

```
---
title: "DCA LS"
author: "Bram"
date: "19/03/2021"
output:
  html_document:
    df_print: paged
---
```

```
``{r setup, include=FALSE}
knitr::opts_chunk$set(echo = F, warning = F, message = F, error = F)
...

``{r pakker, echo = F, warning = F, message = F, error = F, results='hide'}
require(RMySQL) || {install.packages("RMySQL"); require(RMySQL)}
require(corrplot) || {install.packages("corrplot"); require(corrplot)}
require(quantmod) || {install.packages("quantmod"); require(quantmod)}
require(PerformanceAnalytics) || {install.packages("PerformanceAnalytics");
require(PerformanceAnalytics)}
require(dygraphs) || {install.packages("dygraphs"); require(dygraphs)}
require(xts) || {install.packages("xts"); require(xts)}
require(corrplot) || {install.packages("corrplot"); require(corrplot)}
require(dplyr) || {install.packages("dplyr"); require(dplyr)}
require(mosaic) || {install.packages("mosaic"); require(mosaic)}
require(stringi) || {install.packages("stringi"); require(stringi)}
require(jrvFinance) || {install.packages("jrvFinance"); require(jrvFinance)}
require(scales) || {install.packages("scales"); require(scales)}
require(ggplot2) || {install.packages("ggplot2"); require(ggplot2)}
require(tidyr) || {install.packages("tidyr"); require(tidyr)}
require(reticulate) || {install.packages("reticulate"); require(reticulate)}
```

```
require(stargazer) | | {install.packages("tidyr"); require(tidyr)}
```

```
...
```

Datainnhenting

Vi starter med datainnsamling. Kurshistorikk fra OBX er hentet fra Ødegaard sin nettside. Første observasjon dateres til 1980:1. Siste handelsdag 2020:11. Totalt omtrent 40 år med data. Vi inkluderer den månedlige prosentvise endringen i prisen, justert for utbytte og andre handlinger, samt datoen. Det gir oss 491 observasjoner.

```
``{r datainnhenting}
```

```
# Data hentet fra https://ba-odegaard.no/financial_data/ose_asset_pricing_data/index.html
```

```
osebxcsv <- readr::read_csv("https://ba-  
odegaard.no/financial_data/ose_asset_pricing_data/market_portfolios_monthly.txt")
```

```
OBX = data.frame(OBX = osebxcsv$OBX)
```

```
rownames(OBX) = as.Date(as.character(osebxcsv$date), format = "%Y%m%d")
```

```
prices <- data.frame(OBX)
```

```
...
```

OBX er først listet i januar 1987 og prishistorikken for tidligere observasjoner er således null. Den markedsvektede indeksen før den tid er kunstig utledet basert på markedsvekten til de 25 mest omsatte aksjer på Oslo Børs og utelates fra datasettet. Totalt sitter vi da igjen med 407 observasjoner, som omfatter kursutviklingen på OBX fra januar 1987 til og med november 2020.

Opprydding

```
``{r opprydding, echo = T}
```

```
summary(is.na(prices))
```

```
prices_clean <- na.omit(prices)
```

```
nrow(prices_clean)
```

```
...
```

DCA og LS strategien

DCA og LS blir regnet ut med utgangspunkt i listene som inneholder månedlig prisendring. Den totale porteføljeverdien endres hver måned basert på kursutviklingen til underliggende verdipapir. Den

prosentvise endringen i porteføljen hver måned gir grunnlag for beregning av både annualisert avkastning, standardavvik og Sharpe-raten.

Vi tester et utvalg av mulige sparehorisonter i datasettet, i intervaller av 12 måneder

sparing i 12 mnd, som er mulig 396 ganger, i periodene 1987:1 - 1987:12, 1987:2 - 1988:1, osv til 2019:12 - 2020:11

sparing i 24 mnd, som er mulig 384 ganger, i periodene 1987:1 - 1988:12, 1987:2 - 1989:1, osv til 2018:12 - 2020:11

...

sparing i 132 måneder, som er mulig 276 ganger, i perioden 1987:1 - 1997:12, 1987:2 - 1998:1, osv til 2009:12 - 2020:11

sparing i 144 måneder, som er mulig 264 ganger, i perioden 1987:1 - 1998:12, 1987:2 - 1999:1, osv til 2008:12 - 2020:11

Det gir totalt 3960 ulike perioder per strategi.

```
``{r dca og bh}
```

```
# Henter datoer fra datasettet
```

```
Dates <- as.Date(rownames(prices_clean), format = "%Y-%m-%d")
```

```
# resultList tar vare på resultatene
```

```
resultList <- list()
```

```
# brukes til indexing av observasjonene
```

```
o <- 0
```

```
# Logikken er som følgende:
```

```
# Hver runde simulerer den begge strategiene
```

```
# Første runde har oppstart i januar 1987 og sparer i 1 måned
```

```
# Andre runde har oppstart i januar 1987 og sparer i 2 måneder
```

```
# Siste runde har oppstart i januar 1987 og sparer i 407 måneder
```

```
# Når alle simuleringer er fullført starter logikken på nytt
```

```
# Første runde har da oppstart i februar 1987 og sparer i 1 måned
```

```
# Andre runde har oppstart i februar 1987 og sparer i 2 måneder
```

```
# Siste runde har oppstart i februar 1987 og sparer i 406 måneder
```

```

# Hver simulering lagres i listen resultList
# resultList[[1]] inneholder simuleringene som har oppstart i januar 1987 og sparer i 1 måned.
# resultList[[2]] inneholder simuleringen som har oppstart i januar 1987 og sparer i 2 måneder.
# resultList[[407]] inneholder simuleringen som har oppstart i januar 1987 og sparer i 407 måneder
# resultList[[408]] inneholder simuleringen som har oppstart i februar 1987 og sparer i 1 måned.
for (m in 1:nrow(prices_clean)) {
  for (i in m:nrow(prices_clean)) {
    setfour <- new.env() # Lager en ny environment hver runde
    Returns <-
      as.numeric(prices_clean[m:i, 1]) # Hent månedlig avkastning
    frame <- data.frame>Returns) # Putt det i Frame
    rownames(frame) <- Dates[m:i] # Sett radnavnan til Dato
    setfour$frame <- frame # Tildele nytt datasett til enviroment
    frame$period <- as.factor(nrow(frame)) # Tildele en faktorvariabel basert på sparehorisontens
    lengde
    dcasum <- 100 # Tildeler beløpet som investeres månedlig

    # Lump Sum
    frame$Buy_Hold_end <- rep(0, nrow(frame))
    frame$Buy_Hold_end[1] <-
      nrow(frame) * dcasum * (1 + frame>Returns[1])
    frame$Buy_Hold_start <- rep(0, nrow(frame))
    frame$Buy_Hold_start[1] <- (nrow(frame) * dcasum)
    if ((i - m) > 0) {
      for (j in 2:nrow(frame)) {
        frame$Buy_Hold_end[j] <-
          frame$Buy_Hold_end[j - 1] * (1 + frame>Returns[j])
        frame$Buy_Hold_start[j] <- frame$Buy_Hold_end[j - 1]
      }
    }
  }
}

```

```

# DCA
frame$DCA <- rep(0, nrow(frame))
frame$DCA[1] <- dcasum * (1 + frame>Returns[1])
frame$DCA_cash <- rep(0, nrow(frame))
frame$DCA_cash[1] <- (nrow(frame) * dcasum) - dcasum
frame$DCA_portfolio_end <- rep(0, nrow(frame))
frame$DCA_portfolio_end[1] <-
  ((nrow(frame) - 1) * dcasum) + frame$DCA[1]
frame$DCA_portfolio_start <- rep(0, nrow(frame))
frame$DCA_portfolio_start[1] <- (nrow(frame) * dcasum)
if (i > m) {
  for (k in 2:nrow(frame)) {
    frame$DCA[k] <- (frame$DCA[k - 1] + dcasum) * (1 + frame>Returns[k])
    frame$DCA_cash[k] <- frame$DCA_cash[k - 1] - dcasum
    frame$DCA_portfolio_end[k] <-
      frame$DCA_cash[k] + frame$DCA[k]
    frame$DCA_portfolio_start[k] <-
      frame$DCA_portfolio_end[k - 1]
  }
}

# Lagre resultatene
resultList[[o]] <- frame

# Øker indeksen for neste simulering
o = o + 1

}
}
...

```

```
# Sortering av simuleringer
```

```
``{r sort, echo = F}
```

```
# Omstrukturerer resultList basert på sparehorisonten
```

```
mybiglist <- list()
```

```
for (j in 1:407) {
```

```
  k = 1
```

```
  for (i in 1:length(resultlist2)) {
```

```
    if (nrow(resultlist[[i]]) == j) {
```

```
      name <- paste(j, "months", k)
```

```
      mybiglist[[name]] <- resultlist[[i]]
```

```
      k = k + 1
```

```
    }
```

```
  }
```

```
}
```

```
...
```

```
# Beregner nøkkeltall for simuleringene
```

```
``{r tester, echo = F}
```

```
# Kjører tester på et utvalg av tidshorisonter (T = 12, 24 ... 144)
```

```
p <- (seq(12, 144, 12))
```

```
##### IRR
```

```
# k tar hånd om perioden (investere man i 12 mnd, 24 mnd osv)
```

```
for (k in p) {
```

```
  x = 0
```

```
  y = 0
```

```
  xx = 0
```

```
yy = 0
```

```
z = as.factor(0)
```

```
# i tar hånd om alle tilgjengelige porteføljer i k (alle 12 mnd porteføljer, alle 24 mnd porteføljer osv)
```

```
for (i in 1:(407 - k + 1)) {
```

```
  name <- paste(k, "months", i)
```

```
  rep <-
```

```
    c(rep(-100, nrow(mybiglist[[name]])),  
      mybiglist[[name]]$DCA_portfolio_end[nrow(mybiglist[[name]])])
```

```
  rep2 <-
```

```
    c(-100 * nrow(mybiglist[[name]]),  
      rep(0, nrow(mybiglist[[name]]) - 1),  
      mybiglist[[name]]$Buy_Hold_end[nrow(mybiglist[[name]])])
```

```
  x[i] <- irr(rep)
```

```
  y[i] <- irr(rep2)
```

```
  xx[i] <- (((1 + x[i]) ^ 12) - 1) # annualiserer IRR
```

```
  yy[i] <- (((1 + y[i]) ^ 12) - 1) # annualiserer IRR
```

```
  per <- mybiglist[[name]]$period[1] # henter faktorvariabel
```

```
  if (i == (407 - k + 1)) {
```

```
    if (k == 12) {
```

```
      irrdca <-
```

```
        data.frame(  
          irr = x,
```

```
    irr_a = xx,  
    strategi = "DCA",  
    period = per  
  )  
  irrbh <-  
  data.frame(  
    irr = y,  
    irr_a = yy,  
    strategi = "LS",  
    period = per  
  )  
  irrdf <-  
  rbind(irrdca, irrbh)  
  
} else if (k > 12) {  
  irrdca <-  
  data.frame(  
    irr = x,  
    irr_a = xx,  
    strategi = "DCA",  
    period = per  
  )  
  irrbh <-  
  data.frame(  
    irr = y,  
    irr_a = yy,  
    strategi = "LS",  
    period = per  
  )  
  irrdfnew <-  
  rbind(irrdca, irrbh)
```



```

irrdf <- rbind(irrdf, irrdfnew)

}

}

}

}

# Wide format
irrdca <- irrdf[irrdf$strategi=="DCA",]
irrbh <- irrdf[irrdf$strategi=="LS",]
irrdf_wide <- cbind(irrdca,irrbh)
irrdf_wide <- irrdf_wide[,c(1,2,5,6,8)]
colnames(irrdf_wide) <- c("irrdca", "irrdca_a", "irrbh", "irrbh_a",
                        "period")

# Annualisert avkastning
# k tar hånd om periodan (investere man i 12 mnd, 24 mnd osv)
# Lump Sum
for (k in p) {
  statlist <- list()
  temp = 0
  temp2 = 0

  # i tar hånd om alle tilgjengelige porteføljer i k (alle 12 mnd porteføljer osv)
  for (i in 1:(407 - k + 1)) {
    name <- paste(k, "months", i)

    stats <- data.frame(rep(0, nrow(mybiglist[[name]])))

    stats$bh <-
      (mybiglist[[name]]$Buy_Hold_end / mybiglist[[name]]$Buy_Hold_start) - 1

```

```
rownames(stats) <- rownames(mybiglist[[name]])
```

```
statlist[[i]] <- as.xts(stats[-1])
```

```
temp2[i] <- rbind(  
  table.AnnualizedReturns(statlist[[i]]),  
  maxDrawdown(statlist[[i]]),  
  SortinoRatio(statlist[[i]])  
)
```

```
if (i == (407 - k + 1)) {  
  z <- mybiglist[[name]]$period[1]  
  temp <- do.call(rbind, temp2)  
  if (k == 12) {  
    returnbh <-
```

```
    data.frame(  
      return = temp[, 1],  
      sd = temp[, 2],  
      sharpe = temp[, 3],  
      drawdown = temp[, 4],  
      strategi = "LS",  
      sortino = temp[, 5],  
      period = as.factor(z)  
    )
```

```
} else if (k > 12) {  
  returnbhnew <-
```

```

data.frame(
  return = temp[, 1],
  sd = temp[, 2],
  sharpe = temp[, 3],
  drawdown = temp[, 4],
  sortino = temp[, 5],
  strategi = "LS",
  period = as.factor(z)
)

returnbh <- rbind(returnbh, returnbhnew)
}
}
}
}
for (k in p) {
  statlist <- list()
  temp = 0
  temp2 = 0

  # i tar hånd om alle tilgjengelige porteføljer i k (alle 12 mnd porteføljer osv)
  for (i in 1:(407 - k + 1)) {
    name <- paste(k, "months", i)

    stats <- data.frame(rep(0, nrow(mybiglist[[name]])))

    stats$bh <-
      (mybiglist[[name]]$DCA_portfolio_end / mybiglist[[name]]$DCA_portfolio_start) - 1

    rownames(stats) <- rownames(mybiglist[[name]])
  }
}

```

```

statlist[[i]] <- as.xts(stats[-1])

temp2[i] <- rbind(
  table.AnnualizedReturns(statlist[[i]]),
  maxDrawdown(statlist[[i]]),
  SortinoRatio(statlist[[i]])
)

if (i == (407 - k + 1)) {
  z <- mybiglist[[name]]$period[1]
  temp <- do.call(rbind, temp2)
  if (k == 12) {
    returndca <-

    data.frame(
      return = temp[, 1],
      sd = temp[, 2],
      sharpe = temp[, 3],
      drawdown = temp[, 4],
      strategi = "DCA",
      sortino = temp[, 5],
      period = as.factor(z)

    )

  } else if (k > 12) {
    returndcanew <-

    data.frame(
      return = temp[, 1],

```

```

sd = temp[, 2],
sharpe = temp[, 3],
drawdown = temp[, 4],
sortino = temp[, 5],
strategi = "DCA",
period = as.factor(z)
)

returndca <- rbind(returndca, returndcanew)
}
}
}
}
# Long format
returndcabh <- rbind(returndca, returnbh)
returndcabh[mapply(is.infinite, returndcabh)] <- 0
returndcabh[mapply(is.na, returndcabh)] <- 0
# Wide format
returndcabh_wide <- cbind(returndca, returnbh)
returndcabh_wide[mapply(is.infinite, returndcabh_wide)] <- 0
returndcabh_wide[mapply(is.na, returndcabh_wide)] <- 0
returndcabh_wide <- returndcabh_wide[,c(1,2,3,6,8,9,10,13,14)]
colnames(returndcabh_wide) <- c("returndca", "sddca", "sharpedca", "sortinodca",
"returnbh", "sdbh", "sharpebh", "sortinobh", "period")
...

# Plotting
``{r plotting med faktora}
##### BW PLOTS #####
### Avkastning

```

```
ggplot(aes(y = return, x = period, fill = strategi), data = returndcabh) +  
geom_boxplot()+labs(title="Avkastning", x = "Sparehorisont (måneder) ",  
y="")+scale_y_continuous(labels=percent)+ theme_classic() + png("avkastning_bw.png", width =  
7200, height = 3600, res = 600) + theme(panel.spacing = unit(1, "lines"))
```

Standardavvik

```
ggplot(aes(y = sd, x = period, fill = strategi), data = returndcabh) + geom_boxplot()+  
geom_boxplot()+labs(title="Standardavvik", x = "Sparehorisont (måneder) ",  
y="")+scale_y_continuous(labels=percent)+ theme_classic() + png("standardavvik_bw.png", width =  
7200, height = 3600, res = 600) + theme(panel.spacing = unit(1, "lines"))
```

Sharpe

```
ggplot(aes(y = sharpe, x = period, fill = strategi), data = returndcabh) +  
geom_boxplot()+labs(title="Sharpe", x = "Sparehorisont (måneder) ",  
y="")+scale_y_continuous(labels=percent)+ theme_classic() + png("sharpe_bw.png", width = 7200,  
height = 3600, res = 600) + theme(panel.spacing = unit(1, "lines"))
```

IRR

```
ggplot(aes(y = irr_a, x = period, fill = strategi), data = irrdf) +  
geom_boxplot()+labs(title="Internrente", x = "Sparehorisont (måneder) ",  
y="")+scale_y_continuous(labels=percent)+ theme_classic() + png("internrente_bw.png", width =  
7200, height = 3600, res = 600) + theme(panel.spacing = unit(1, "lines"))
```

Sortino

```
ggplot(aes(y = sortino, x = period, fill = strategi), data = returndcabh) +  
geom_boxplot()+labs(title="Sortino", x = "Sparehorisont (måneder) ",  
y="")+scale_y_continuous(labels=percent)+ theme_classic() + png("sortino_bw.png", width = 7200,  
height = 3600, res = 600) + theme(panel.spacing = unit(1, "lines"))
```

HISTOGRAM PLOTS

Avkastning

```
ggplot(returndcabh, aes(return, fill = strategi)) + geom_histogram( color ="grey30", binwidth=0.02) +  
facet_wrap(~period) + labs(title="Avkastning per sparehorisont", x = "%Avkastning", y= "") +  
scale_x_continuous(labels=percent) + geom_vline(xintercept=0, linetype="dashed", color="red") +  
png("avkastning.png", width = 7200, height = 3600, res = 600) + theme(panel.spacing = unit(1,  
"lines")) + theme_classic()
```

Standardavvik

```
ggplot(returndcabh, aes(sd, fill = strategi)) + geom_histogram( color ="grey30", binwidth=0.02) +  
facet_wrap(~period) + labs(title="Standardavvik per sparehorisont", x = "%Standardavvik", y= "") +  
scale_x_continuous(labels=percent) + geom_vline(xintercept=0, linetype="dashed", color="red") +  
scale_fill_discrete(name = "Strategi", labels = c("Lump Sum", "Dollar Cost Averaging")) +  
png("standardavvik.png", width = 7200, height = 3600, res = 600) + theme(panel.spacing = unit(1,  
"lines"))
```

Sharpe

```
ggplot(returndcabh, aes(sharpe, fill = strategi)) + geom_histogram( color ="grey30", binwidth=0.15)
+ facet_wrap(~period) + labs(title="Sharpe per sparehorisont", x = "%Sharpe", y= "") +
geom_vline(xintercept=0, linetype="dashed", color="red") + scale_fill_discrete(name = "Strategi",
labels = c("Lump Sum", "Dollar Cost Averaging")) +
```

```
png("sharpe.png", width = 7200, height = 3600, res = 600) + theme(panel.spacing = unit(1, "lines"))
```

```
### Sortino
```

```
ggplot(returndcabh, aes(sortino, fill = strategi)) + geom_histogram( color ="grey30", binwidth=0.15)
+ facet_wrap(~period, scales="free") + labs(title="Sortino per sparehorisont", x = "%Sortino", y= "") +
geom_vline(xintercept=0, linetype="dashed", color="red") + scale_fill_discrete(name = "Strategi",
labels = c("Lump Sum", "Dollar Cost Averaging")) +
```

```
png("sortino.png", width = 7200, height = 3600, res = 600) + theme(panel.spacing = unit(1, "lines"))
```

```
### Internrente
```

```
ggplot(irrdf, aes(irr_a, fill = strategi)) + geom_histogram(color ="grey30", binwidth=0.05) +
facet_wrap(~period)+ labs(title="Internrente", x = "%Internrente", y= "") +
scale_x_continuous(labels=percent) + geom_vline(xintercept=0, linetype="dashed", color="red") +
```

```
png("internrente.png", width = 7200, height = 3600, res = 600) + theme(panel.spacing = unit(1,
"lines"))
```

```
#### PLOTTE ET MER SPESIFISERT UTVALG AV PERIODER (T = 12, 60, 120 måneder)
```

```
# Subset
```

```
returndcabh_subset <- returndcabh %>% filter(period %in% c(12,60,120))
```

```
irrdf_subset <- irrdf %>% filter(period %in% c(12,60,120))
```

```
### DENSITY PLOTS
```

```
### Avkastning (12,60,120)
```

```
mu <- plyr::ddply(returndcabh_subset, "strategi", summarise, grp.mean=mean(return))
```

```
ggplot(returndcabh_subset, aes(x = return, fill = strategi)) + geom_density(alpha=0.5) +
facet_wrap(~period, scales="free") + labs(title="Avkastning per sparehorisont", x = "%Avkastning",
y= "") + scale_x_continuous(labels=percent) + geom_vline(data=mu, aes(xintercept=grp.mean,
color=strategi), linetype="dashed") +
```

```
png("avkastning_subset.png", width = 7200, height = 3600, res = 600) + theme_classic()
```

```
### Standardavvik (12,60,120)
```

```
mu <- plyr::ddply(returndcabh_subset, "strategi", summarise, grp.mean=mean(sd))
```

```
ggplot(returndcabh_subset, aes(x = sd, fill = strategi)) + geom_density(alpha=0.5) +
facet_wrap(~period, scales="free") + labs(title="Standardavvik per sparehorisont", x =
"%Standardavvik", y= "") + scale_x_continuous(labels=percent) + geom_vline(data=mu,
aes(xintercept=grp.mean, color=strategi), linetype="dashed") +
```

```
png("standardavvik_subset.png", width = 7200, height = 3600, res = 600) + theme_classic()
```

```

### Sharpe (12,60,120)

mu <- plyr::ddply(returndcabh_subset, "strategi", summarise, grp.mean=mean(sharpe))

ggplot(returndcabh_subset, aes(x = sharpe, fill = strategi)) + geom_density(alpha=0.5) +
facet_wrap(~period, scales="free") + labs(title="Sharpe per sparehorisont", x = "%Sharpe", y= "") +
scale_x_continuous(labels=percent) + geom_vline(data=mu, aes(xintercept=grp.mean,
color=strategi), linetype="dashed") +

png("sharpe_subset.png", width = 7200, height = 3600, res = 600) + theme_classic()

### Sortino (12,60,120)

returndcabh_subset$sortino[is.na(returndcabh_subset$sortino)] = 0

mu <- plyr::ddply(returndcabh_subset, "strategi", summarise, grp.mean=mean(sortino))

ggplot(returndcabh_subset, aes(x = sortino, fill = strategi)) + geom_density(alpha=0.5) +
facet_wrap(~period, scales="free") + labs(title="Sortino per sparehorisont", x = "%Sortino", y= "") +
scale_x_continuous(labels=percent) + geom_vline(data=mu, aes(xintercept=grp.mean,
color=strategi), linetype="dashed") +

png("sortino_subset.png", width = 7200, height = 3600, res = 600) + theme_classic()

### Internrente (12,60,120)

mu <- plyr::ddply(irrdf_subset, "strategi", summarise, grp.mean=mean(irr_a))

ggplot(irrdf_subset, aes(irr_a, fill = strategi)) + geom_density(alpha = 0.5) + facet_wrap(~period,
scales="free")+ labs(title="Internrente per sparehorisont", x = "%Internrente", y= "") +
scale_x_continuous(labels=percent) + geom_vline(data=mu, aes(xintercept=grp.mean,
color=strategi), linetype="dashed") +

  png("internrente_subset.png", width = 7200, height = 3600, res = 600) + theme(panel.spacing =
unit(1, "lines"))

#### SCATTERPLOT FOR Å VISE PAIRED OBSERVASJONA

# Lage wide-format for scatterplot

returndcabh_subset_wide <- returndcabh_wide %>% filter(period %in% c(12,60,120))

irrdf_subset_wide <- irrdf_wide %>% filter(period %in% c(12,60,120))

### Avkastning (12, 60, 120)

ggplot(returndcabh_subset_wide, aes(x=returndca, y = returnbh)) + geom_point(size=0.5) + labs(title =
=
"Avkastning per sparehorisont", x = "DCA", y =
"LS") + geom_abline(slope = 1, intercept = 0, color="red") + facet_wrap(~period, scales = "free") +
png("avkastning_scatter.png", width = 7200, height = 3600, res = 600)

### Standardavvik (12, 60, 120)

ggplot(returndcabh_subset_wide, aes(x = sddca, y = sdbh)) + geom_point(size=0.5) + labs(title =

```



```
      "Standardavvik per sparehorisont", x = "DCA", y =
"LS") + geom_abline(slope = 1, intercept = 0, color="red") + facet_wrap(~period, scales = "free") +
png("standardavvik_scatter.png", width = 7200, height = 3600, res = 600)
```

```
### Sharpe (12, 60, 120)
```

```
ggplot(returndcabh_subset_wide, aes(x = sharpedca, y = sharpebh)) + geom_point(size=0.5) +
labs(title =
```

```
      "Sharpe per sparehorisont", x = "DCA", y = "LS") +
geom_abline(slope = 1, intercept = 0, color="red") + facet_wrap(~period, scales = "free") +
png("sharpe_scatter.png", width = 7200, height = 3600, res = 600)
```

```
### Sortino (12, 60, 120)
```

```
ggplot(returndcabh_subset_wide, aes(x = sortinodca, y = sortinobh)) + geom_point(size=0.5) +
labs(title =
```

```
      "Sortino per sparehorisont", x = "DCA", y = "LS") +
geom_abline(slope = 1, intercept = 0, color="red") + facet_wrap(~period, scales = "free") +
png("sortino_scatter.png", width = 7200, height = 3600, res = 600)
```

```
### Internrente (12, 60, 120)
```

```
ggplot(irrdf_subset_wide, aes(x = irrdca_a, y = irrbh_a)) + geom_point(size=0.5) + labs(title =
```

```
      "Internrente per sparehorisont", x = "DCA", y =
"LS") + geom_abline(slope = 1, intercept = 0, color="red") + facet_wrap(~period, scales = "free") +
png("internrente_scatter.png", width = 7200, height = 3600, res = 600)
```

```
...
```

```
# Parede t-tester på nøkkeltallene
```

```
``{r hypotese}
```

```
## avkastning
```

```
# For samtlige perioder - LS har høyere avkastning utover hele
```

```
all.test <- t.test(return~strategi, returndcabh, paired=T, alternative = "greater")
```

```
# teste per periode
```

```
my.t = function(fac1) {
```

```
  t.test(return[returndcabh$period == fac1] ~ strategi[returndcabh$period == fac1], data =
    returndcabh, paired = T, alternative = "greater")
```

```
}
```

```
my.data <- lapply(p, my.t)
```

```
my.data[[13]] <- all.test
```

```

my.estimate <- data.frame(t(matrix(unlist(my.data), ncol = length(p)+1)))
colnames(my.estimate) <- names(my.t(12))
my.estimate <- my.estimate[,c(1,2,3,6,7)]
my.estimate <- mapply(my.estimate, FUN=as.numeric)
my.matrix <- as.matrix(my.estimate)
rownames(my.matrix) <- c(p, "sum")
colnames(my.matrix) <- c("t-verdi", "df", "p-verdi", "estimat", "stdfeil")
stargazer(my.matrix, digits = 4, type = "text", out="avkastning_t-test.txt", title = "Avkastning")
# standardavvik
# For samtlige perioder - LS har høyere SD utover hele
all.test <- t.test(sd~strategi, returndcabh, paired=T)
# teste per periode
my.t = function(fac1) {
  t.test(sd[returndcabh$period == fac1] ~ strategi[returndcabh$period == fac1], data =
    returndcabh, paired = T)
}
my.data <- lapply(p, my.t)
my.data[[13]] <- all.test
my.estimate <- data.frame(t(matrix(unlist(my.data), ncol = length(p)+1)))
colnames(my.estimate) <- names(my.t(12))
my.estimate <- my.estimate[,c(1,2,3,6,7)]
my.estimate <- mapply(my.estimate, FUN=as.numeric)
my.matrix <- as.matrix(my.estimate)
rownames(my.matrix) <- c(p, "sum")
colnames(my.matrix) <- c("t-verdi", "df", "p-verdi", "estimat", "stdfeil")
stargazer(my.matrix, digits = 4, type = "text", out="sd_t-test.txt", title = "Standardavvik")
# Sharpe
# For samtlige perioder - Ingen signifikant forskjell
all.test <- t.test(sharpe~strategi, returndcabh, paired=T, alternative = "greater")
all.test
# teste per periode

```

```

my.t = function(fac1) {
  t.test(sharpe[returndcabh$period == fac1] ~ strategi[returndcabh$period == fac1], data =
    returndcabh, paired = T, alternative = "greater")
}
my.data <- lapply(p, my.t)
my.data[[13]] <- all.test
my.estimate <- data.frame(t(matrix(unlist(my.data), ncol = length(p)+1)))
colnames(my.estimate) <- names(my.t(12))
my.estimate <- my.estimate[,c(1,2,3,6,7)]
my.estimate <- mapply(my.estimate, FUN=as.numeric)
my.matrix <- as.matrix(my.estimate)
rownames(my.matrix) <- c(p, "sum")
colnames(my.matrix) <- c("t-verdi", "df", "p-verdi", "estimat", "stdfeil")
stargazer(my.matrix, digits = 3, type = "text", out="sharpe_t-test.txt", title = "Sharpe")
# Sortino
all.test <- t.test(sortino~strategi, returndcabh, paired=T, alternative = "greater")
my.t = function(fac1) {
  t.test(sortino[returndcabh$period == fac1] ~ strategi[returndcabh$period == fac1], data =
    returndcabh, paired = T, alternative = "greater")
}
my.data <- lapply(p, my.t)
my.data[[13]] <- all.test
my.estimate <- data.frame(t(matrix(unlist(my.data), ncol = length(p)+1)))
colnames(my.estimate) <- names(my.t(12))
my.estimate <- my.estimate[,c(1,2,3,6,7)]
my.estimate <- mapply(my.estimate, FUN=as.numeric)
my.matrix <- as.matrix(my.estimate)
rownames(my.matrix) <- c(p, "sum")
colnames(my.matrix) <- c("t-verdi", "df", "p-verdi", "estimat", "stdfeil")
stargazer(my.matrix, digits = 3, type = "text", out="sortino_t-test.txt", title = "Sortino")
# Internrente

```

```

all.test <- t.test(irr_a~strategi, irrdf, paired=T, alternative = "greater")
my.t = function(fac1) {
  t.test(irr_a[irrdf$period == fac1] ~ strategi[irrdf$period == fac1], data =
    irrdf, paired = T, alternative = "greater")
}
plot(lm(irr_a~strategi, irrdf))
my.data <- lapply(p, my.t)
my.data[[13]] <- all.test
my.estimate <- data.frame(t(matrix(unlist(my.data), ncol = length(p)+1)))
colnames(my.estimate) <- names(my.t(12))
my.estimate <- my.estimate[,c(1,2,3,6,7)]
my.estimate <- mapply(my.estimate, FUN=as.numeric)
my.matrix <- as.matrix(my.estimate)
rownames(my.matrix) <- c(p, "sum")
colnames(my.matrix) <- c("t-verdi", "df", "p-verdi", "estimat", "stdfeil")
stargazer(my.matrix, digits = 3, type = "text", out="internrente_t-test.txt", title = "Internrente")
test <- returndcabh[returndcabh$period==144,]
mean(test$return~test$strategi)
mean(returndcabh[,1]~returndcabh$period==144&returndcabh$strategi=="LS")
# Summere opp nøkkeltallene
# 1 = avkastning, 2 = standardavvik, 3 = sharpe, 6 = sortino
q = c(1,2,3,6)
my.mean = function(fac1) {
  mean(returndcabh[,fac1]~returndcabh$strategi+returndcabh$period)
}
my.max = function(fac1) {
  max(returndcabh[,fac1]~returndcabh$strategi+returndcabh$period)
}
my.data <- lapply(q, my.mean)
my.frame <- as.data.frame(my.data)
colnames(my.frame) <- colnames(returndcabh[,q])

```

```

temp.matrix <- t(matrix(unlist(my.frame), nrow=2))
result.matrix <- reticulate::array_reshape(temp.matrix, c(12, 8), order = "F")
result.matrix <- result.matrix[,c(1,5,2,6,3,7,4,8)] # Fikser rekkefølgen
rownames(result.matrix) <- p
colnames(result.matrix) <- c("Avkastning DCA", "Avkastning LS", "Sd DCA", "Sd LS", "Sharpe DCA",
"Sharpe LS", "Sortino DCA", "Sortino LS")
stargazer(t(result.matrix), digits = 3, type = "text", out="Resultater.txt", title = "Resultater")
# BEREGNER DEN RELATIVE ANDELEN AV SIMULERINGER DER DCA "VINNER"
returndcabh2 <- returndcabh[,c(q,7,8)]
returncompare <- cbind(returndca, returnbh)
returncompare$swinsreturn <- 0
returncompare$swinssharpe <- 0
mean(returndcabh2$return~returndcabh2$strategi)
for (i in 1:nrow(returncompare)) {
  if (returncompare[i,1]<returncompare[i,8]) {
    returncompare[i,15] <- 1
  }
  if (returncompare[i,3]<returncompare[i,10]) {
    returncompare[i,16] <- 1
  }
}
l <- split(returncompare, returncompare$period)
x <- 0
y <- 0
for (i in 1:12) {
  x[i] <- 1 - (sum(l[[i]]$swinsreturn)/nrow(l[[i]]))
  y[i] <- 1 - (sum(l[[i]]$swinssharpe)/nrow(l[[i]]))
}
wins <- data.frame(x,y)
colnames(wins) <- c("Avkastning", "Sharpe")
rownames(wins) <- p

```

```
stargazer(t(t(wins)), type = "text")
```

```
...
```

Over tid vil DCA begynne å underprestere LS men spesielt på kort sikt (1-3 år) har DCA mye mindre drawdowns og svingninger. Etterhvert som fondet øker i pris og færre fondsandeler kjøpes vil utviklingen til DCA portefølje gradvis begynne å ligne den for LS, fordi hvert innskudd i de siste periodene utgjør fondsandeler tilnærmet 0.

```
# Eksempel av porteføljeutvikling over tid for begge strategiene
``{r plot}
# Plot porteføljeverdi
t <- cbind(mybiglist[["120 months 100"]][3],mybiglist[["120 months 100"]][7])
plot(t)
# Prøve %vis avkastning
x <- as.data.frame(mybiglist[["120 months 100"]]$DCA_portfolio_end)
rownames(x) <- rownames(mybiglist[["120 months 100"]])
xxx <- periodReturn(as.xts(x), period = "monthly", type = "arithmetic")
x2 <- as.data.frame(mybiglist[["120 months 100"]]$Buy_Hold_end)
rownames(x2) <- rownames(mybiglist[["120 months 100"]])
xxx2 <- periodReturn(as.xts(x2), period = "monthly", type = "arithmetic")
t <- as.data.frame(cbind(xxx, xxx2))
rownames(t) <- rownames(mybiglist[["120 months 100"]])
t$Date <- index(t)
colnames(t) <- c("DCA", "LS", "Date")
f <- gather(t, key = Strategi, value = Porteføljeverdi,
c("LS", "DCA"))
ggplot(f, aes(x=Date, y = Porteføljeverdi, group = Strategi, colour = Strategi)) +
geom_line(size = 0.3) + labs(title="Porteføljeutvikling", x = "Sparehorisont (måneder)", y =
"%Endring") + scale_color_brewer(palette="Set1") + png("prosent_sammenligning.png", width =
3600, height = 1800, res = 600)
```

...

```
``{r differanseinvestering}
##### IRR differanseinvestering
z = 0
zz = 0
per = as.factor(0)
# k tar hånd om periodan (investere man i 12 mnd, 24 mnd osv)
for (k in p) {
  x = 0
  y = 0
  xx = 0
  yy = 0
  z = 0
  zz = 0
  lswins = 0
  per = as.factor(0)

  # i tar hånd om alle tilgjengelige porteføljer i k (alle 12 mnd porteføljer, alle 24 mnd porteføljer osv)
  for (i in 1:(407 - k + 1)) {
    rep = 0
    rep2 = 0
    name <- paste(k, "months", i)

    rep <-
      c(rep(-100, nrow(mybiglist[[name]])),
        mybiglist[[name]]$DCA_portfolio_end[nrow(mybiglist[[name]])])
```

```
rep2 <-  
c(-100 * nrow(mybiglist[[name]]),  
  rep(0, nrow(mybiglist[[name]])-1),  
  mybiglist[[name]]$Buy_Hold_end[nrow(mybiglist[[name]])])
```

```
x[i] <- irr(rep)  
y[i] <- irr(rep2)  
lswins[i] <- 0  
if (sum(rep)>sum(rep2)) {  
  z[i] <- irr(rep - rep2)  
  if (!is.na(z[i])) {  
    if (z[i]>0) {  
      lswins[i] <- 0  
    }  
  }  
}
```

```
} else if (sum(rep)<sum(rep2)) {  
  z[i] <- irr(rep2 - rep)  
  if (!is.na(z[i])) {  
    if (z[i]>0) {  
      lswins[i] <- 1  
    }  
  }  
}
```

```
xx[i] <- (((1 + x[i]) ^ 12) - 1) # annualiserer IRR  
yy[i] <- (((1 + y[i]) ^ 12) - 1) # annualiserer IRR  
zz[i] <- (((1 + z[i]) ^ 12) - 1) # annualiserer IRR
```



```
per <- mybiglist[[name]]$period[1] # henter faktorvariabel
```

```
if (i == (407 - k + 1)) {
```

```
  if (k == 12) {
```

```
    irrdca <-
```

```
      data.frame(irr = x, irr_a = xx, strategi = "DCA", period = per)
```

```
    irrbh <-
```

```
      data.frame(irr = y, irr_a = yy, strategi = "LS", period = per)
```

```
    irrdiff <-
```

```
      data.frame(irr = z, irr_a = zz, win = lswins, strategi = "DIFF", period = per)
```

```
  irrdf <-
```

```
    rbind(irrdca, irrbh)
```

```
} else if (k > 12) {
```

```
  irrdca <-
```

```
    data.frame(
```

```
      irr = x,
```

```
      irr_a = xx,
```

```
      strategi = "DCA",
```

```
      period = per
```

```
    )
```

```
  irrbh <-
```

```
    data.frame(
```

```
      irr = y,
```

```
      irr_a = yy,
```

```
      strategi = "LS",
```

```
      period = per
```

```

)
irrdiffnew <-
  data.frame(
    irr = z,
    irr_a = zz,
    win = lswins,
    strategi = "DIFF",
    period = per
  )
irrdfnew <-
  rbind(irrdca, irrbh)

irrdiff <- rbind(irrdiff, irrdiffnew)
irrdf <- rbind(irrdf, irrdfnew)

}
}
}
}
print(table(irrdiff$win, irrdiff$period))
```

```

```
ROBUSTNES
```

```
```{r robustnes, RF, transaksjon, forvaltning}
```

```
# NY LOGIKK MED RISIKOFRI RENTE, TRANSAKSJONSKOSTNADER OG FORVALTNINGSHONORAR
```

```
o <- 1
```

```
m <- 1
```

```
i <- 1
```

```
k <- 1
```

```

robustList <- list()
robustList2 <- list()
for (m in 1:nrow(prices_clean)) {
  for (i in m:nrow(prices_clean)) {
    setfour <- new.env() # Lager en ny environment hver runde
    Returns <-
      as.numeric(prices_clean[m:i, 1]) # Hent månedlig avkastning
    frame <- data.frame>Returns) # Putt det i Frame
    rownames(frame) <- Dates[m:i] # Sett radnavnan til Dato
    setfour$frame <- frame # Tildele nytt datasett til enviroment
    frame$period <- as.factor(nrow(frame)) # Tildele en faktorvariabel basert på sparehorisontens
    lengde
    dcasum <- 100 # Tildeler beløpet som investeres månedlig
    transaksjonskost <- 0.002
    forvaltningshonorar <- 0.002
    risikofri <- 0.0005

    # DCA
    frame$DCA <- rep(0, nrow(frame))
    frame$DCA[1] <- dcasum * (1 + frame>Returns[1] - transaksjonskost) #transaksjonskostnad
    frame$DCA_cash <- rep(0, nrow(frame))
    frame$DCA_cash[1] <- ((nrow(frame) * dcasum) - dcasum) * (1+risikofri)
    frame$DCA_portfolio_end <- rep(0, nrow(frame))
    frame$DCA_portfolio_end[1] <-
      frame$DCA_cash[1] + frame$DCA[1]
    frame$DCA_portfolio_start <- rep(0, nrow(frame))
    frame$DCA_portfolio_start[1] <- (nrow(frame) * dcasum)

    # Lump Sum
    frame$Buy_Hold_end <- rep(0, nrow(frame))
    frame$Buy_Hold_end[1] <-

```

```

nrow(frame) * dcasum * (1 + frame$Returns[1] - transaksjonskost) #transaksjonskostnad
frame$Buy_Hold_start <- rep(0, nrow(frame))
frame$Buy_Hold_start[1] <- (nrow(frame) * dcasum)

if ((i - m) > 0) {
  for (j in 2:nrow(frame)) {
    if (j %% 12 == 0) {
      frame$Buy_Hold_end[j] <-
        frame$Buy_Hold_end[j - 1] * (1 + frame$Returns[j] - forvaltningshonorar) #forvaltningshonorar
      frame$Buy_Hold_start[j] <- frame$Buy_Hold_end[j - 1]
    } else {
      frame$Buy_Hold_end[j] <-
        frame$Buy_Hold_end[j - 1] * (1 + frame$Returns[j])
      frame$Buy_Hold_start[j] <- frame$Buy_Hold_end[j - 1]
    }
  }
}

if (i > m) {
  for (k in 2:nrow(frame)) {
    if (k %% 12 == 0) {
      # Hver 12. måned...
      frame$DCA[k] <-
        (frame$DCA[k - 1] + dcasum) * (1 + frame$Returns[k] - forvaltningshonorar - transaksjonskost)
# forvaltningshonorar
      frame$DCA_cash[k] <-
        (frame$DCA_cash[k - 1] - dcasum) * (1 + risikofri)
      frame$DCA_portfolio_end[k] <-
        frame$DCA_cash[k] + frame$DCA[k]
      frame$DCA_portfolio_start[k] <-

```

```

    frame$DCA_portfolio_end[k - 1]
  } else {
    frame$DCA[k] <-
      (frame$DCA[k - 1] + dcasum) * (1 + frame>Returns[k] - transaksjonskost)
    frame$DCA_cash[k] <-
      (frame$DCA_cash[k - 1] - dcasum) * (1 + risikofri)
    frame$DCA_portfolio_end[k] <-
      frame$DCA_cash[k] + frame$DCA[k]
    frame$DCA_portfolio_start[k] <-
      frame$DCA_portfolio_end[k - 1]
  }
}
}

```

```

# Lagre resultatene

```

```

robustList2[[o]] <- frame

```

```

# Øker indeksen for neste simulering

```

```

o = o + 1

```

```

}

```

```

}

```

```

robust2 <- list()

```

```

for (j in 1:407) {

```

```

  k = 1

```

```

  for (i in 1:length(robustList2)) {

```

```

    if (nrow(robustList2[[i]]) == j) {

```

```

      name <- paste(j, "months", k)

```

```

      robust2[[name]] <- robustList2[[i]]

```

```

      k = k + 1

```

```

    }

```

```

}
}
...

```{r robust RF, Transaksjon, Forvaltning test}
Annualisert avkastning
k tar hånd om periodan (investere man i 12 mnd, 24 mnd osv)
Lump Sum
for (k in p) {
 statlist <- list()
 temp = 0
 temp2 = 0

 # i tar hånd om alle tilgjengelige porteføljer i k (alle 12 mnd porteføljer, alle 24 mnd porteføljer osv)
 for (i in 1:(407 - k + 1)) {
 name <- paste(k, "months", i)

 stats <- data.frame(rep(0, nrow(robust2[[name]])))

 stats$bh <-
 (robust2[[name]]$Buy_Hold_end / robust2[[name]]$Buy_Hold_start) - 1

 rownames(stats) <- rownames(robust2[[name]])

 statlist[[i]] <- as.xts(stats[-1])

 temp2[i] <- rbind(
 table.AnnualizedReturns(statlist[[i]]),
 maxDrawdown(statlist[[i]]),
 SortinoRatio(statlist[[i]])
)
 }
}

```

```

if (i == (407 - k + 1)) {
 z <- robust2[[name]]$period[1]
 temp <- do.call(rbind, temp2)
 if (k == 12) {
 returnbhrobust <-

 data.frame(
 return = temp[, 1],
 sd = temp[, 2],
 sharpe = temp[, 3],
 drawdown = temp[, 4],
 strategi = "LS",
 sortino = temp[, 5],
 period = as.factor(z)

)

 } else if (k > 12) {
 returnbhnew <-

 data.frame(
 return = temp[, 1],
 sd = temp[, 2],
 sharpe = temp[, 3],
 drawdown = temp[, 4],
 sortino = temp[, 5],
 strategi = "LS",
 period = as.factor(z)

)
 }
}

```

```

returnbhrobust <- rbind(returnbhrobust, returnbhnew)
}
}
}
}
for (k in p) {
 statlist <- list()
 temp = 0
 temp2 = 0

 # i tar hånd om alle tilgjengelige porteføljer i k (alle 12 mnd porteføljer, alle 24 mnd porteføljer osv)
 for (i in 1:(407 - k + 1)) {
 name <- paste(k, "months", i)

 stats <- data.frame(rep(0, nrow(robust2[[name]])))

 stats$bh <-
 (robust2[[name]]$DCA_portfolio_end / robust2[[name]]$DCA_portfolio_start) - 1

 rownames(stats) <- rownames(robust2[[name]])

 statlist[[i]] <- as.xts(stats[-1])

 temp2[i] <- rbind(
 table.AnnualizedReturns(statlist[[i]]),
 maxDrawdown(statlist[[i]]),
 SortinoRatio(statlist[[i]])
)

 if (i == (407 - k + 1)) {

```



```

z <- robust2[[name]]$period[1]
temp <- do.call(rbind, temp2)
if (k == 12) {
 returndcarobust <-

 data.frame(
 return = temp[, 1],
 sd = temp[, 2],
 sharpe = temp[, 3],
 drawdown = temp[, 4],
 strategi = "DCA",
 sortino = temp[, 5],
 period = as.factor(z)

)

} else if (k > 12) {
 returndcanew <-

 data.frame(
 return = temp[, 1],
 sd = temp[, 2],
 sharpe = temp[, 3],
 drawdown = temp[, 4],
 sortino = temp[, 5],
 strategi = "DCA",
 period = as.factor(z)

)

 returndcarobust <- rbind(returndcarobust, returndcanew)

```

```

 }
 }
}

Long format
returndcabhrobust <- rbind(returndcarobust, returnbhrobust)
returndcabhrobust[mapply(is.infinite, returndcabhrobust)] <- 0
returndcabhrobust[mapply(is.na, returndcabhrobust)] <- 0

Wide format
returndcabh_robust_wide <- cbind(returndcarobust, returnbhrobust)
returndcabh_robust_wide[mapply(is.infinite, returndcabh_robust_wide)] <- 0
returndcabh_robust_wide[mapply(is.na, returndcabh_robust_wide)] <- 0
returndcabh_robust_wide <- returndcabh_robust_wide[,c(1,2,3,6,8,9,10,13,14)]
colnames(returndcabh_robust_wide) <- c("returndca", "sddca", "sharpedca", "sortinodca",
 "returnbh", "sdbh", "sharpebh", "sortinobh", "period")

IRR ROBUST
for (k in p) {
 x = 0
 y = 0
 xx = 0
 yy = 0
 z = as.factor(0)

 # i tar hånd om alle tilgjengelige porteføljer i k (alle 12 mnd porteføljer, alle 24 mnd porteføljer osv)
 for (i in 1:(407 - k + 1)) {
 name <- paste(k, "months", i)

 rep <-
 c(rep(-100, nrow(robust2[[name]])),
 robust2[[name]]$DCA_portfolio_end[nrow(robust2[[name]])])
 }
}

```

```
rep2 <-
 c(-100 * nrow(robust2[[name]]),
 rep(0, nrow(robust2[[name]] - 1),
 robust2[[name]]$Buy_Hold_end[nrow(robust2[[name]])])
```

```
x[i] <- irr(rep)
y[i] <- irr(rep2)
```

```
xx[i] <- (((1 + x[i]) ^ 12) - 1) # annualiserer IRR
yy[i] <- (((1 + y[i]) ^ 12) - 1) # annualiserer IRR
```

```
per <- robust2[[name]]$period[1] # henter faktorvariabel
```

```
if (i == (407 - k + 1)) {
```

```
 if (k == 12) {
```

```
 irrdcarobust <-
```

```
 data.frame(
 irr = x,
```

```
 irr_a = xx,
```

```
 strategi = "DCA",
```

```
 period = per
)
```

```
 irrbhrobust <-
```

```
 data.frame(
 irr = y,
```

```
 irr_a = yy,
```

```
 strategi = "LS",
```

```
 period = per
)
```

```

irrdfrobust <-
 rbind(irrdcarobust, irrbhrobust)

} else if (k > 12) {
 irrdcarobust <-
 data.frame(
 irr = x,
 irr_a = xx,
 strategi = "DCA",
 period = per
)
 irrbhrobust <-
 data.frame(
 irr = y,
 irr_a = yy,
 strategi = "LS",
 period = per
)
 irrdfnewrobust <-
 rbind(irrdcarobust, irrbhrobust)

 irrdfrobust <- rbind(irrdfrobust, irrdfnewrobust)

}
}
}
}

Wide format
irrdcarobust <- irrdfrobust[irrdfrobust$strategi=="DCA",]
irrbhrobust <- irrdfrobust[irrdfrobust$strategi=="LS",]
irrdf_robust_wide <- cbind(irrdcarobust,irrbhrobust)

```

```

irrdf_robust_wide <- irrdf_robust_wide[,c(1,2,5,6,8)]
colnames(irrdf_robust_wide) <- c("irrdca", "irrdca_a", "irrbh", "irrbh_a",
 "period")
BEREGNER DEN PROSENTWISE ANDELEN DER DCA "VINNER"
returncomparerobust <- cbind(returndcarobust, returnbhrobust)
returncomparerobust$lswinsreturn <- 0
returncomparerobust$lswinssharpe <- 0
for (i in 1:nrow(returncomparerobust)) {
 if (returncomparerobust[i,1]<returncomparerobust[i,8]) {
 returncomparerobust[i,15] <- 1
 }
 if (returncomparerobust[i,3]<returncomparerobust[i,10]) {
 returncomparerobust[i,16] <- 1
 }
}
l <- split(returncomparerobust, returncomparerobust$period)
x = 0
y = 0
for (i in 1:12) {
 x[i] <- 1 - (sum(l[[i]]$lswinsreturn)/nrow(l[[i]]))
 y[i] <- 1 - (sum(l[[i]]$lswinssharpe)/nrow(l[[i]]))
}
winsrobust <- data.frame(x,y)
colnames(winsrobust) <- c("Avkastning", "Sharpe")
rownames(winsrobust) <- p
stargazer(t(t(winsrobust)), type = "text")
...

``{r robust t-test}
Parede t-tester
avkastning

```

```

For samtlige perioder - LS har høyere avkastning utover hele
all.test <- t.test(return~strategi, returndcabhrobust, paired=T)

teste per periode
my.t = function(fac1) {
 t.test(return[returndcabhrobust$period == fac1] ~ strategi[returndcabhrobust$period == fac1],
 data =
 returndcabhrobust, paired = T)
}

my.data <- lapply(p, my.t)
my.data[[13]] <- all.test

my.estimate <- data.frame(t(matrix(unlist(my.data), ncol = length(p)+1)))
colnames(my.estimate) <- names(my.t(12))
my.estimate <- my.estimate[,c(1,2,3,6,7)]
my.estimate <- mapply(my.estimate, FUN=as.numeric)
my.matrix <- as.matrix(my.estimate)
rownames(my.matrix) <- c(p, "sum")
colnames(my.matrix) <- c("t-verdi", "df", "p-verdi", "estimat", "stdfeil")
stargazer(my.matrix, digits = 4, type = "text", out="avkastning_t-test.txt", title = "Avkastning")

standardavvik

For samtlige perioder - LS har høyere SD utover hele
all.test <- t.test(sd~strategi, returndcabhrobust, paired=T)

teste per periode
my.t = function(fac1) {
 t.test(sd[returndcabhrobust$period == fac1] ~ strategi[returndcabhrobust$period == fac1], data =
 returndcabhrobust, paired = T)
}

my.data <- lapply(p, my.t)
my.data[[13]] <- all.test

my.estimate <- data.frame(t(matrix(unlist(my.data), ncol = length(p)+1)))
colnames(my.estimate) <- names(my.t(12))
my.estimate <- my.estimate[,c(1,2,3,6,7)]

```

```

my.estimate <- mapply(my.estimate, FUN=as.numeric)
my.matrix <- as.matrix(my.estimate)
rownames(my.matrix) <- c(p, "sum")
colnames(my.matrix) <- c("t-verdi", "df", "p-verdi", "estimat", "stdfeil")
stargazer(my.matrix, digits = 4, type = "text", out="sd_t-test.txt", title = "Standardavvik")

Sharpe
For samtlige perioder - Ingen signifikant forskjell
all.test <- t.test(sharpe~strategi, returndcabhrobust, paired=T)

teste per periode
my.t = function(fac1) {
 t.test(sharpe[returndcabhrobust$period == fac1] ~ strategi[returndcabhrobust$period == fac1],
 data =
 returndcabhrobust, paired = T)
}

my.data <- lapply(p, my.t)
my.data[[13]] <- all.test
my.estimate <- data.frame(t(matrix(unlist(my.data), ncol = length(p)+1)))
colnames(my.estimate) <- names(my.t(12))
my.estimate <- my.estimate[,c(1,2,3,6,7)]
my.estimate <- mapply(my.estimate, FUN=as.numeric)
my.matrix <- as.matrix(my.estimate)
rownames(my.matrix) <- c(p, "sum")
colnames(my.matrix) <- c("t-verdi", "df", "p-verdi", "estimat", "stdfeil")
stargazer(my.matrix, digits = 3, type = "text", out="sharpe_t-test.txt", title = "Sharpe")

Sortino
all.test <- t.test(sortino~strategi, returndcabhrobust, paired=T)
my.t = function(fac1) {
 t.test(sortino[returndcabhrobust$period == fac1] ~ strategi[returndcabhrobust$period == fac1],
 data =
 returndcabhrobust, paired = T)
}

my.data <- lapply(p, my.t)

```

```

my.data[[13]] <- all.test
my.estimate <- data.frame(t(matrix(unlist(my.data), ncol = length(p)+1)))
colnames(my.estimate) <- names(my.t(12))
my.estimate <- my.estimate[,c(1,2,3,6,7)]
my.estimate <- mapply(my.estimate, FUN=as.numeric)
my.matrix <- as.matrix(my.estimate)
rownames(my.matrix) <- c(p, "sum")
colnames(my.matrix) <- c("t-verdi", "df", "p-verdi", "estimat", "stdfeil")
stargazer(my.matrix, digits = 3, type = "text", out="sortino_t-test.txt", title = "Sortino")
Internrente
all.test <- t.test(irr_a~strategi, irrdfrobust, paired=T)
my.t = function(fac1) {
 t.test(irr_a[irrdfrobust$period == fac1] ~ strategi[irrdfrobust$period == fac1], data =
 irrdfrobust, paired = T)
}
plot(lm(irr_a~strategi, irrdf))
my.data <- lapply(p, my.t)
my.data[[13]] <- all.test
my.estimate <- data.frame(t(matrix(unlist(my.data), ncol = length(p)+1)))
colnames(my.estimate) <- names(my.t(12))
my.estimate <- my.estimate[,c(1,2,3,6,7)]
my.estimate <- mapply(my.estimate, FUN=as.numeric)
my.matrix <- as.matrix(my.estimate)
rownames(my.matrix) <- c(p, "sum")
colnames(my.matrix) <- c("t-verdi", "df", "p-verdi", "estimat", "stdfeil")
stargazer(my.matrix, digits = 3, type = "text", out="internrente_t-test.txt", title = "Internrente")
Summere opp nøkkeltallene
1 = avkastning, 2 = standardavvik, 3 = sharpe, 6 = sortino
q = c(1,2,3,6)
my.mean = function(fac1) {
 mean(returndcabhrobust[,fac1]~returndcabhrobust$strategi+returndcabhrobust$period)
}

```



```

}
my.max = function(fac1) {
 max(returndcabhrobust[,fac1]~returndcabhrobust$strategi+returndcabhrobust$period)
}
my.data <- lapply(q, my.mean)
my.frame <- as.data.frame(my.data)
colnames(my.frame) <- colnames(returndcabhrobust[,q])
temp.matrix <- t(matrix(unlist(my.frame), nrow=2))
result.matrix <- reticulate::array_reshape(temp.matrix, c(12, 8), order = "F")
result.matrix <- result.matrix[,c(1,5,2,6,3,7,4,8)] # Fikser rekkefølgen
rownames(result.matrix) <- p
colnames(result.matrix) <- c("Avkastning DCA", "Avkastning LS", "Sd DCA", "Sd LS", "Sharpe DCA",
"Sharpe LS", "Sortino DCA", "Sortino LS")
stargazer(t(result.matrix), digits = 3, type = "text", out="Resultater.txt", title = "Resultater")
...

``{r robust RF, Forvaltning, ingen transkasjonskost}
Gjentar logikken for å simulere DCA og LS, med risikofri rente og forvaltningshonorar inkludert
o <- 1
m <- 1
i <- 1
k <- 1
robustList3 <- list()
for (m in 1:nrow(prices_clean)) {
 for (i in m:nrow(prices_clean)) {
 setfour <- new.env() # Lager en ny environment hver runde
 Returns <-
 as.numeric(prices_clean[m:i, 1]) # Hent månedlig avkastning
 frame <- data.frame>Returns) # Putt det i Frame
 rownames(frame) <- Dates[m:i] # Sett radnavnan til Dato
 setfour$frame <- frame # Tildele nytt datasett til enviroment
 }
}

```

```

frame$period <- as.factor(nrow(frame)) # Tildeler en faktorvariabel basert på sporehorisontens
lengde

dcasum <- 100 # Tildeler beløpet som investeres månedlig

transaksjonskost <- 0.000

forvaltningshonorar <- 0.002

risikofri <- 0.0005

DCA
frame$DCA <- rep(0, nrow(frame))
frame$DCA[1] <- dcasum * (1 + frame$Returns[1] - transaksjonskost) #transaksjonskostnad
frame$DCA_cash <- rep(0, nrow(frame))
frame$DCA_cash[1] <- ((nrow(frame) * dcasum) - dcasum) * (1+risikofri)
frame$DCA_portfolio_end <- rep(0, nrow(frame))
frame$DCA_portfolio_end[1] <-
 frame$DCA_cash[1] + frame$DCA[1]
frame$DCA_portfolio_start <- rep(0, nrow(frame))
frame$DCA_portfolio_start[1] <- (nrow(frame) * dcasum)

Lump Sum
frame$Buy_Hold_end <- rep(0, nrow(frame))
frame$Buy_Hold_end[1] <-
 nrow(frame) * dcasum * (1 + frame$Returns[1] - transaksjonskost) #transaksjonskostnad
frame$Buy_Hold_start <- rep(0, nrow(frame))
frame$Buy_Hold_start[1] <- (nrow(frame) * dcasum)

if ((i - m) > 0) {
 for (j in 2:nrow(frame)) {
 if (j %% 12 == 0) {
 frame$Buy_Hold_end[j] <-
 frame$Buy_Hold_end[j - 1] * (1 + frame$Returns[j] - forvaltningshonorar) #forvaltningshonorar
 frame$Buy_Hold_start[j] <- frame$Buy_Hold_end[j - 1]
 }
 }
}

```

```

} else {
frame$Buy_Hold_end[j] <-
 frame$Buy_Hold_end[j - 1] * (1 + frame>Returns[j])
frame$Buy_Hold_start[j] <- frame$Buy_Hold_end[j - 1]
}
}
}

if (i > m) {
 for (k in 2:nrow(frame)) {
 if (k %% 12 == 0) {
 # Hver 12. måned...
 frame$DCA[k] <-
 (frame$DCA[k - 1] + dcasum) * (1 + frame>Returns[k] - forvaltningshonorar - transaksjonskost)
forvaltningshonorar
 frame$DCA_cash[k] <-
 (frame$DCA_cash[k - 1] - dcasum) * (1 + risikofri)
 frame$DCA_portfolio_end[k] <-
 frame$DCA_cash[k] + frame$DCA[k]
 frame$DCA_portfolio_start[k] <-
 frame$DCA_portfolio_end[k - 1]
 } else {
 frame$DCA[k] <-
 (frame$DCA[k - 1] + dcasum) * (1 + frame>Returns[k] - transaksjonskost)
 frame$DCA_cash[k] <-
 (frame$DCA_cash[k - 1] - dcasum) * (1 + risikofri)
 frame$DCA_portfolio_end[k] <-
 frame$DCA_cash[k] + frame$DCA[k]
 frame$DCA_portfolio_start[k] <-
 frame$DCA_portfolio_end[k - 1]
 }
 }
}

```

```

 }
 }
}

Lagre resultatene
robustList3[[o]] <- frame
Øker indeksen for neste simulering
o = o + 1

}
}
robust3 <- list()
for (j in 1:407) {
 k = 1
 for (i in 1:length(robustList3)) {
 if (nrow(robustList3[[i]]) == j) {
 name <- paste(j, "months", k)
 robust3[[name]] <- robustList3[[i]]
 k = k + 1
 }
 }
}

Annualisert avkastning
k tar hånd om periodan (investere man i 12 mnd, 24 mnd osv)
Lump Sum
for (k in p) {
 statlist <- list()
 temp = 0
 temp2 = 0

```

```

i tar hånd om alle tilgjengelige porteføljer i k (alle 12 mnd porteføljer osv)
for (i in 1:(407 - k + 1)) {
 name <- paste(k, "months", i)

 stats <- data.frame(rep(0, nrow(robust3[[name]])))

 stats$bh <-
 (robust3[[name]]$Buy_Hold_end / robust3[[name]]$Buy_Hold_start) - 1

 rownames(stats) <- rownames(robust3[[name]])

 statlist[[i]] <- as.xts(stats[-1])

 temp2[i] <- rbind(
 table.AnnualizedReturns(statlist[[i]]),
 maxDrawdown(statlist[[i]]),
 SortinoRatio(statlist[[i]])
)

 if (i == (407 - k + 1)) {
 z <- robust3[[name]]$period[1]
 temp <- do.call(rbind, temp2)
 if (k == 12) {
 returnbhrobust2 <-

 data.frame(
 return = temp[, 1],
 sd = temp[, 2],
 sharpe = temp[, 3],
 drawdown = temp[, 4],
 strategi = "LS",

```

```

 sortino = temp[, 5],
 period = as.factor(z)

)

} else if (k > 12) {
 returnbhnew2 <-

 data.frame(
 return = temp[, 1],
 sd = temp[, 2],
 sharpe = temp[, 3],
 drawdown = temp[, 4],
 sortino = temp[, 5],
 strategi = "LS",
 period = as.factor(z)
)

 returnbhrobust2 <- rbind(returnbhrobust2, returnbhnew2)
}
}
}
}

for (k in p) {
 statist <- list()
 temp = 0
 temp2 = 0

 # i tar hånd om alle tilgjengelige porteføljer i k (alle 12 mnd porteføljer osv)
 for (i in 1:(407 - k + 1)) {

```

```

name <- paste(k, "months", i)

stats <- data.frame(rep(0, nrow(robust3[[name]])))

stats$bh <-
 (robust3[[name]]$DCA_portfolio_end / robust3[[name]]$DCA_portfolio_start) - 1

rownames(stats) <- rownames(robust3[[name]])

statlist[[i]] <- as.xts(stats[-1])

temp2[i] <- rbind(
 table.AnnualizedReturns(statlist[[i]]),
 maxDrawdown(statlist[[i]]),
 SortinoRatio(statlist[[i]])
)

if (i == (407 - k + 1)) {
 z <- robust3[[name]]$period[1]
 temp <- do.call(rbind, temp2)
 if (k == 12) {
 returndcarobust2 <-

 data.frame(
 return = temp[, 1],
 sd = temp[, 2],
 sharpe = temp[, 3],
 drawdown = temp[, 4],
 strategi = "DCA",
 sortino = temp[, 5],
 period = as.factor(z)
)
 }
}

```

```

)

} else if (k > 12) {
 returndcanew2 <-

 data.frame(
 return = temp[, 1],
 sd = temp[, 2],
 sharpe = temp[, 3],
 drawdown = temp[, 4],
 sortino = temp[, 5],
 strategi = "DCA",
 period = as.factor(z)
)

 returndcarobust2 <- rbind(returndcarobust2, returndcanew2)
}
}
}

Long format
returndcabhrobust2 <- rbind(returndcarobust2, returnbhrobust2)
returndcabhrobust2[mapply(is.infinite, returndcabhrobust2)] <- 0
returndcabhrobust2[mapply(is.na, returndcabhrobust2)] <- 0

Wide format
returndcabh_robust_wide2 <- cbind(returndcarobust2, returnbhrobust2)
returndcabh_robust_wide2[mapply(is.infinite, returndcabh_robust_wide2)] <- 0
returndcabh_robust_wide2[mapply(is.na, returndcabh_robust_wide2)] <- 0
returndcabh_robust_wide2 <- returndcabh_robust_wide2[,c(1,2,3,6,8,9,10,13,14)]

```



```
colnames(returndcabh_robust_wide2) <- c("returndca", "sddca", "sharpedca", "sortinodca",
 "returnbh", "sdbh", "sharpebh", "sortinobh", "period")
```

```
IRR ROBUST
```

```
for (k in p) {
```

```
 x = 0
```

```
 y = 0
```

```
 xx = 0
```

```
 yy = 0
```

```
 z = as.factor(0)
```

```
 # i tar hånd om alle tilgjengelige porteføljer i k (alle 12 mnd porteføljer når k = 12, alle 13 mnd
 porteføljer når k = 13, osv)
```

```
 for (i in 1:(407 - k + 1)) {
```

```
 name <- paste(k, "months", i)
```

```
 rep <-
```

```
 c(rep(-100, nrow(robust3[[name]])),
 robust2[[name]]$DCA_portfolio_end[nrow(robust3[[name]])])
```

```
 rep2 <-
```

```
 c(-100 * nrow(robust2[[name]]),
```

```
 rep(0, nrow(robust2[[name]]) - 1),
```

```
 robust3[[name]]$Buy_Hold_end[nrow(robust3[[name]])])
```

```
 x[i] <- irr(rep)
```

```
 y[i] <- irr(rep2)
```

```
 xx[i] <- (((1 + x[i]) ^ 12) - 1) # annualiserer IRR
```

```
 yy[i] <- (((1 + y[i]) ^ 12) - 1) # annualiserer IRR
```

```
per <- robust3[[name]]$period[1] # henter faktorvariabel
```

```
if (i == (407 - k + 1)) {
```

```
 if (k == 12) {
```

```
 irrdcarobust2 <-
```

```
 data.frame(
```

```
 irr = x,
```

```
 irr_a = xx,
```

```
 strategi = "DCA",
```

```
 period = per
```

```
)
```

```
 irrbhrobust2 <-
```

```
 data.frame(
```

```
 irr = y,
```

```
 irr_a = yy,
```

```
 strategi = "LS",
```

```
 period = per
```

```
)
```

```
 irrdfrobust2 <-
```

```
 rbind(irrdcarobust2, irrbhrobust2)
```

```
 } else if (k > 12) {
```

```
 irrdcarobust2 <-
```

```
 data.frame(
```

```
 irr = x,
```

```
 irr_a = xx,
```

```
 strategi = "DCA",
```

```
 period = per
```

```
)
```

```
 irrbhrobust2 <-
```

```
 data.frame(
```

```

 irr = y,
 irr_a = yy,
 strategi = "LS",
 period = per
)
 irrdfnewrobust2 <-
 rbind(irrdcarobust2, irrbhrobust2)

 irrdfrobust2 <- rbind(irrdfrobust2, irrdfnewrobust2)

}
}
}
}
Wide format
irrdcarobust2 <- irrdfrobust2[irrdfrobust2$strategi=="DCA",]
irrbhrobust2 <- irrdfrobust2[irrdfrobust2$strategi=="LS",]
irrdf_robust_wide2 <- cbind(irrdcarobust2,irrbhrobust2)
irrdf_robust_wide2 <- irrdf_robust_wide2[,c(1,2,5,6,8)]
colnames(irrdf_robust_wide2) <- c("irrdca", "irrdca_a", "irrbh", "irrbh_a",
 "period")
BEREGNER PROSENTVIS ANDEL AV SIMULERINGER DER DCA "VINNER"
returncomparerobust2 <- cbind(returndcarobust2, returnbhrobust2)
returncomparerobust2$lswinsreturn <- 0
returncomparerobust2$lswinssharpe <- 0
for (i in 1:nrow(returncomparerobust2)) {
 if (returncomparerobust2[i,1]<returncomparerobust2[i,8]) {
 returncomparerobust2[i,15] <- 1
 }
 if (returncomparerobust2[i,3]<returncomparerobust2[i,10]) {
 returncomparerobust2[i,16] <- 1
 }
}

```

```
}
}
l <- split(returncomparerobust2, returncomparerobust2$period)
x = 0
y = 0
for (i in 1:12) {
 x[i] <- 1 - (sum(l[[i]]$lswinsreturn)/nrow(l[[i]]))
 y[i] <- 1 - (sum(l[[i]]$lswinssharpe)/nrow(l[[i]]))
}
winsrobust2 <- data.frame(x,y)
colnames(winsrobust2) <- c("Avkastning", "Sharpe")
rownames(winsrobust2) <- p
stargazer(t(t(winsrobust2)), type = "text")
``
```