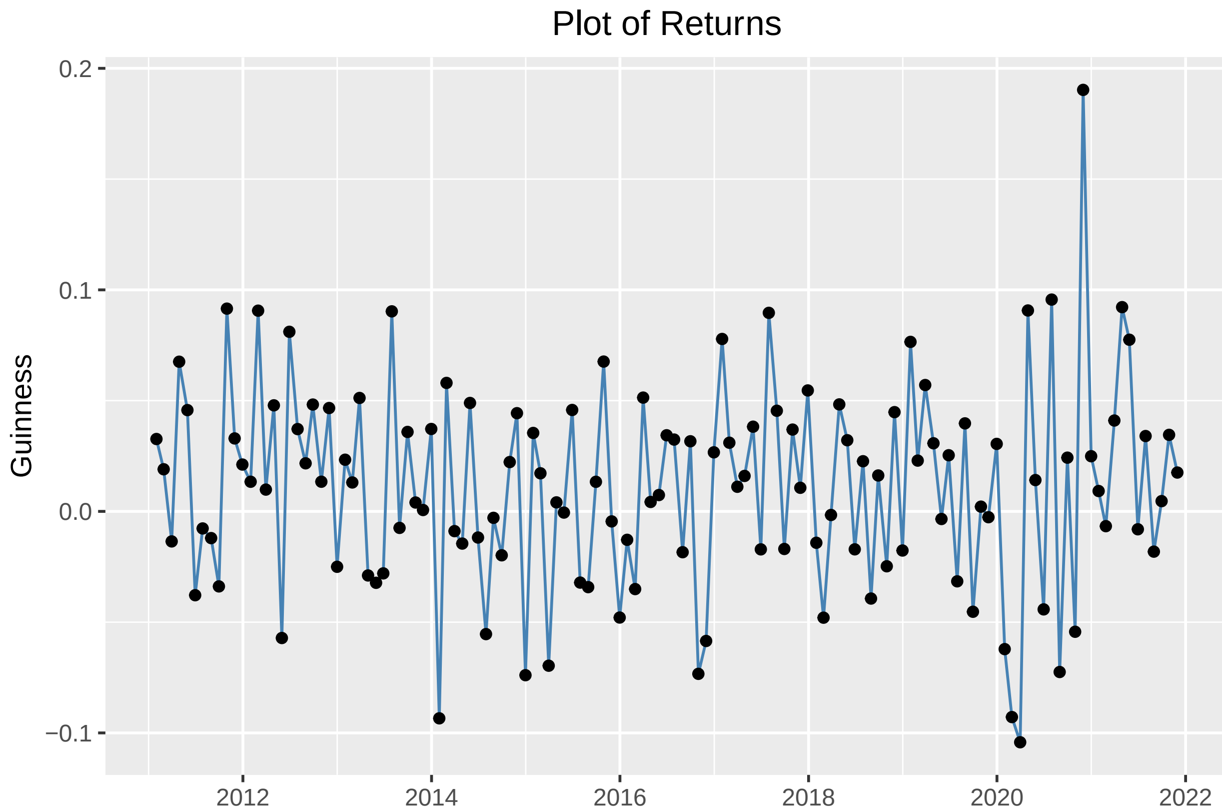

Plot of Returns
Carlsberg
0.1
0.0
−0.1
−0.2
2012
2014
2016
2018
2020
2022

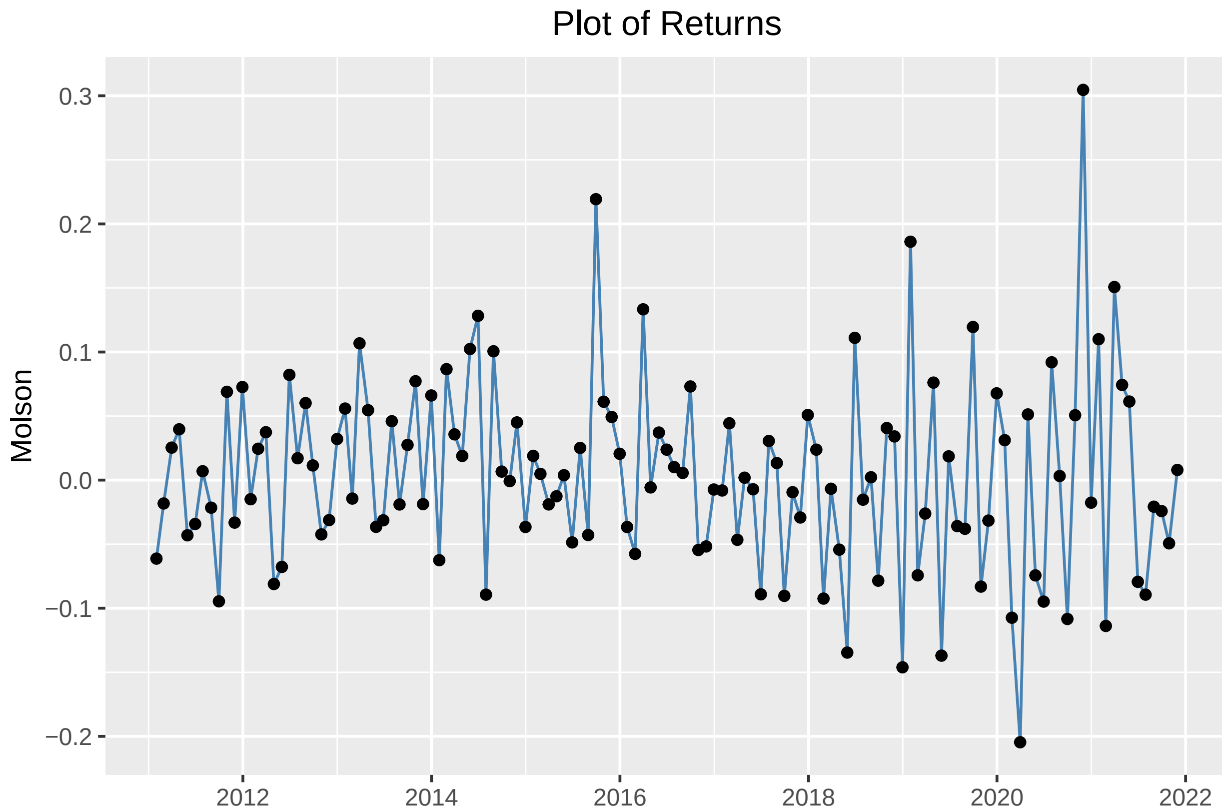

Plot of Returns
Heineken
0.2
0.1
0.0
−0.1
2012
2014
2016
2018
2020
2022

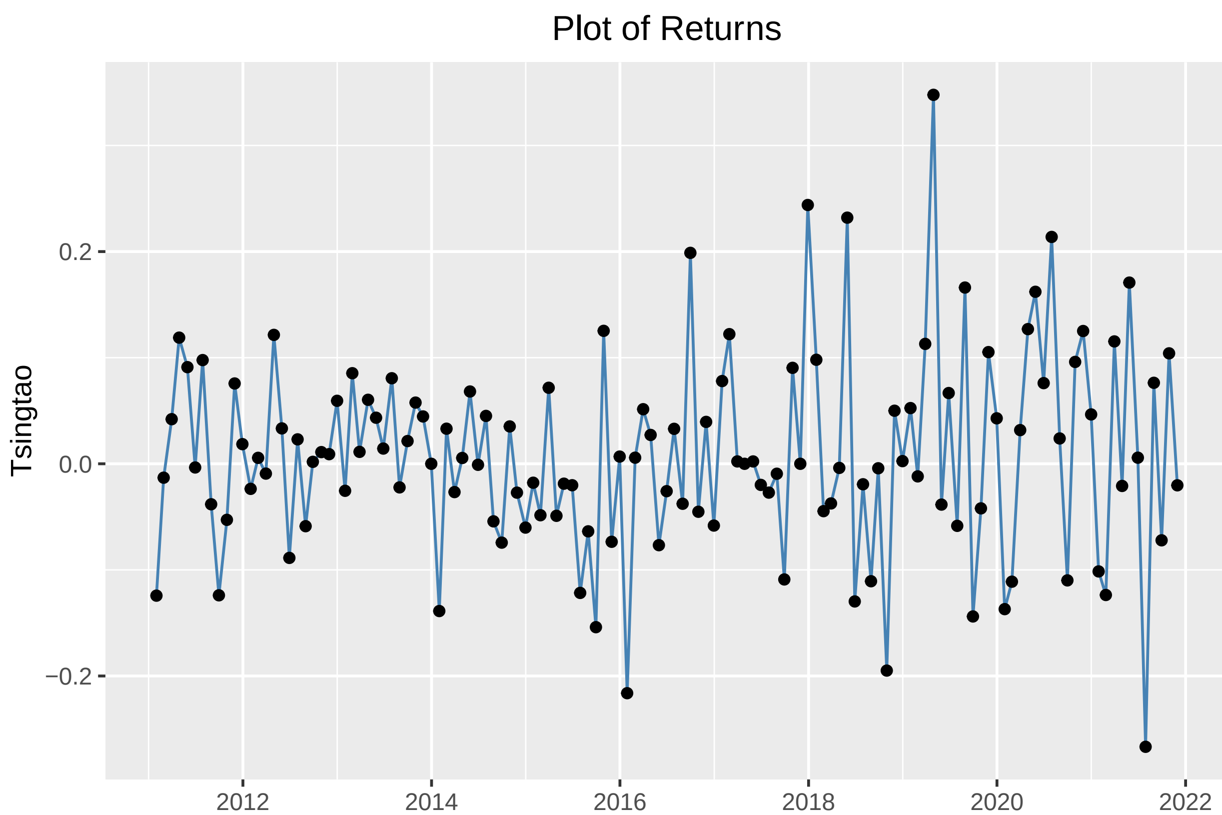


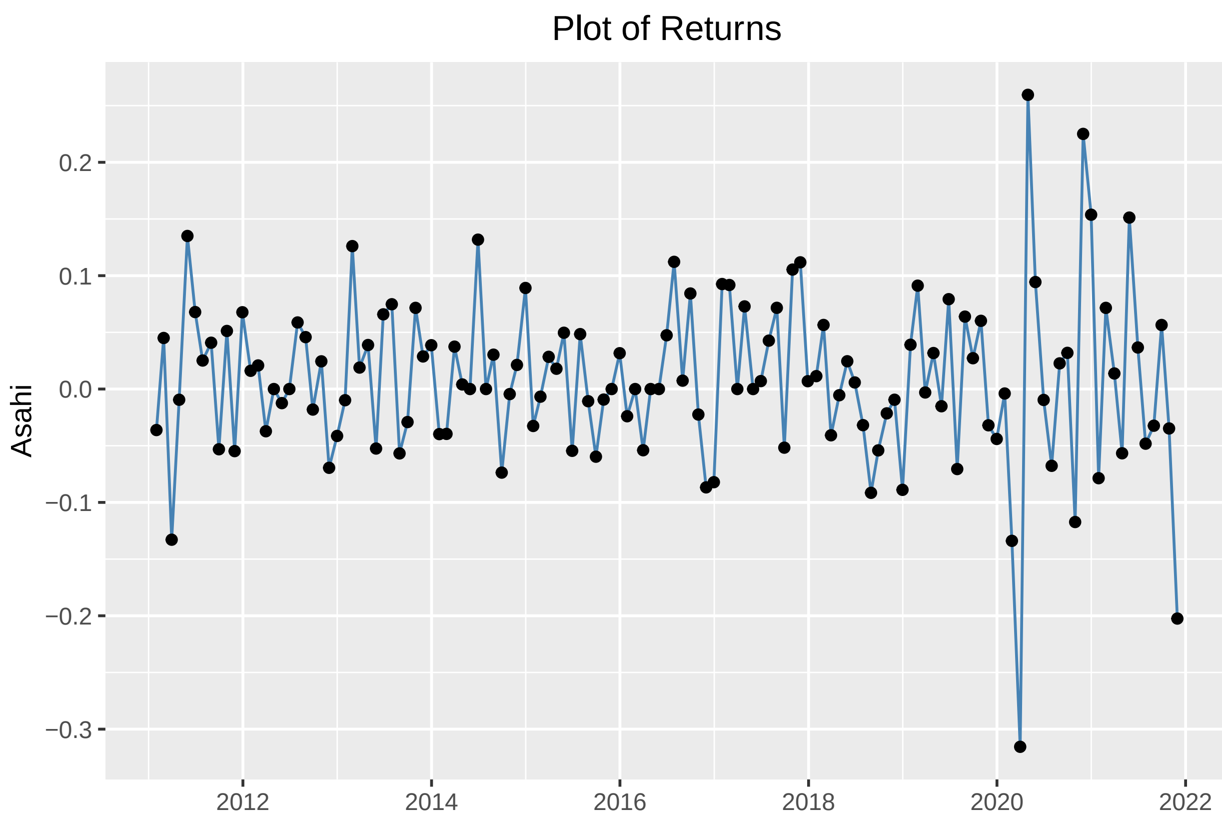

```
## Warning: Removed 14 rows containing missing values (geom_point).
```


Plot of Returns
Harboe
0.2
0.1
0.0
−0.1
2012
2014
2016
2018
2020
2022

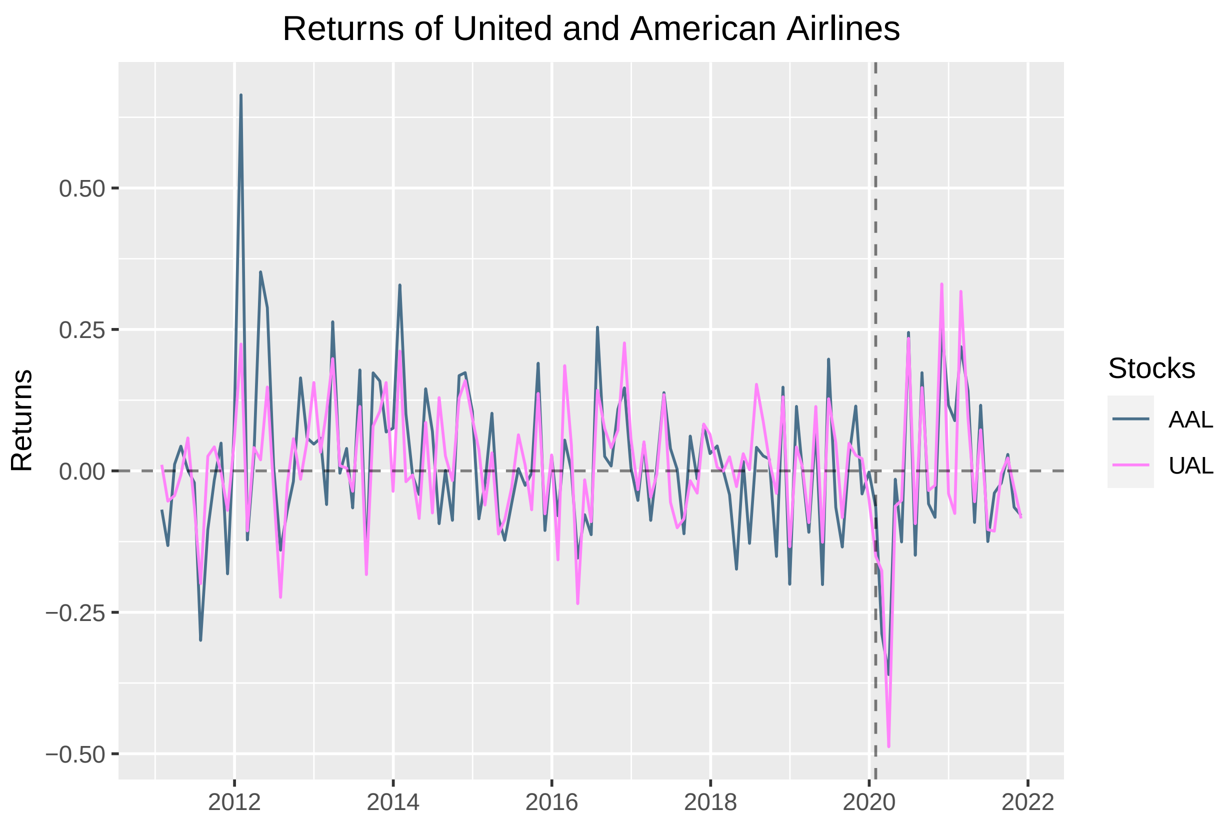

Plot of Returns
Guinness
0.2
0.1
0.0
−0.1
2012
2014
2016
2018
2020
2022


Plot of Returns
Molson
0.3
0.2
0.1
0.0
−0.1
−0.2
2012
2014
2016
2018
2020
2022

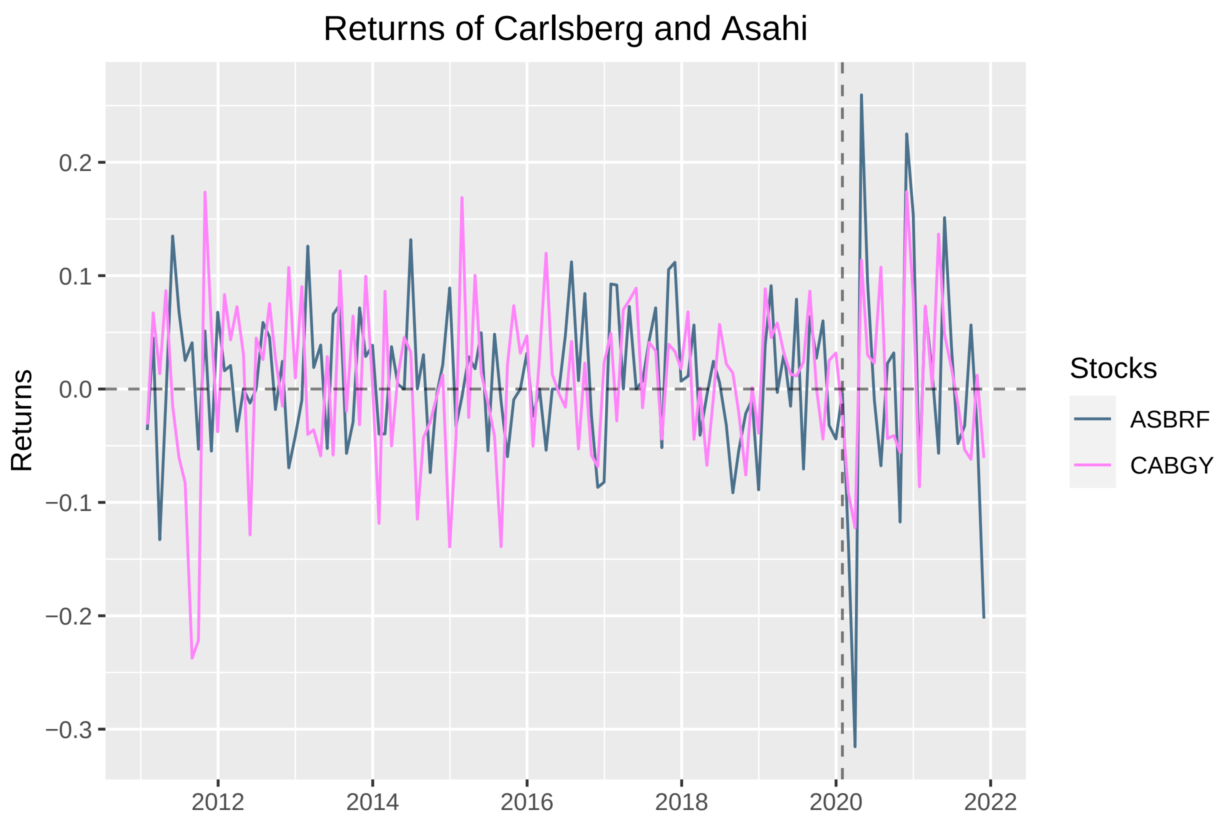

Plot of Returns
Tsingtao
0.2
0.0
−0.2
2012
2014
2016
2018
2020
2022




```r
#united airlines and american airlines

airlines <- returns %>%
            select(date, UAL, AAL) %>%
            gather(key="Stocks", value = "Returns", -date)

ggplot(airlines, aes(x = date, y = Returns)) +
  geom_line(aes(color = Stocks)) +
  scale_color_manual(values = c("skyblue4", "orchid1")) +
  geom_hline(yintercept=0,linetype="dashed", alpha=0.5)+
  geom_vline(xintercept = as.Date("2020-01-31"), linetype="dashed", alpha=0.5) +
  ggtitle("Returns of United and American Airlines") +
  theme(plot.title = element_text(hjust = 0.5)) +
  xlab("")
```




```
rm(airlines)

#carslberg and asahi

beers <- returns %>%
         select(date, CABGY, ASBRF ) %>%
         gather(key="Stocks", value = "Returns", -date)

ggplot(beers, aes(x = date, y = Returns)) +
  geom_line(aes(color = Stocks)) +
  scale_color_manual(values = c("skyblue4", "orchid1")) +
  geom_hline(yintercept=0,linetype="dashed", alpha=0.5)+
  geom_vline(xintercept = as.Date("2020-01-31"), linetype="dashed", alpha=0.5) +
  ggtitle("Returns of Carlsberg and Asahi") +
  theme(plot.title = element_text(hjust = 0.5)) +
  xlab("")
```

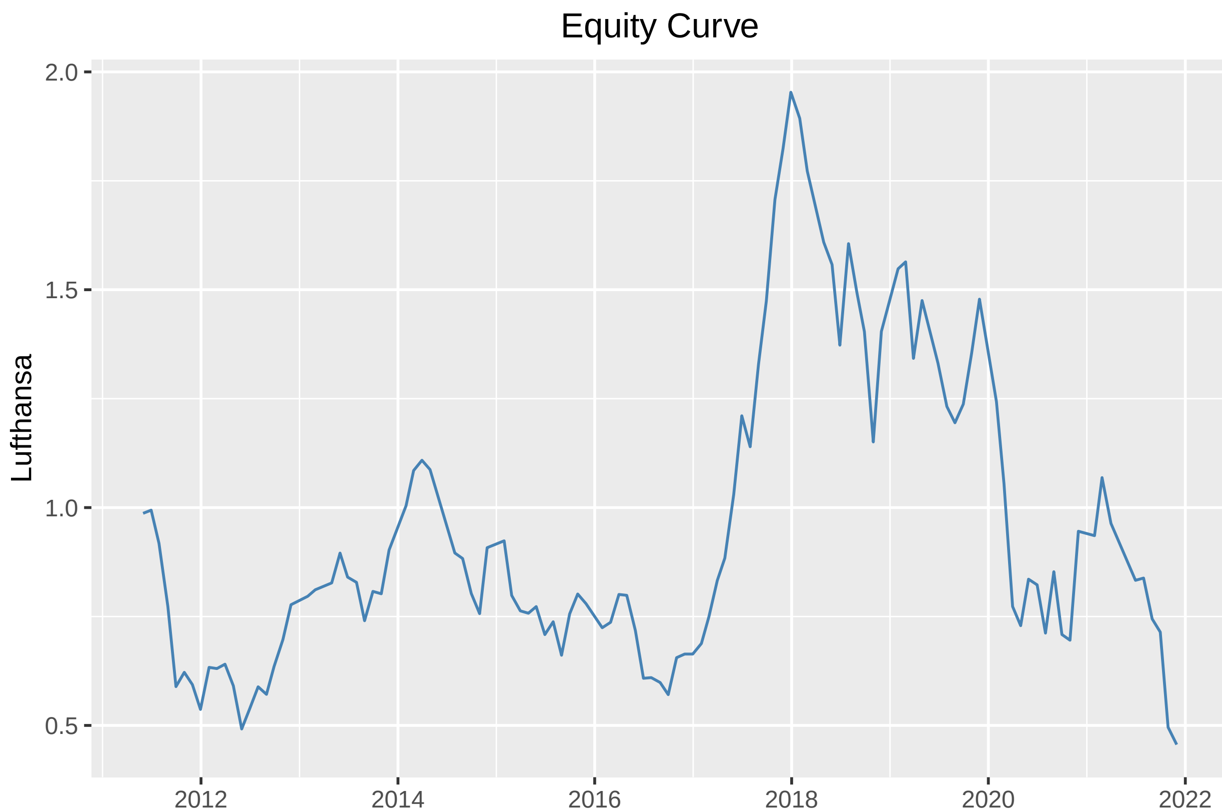


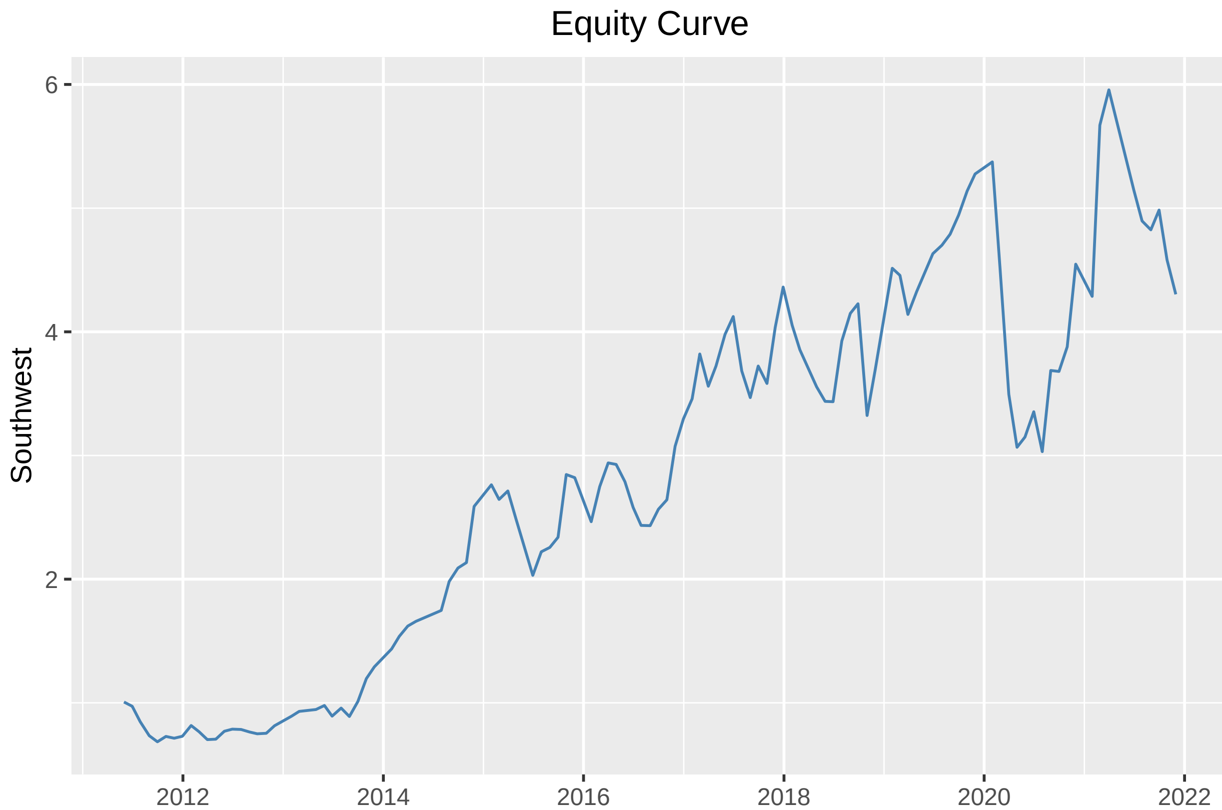

```
rm(beers)
```

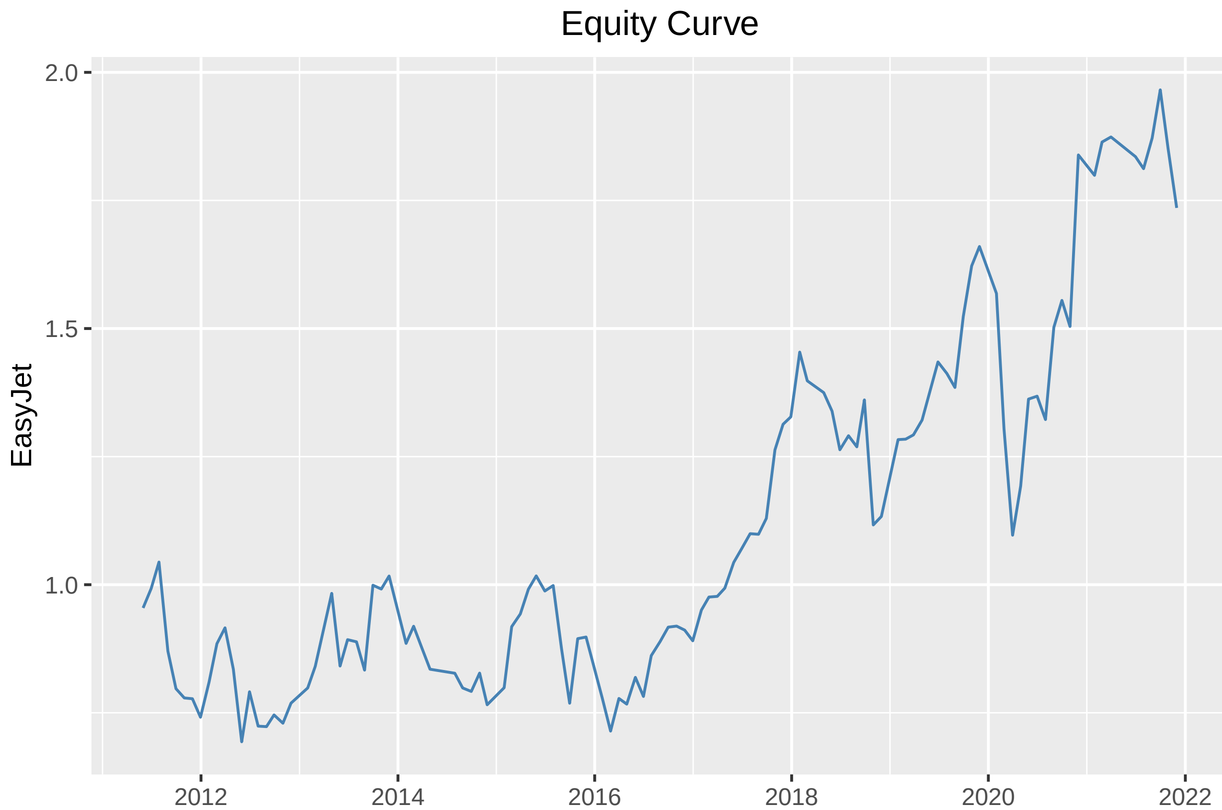
3. Plot equity curve
   https://bookdown.org/compfinezbook/introcompfinr/ReturnCalculationsR.html#

```
returnsNA <- as.data.frame(na.omit(returns))    #doesn't work with NA values

equityCurve <- as.data.frame(matrix(c(rep(NA,113*15)), nrow= 113))
for (i in 1:15)
{
  equityCurve[,i] <- cumprod(1 + returnsNA[,i+1])
}


equityCurve <- cbind(returnsNA[,1], equityCurve)
colnames(equityCurve)[1] <- "date"

#airlines 7
for (i in 1:7)
{
  names <- list(stock_names)
  print(ggplot(equityCurve, aes(x=date, y=equityCurve[,i+1])) +
  ggtitle("Equity Curve") +
  theme(plot.title = element_text(hjust = 0.5))+
  geom_line(color="steelblue") +
```

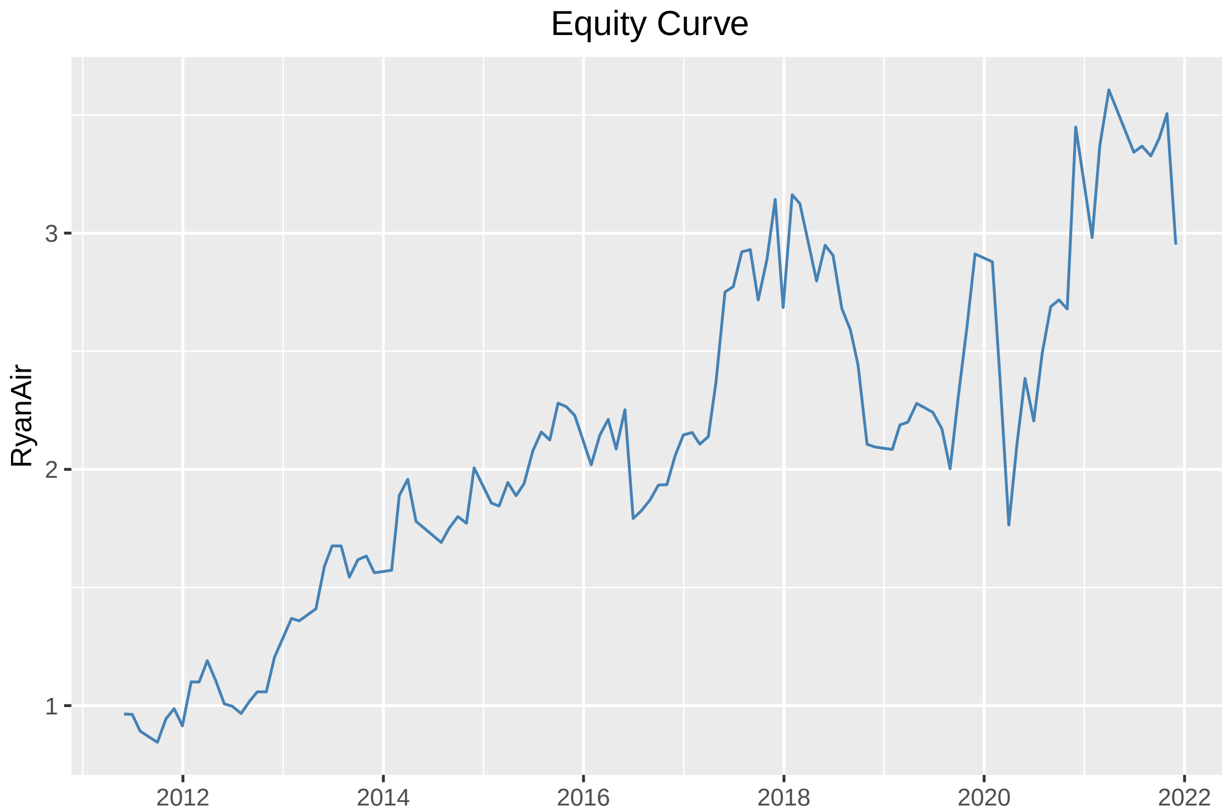
```
  xlab("") +
  ylab(names[[1]][i]))
}
```



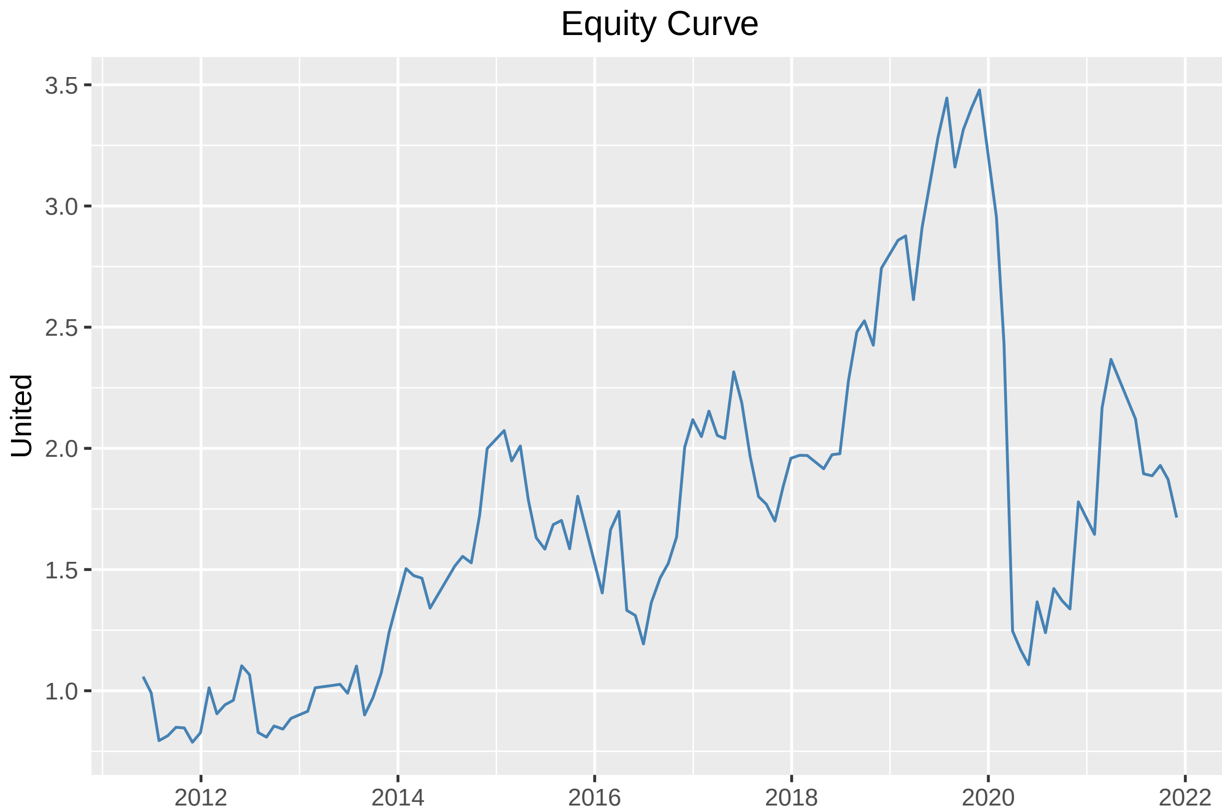

Equity Curve
Southwest
6
4
2
2012
2014
2016
2018
2020
2022

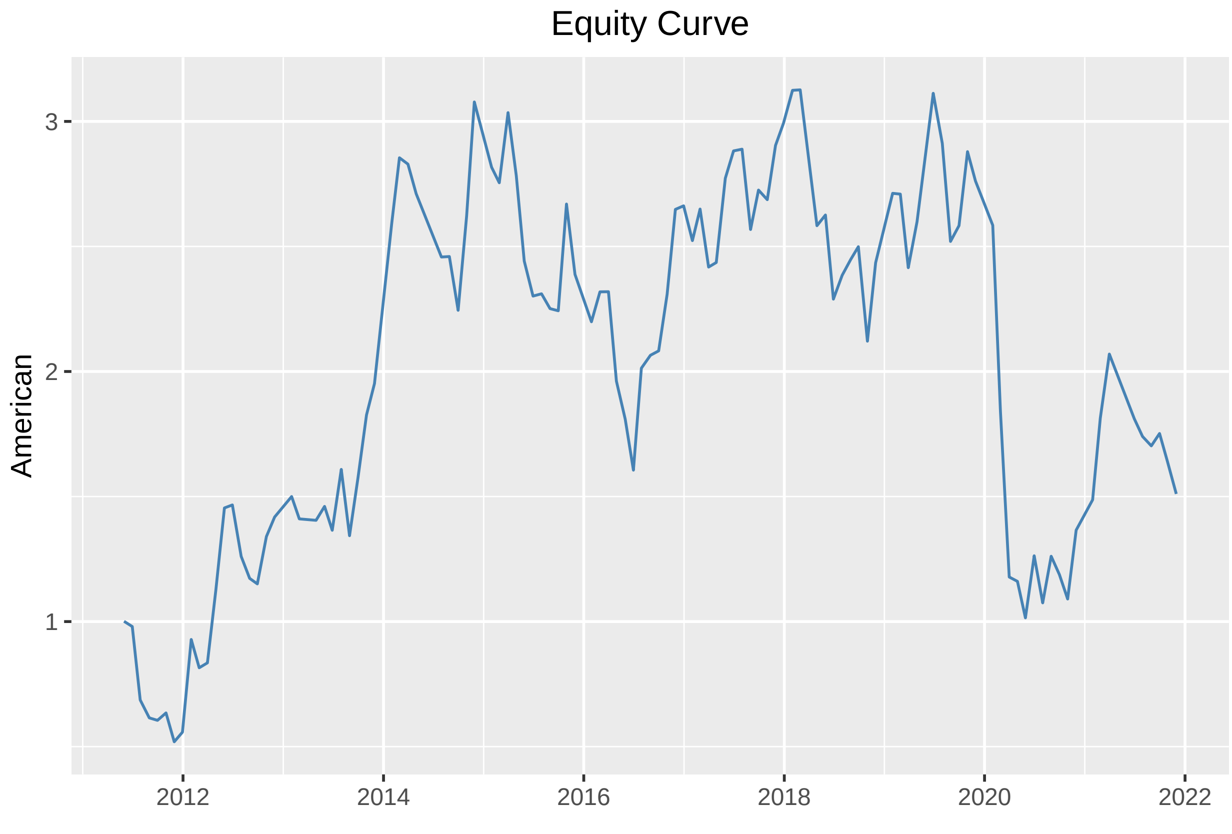


Equity Curve
EasyJet
2.0
1.5
1.0
2012
2014
2016
2018
2020
2022

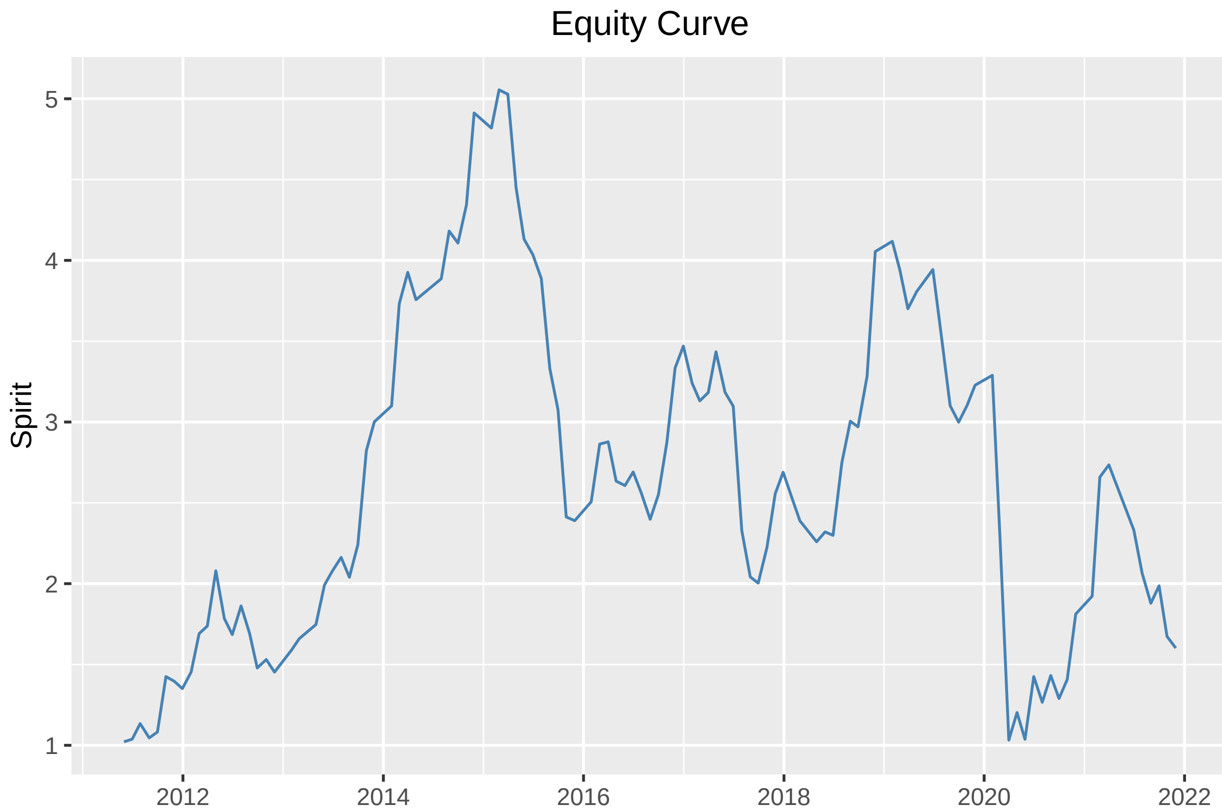


Equity Curve
RyanAir
3
2
1
2012
2014
2016
2018
2020
2022

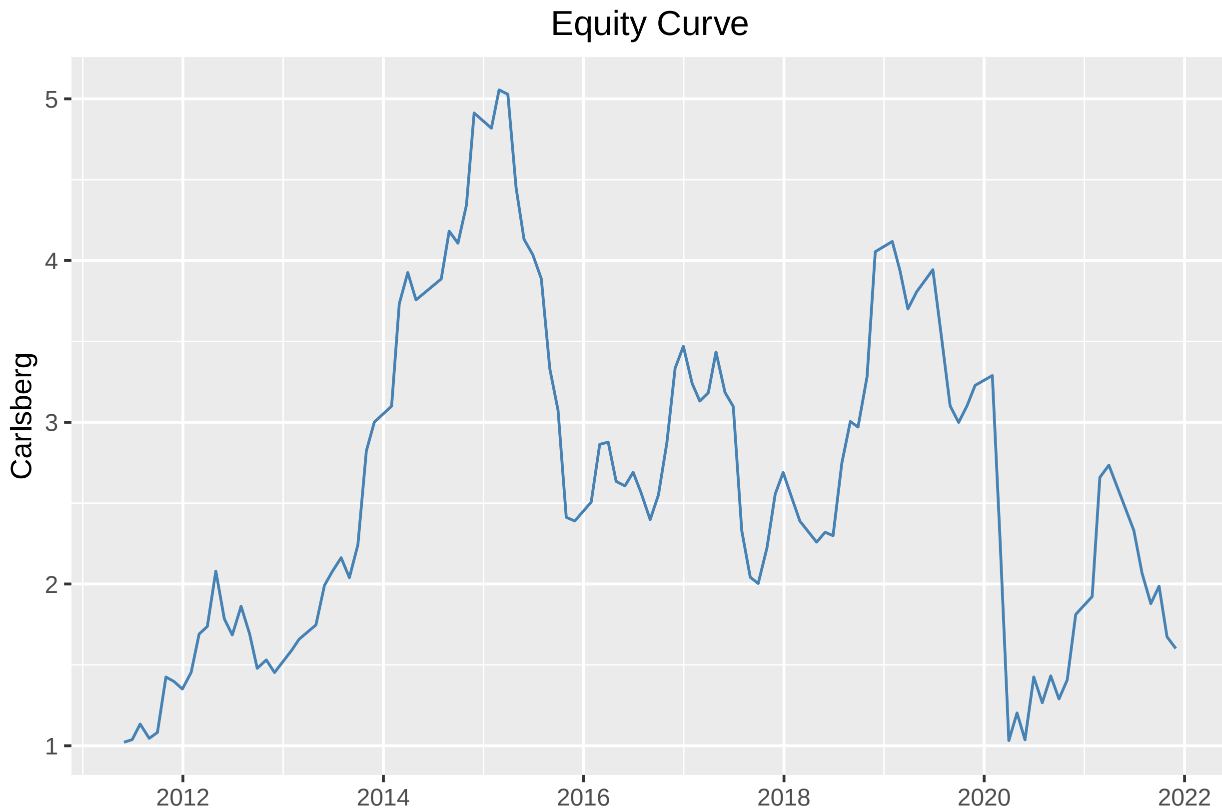

Equity Curve
United
3.5
3.0
2.5
2.0
1.5
1.0
2012
2014
2016
2018
2020
2022

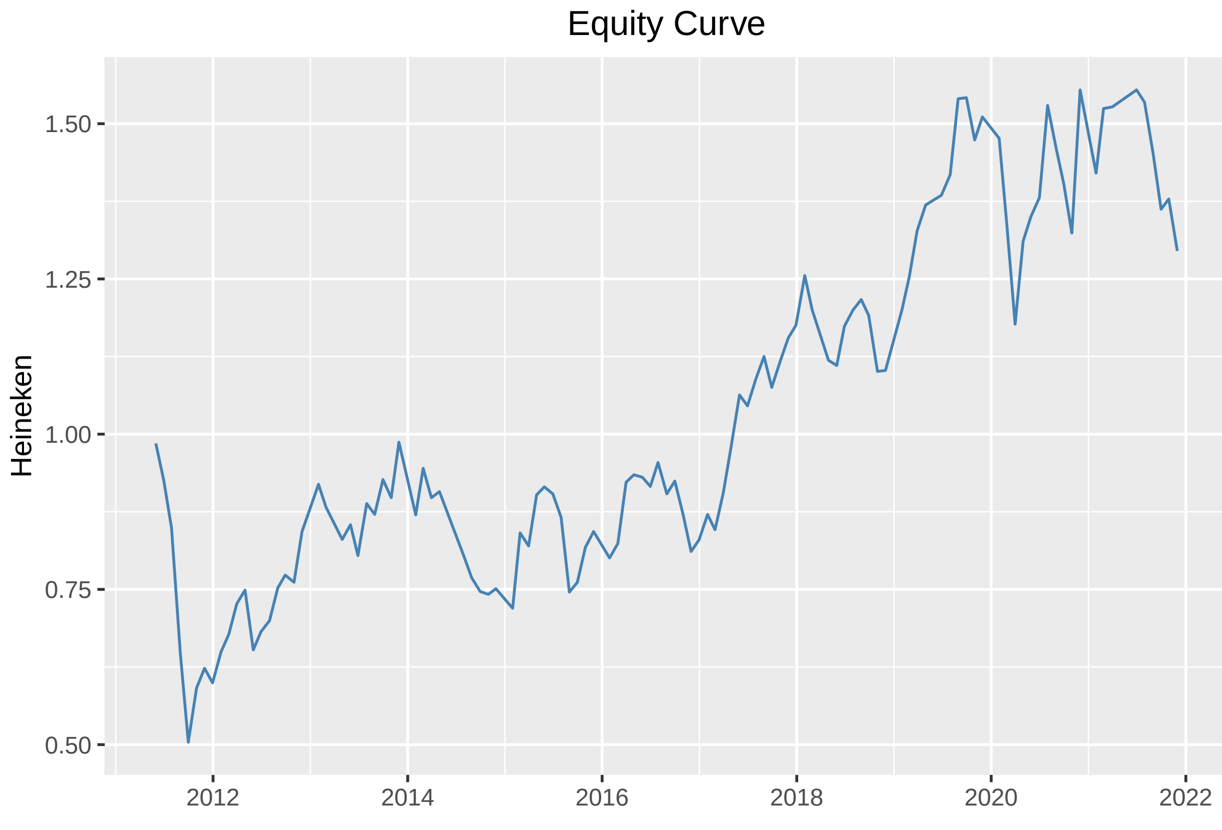


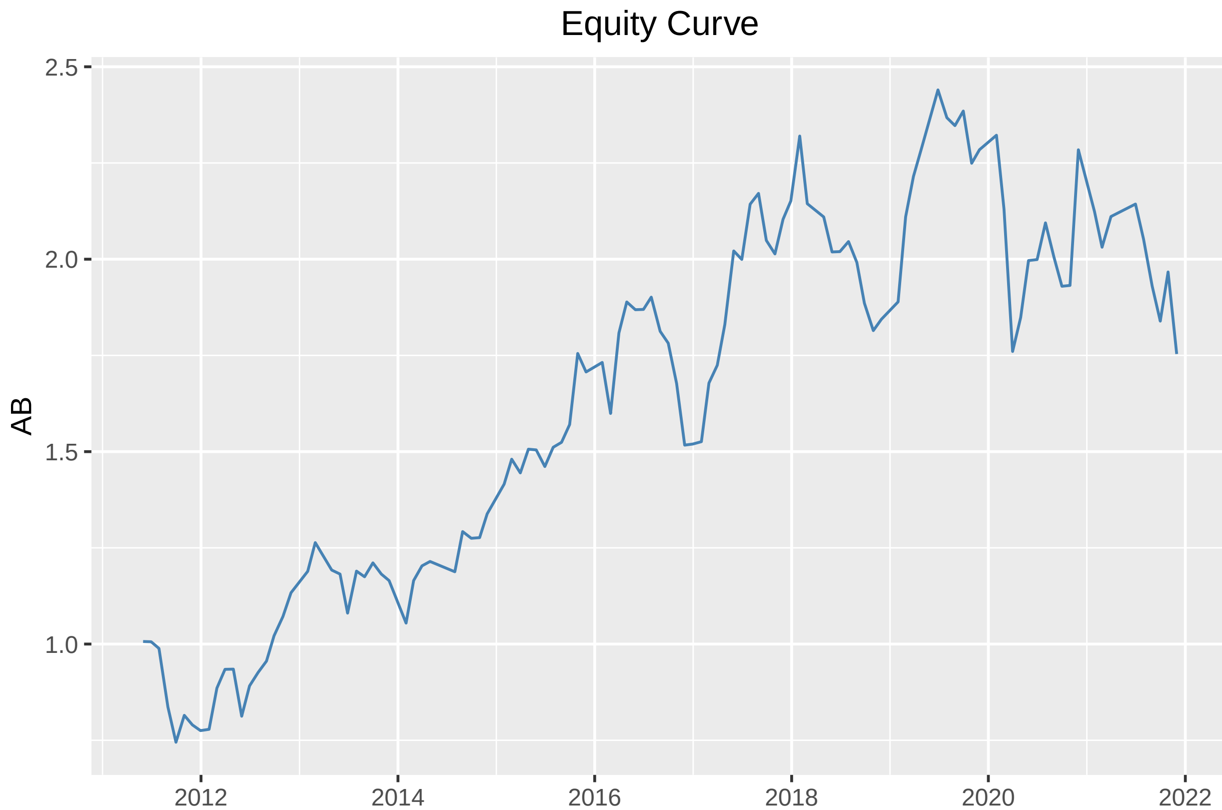
Equity Curve
American
3
2
1
2012
2014
2016
2018
2020
2022



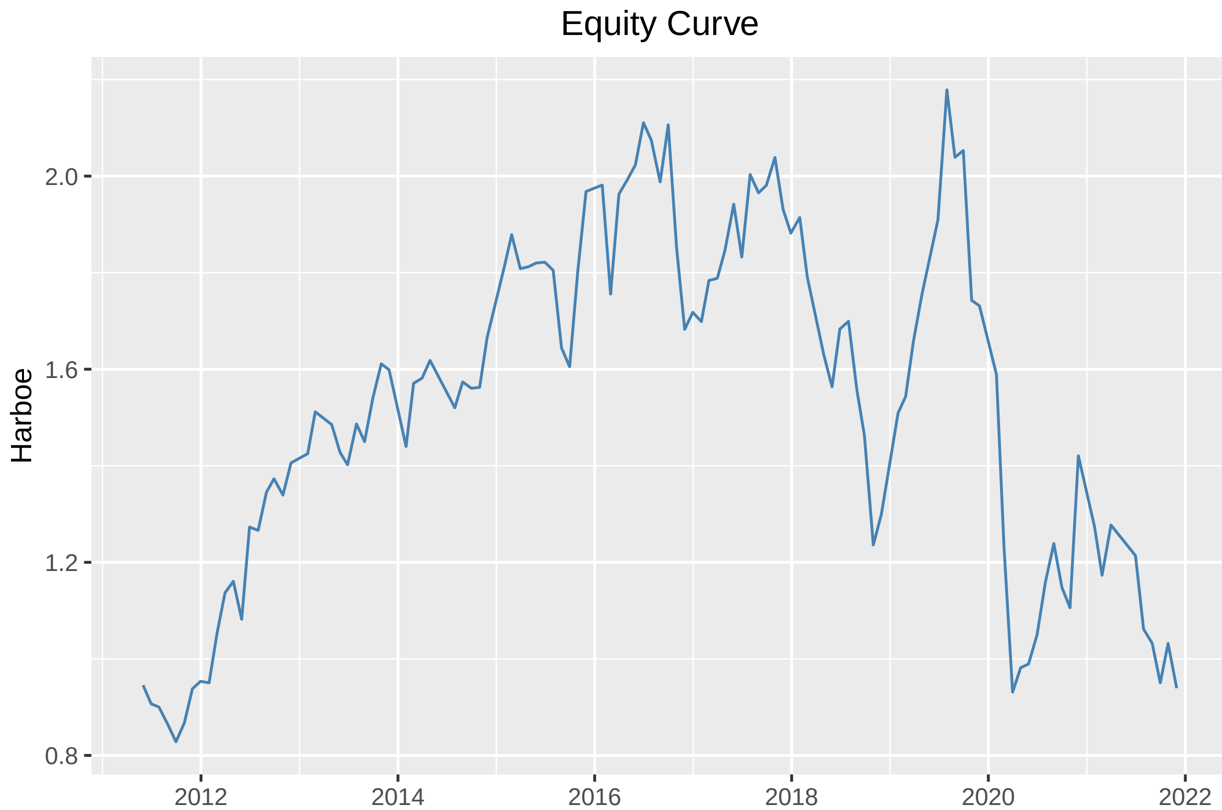

```
#beers
for (i in 1:8)
{
  names <- list(stock_names)
  print(ggplot(equityCurve, aes(x=date, y=equityCurve[,i+7])) +
  ggtitle("Equity Curve") +
  theme(plot.title = element_text(hjust = 0.5)) +
  geom_line(color="steelblue") +
  xlab("") +
  ylab(names[[1]][i+7]))
}
```

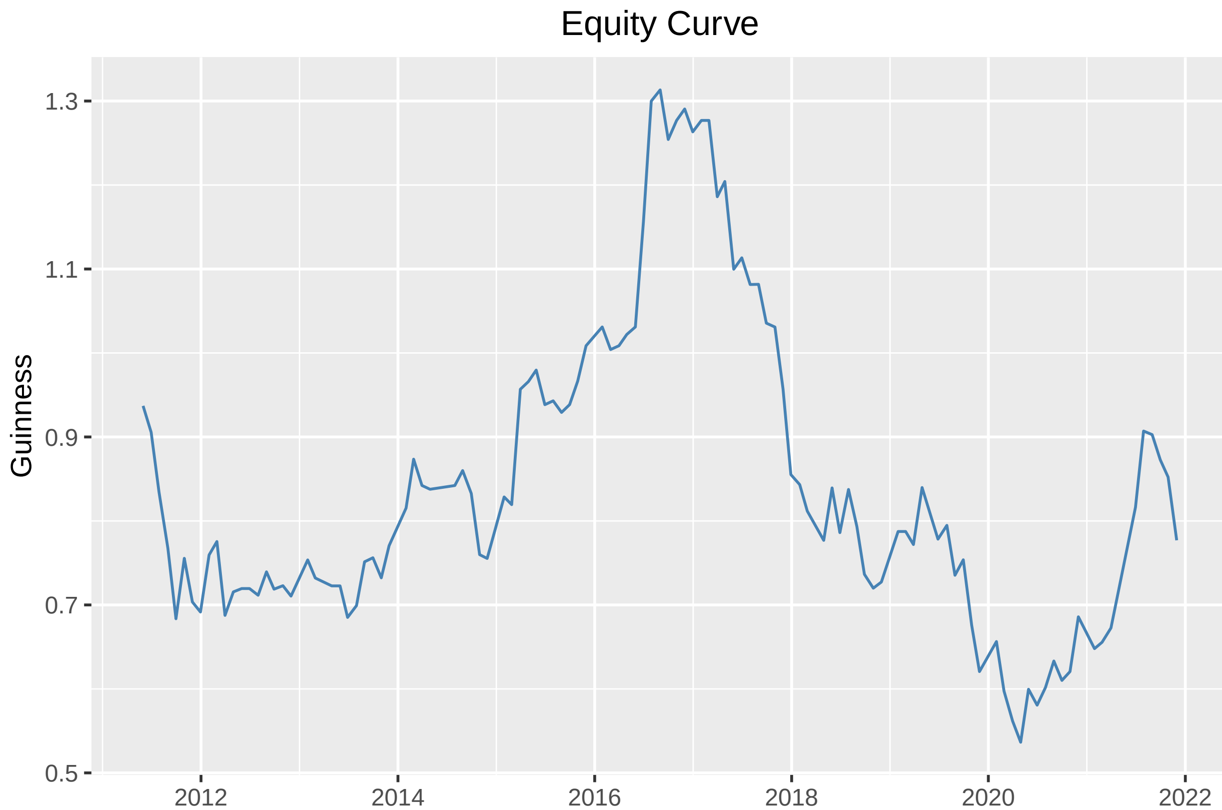

Equity Curve
Carlsberg
5
4
3
2
1
2012
2014
2016
2018
2020
2022


Equity Curve
Heineken
1.50
1.25
1.00
0.75
0.50
2012
2014
2016
2018
2020
2022

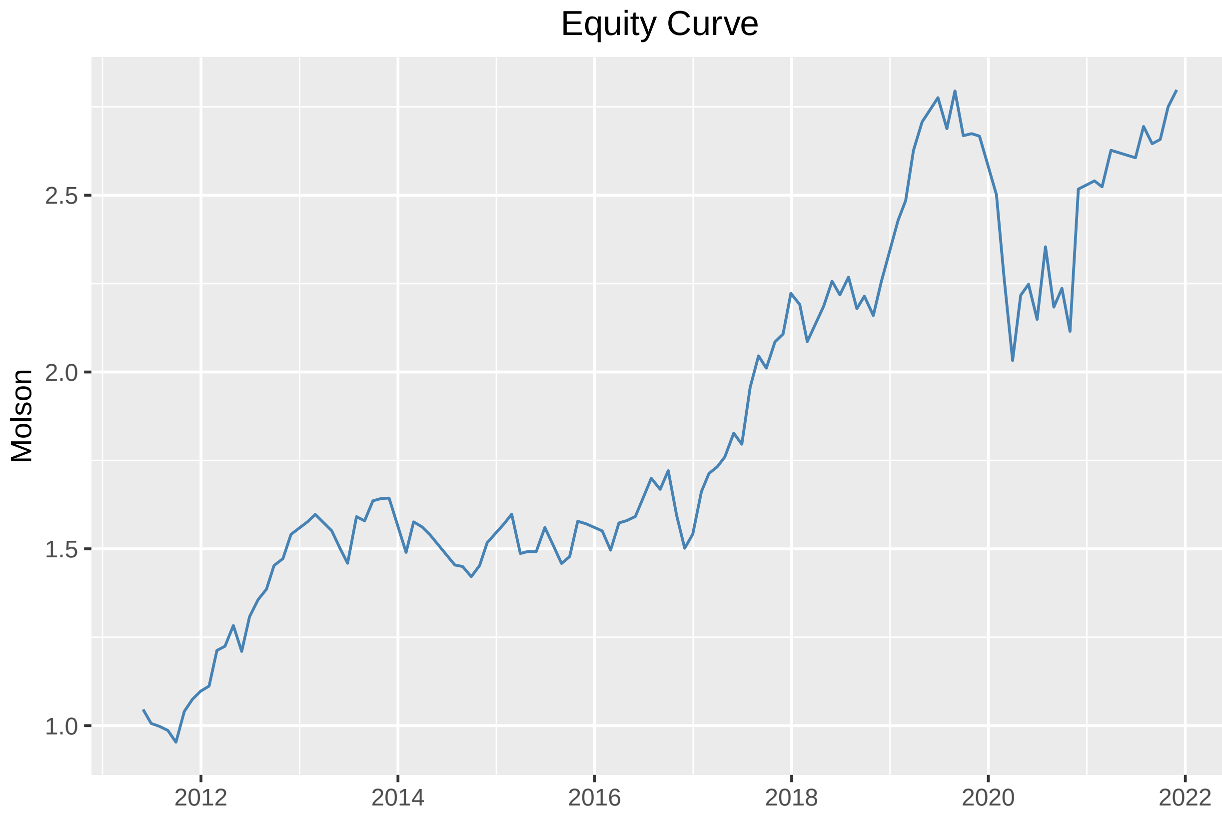


Equity Curve
AB
2.5
2.0
1.5
1.0
2012
2014
2016
2018
2020
2022

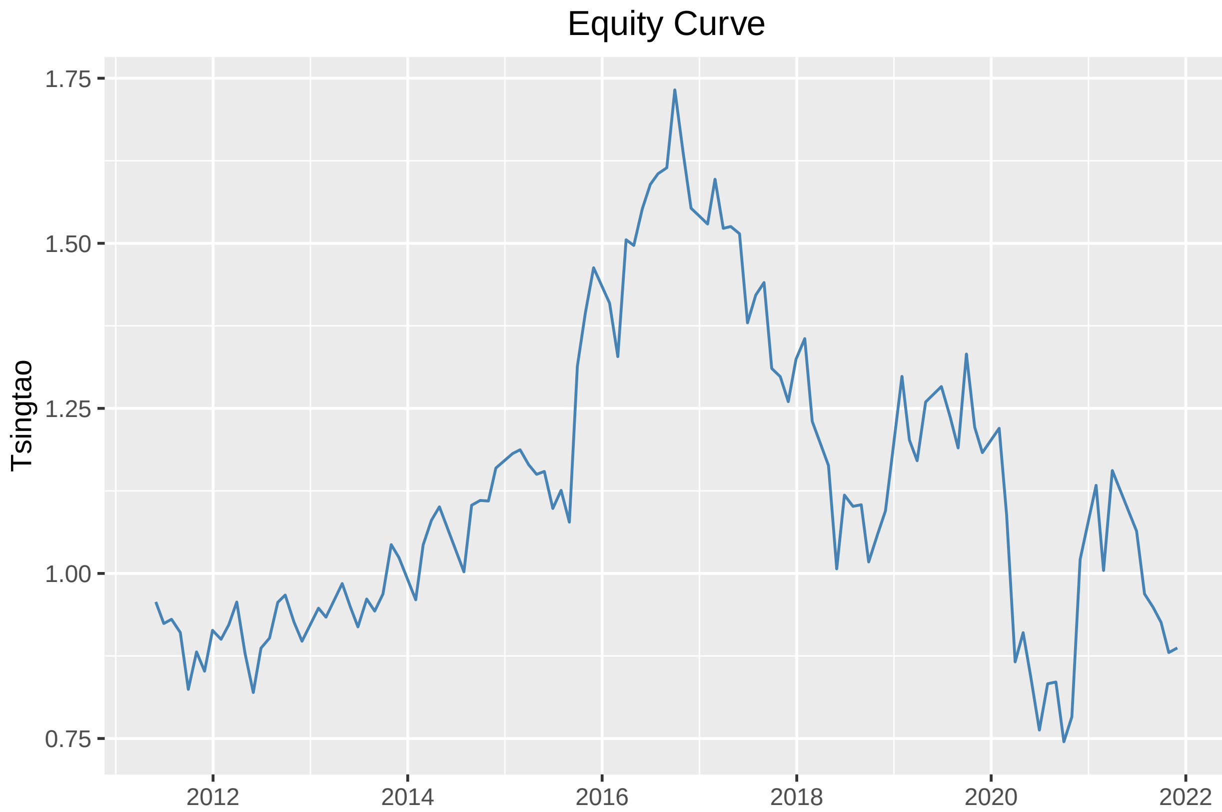

Equity Curve
Harboe
2.0
1.6
1.2
0.8
2012
2014
2016
2018
2020
2022

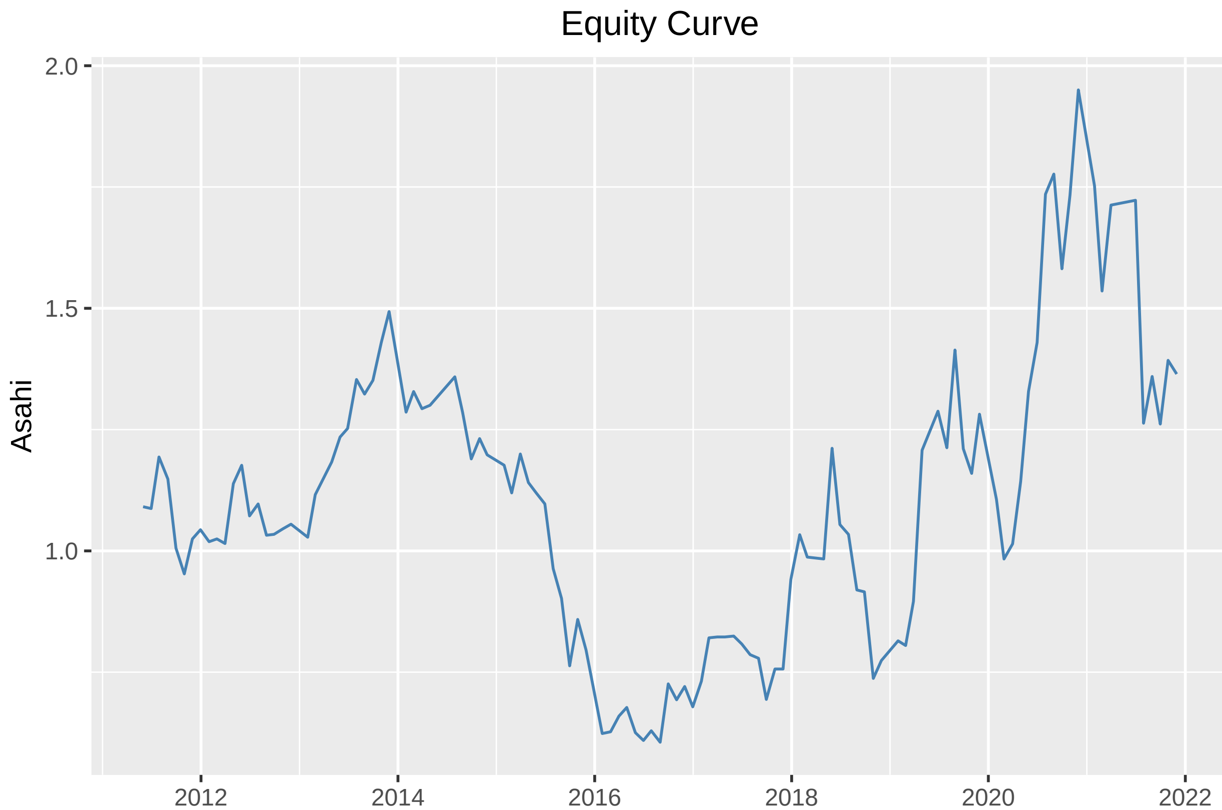


Equity Curve
Guinness
1.3
1.1
0.9
0.7
0.5
2012
2014
2016
2018
2020
2022



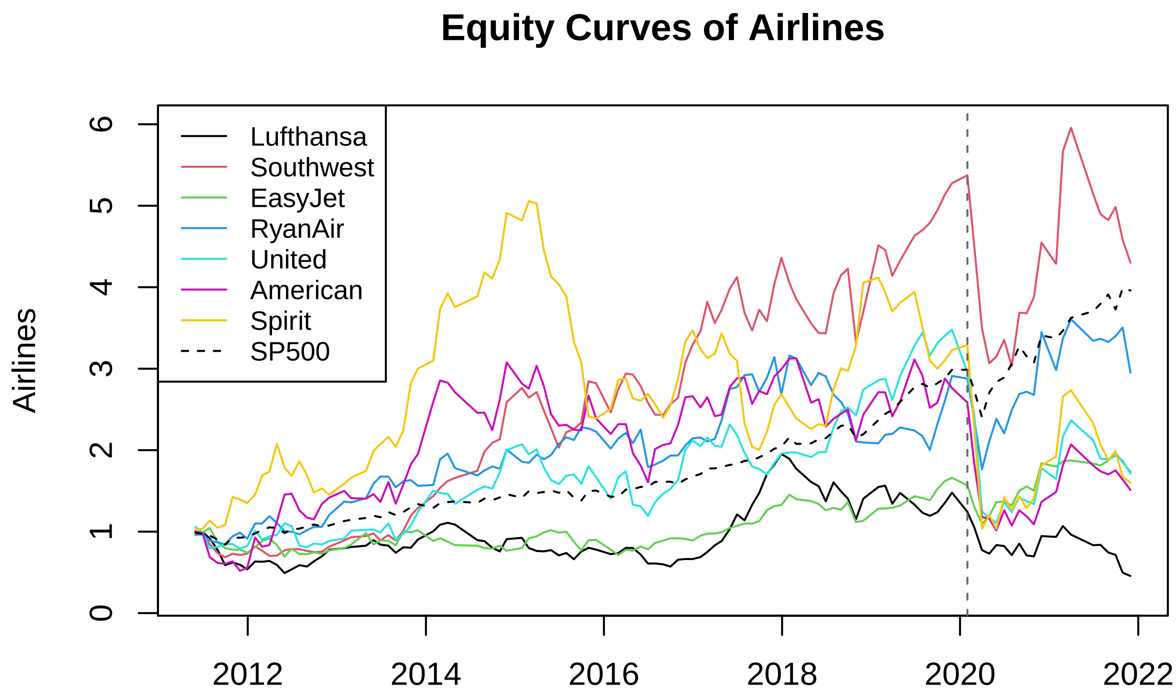
Equity Curve
Molson
2.5
2.0
1.5
1.0
2012
2014
2016
2018
2020
2022


Equity Curve
Tsingtao
1.75
1.50
1.25
1.00
0.75
2012
2014
2016
2018
2020
2022

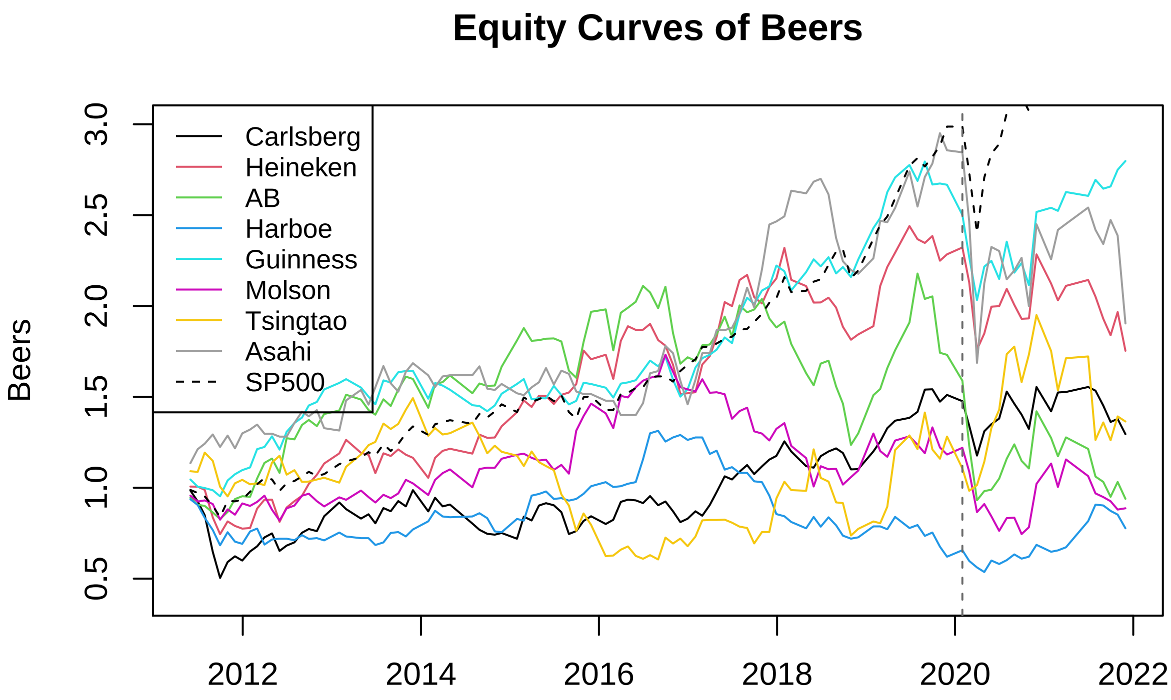


You should also provide an equity curve for each asset (that is, a curve that shows the growth of a $1 in each of the asset over the time period you chose) and comment of your results. You should do the same for S&P 500 and compare it with the assets.

```
equityCurveSP <- cumprod(1 + returnsNA[,17])

namesair <- names[[1]][1:7]
namesair <- append(namesair, "SP500")

colors <- c(1,2,3,4,5,6,7,1)
lt <- c(1,1,1,1,1,1,1,2)

#airlines
plot(equityCurve[,1], equityCurve[,2], type = "l", ylim= c(0.2,6), xlab="", ylab="Airlines", main="Equi
for(i in 1:6)
{
  lines(equityCurve[,1], equityCurve[,i+2], type = "l", col = i+1)
}
lines(equityCurve[,1],equityCurveSP,type="l", lty=2, lwd=1, col="black")
abline(v=as.Date("2020-01-31"), lty=2, col="dimgrey")
legend(x='topleft', legend=namesair,col=colors, lty=lt, cex=0.8)
```



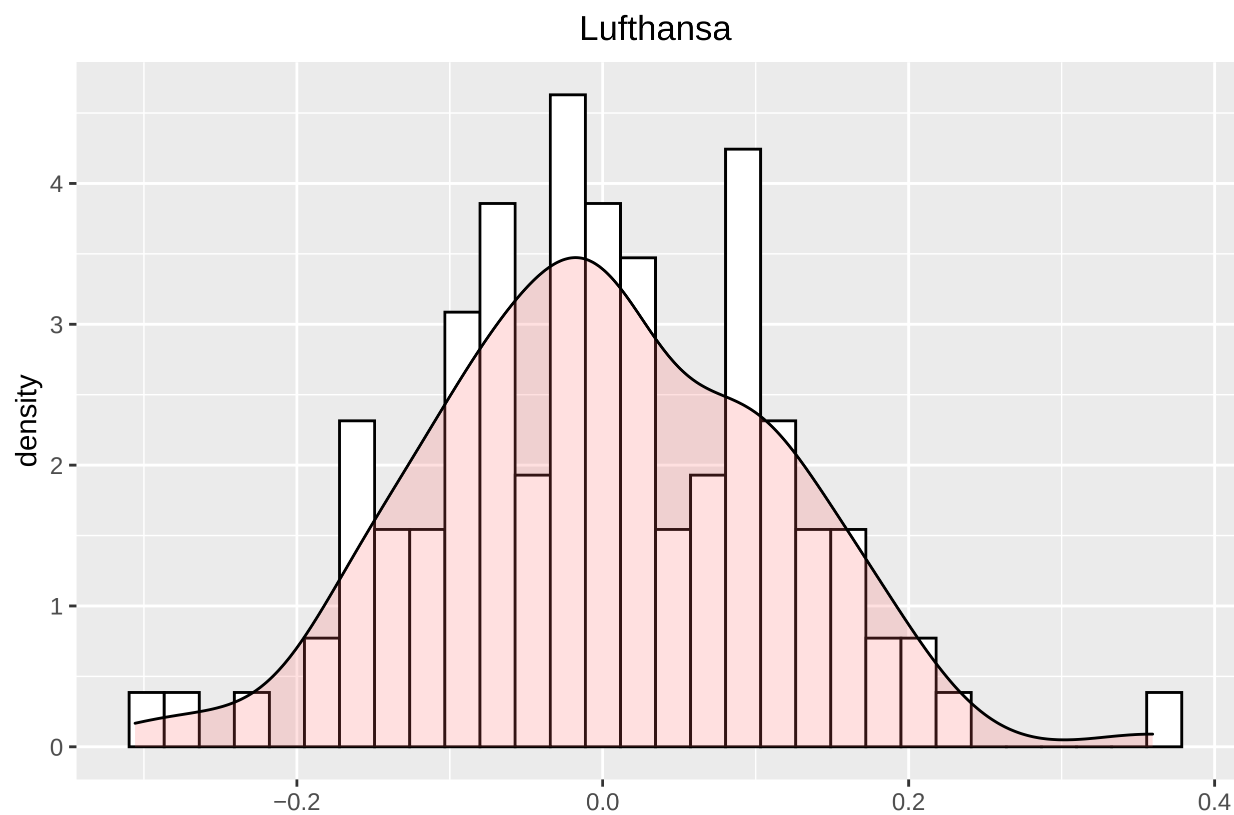

```
##################################################################

colors <- c(1,2,3,4,5,6,7,8,1)
lt <- c(1,1,1,1,1,1,1,1,2)

#beers
plot(equityCurve[,1], equityCurve[,9], type = "l", ylim= c(0.4,3), xlab="", ylab="Beers", main="Equity C
for(i in 1:7)
{
  lines(equityCurve[,1], equityCurve[,i+9], type = "l", col = i+1)
}
lines(equityCurve[,1],equityCurveSP,type="l", lty=2, lwd=1, col="black")
abline(v=as.Date("2020-01-31"), lty=2, col="dimgrey")
legend(x='topleft', legend=names[[1]][8:16],col=colors, lty=lt, cex=0.8)
```



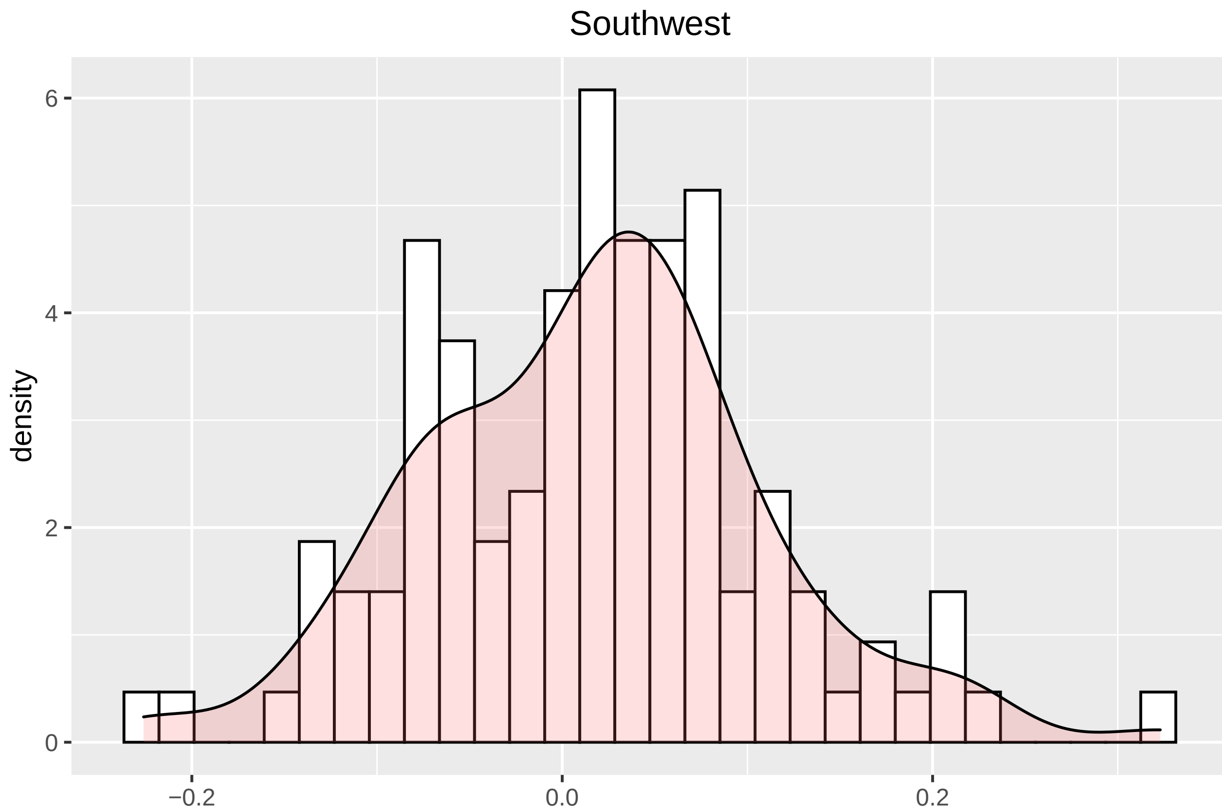
4. Histograms for each return series

```
for (i in 1:15)
{
  names <- list(stock_names)
  print(ggplot(returnsNA, aes(x=returnsNA[,i+1])) +
  geom_histogram(aes(y=..density..), colour="black", fill="white") +
  geom_density(alpha=.2, fill="#FF6666")+
  xlab("") +
  ggtitle(names[[1]][i])+
  theme(plot.title = element_text(hjust = 0.5)))
}
```

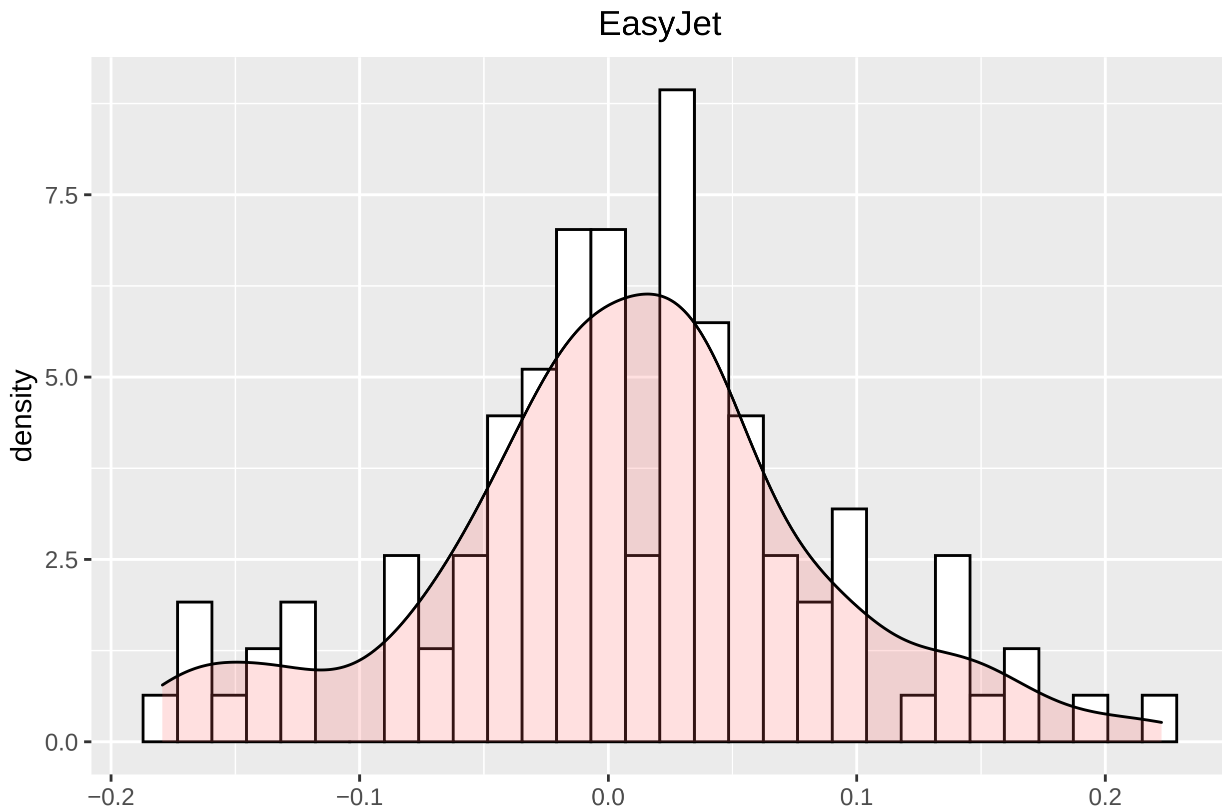
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



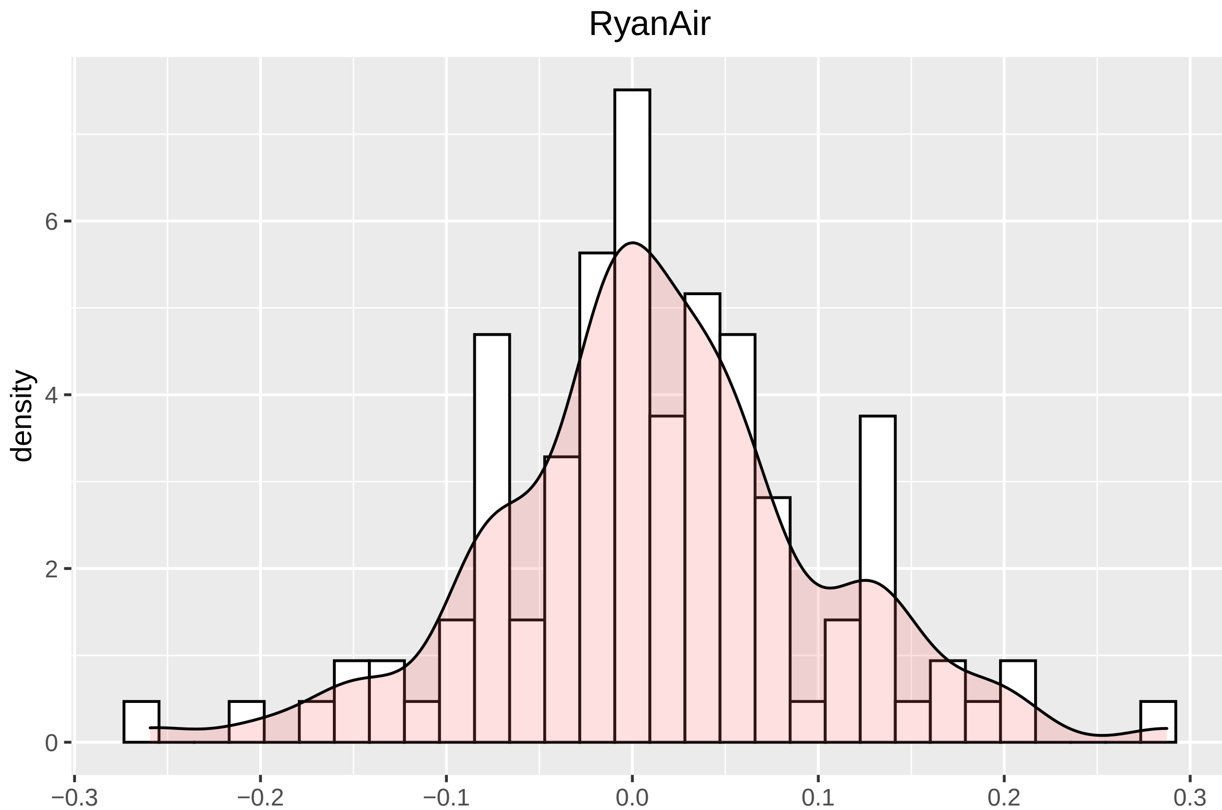

```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```



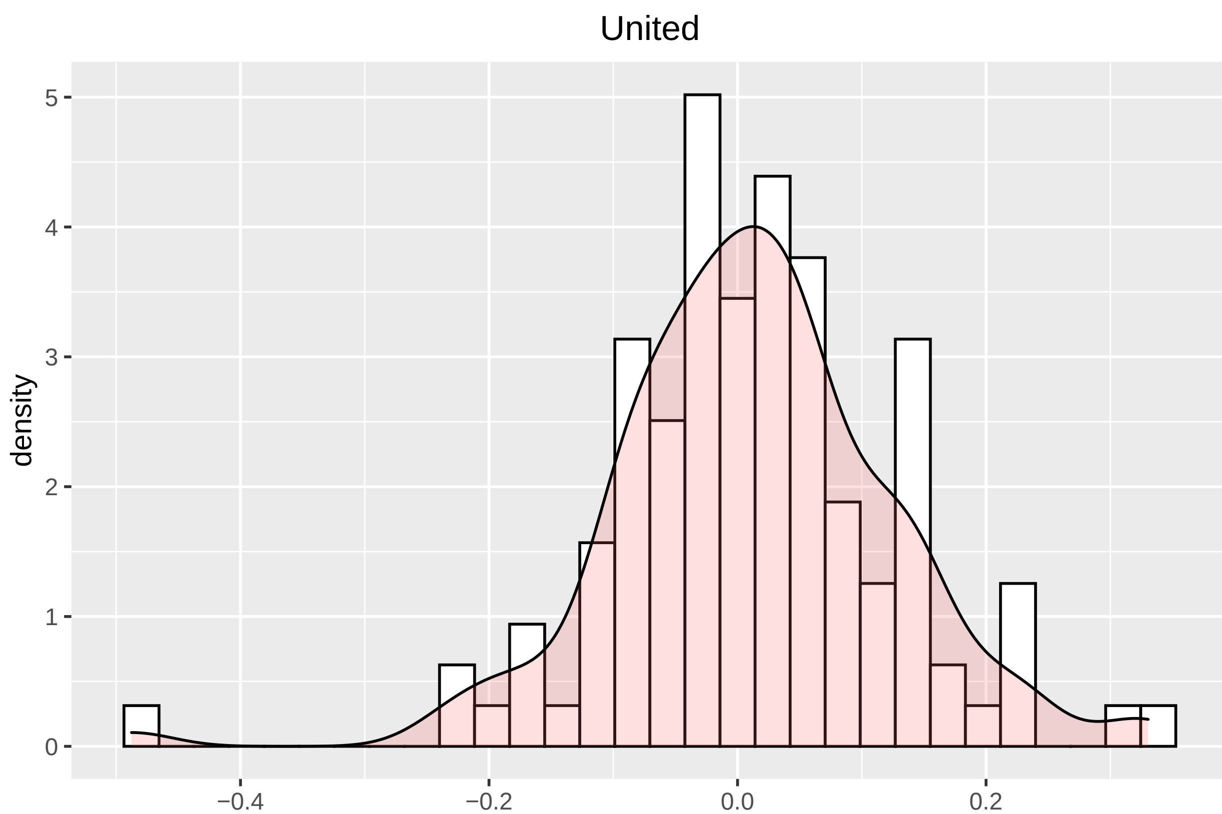

```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```



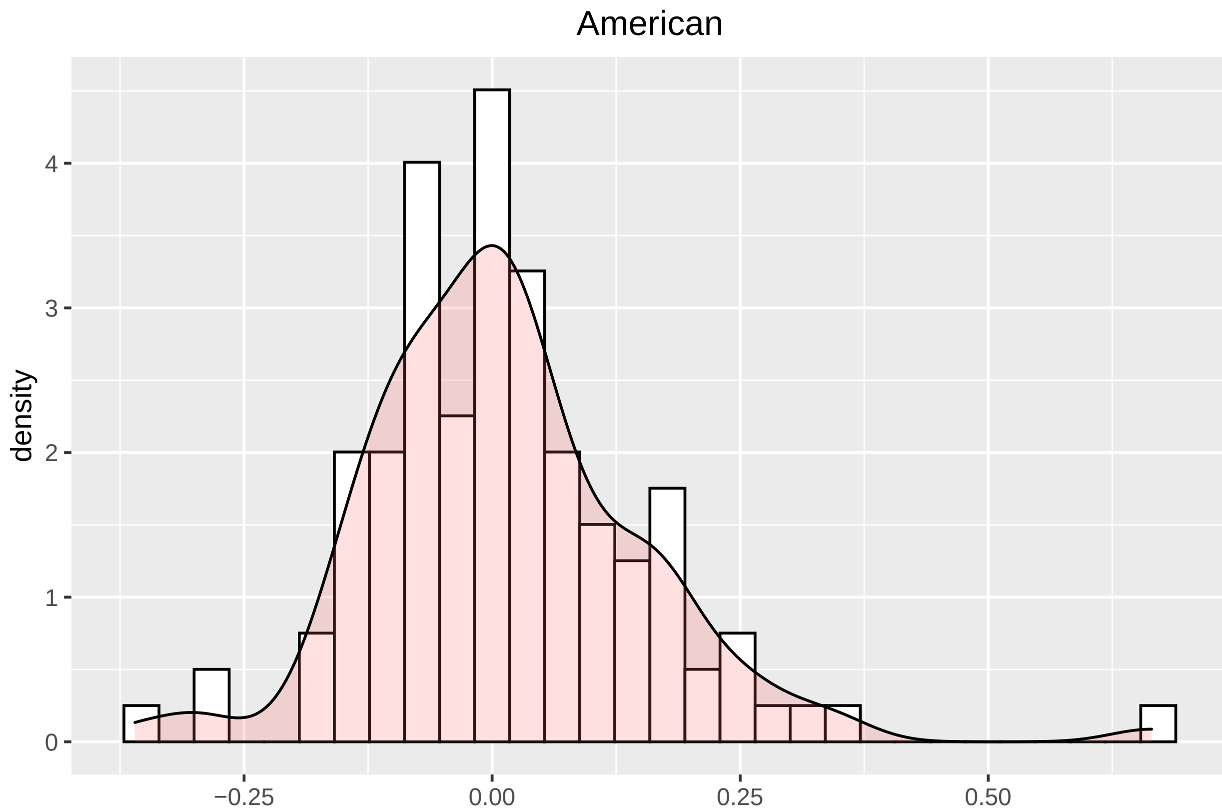

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



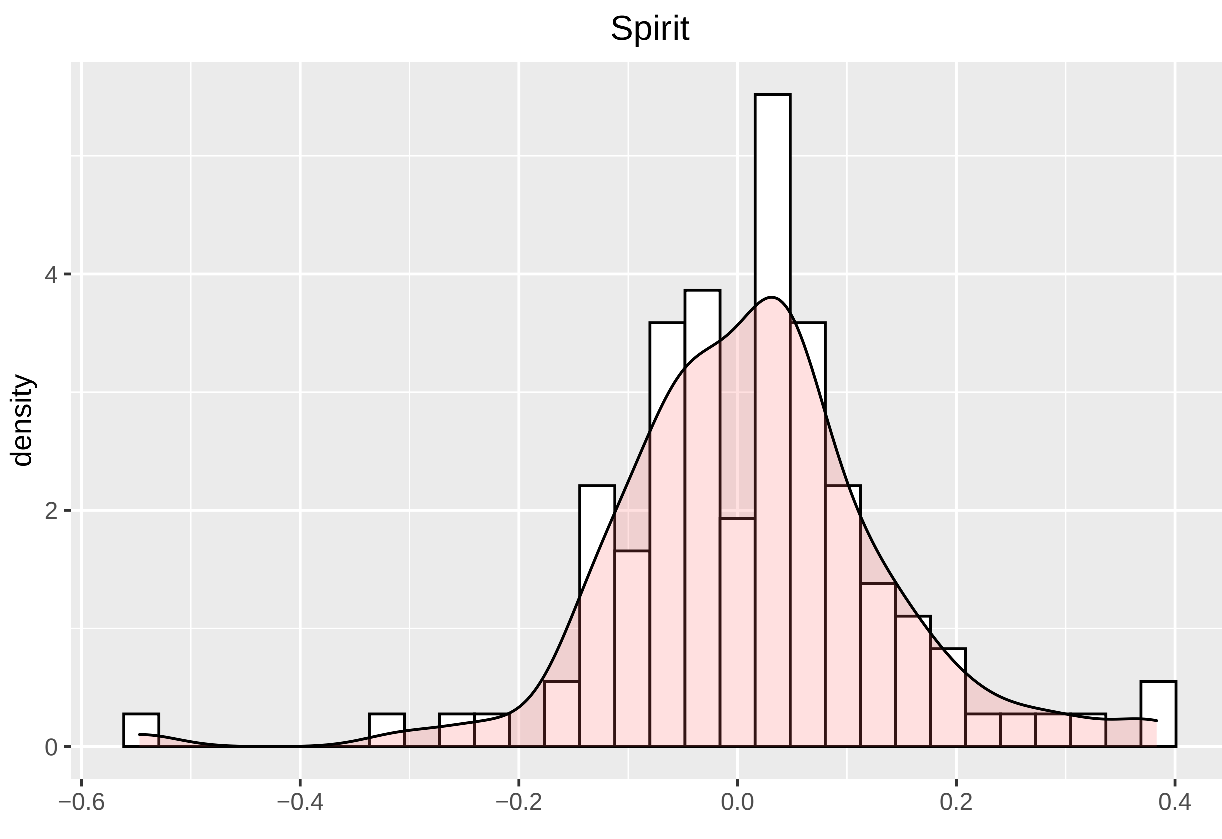

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



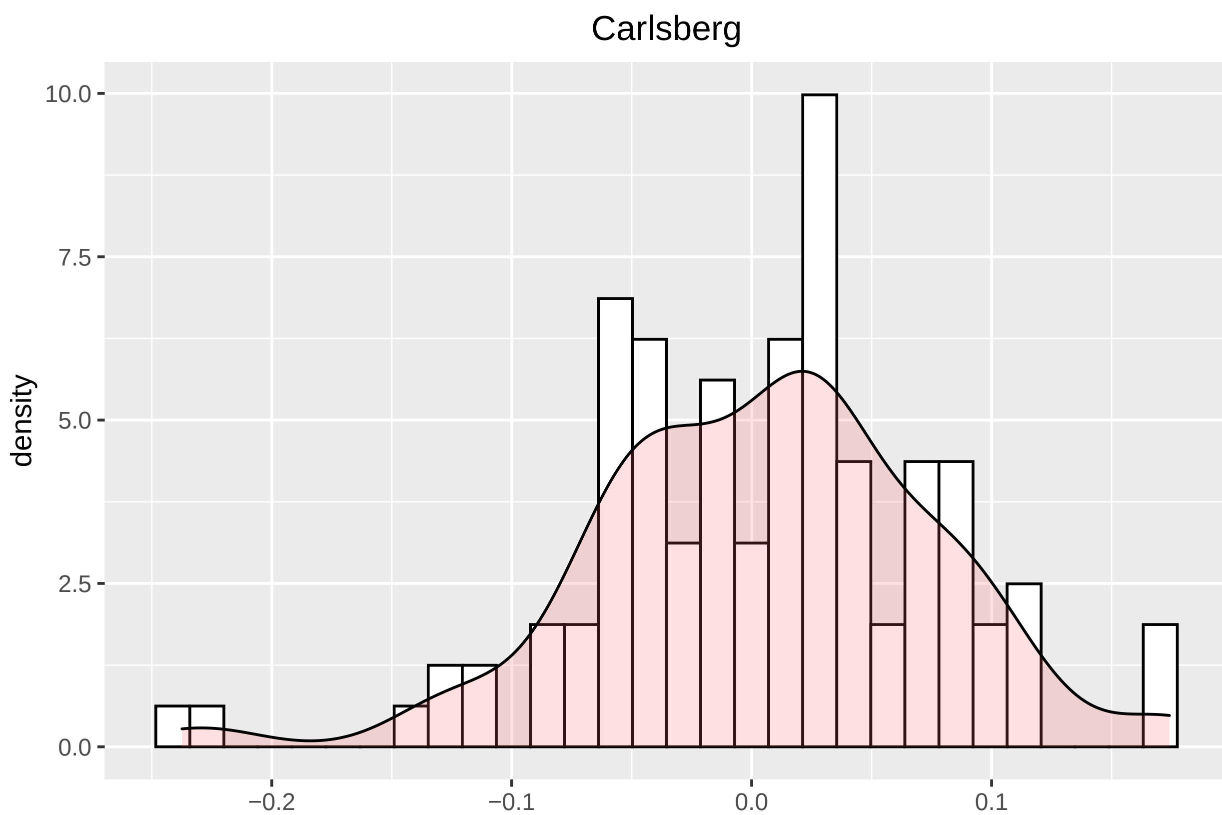

```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```



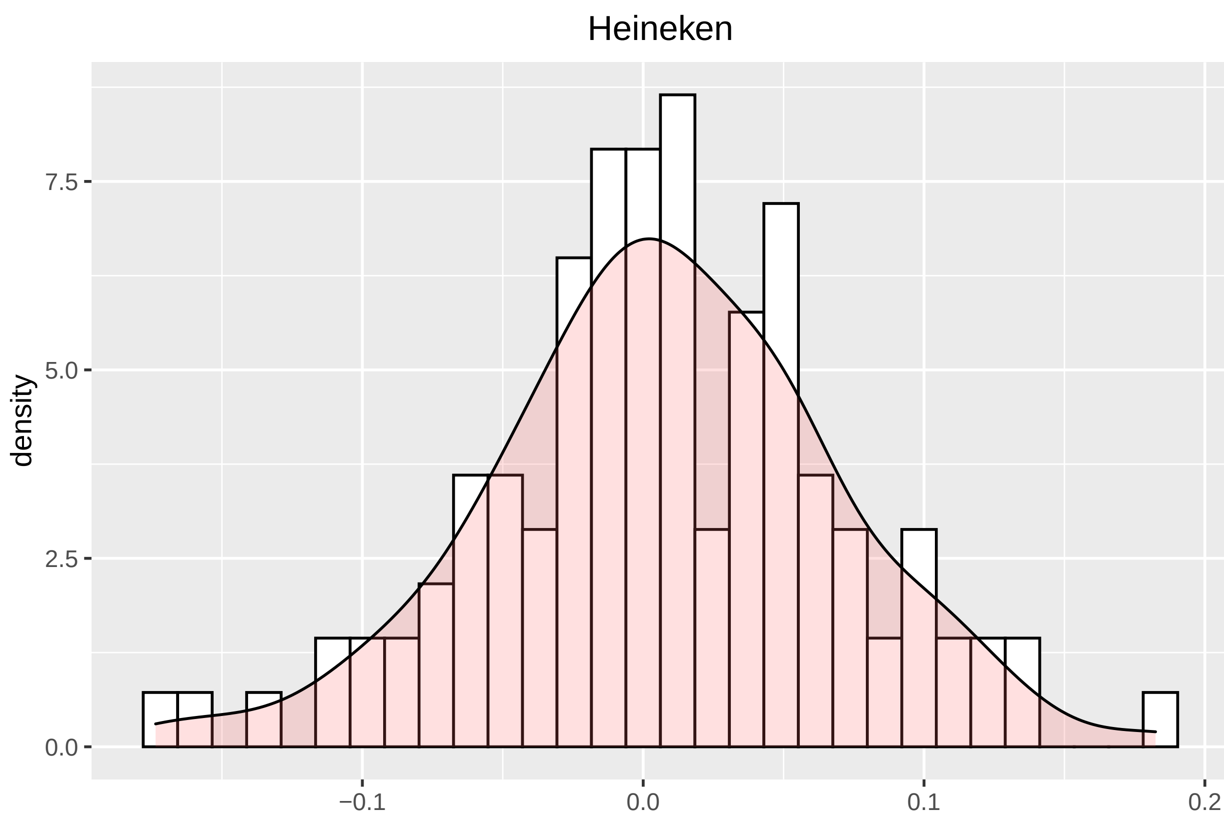

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



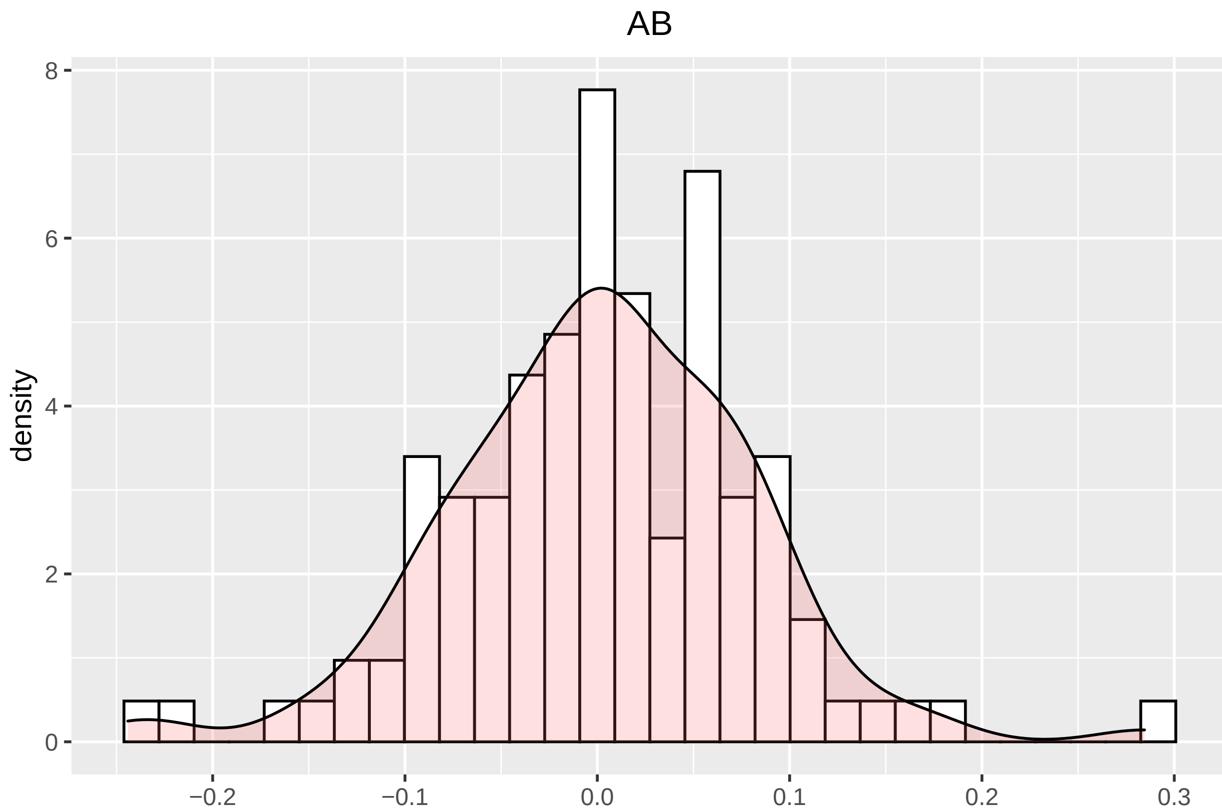

```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```



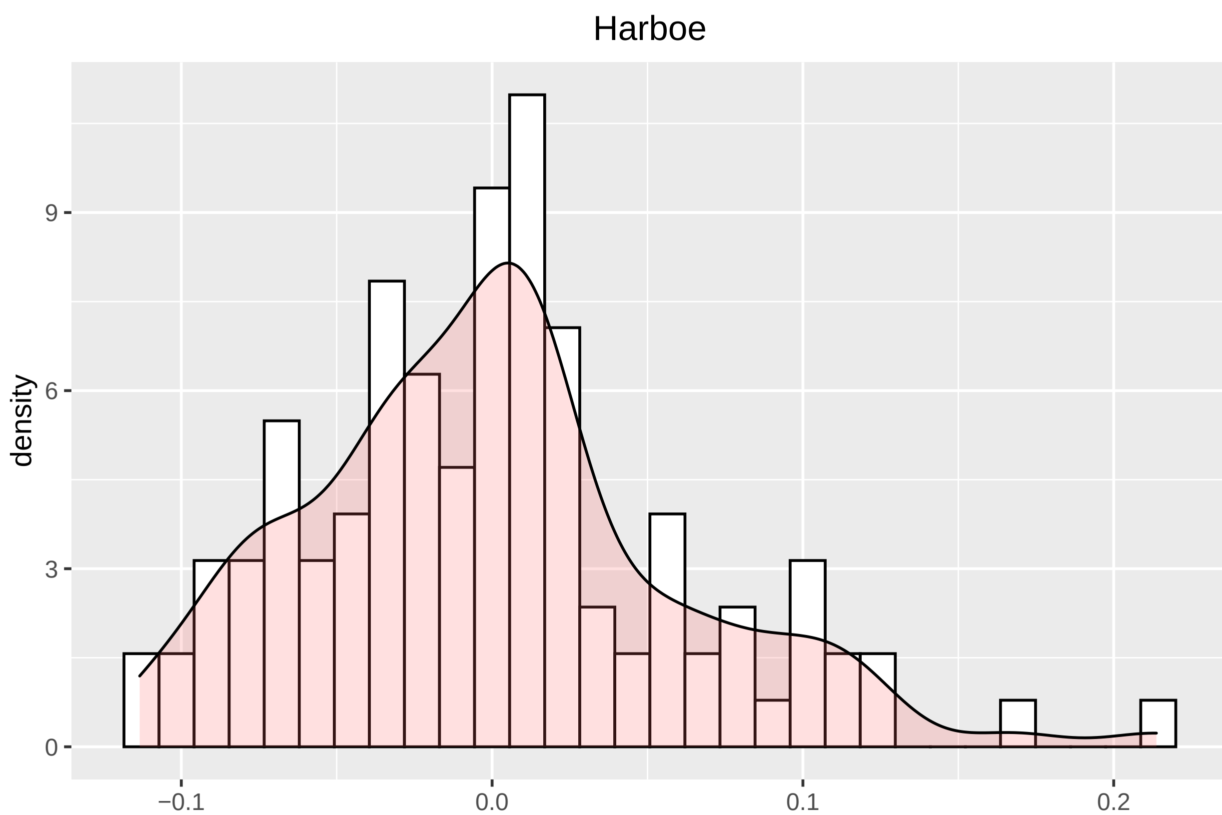

```
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
```



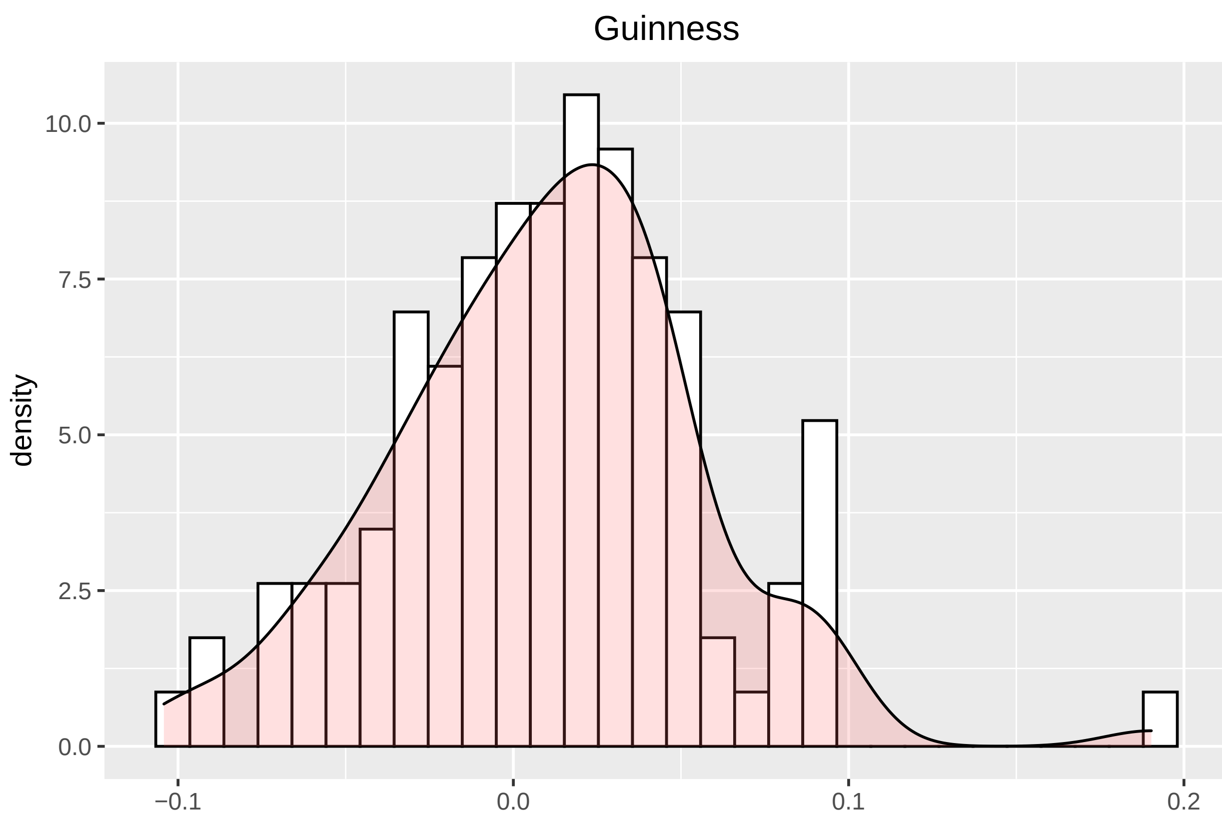

```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```



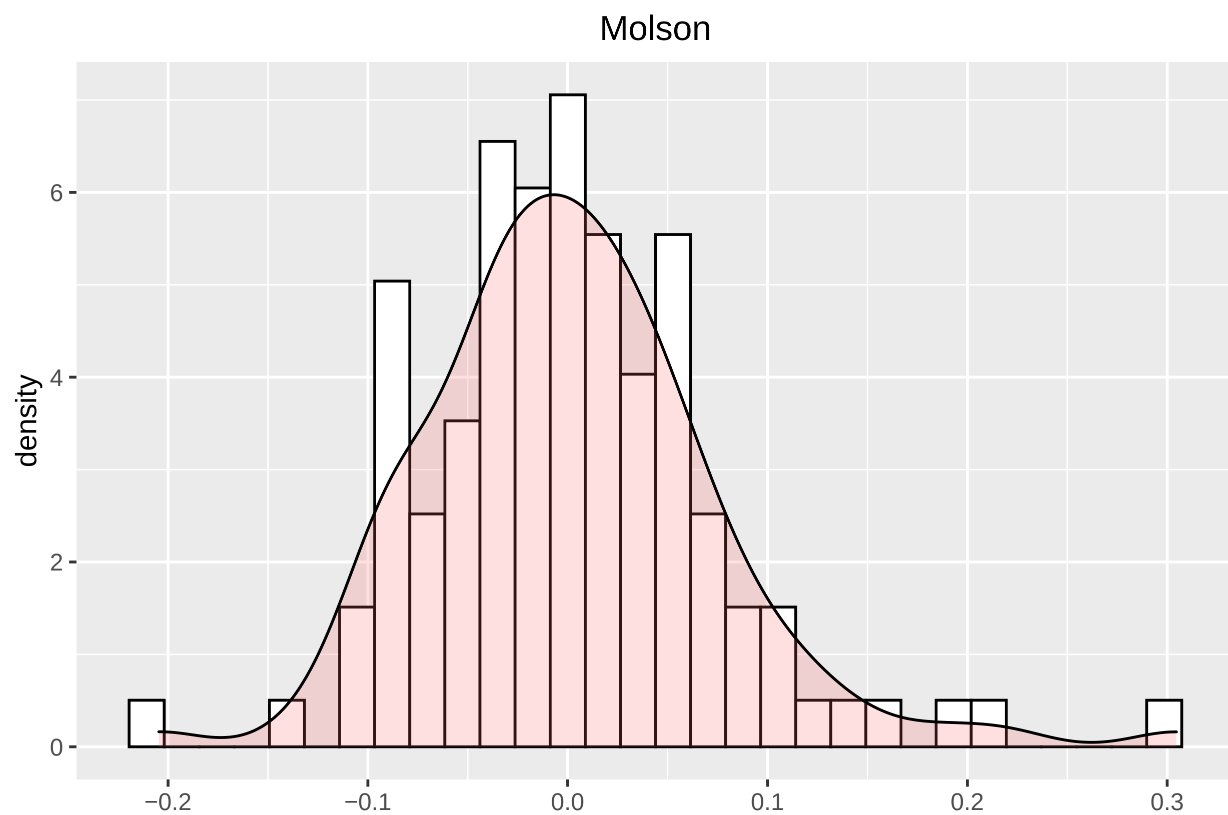

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



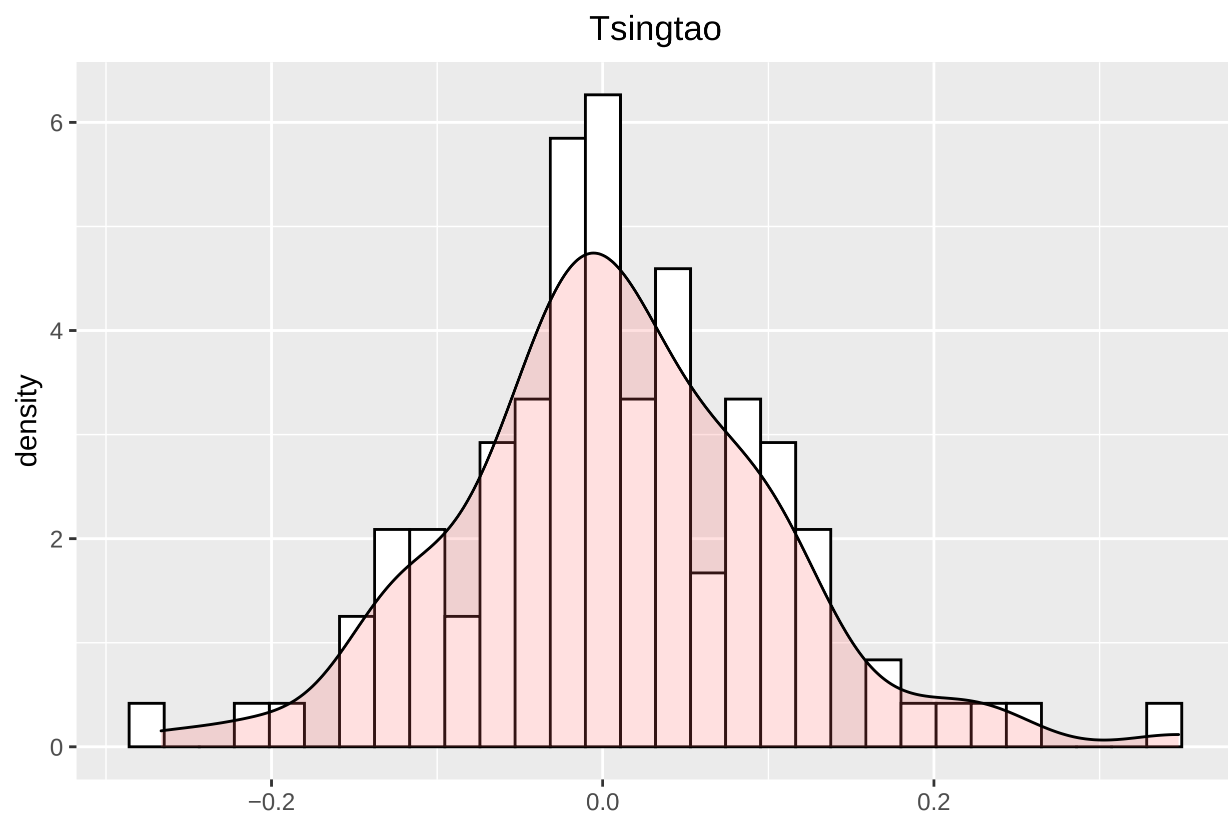

```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```



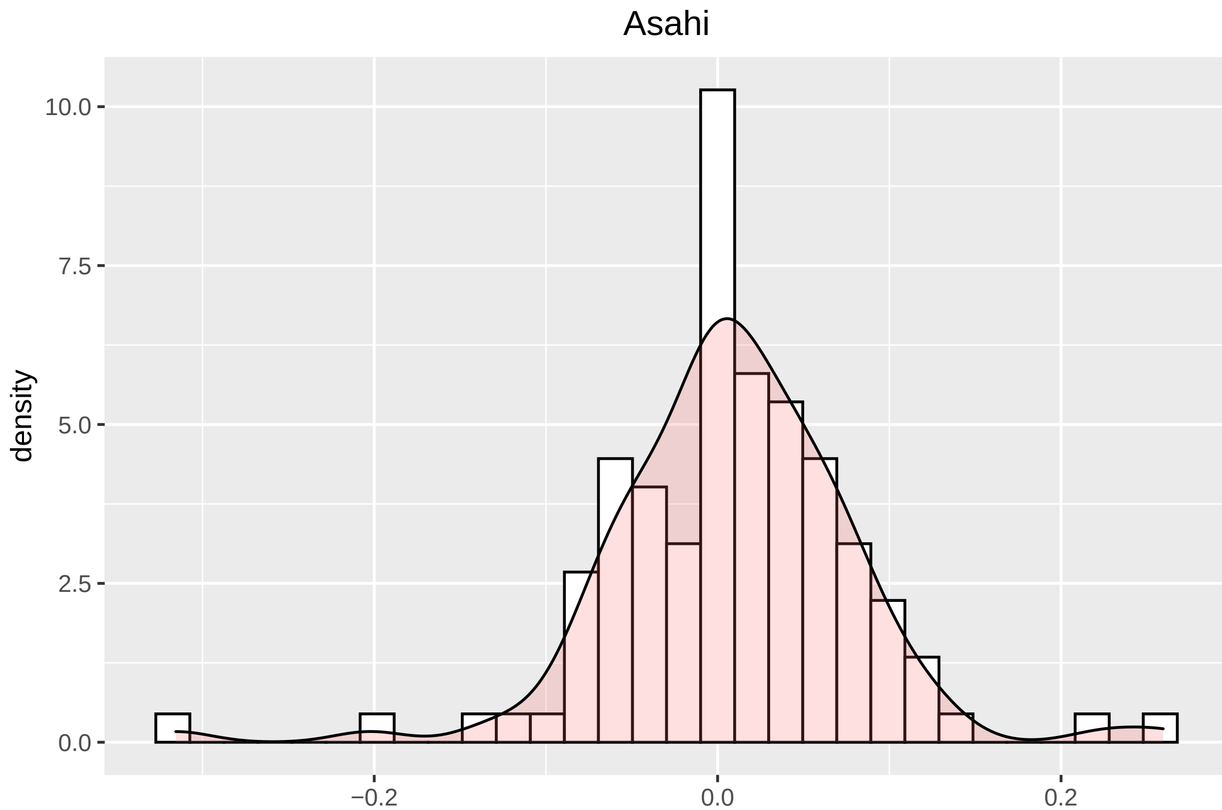

```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```




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

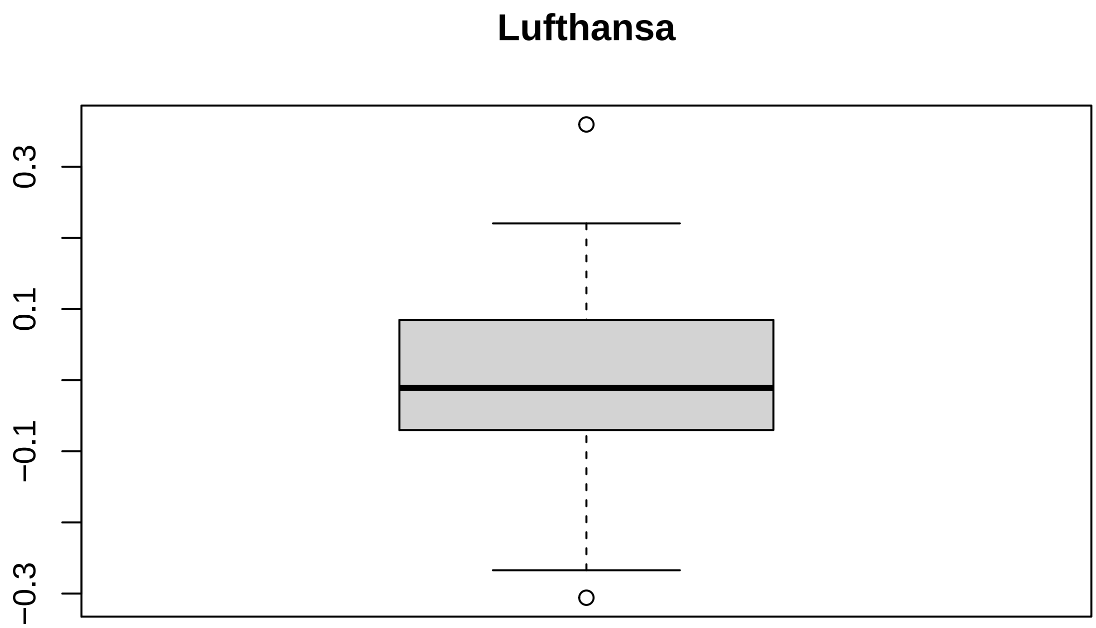



```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```

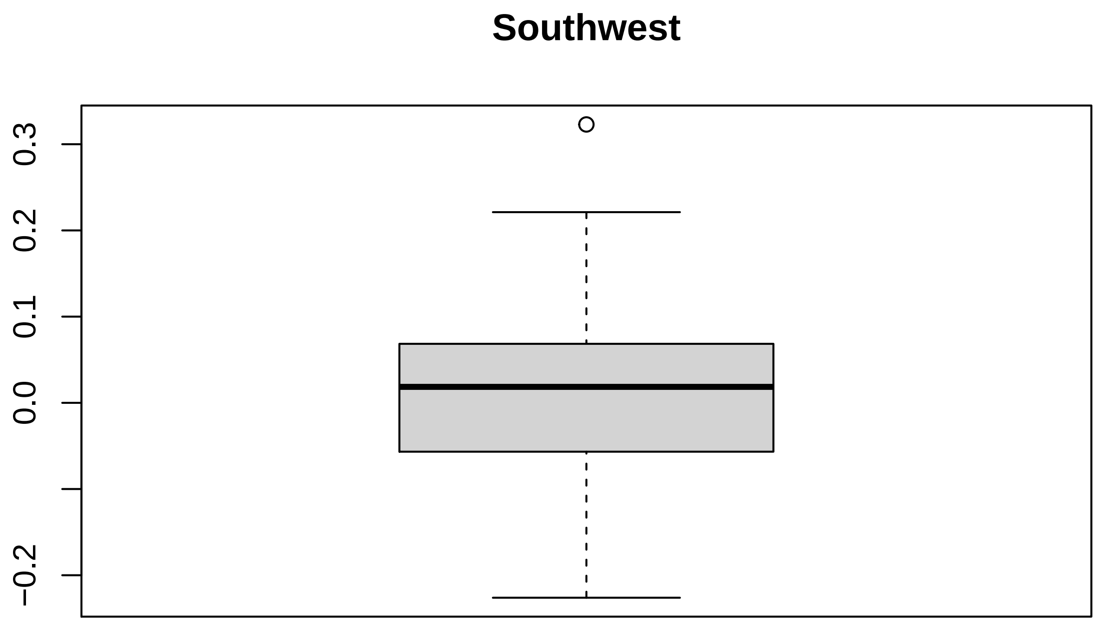


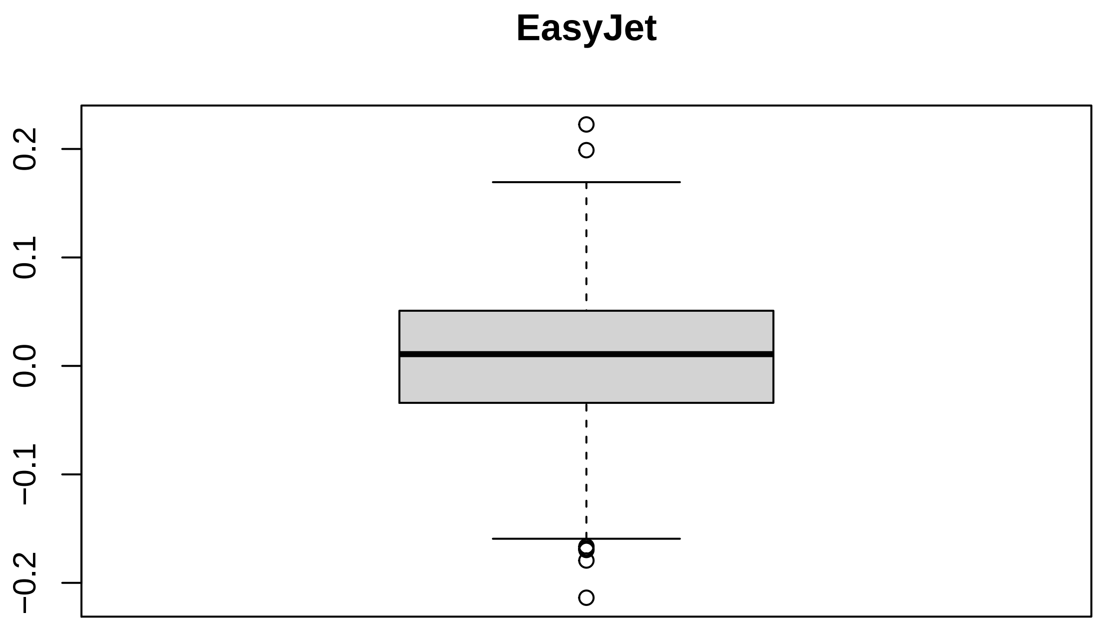
5. Boxplots for each return series

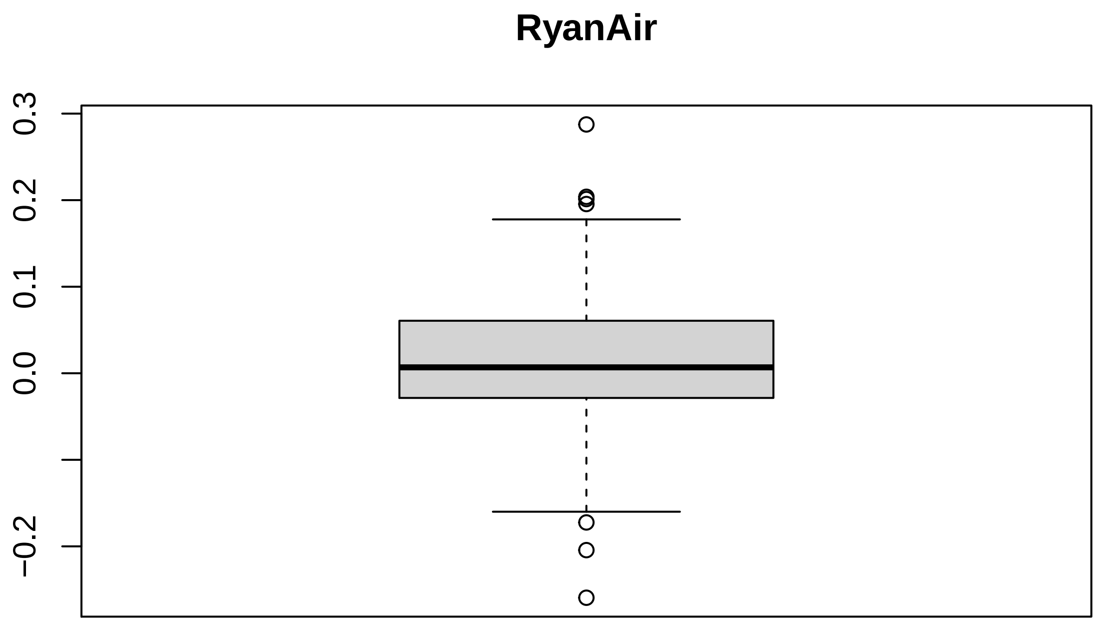
```
#individual boxplots

for (i in 1:15)
{
  names <- list(stock_names)
  boxplot(returnsIVV[,i+1], xlab="", main = names[[1]][i])
}
```

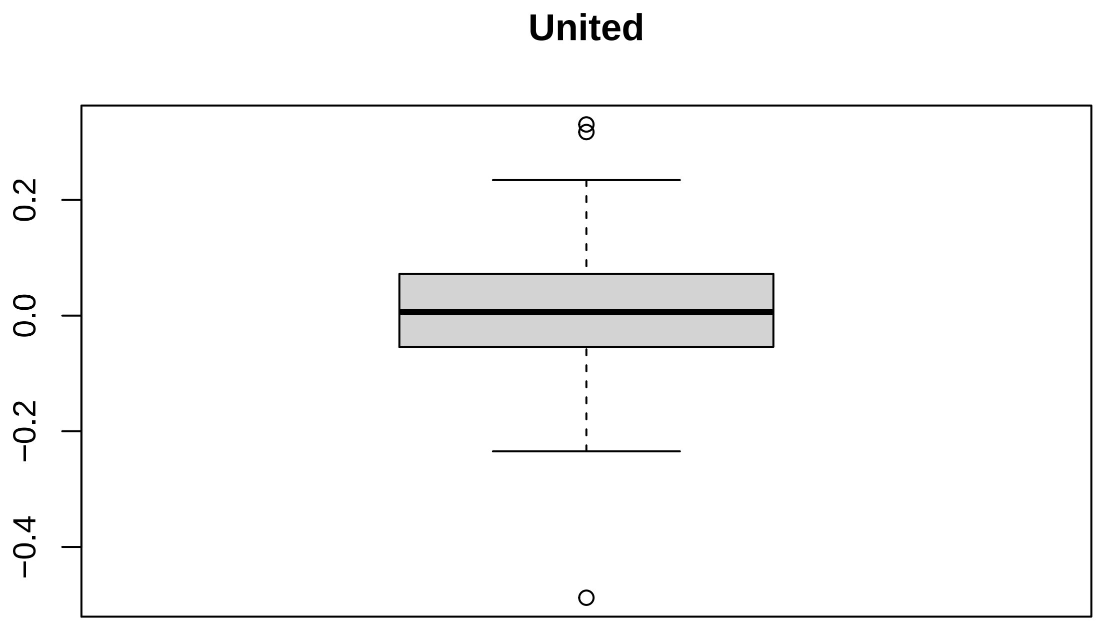


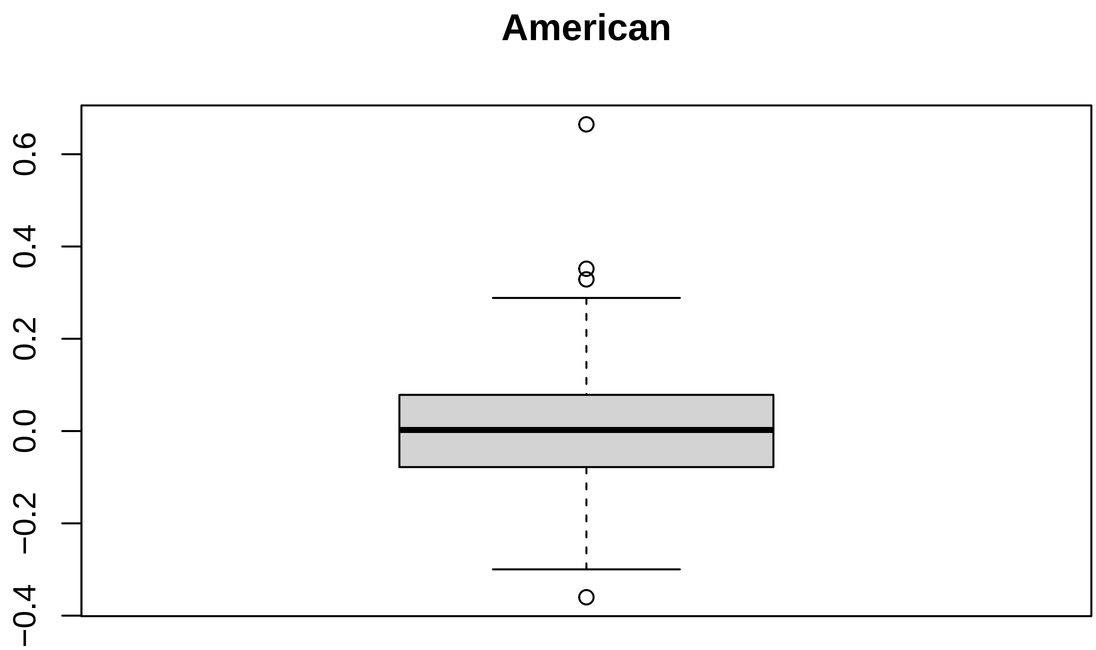
Lufthansa
0.3
0.1
−0.1
−0.3

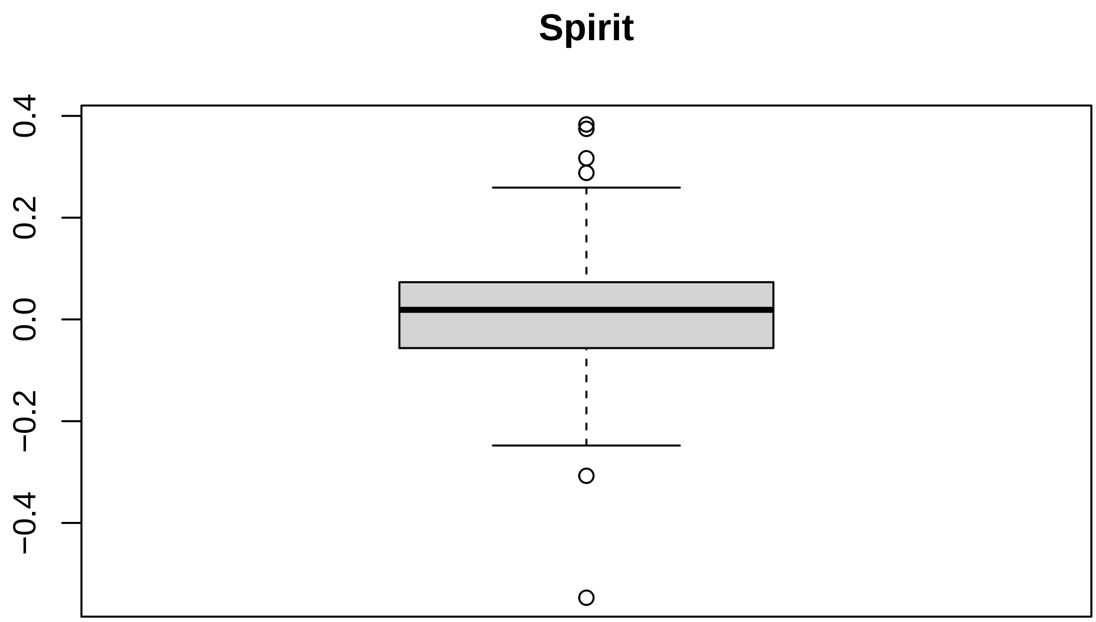

Southwest
0.3
0.2
0.1
0.0
−0.2

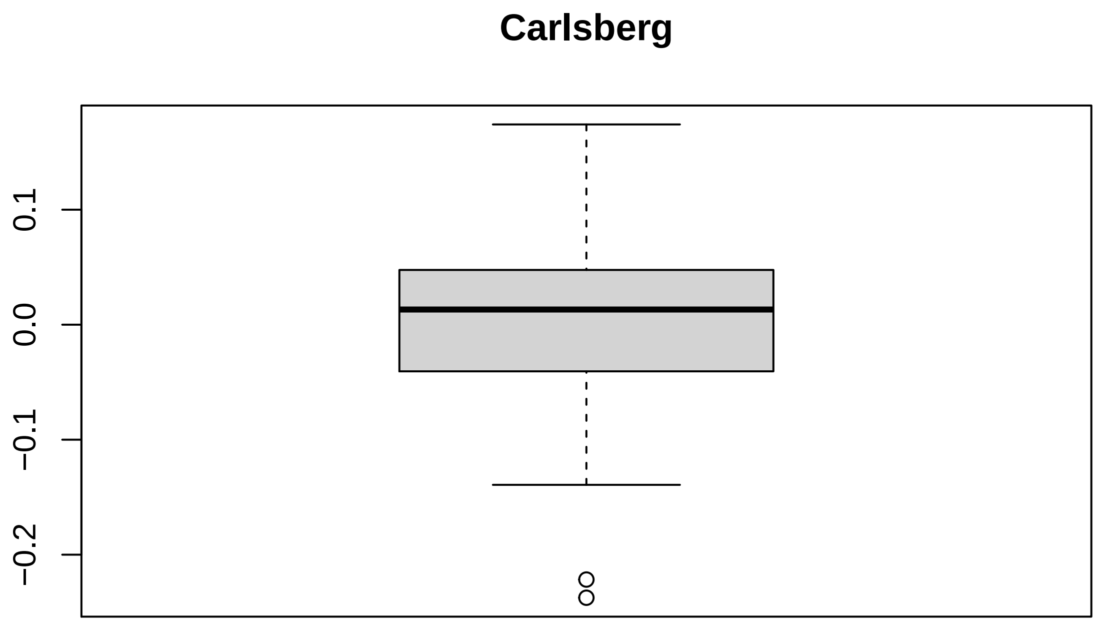

EasyJet
0.2
0.1
0.0
−0.1
−0.2

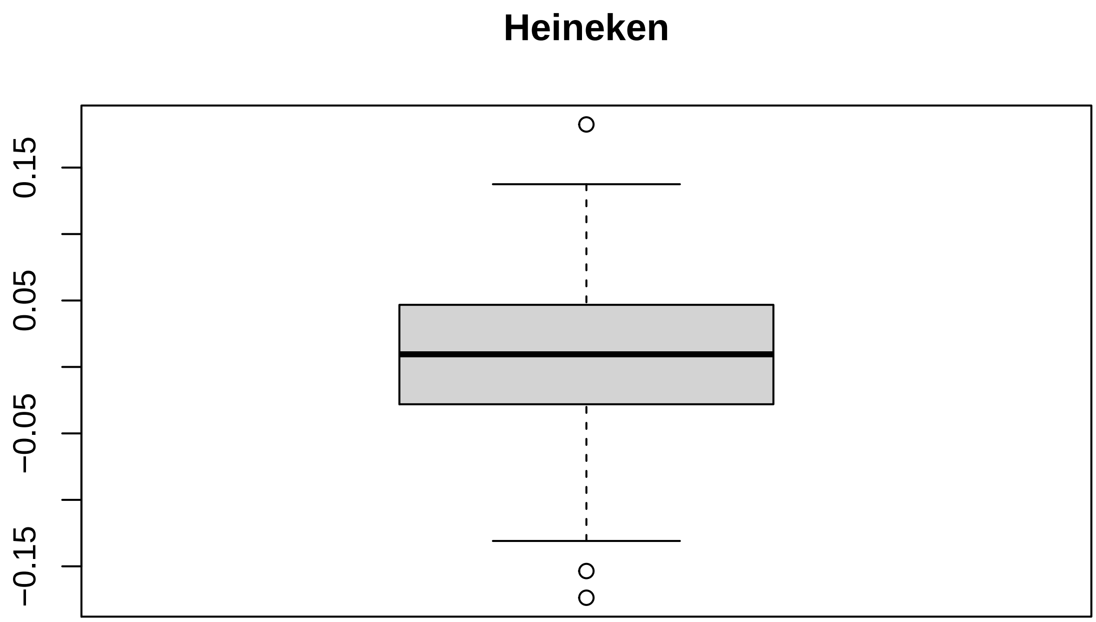

RyanAir
0.3
0.2
0.1
0.0
−0.2

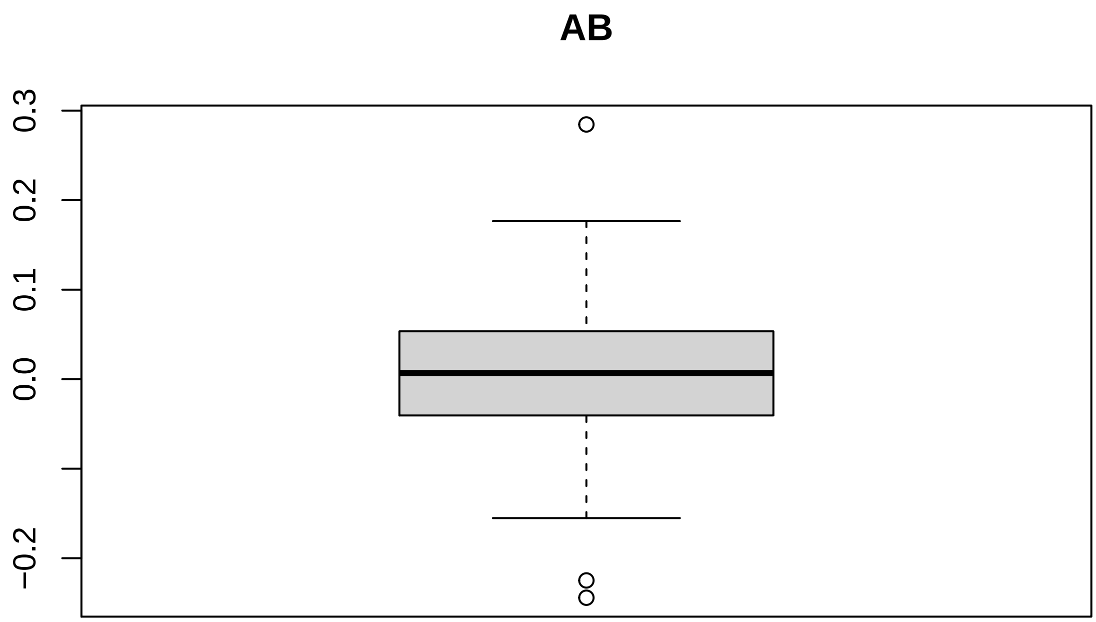

United
0.2
0.0
−0.2
−0.4


American
0.6
0.4
0.2
0.0
−0.2
−0.4

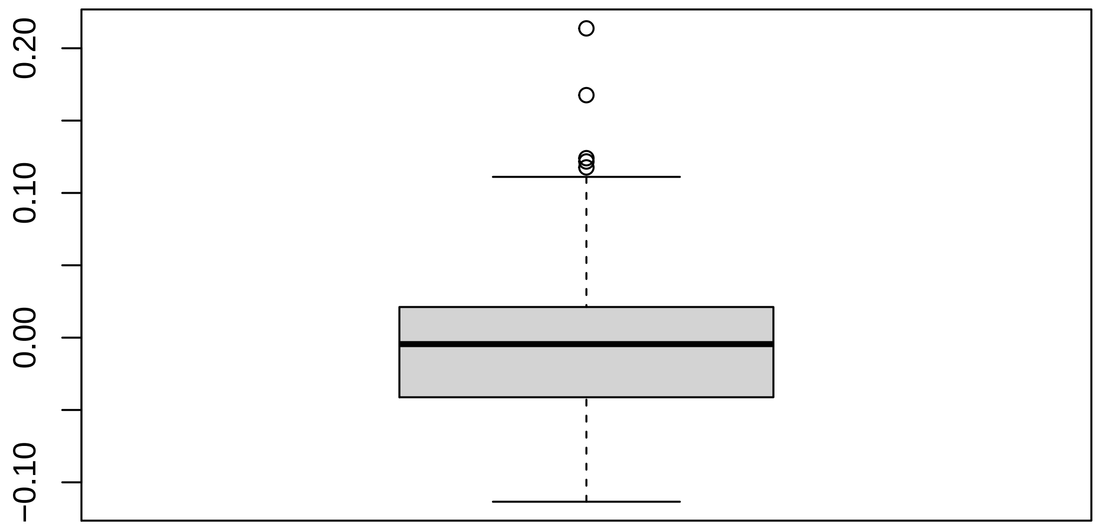

Spirit
0.4
0.2
0.0
−0.2
−0.4

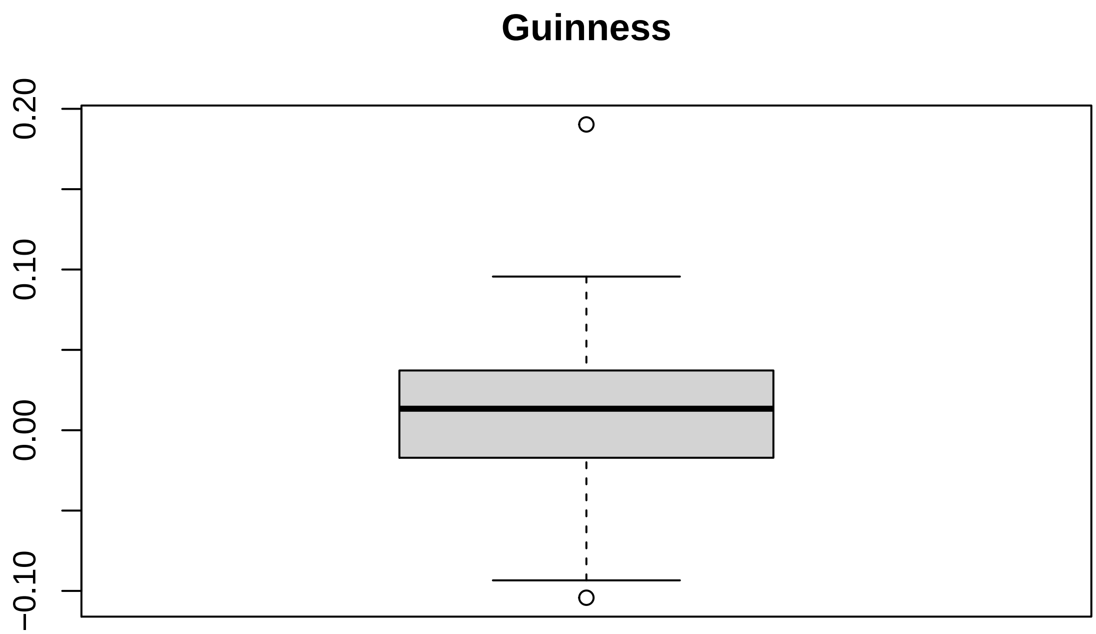

Carlsberg
0.1
0.0
−0.1
−0.2

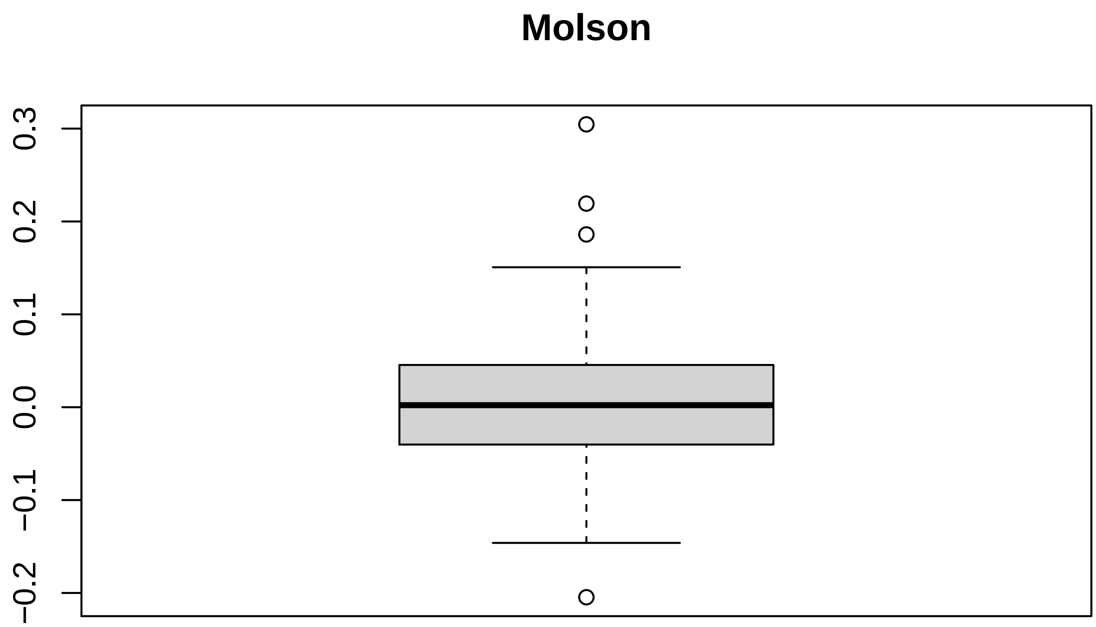

Heineken
0.15
0.05
−0.05
−0.15


AB
0.3
0.2
0.1
0.0
−0.2

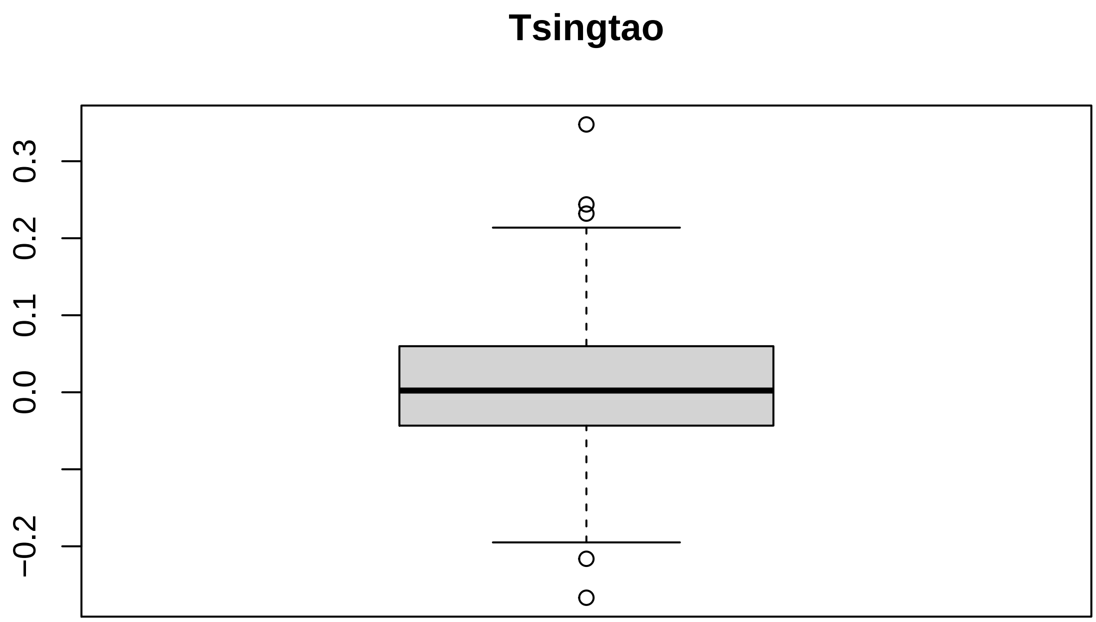
Harboe

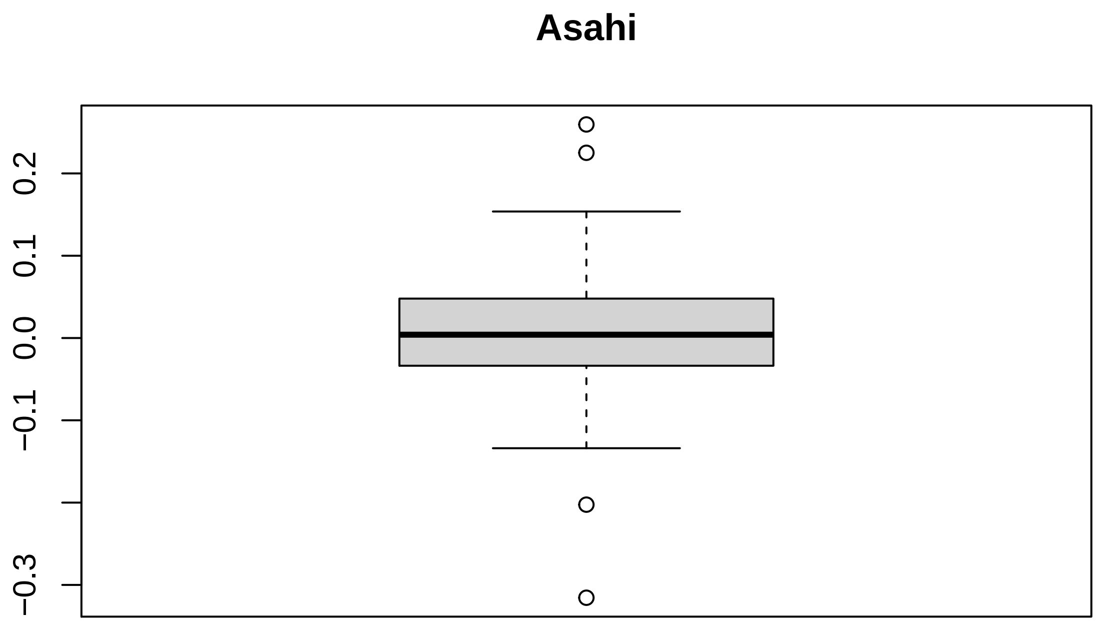



Guinness
0.20
0.10
0.00
−0.10



Molson
0.3
0.2
0.1
0.0
−0.1
−0.2

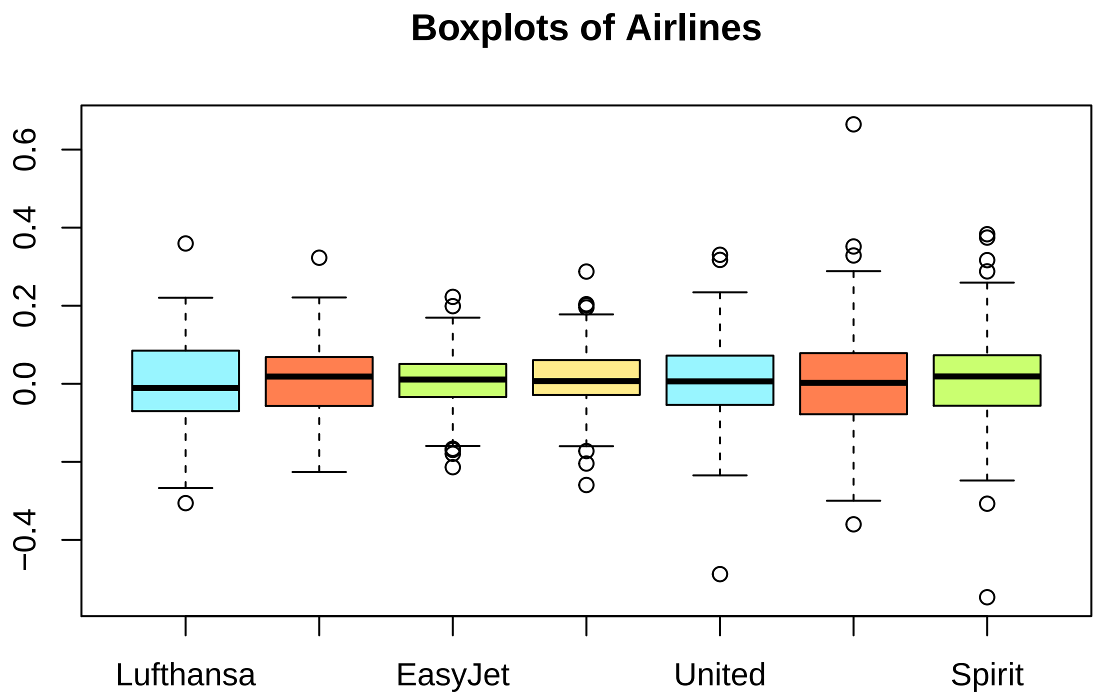

Tsingtao
0.3
0.2
0.1
0.0
−0.2



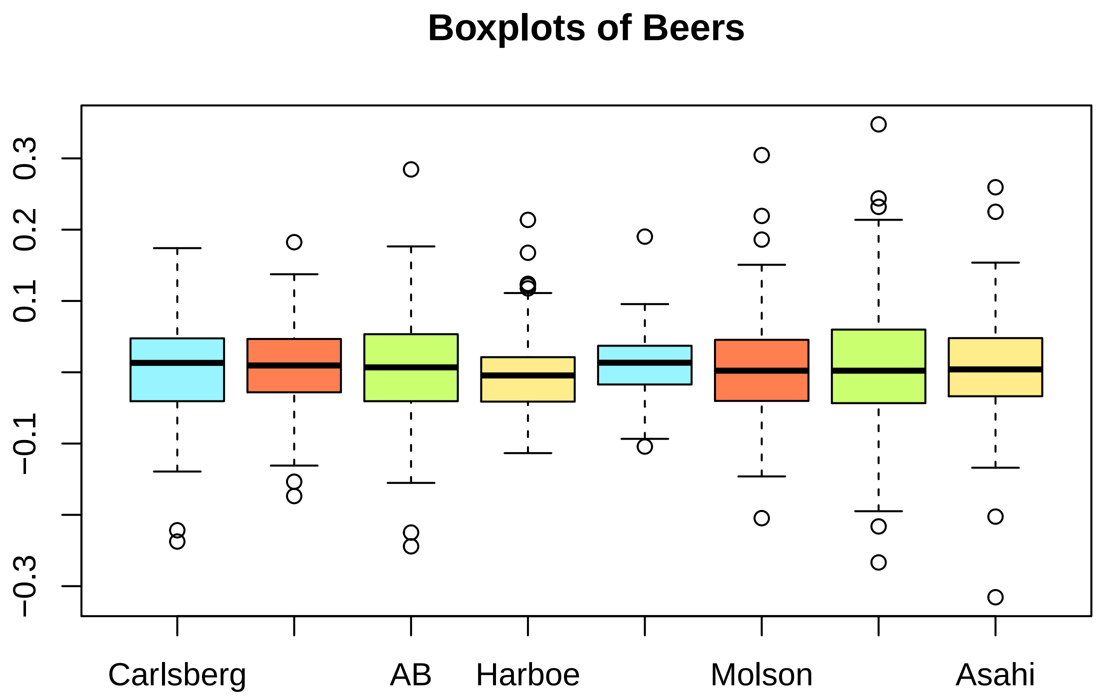

```
returnsbox <- returnsIVV
colnames(returnsbox)[2:16] <- names[[1]]
```

```
## Warning in colnames(returnsbox)[2:16] <- names[[1]]: number of items to replace
## is not a multiple of replacement length
```

```
colors <- c('cadetblue1', 'coral', 'darkolivegreen1', 'lightgoldenrod1')

#airlines
boxplot(returnsbox[,2:8], col = colors, main= "Boxplots of Airlines")
```

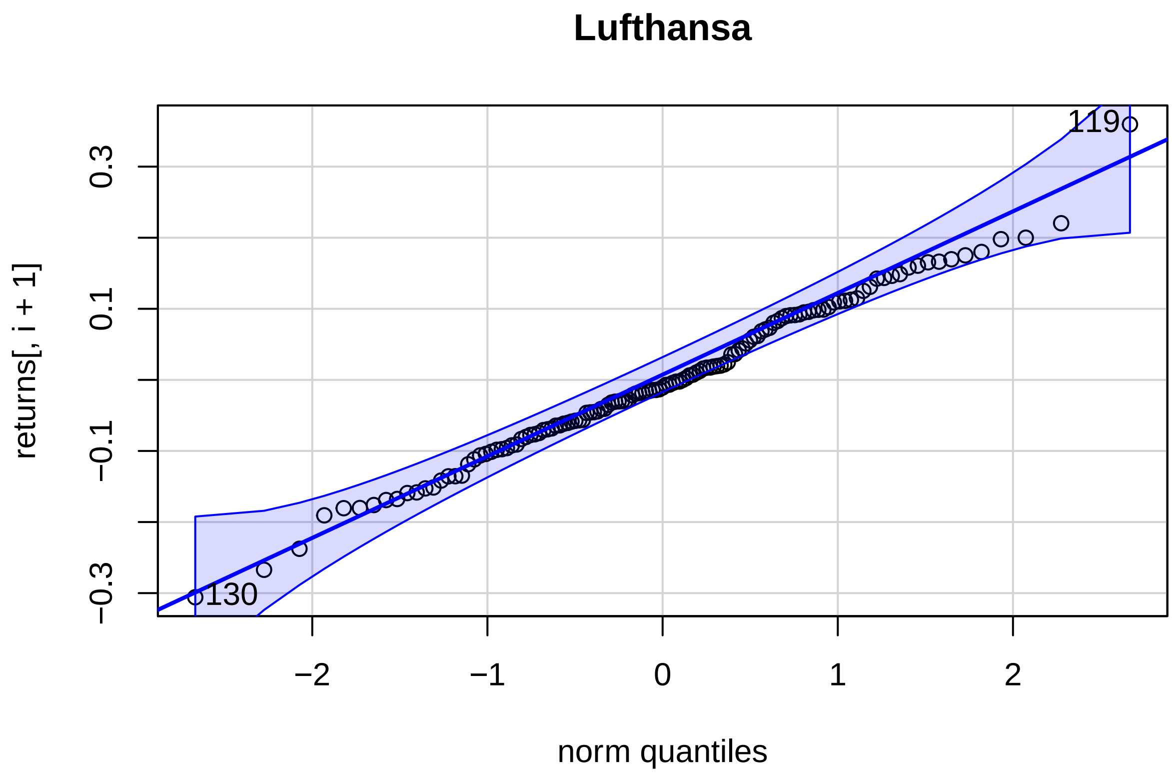


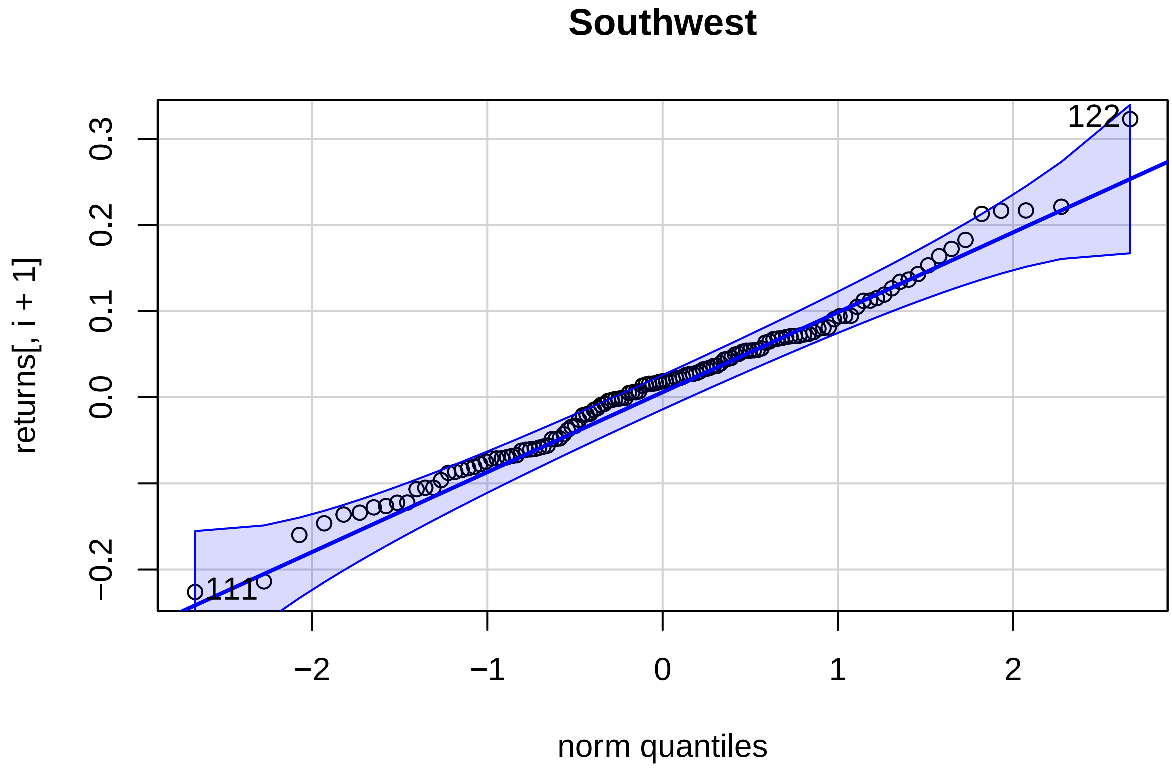

```
#beers
boxplot(returnsbox[,9:16], col= colors, main = "Boxplots of Beers")
```

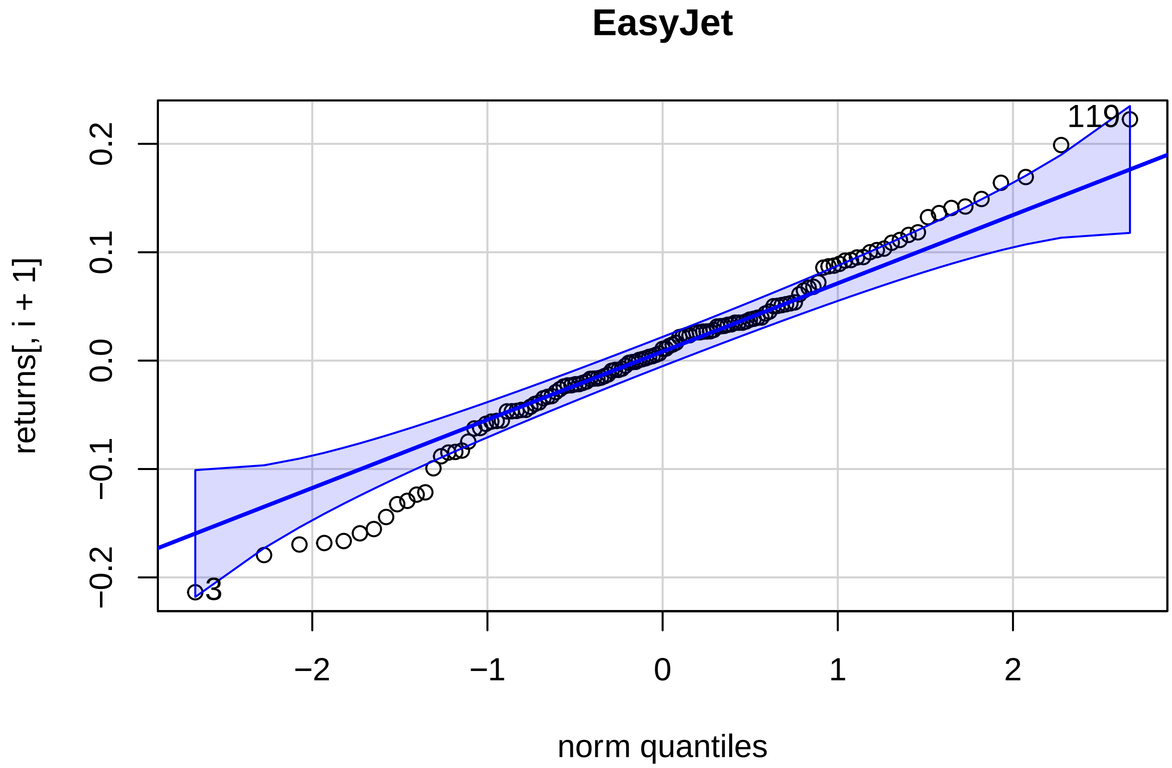


6. qqplots for each return series

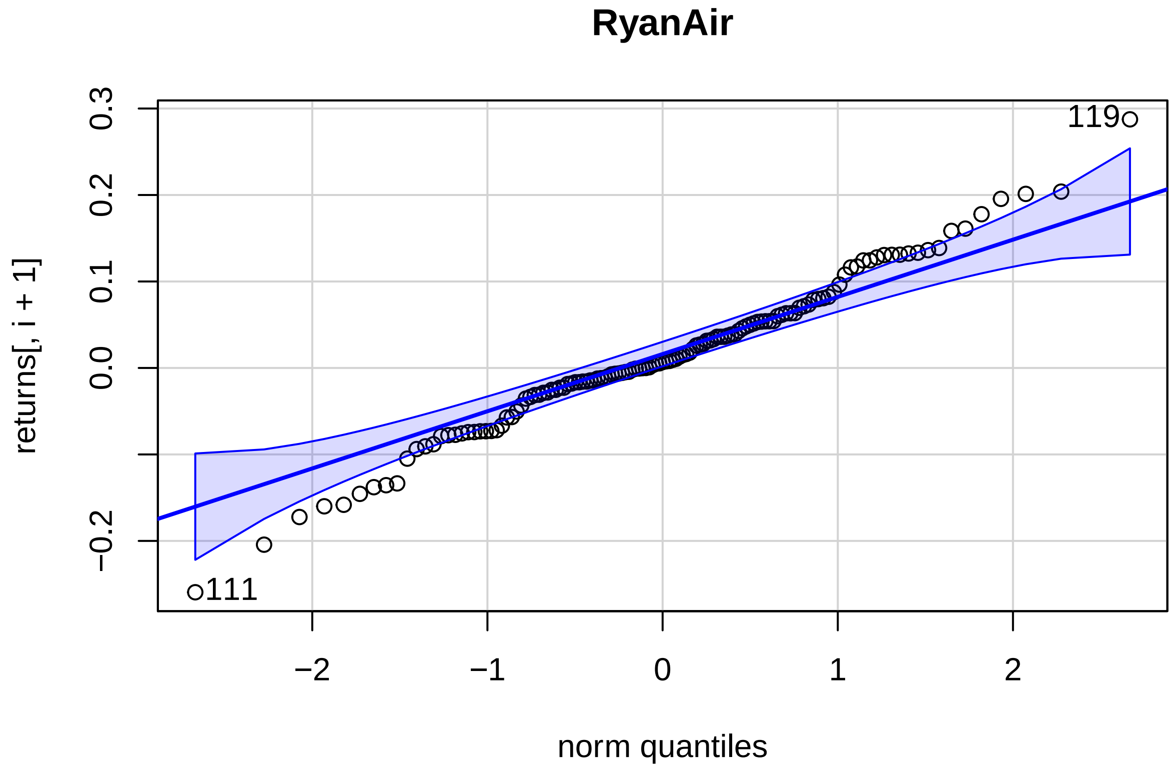
```
for (i in 1:15)
{
  qqPlot(returns[,i+1], main = names[[1]][i])
}
```

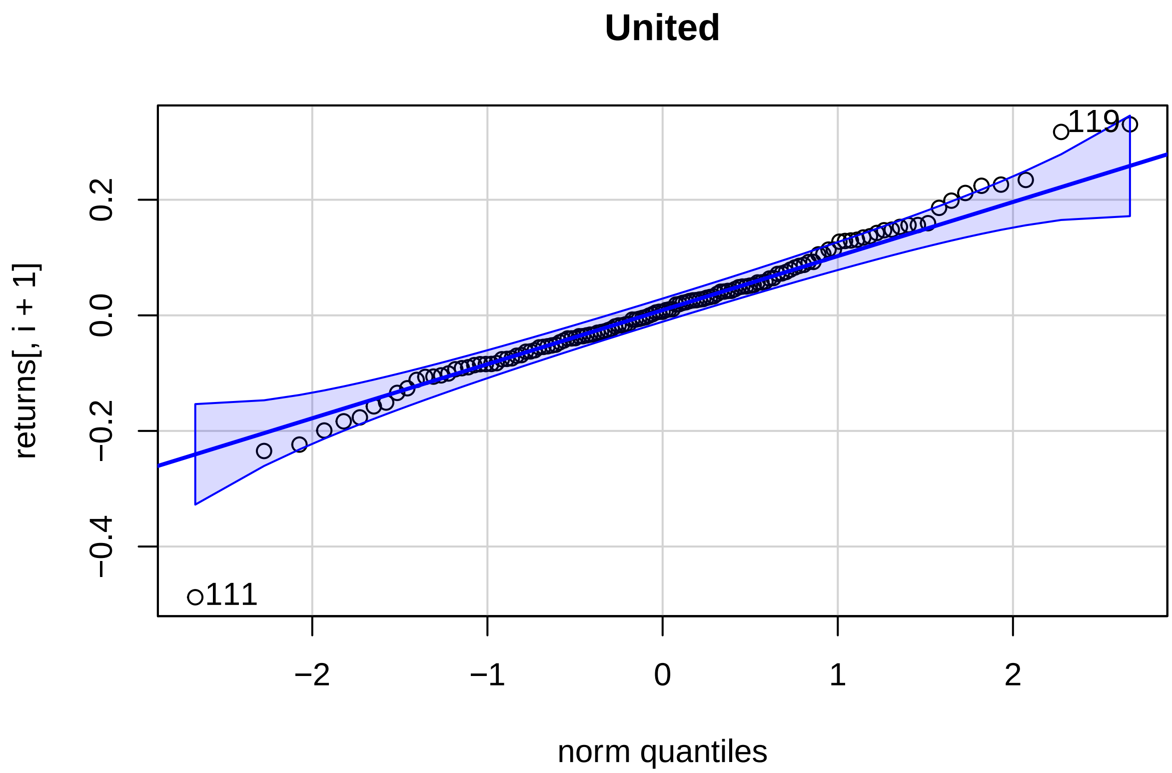

Lufthansa
returns[, i + 1]
0.3
0.1
−0.1
−0.3
119
130
−2
−1
0
1
2
norm quantiles

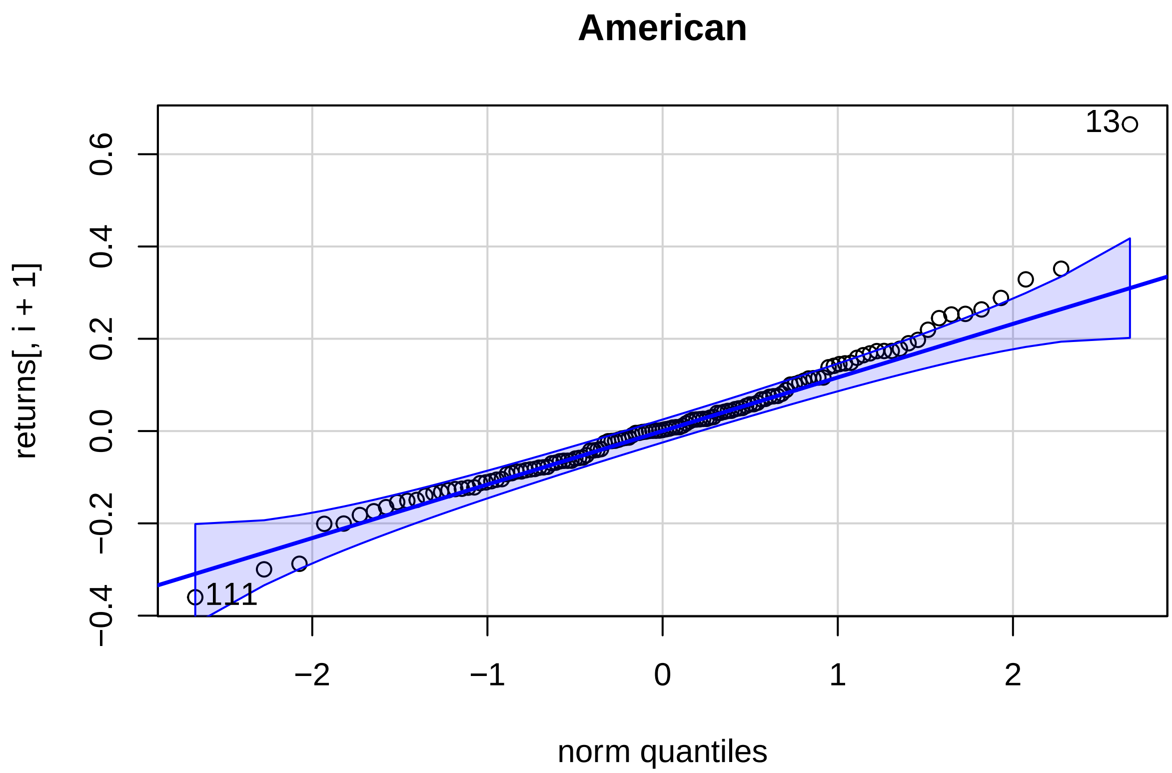

Southwest
returns[, i + 1]
0.3
0.2
0.1
0.0
−0.2
−2
−1
0
1
2
norm quantiles
122
111

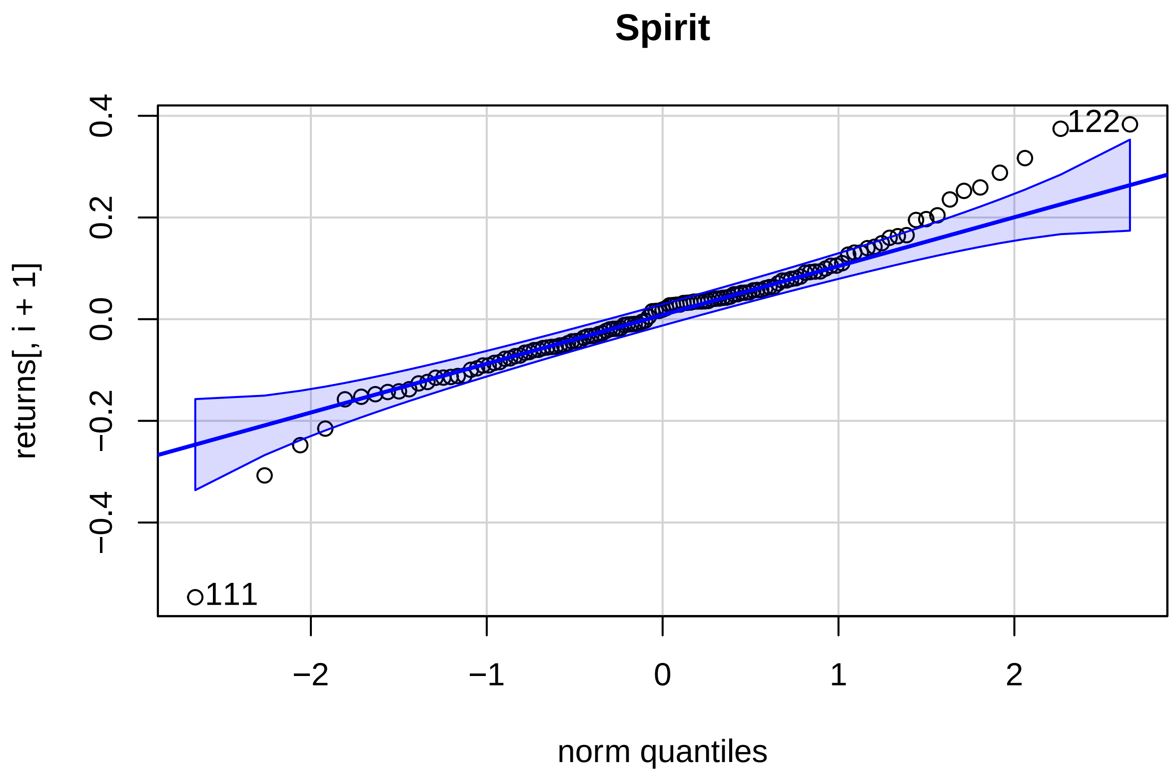


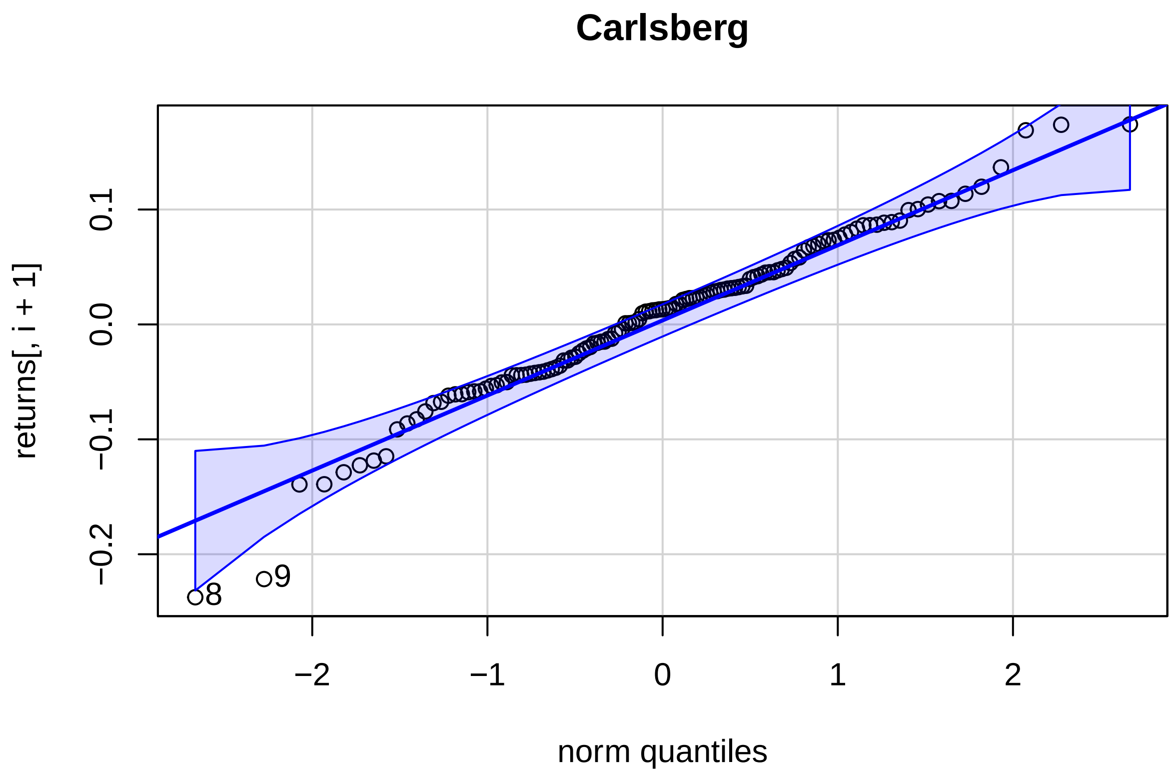
EasyJet
119
3
0.2
0.1
0.0
−0.1
−0.2
returns[, i + 1]
−2
−1
0
1
2
norm quantiles


RyanAir
119
111
0.3
0.2
0.1
0.0
−0.2
returns[, i + 1]
−2
−1
0
1
2
norm quantiles

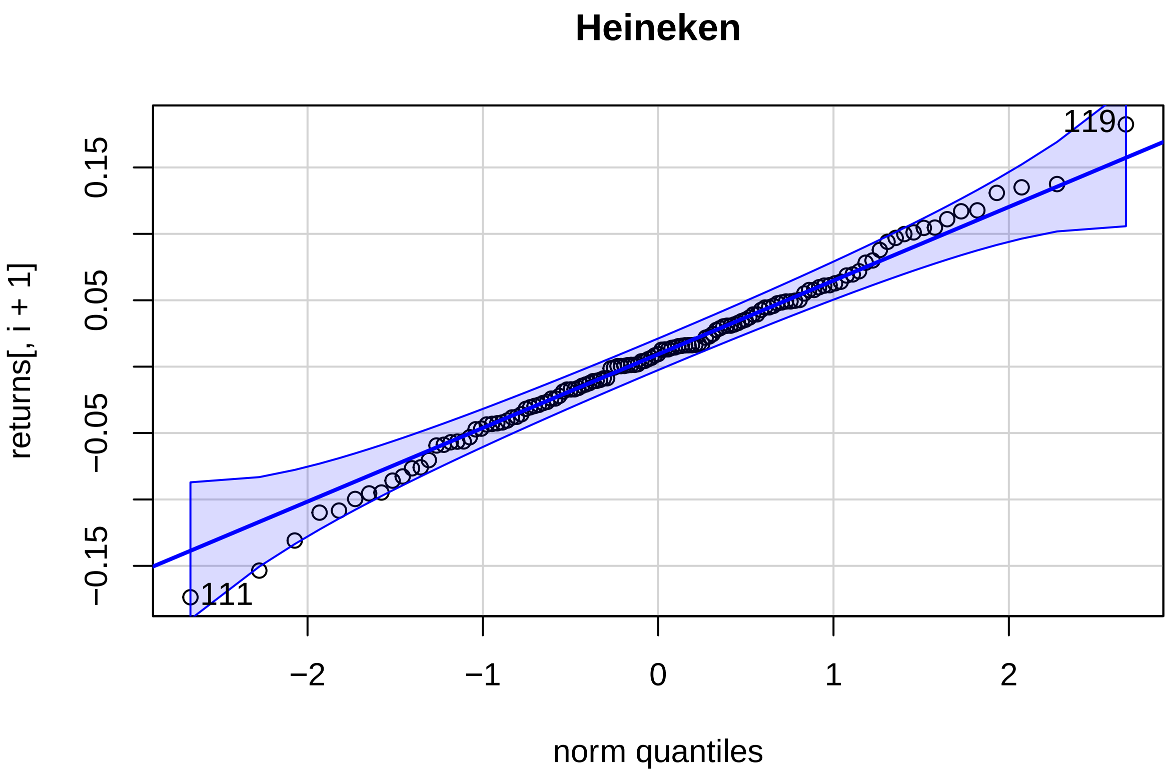

United
119
111
0.2
0.0
−0.2
−0.4
returns[, i + 1]
−2
−1
0
1
2
norm quantiles


American
returns[, i + 1]
0.6
0.4
0.2
0.0
−0.2
−0.4
13
111
−2
−1
0
1
2
norm quantiles

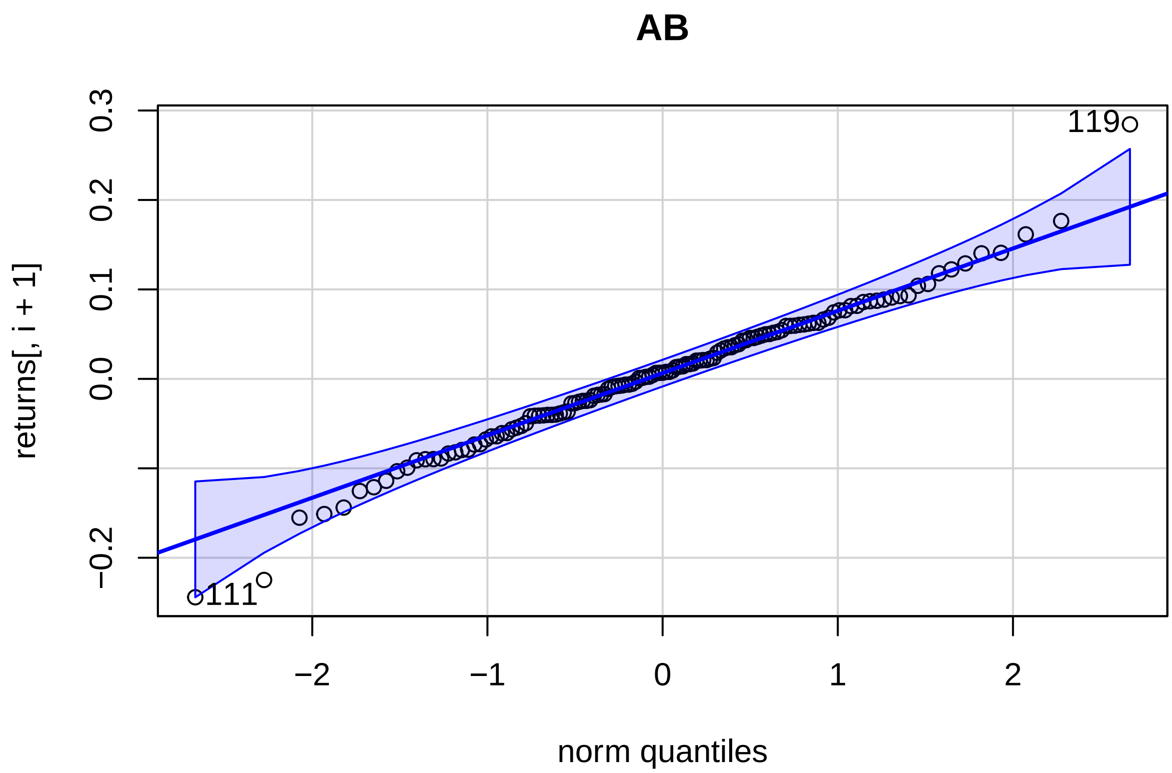

Spirit
returns[, i + 1]
0.4
0.2
0.0
−0.2
−0.4
122
111
−2
−1
0
1
2
norm quantiles

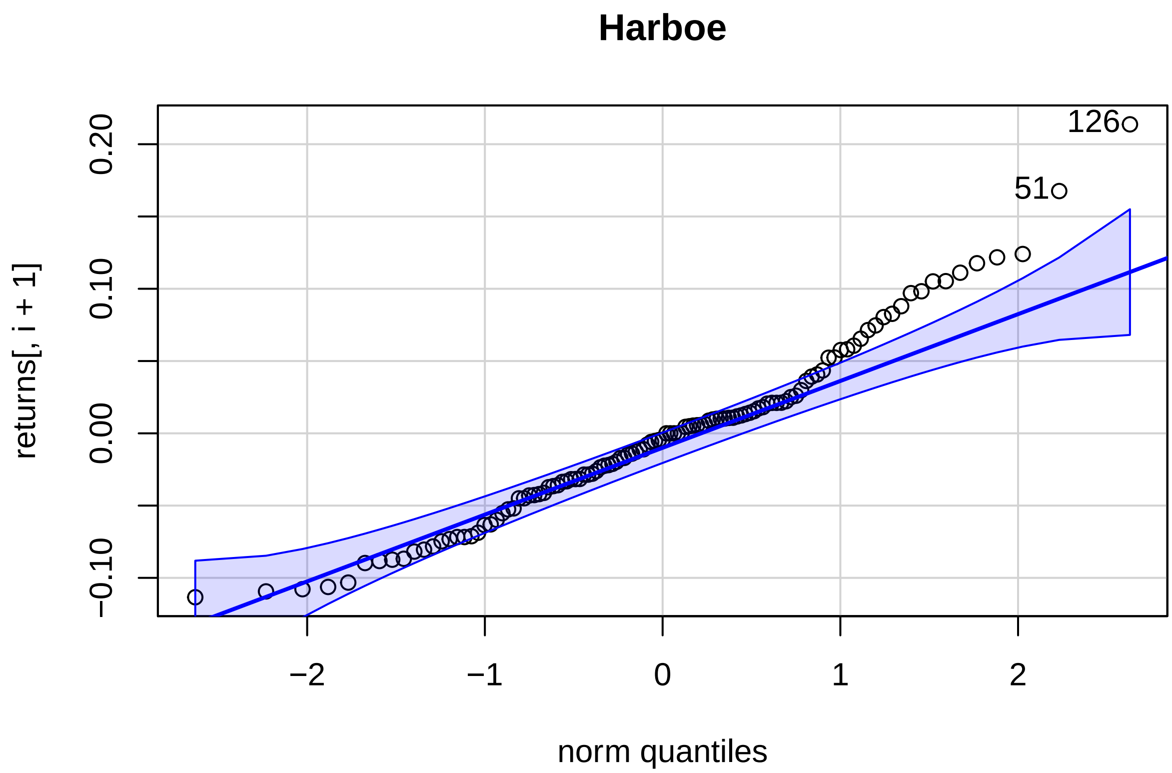


Carlsberg
returns[, i + 1]
0.1
0.0
−0.1
−0.2
8
9
−2
−1
0
1
2
norm quantiles

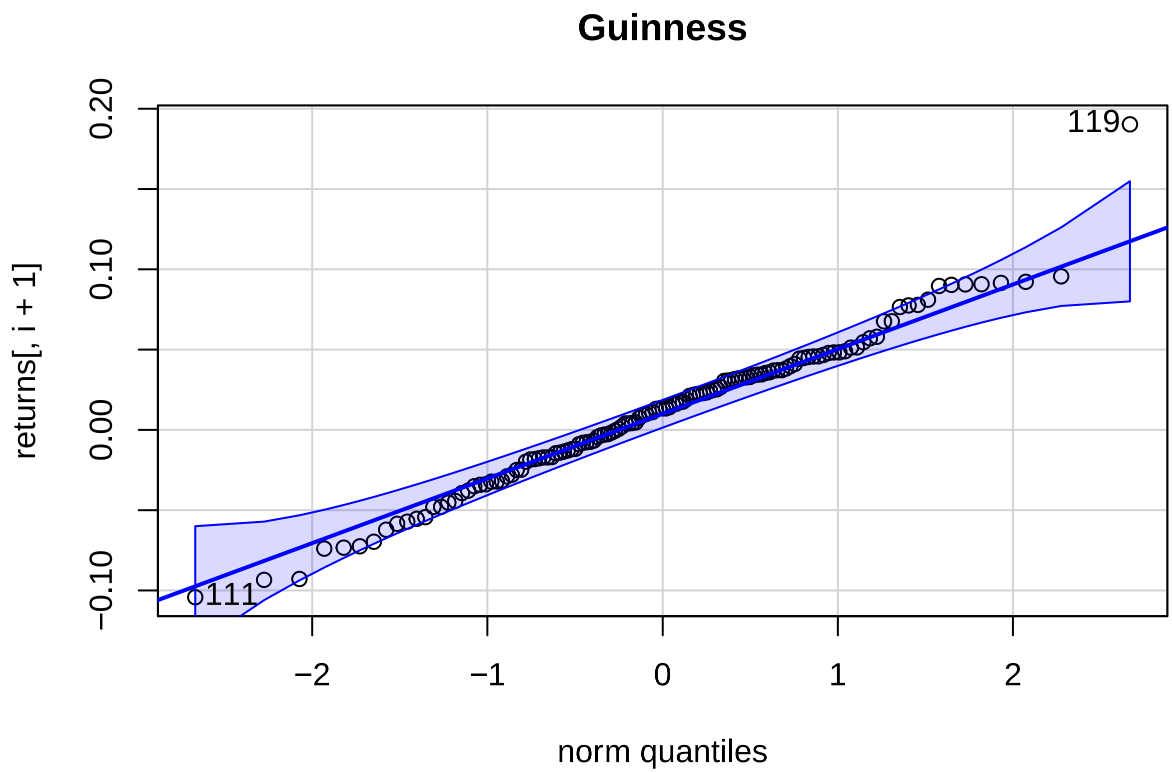


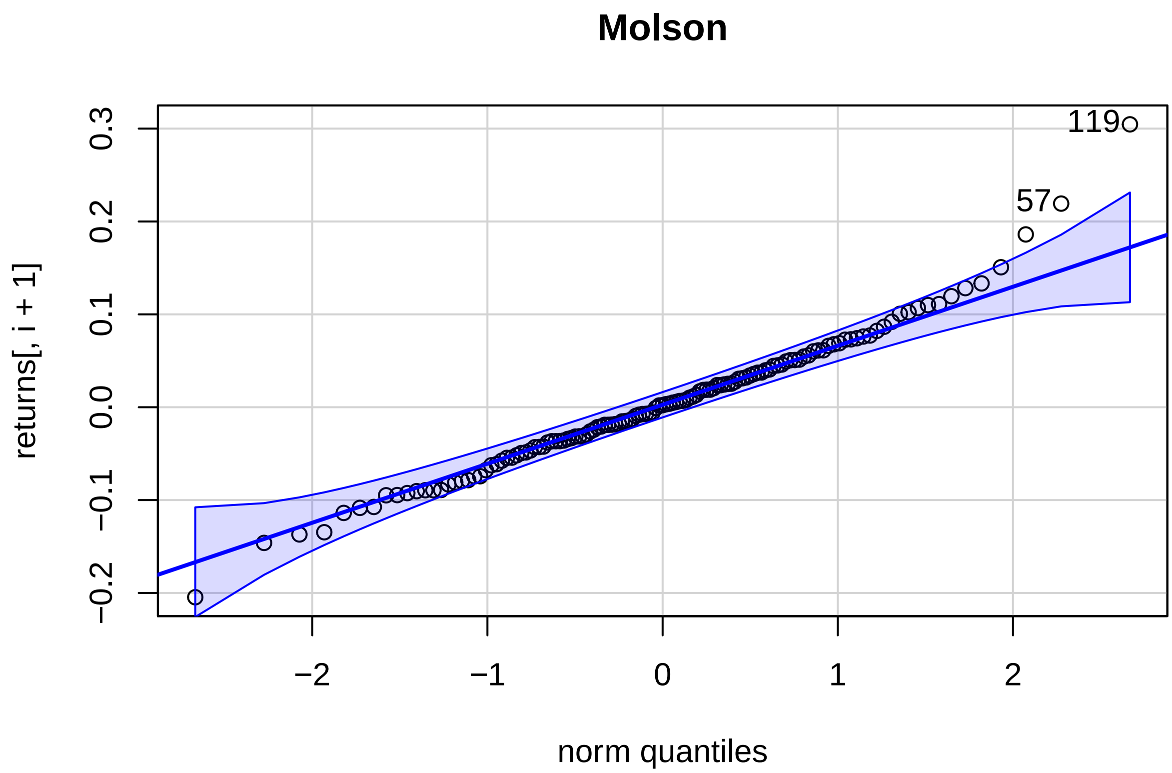
Heineken
0.15
0.05
−0.05
−0.15
returns[, i + 1]
−2
−1
0
1
2
norm quantiles
119
111

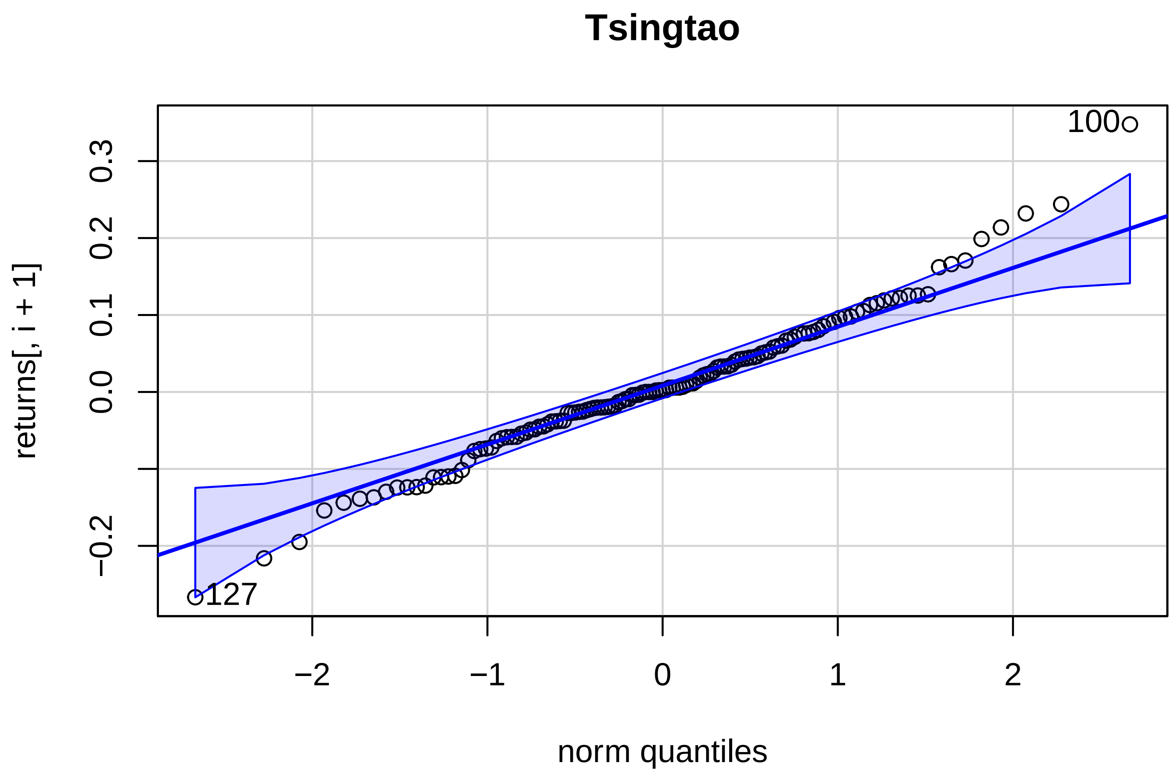

AB
119
111
0.3
0.2
0.1
0.0
−0.2
returns[, i + 1]
−2
−1
0
1
2
norm quantiles

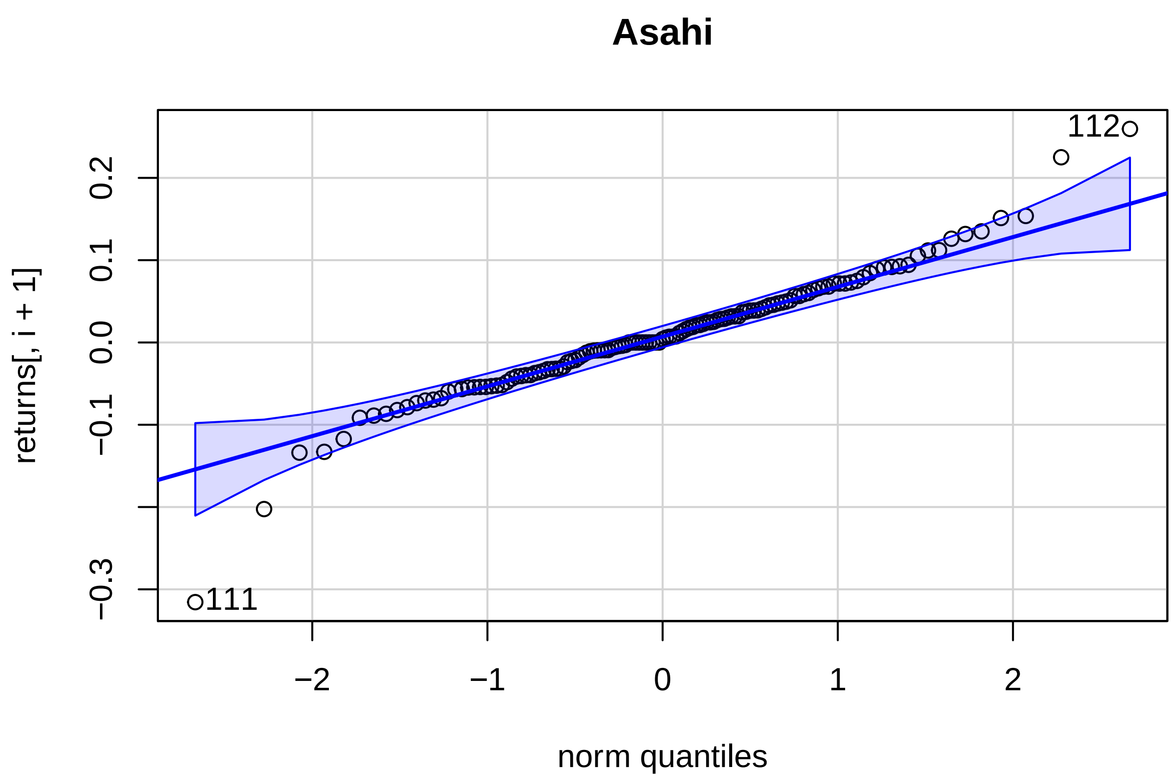


Harboe
126
51
0.20
0.10
0.00
−0.10
returns[, i + 1]
−2
−1
0
1
2
norm quantiles


Guinness
returns[, i + 1]
0.20
0.10
0.00
−0.10
119
111
−2
−1
0
1
2
norm quantiles


Molson
119
57
returns[, i + 1]
0.3
0.2
0.1
0.0
−0.1
−0.2
−2
−1
0
1
2
norm quantiles


Tsingtao
returns[, i + 1]
0.3
0.2
0.1
0.0
−0.2
−2
−1
0
1
2
norm quantiles
100
127



7. Run tests for stationarity

```r
#null hypothesis is series is non stationary

for (i in 1:15)
{
 print(adf.test(returnsNA[,i+1]))
}
```

```
## Warning in adf.test(returnsNA[, i + 1]): p-value smaller than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  returnsNA[, i + 1]
## Dickey-Fuller = -4.4488, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary

## Warning in adf.test(returnsNA[, i + 1]): p-value smaller than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  returnsNA[, i + 1]
## Dickey-Fuller = -4.8884, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

```
## Warning in adf.test(returnsNA[, i + 1]): p-value smaller than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  returnsNA[, i + 1]
## Dickey-Fuller = -5.0885, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary

## Warning in adf.test(returnsNA[, i + 1]): p-value smaller than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  returnsNA[, i + 1]
## Dickey-Fuller = -5.7205, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary

## Warning in adf.test(returnsNA[, i + 1]): p-value smaller than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  returnsNA[, i + 1]
## Dickey-Fuller = -4.6744, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary

## Warning in adf.test(returnsNA[, i + 1]): p-value smaller than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  returnsNA[, i + 1]
## Dickey-Fuller = -4.9571, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary

## Warning in adf.test(returnsNA[, i + 1]): p-value smaller than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  returnsNA[, i + 1]
## Dickey-Fuller = -4.2524, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary

## Warning in adf.test(returnsNA[, i + 1]): p-value smaller than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  returnsNA[, i + 1]
## Dickey-Fuller = -7.2601, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

```
## Warning in adf.test(returnsNA[, i + 1]): p-value smaller than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  returnsNA[, i + 1]
## Dickey-Fuller = -5.5502, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary

## Warning in adf.test(returnsNA[, i + 1]): p-value smaller than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  returnsNA[, i + 1]
## Dickey-Fuller = -4.8581, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary

## Warning in adf.test(returnsNA[, i + 1]): p-value smaller than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  returnsNA[, i + 1]
## Dickey-Fuller = -5.1527, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary

## Warning in adf.test(returnsNA[, i + 1]): p-value smaller than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  returnsNA[, i + 1]
## Dickey-Fuller = -4.5603, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary

## Warning in adf.test(returnsNA[, i + 1]): p-value smaller than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  returnsNA[, i + 1]
## Dickey-Fuller = -5.0268, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary

## Warning in adf.test(returnsNA[, i + 1]): p-value smaller than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  returnsNA[, i + 1]
## Dickey-Fuller = -4.9261, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

```
## Warning in adf.test(returnsNA[, i + 1]): p-value smaller than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  returnsNA[, i + 1]
## Dickey-Fuller = -5.4737, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

```
#all pvalues are less than 0.01 and so p-value<0.05 hence we reject the null hypothesis and conclude th

#graphically we can see that all the returns have no trends and the oscillations between the returns se
```

8. Check if returns are normally distributed

```
#From the histograms the returns appear to have the normal distribution shape, where most of them appea
#Looking at the qqplots we can see a few of the returns have values outside the interval which means th
#lastly we do the Shapiro test
#null hypothesis: normal

for (i in 1:15)
{
 print(shapiro.test(returnsNA[,i+1]))
}
```

```
##
##  Shapiro-Wilk normality test
##
## data:  returnsNA[, i + 1]
## W = 0.99245, p-value = 0.7962
##
##
##  Shapiro-Wilk normality test
##
## data:  returnsNA[, i + 1]
## W = 0.98699, p-value = 0.3485
##
##
##  Shapiro-Wilk normality test
##
## data:  returnsNA[, i + 1]
## W = 0.9802, p-value = 0.09229
##
##
##  Shapiro-Wilk normality test
##
## data:  returnsNA[, i + 1]
## W = 0.9842, p-value = 0.2053
##
##
##  Shapiro-Wilk normality test
##
## data:  returnsNA[, i + 1]
```

```
## W = 0.96322, p-value = 0.003369
##
##
##  Shapiro-Wilk normality test
##
## data:  returnsNA[, i + 1]
## W = 0.94478, p-value = 0.0001491
##
##
##  Shapiro-Wilk normality test
##
## data:  returnsNA[, i + 1]
## W = 0.94954, p-value = 0.0003182
##
##
##  Shapiro-Wilk normality test
##
## data:  returnsNA[, i + 1]
## W = 0.97806, p-value = 0.05991
##
##
##  Shapiro-Wilk normality test
##
## data:  returnsNA[, i + 1]
## W = 0.9946, p-value = 0.9422
##
##
##  Shapiro-Wilk normality test
##
## data:  returnsNA[, i + 1]
## W = 0.98093, p-value = 0.1071
##
##
##  Shapiro-Wilk normality test
##
## data:  returnsNA[, i + 1]
## W = 0.96791, p-value = 0.008096
##
##
##  Shapiro-Wilk normality test
##
## data:  returnsNA[, i + 1]
## W = 0.9787, p-value = 0.06814
##
##
##  Shapiro-Wilk normality test
##
## data:  returnsNA[, i + 1]
## W = 0.96058, p-value = 0.002092
##
##
##  Shapiro-Wilk normality test
##
## data:  returnsNA[, i + 1]
```

```
## W = 0.98392, p-value = 0.1943
##
##
##  Shapiro-Wilk normality test
##
## data:  returnsNA[, i + 1]
## W = 0.93592, p-value = 3.924e-05
```

```
#normal <- 1,2,3,4,9,10,12,14,8 barely with 0.05991
#not normal <- 5,6,7,11,13,15
```

9. Fit different distributions to your data, which one fits better?

```
#t distribution

AIC_t <- c(rep(NA), 15)
BIC_t <- c(rep(NA), 15)

for (i in 1:15)
{
  x <- diff(returnsNA[,i+1])
  n = length(x)
  start = c(mean(x), sd(x), 5)
  loglik_t = function(beta) sum( - dt((x - beta[1]) / beta[2], beta[3], log = TRUE) + log(beta[2]) )
  fit_t = optim(start, loglik_t, hessian = T, method = "L-BFGS-B", lower = c(-1, 0.001, 1))
  AIC_t[i] = 2 * fit_t$value + 2 * 3
  BIC_t[i] = 2 * fit_t$value + log(n) * 3
}

#skewed student t
#not work: Harboe=11

AIC_sstd <- c(rep(NA), 15)
BIC_sstd <- c(rep(NA), 15)

for (i in 1:10)
{
  x <- diff(returnsNA[,i+1])
  n = length(x)
  loglik_sstd = function(beta) sum(- dsstd(x, mean = beta[1], sd = beta[2], nu = beta[3], xi = beta[4],
  start = c(mean(x), sd(x), 5, 1)
  fit_sstd = optim(start, loglik_sstd, hessian = T, method = "L-BFGS-B", lower = c(-0.1, 0.01, 2.1, -2)
  AIC_sstd[i] = 2*fit_sstd$value + 2 * 4
  BIC_sstd[i]= 2*fit_sstd$value + log(n) * 4
}

for (i in 1:4)
{
  x <- diff(returnsNA[,i+12])
  n = length(x)
  loglik_sstd = function(beta) sum(- dsstd(x, mean = beta[1], sd = beta[2], nu = beta[3], xi = beta[4],
  start = c(mean(x), sd(x), 5, 1)
  fit_sstd = optim(start, loglik_sstd, hessian = T, method = "L-BFGS-B", lower = c(-0.1, 0.01, 2.1, -2)
```

```r
  AIC_sstd[i+11] = 2*fit_sstd$value + 2 * 4
  BIC_sstd[i+11]= 2*fit_sstd$value + log(n) * 4
}

#ged

AIC_ged <- c(rep(NA), 15)
BIC_ged <- c(rep(NA), 15)

for (i in 1:15)
{
x <- diff(returnsNA[,i+1])
n = length(x)
loglik_ged = function(beta) sum( - dged(x,mean=beta[1],sd=beta[2],nu=beta[3],log=TRUE) )
start = c(mean(x),sd(x),1)
fit_ged = optim(start,loglik_ged,hessian=T,method="L-BFGS-B",lower = c(-.1,.01,1))
AIC_ged[i] = 2*fit_ged$value+2*4
BIC_ged[i] = 2*fit_ged$value+log(n)*4
}

#normal
AIC_n <- c(rep(NA), 15)
BIC_n <- c(rep(NA), 15)

for (i in 1:15)
{
x <- diff(returnsNA[,i+1])
n = length(x)
start = c(mean(x),sd(x),1)
loglik_n = function(beta) sum( - dnorm(x,mean=beta[1],sd=beta[2],log=TRUE) )
fit_n = optim(start, loglik_n, hessian = T,
method = "L-BFGS-B", lower = c(-1, 0.001, 1))
AIC_n[i] = 2 * fit_n$value + 2 * 3
BIC_n[i] = 2 * fit_n$value + log(n) * 3
}

########################

table <- data.frame(AIC_t,AIC_sstd,AIC_ged,AIC_n,BIC_t,BIC_sstd,BIC_ged,BIC_n)
table <- cbind(Stocks, table)
table
```

```
##       Stocks      AIC_t   AIC_sstd    AIC_ged      AIC_n      BIC_t   BIC_sstd
## 1  Lufthansa -101.29256 -103.37211  -99.83794 -101.18741  -93.13707  -92.49812
## 2  Southwest -153.09917 -151.38061 -151.98968 -151.38767 -144.94367 -140.50662
## 3    EasyJet -166.23519 -164.47357 -166.30879 -164.09265 -158.07969 -153.59958
## 4    RyanAir -143.43542 -143.47498 -141.94782 -135.58773 -135.27993 -132.60098
## 5     United  -92.86558  -91.84959  -91.88617  -92.40196  -84.71008  -80.97559
## 6   American  -54.19922  -52.85083  -51.01658  -48.25515  -46.04373  -41.97684
## 7     Spirit  -84.69354  -84.58365  -79.86794  -75.45280  -76.53804  -73.70966
## 8  Carlsberg -195.03689 -195.95710 -192.57583 -190.49990 -186.88139 -185.08310
## 9   Heineken -228.57643 -227.35985 -226.72998 -225.91525 -220.42094 -216.48586
## 10        AB -184.98238 -184.19720 -181.40210 -179.18697 -176.82689 -173.32321
```

```
## 11    Harboe -233.81981        NA -231.94401 -233.82148 -225.66431        NA
## 12  Guinness -281.16333 -282.96831 -278.27527 -277.26795 -273.00783 -272.09431
## 13    Molson -177.16594 -177.19465 -176.32012 -176.78077 -169.01044 -166.32066
## 14  Tsingtao -120.71342 -118.76709 -118.78592 -120.71458 -112.55792 -107.89309
## 15     Asahi -210.03316 -208.07201 -201.06966 -184.41847 -201.87767 -197.19801
##       BIC_ged      BIC_n
## 1   -88.96394  -93.03191
## 2  -141.11568 -143.23217
## 3  -155.43480 -155.93715
## 4  -131.07383 -127.43223
## 5   -81.01218  -84.24647
## 6   -40.14258  -40.09965
## 7   -68.99394  -67.29730
## 8  -181.70183 -182.34440
## 9  -215.85599 -217.75975
## 10 -170.52811 -171.03148
## 11 -221.07002 -225.66598
## 12 -267.40128 -269.11246
## 13 -165.44613 -168.62527
## 14 -107.91192 -112.55908
## 15 -190.19567 -176.26297
```

```r
AIC_result <- c("Skewed Sd t", "t Dist", "Ged", "Skewed Sd t", "t Dist", "t Dist", "t Dist", "Skewed Sd
BIC_result <- c("t Dist","t Dist","t Dist","t Dist","t Dist","t Dist","t Dist","t Dist","t Dist","t Dis

table <- cbind(table, AIC_result, BIC_result)
colnames(table)[2:11] <- c("t AIC", "Skewed t AIC", "Ged AIC", "Normal AIC", "t BIC", "Skewed t BIC", "
table
```

```
##       Stocks      t AIC Skewed t AIC    Ged AIC Normal AIC      t BIC
## 1  Lufthansa -101.29256   -103.37211  -99.83794 -101.18741  -93.13707
## 2  Southwest -153.09917   -151.38061 -151.98968 -151.38767 -144.94367
## 3    EasyJet -166.23519   -164.47357 -166.30879 -164.09265 -158.07969
## 4    RyanAir -143.43542   -143.47498 -141.94782 -135.58773 -135.27993
## 5     United  -92.86558    -91.84959  -91.88617  -92.40196  -84.71008
## 6   American  -54.19922    -52.85083  -51.01658  -48.25515  -46.04373
## 7     Spirit  -84.69354    -84.58365  -79.86794  -75.45280  -76.53804
## 8  Carlsberg -195.03689   -195.95710 -192.57583 -190.49990 -186.88139
## 9   Heineken -228.57643   -227.35985 -226.72998 -225.91525 -220.42094
## 10        AB -184.98238   -184.19720 -181.40210 -179.18697 -176.82689
## 11    Harboe -233.81981           NA -231.94401 -233.82148 -225.66431
## 12  Guinness -281.16333   -282.96831 -278.27527 -277.26795 -273.00783
## 13    Molson -177.16594   -177.19465 -176.32012 -176.78077 -169.01044
## 14  Tsingtao -120.71342   -118.76709 -118.78592 -120.71458 -112.55792
## 15     Asahi -210.03316   -208.07201 -201.06966 -184.41847 -201.87767
##    Skewed t BIC    Ged BIC Normal BIC    Best AIC Best BIC
## 1     -92.49812  -88.96394  -93.03191 Skewed Sd t   t Dist
## 2    -140.50662 -141.11568 -143.23217      t Dist   t Dist
## 3    -153.59958 -155.43480 -155.93715         Ged   t Dist
## 4    -132.60098 -131.07383 -127.43223 Skewed Sd t   t Dist
## 5     -80.97559  -81.01218  -84.24647      t Dist   t Dist
## 6     -41.97684  -40.14258  -40.09965      t Dist   t Dist
## 7     -73.70966  -68.99394  -67.29730      t Dist   t Dist
## 8    -185.08310 -181.70183 -182.34440 Skewed Sd t   t Dist
```

```
## 9    -216.48586 -215.85599 -217.75975      t Dist   t Dist
## 10   -173.32321 -170.52811 -171.03148      t Dist   t Dist
## 11           NA -221.07002 -225.66598      Normal   Normal
## 12   -272.09431 -267.40128 -269.11246 Skewed Sd t   t Dist
## 13   -166.32066 -165.44613 -168.62527 Skewed Sd t   t Dist
## 14   -107.89309 -107.91192 -112.55908      Normal   Normal
## 15   -197.19801 -190.19567 -176.26297      t Dist   t Dist
```

10. Compute Sharpe's slope for each asset. Which asset has the highest slope? If E(RP) and σ_{R_p} are the expected return and standard deviation of the return on a portfolio and μ_f is the risk-free rate, then Sharpe's slope = $\frac{(E(R_P)-\mu_f)}{\sigma_{Rp}}$

```
riskfree <- 0.04

SharpesSlope <- rep(NA,15)

for (i in 1:15)
{
  SharpesSlope[i] <- (mean(returnsNA[,i+1]) - riskfree)/sd(returnsNA[,i+1])
}

#If the analysis results in a negative Sharpe ratio, it either means the risk-free rate is greater than

SharpesSlope
```

```
##  [1] -0.3611148 -0.2458842 -0.4103832 -0.3012443 -0.2466343 -0.1938094
##  [7] -0.2113170 -0.4910430 -0.5343287 -0.4788517 -0.6687074 -0.6470281
## [13] -0.5263983 -0.3381888 -0.4398642
```

```
max(SharpesSlope) #highest slope
```

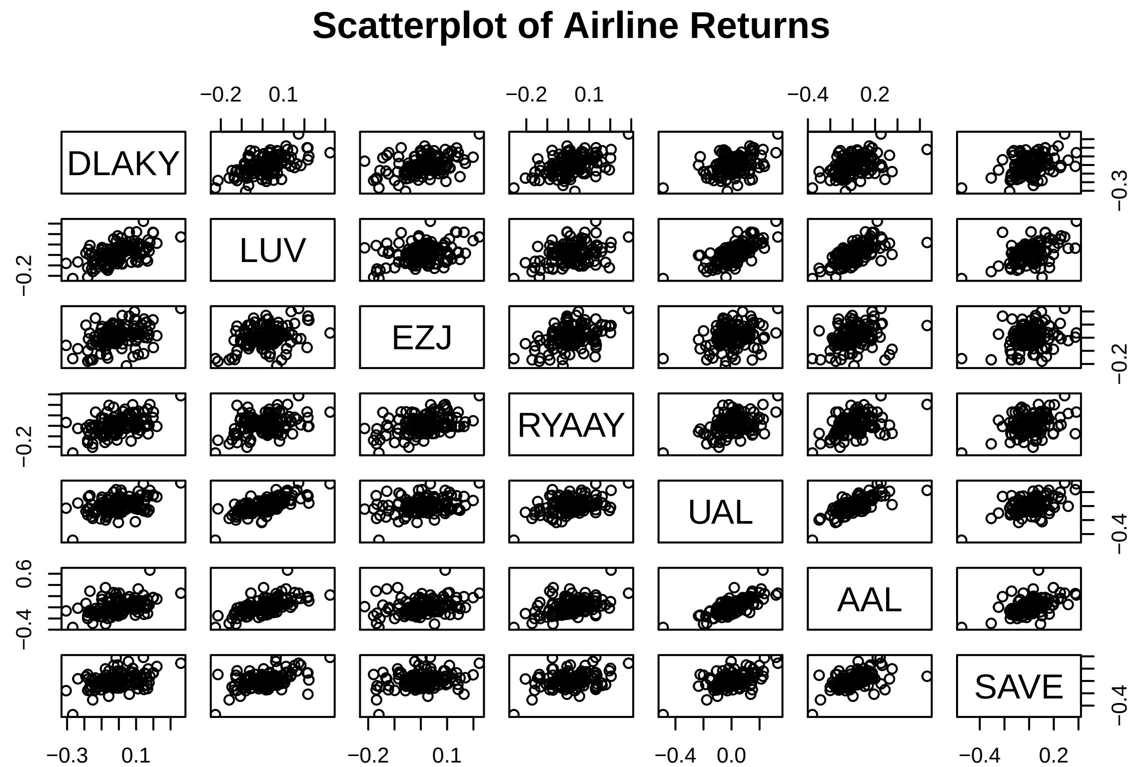
```
## [1] -0.1938094
```

11. Convert the monthly sample means into annual estimates by multiplying by 12 and convert the monthly sample SDs into annual estimates by multiplying by the square root of 12. Comment on the values of these annual numbers.

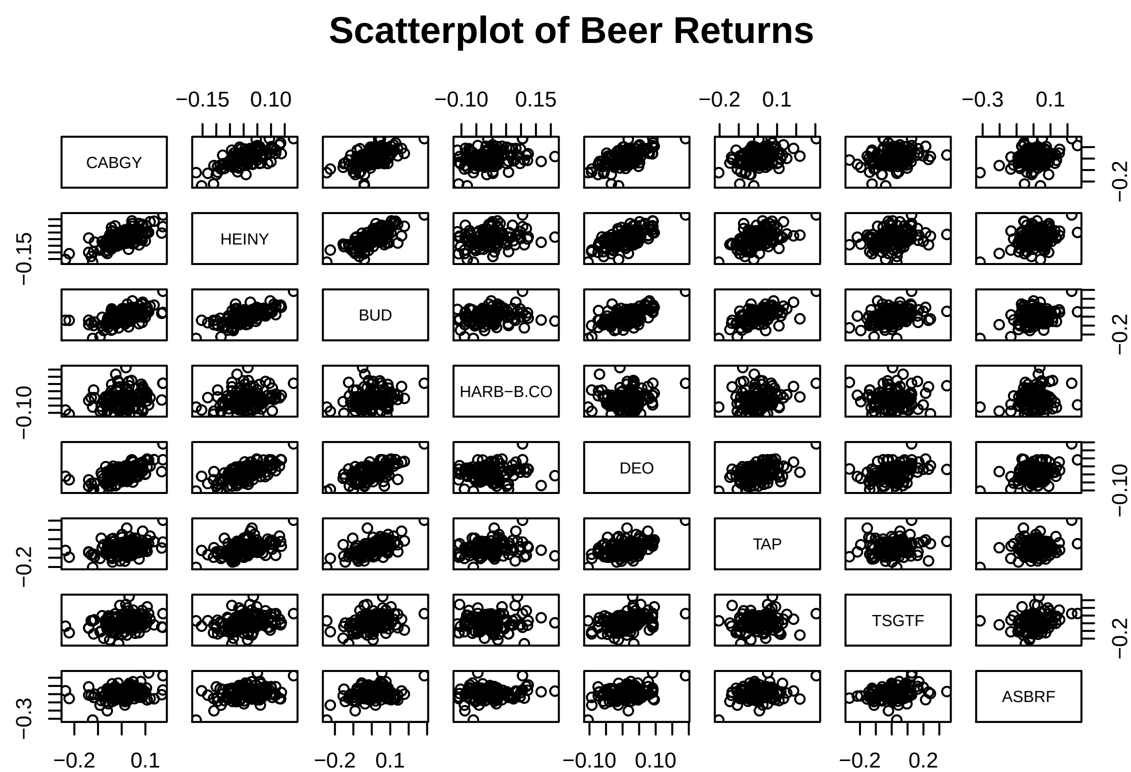
```
#?

#annualsamplemeans <- returns[,2:16] *12


#annualsamplesd <- sd(returns[,2:16])*sqrt(12)
```

12. Construct pairwise scatter plots between your assets returns and comment on any relationships you see.

```
#airlines
pairs(returns[,2:8], main="Scatterplot of Airline Returns")
```




```
#beers
pairs(returns[,9:16], main="Scatterplot of Beer Returns")
```



13. You should also compute the sample covariance matrix of the returns on your assets and comment on the direction of linear association between the asset returns.

```
# covariance monthly data

Covariances <- as.data.frame(cov(returnsNA[2:16]))

#they all have positive covariances hence the returns all move together, vary in the same direction.

Covariancesair <- Covariances[1:7,1:7]
Covariancesbeers <- Covariances[8:15,8:15]

# covariance annualized data

annualreturns <- returnsNA[,2:16]*12

Covariancesannual <- as.data.frame(cov(annualreturns[1:15]))

#they all have positive covariances hence the returns all move together, vary in the same direction.

Covariancesairannual <- Covariancesannual[1:7,1:7]
Covariancesbeersannual <- Covariancesannual[8:15,8:15]


##########################
```

```r
#correlation for monthly data

Correlations <- as.data.frame(cor(returnsNA[2:16]))

#they all have positive covariances hence the returns all move together, vary in the same direction.

Correlationsair <- Correlations[1:7,1:7]
Correlationsbeers <- Correlations[8:15,8:15]
```