

finans_ex1

Silje Marie Anfindsen

9/6/2021

Oppgaver til første samling

oppgave 1)

Gjør dere kjent med dataene, se på tidsseriene, sammenlign volatilitet og andre karakteristika så som hvordan serienes variasjon varierer over tid... og prøv å si noe om det relative forholdet mellom dem - og hva det kan skyldes. For å gjøre det litt morsomt for dere selv - IKKE prøv å matche tidsseriene med tilgjengelige børsserier... da får dere ikke brynet dere på samme måten. Men knytt gjerne skifter i variasjonsmønster til kjente datoer/perioder..

Hvor lenge skal man se på volatilitet over?

historisk er utrolig lave forhold til forventet volatilitet på 15-20 prosent.

```
# read in the first worksheet from the workbook myexcel.xlsx
mydata <- read.xlsx("aksjedata.xlsx", 1)
```

```
manipulateSeries <- function(serie){

  date <- serie[,1]
  value <- serie[,2]

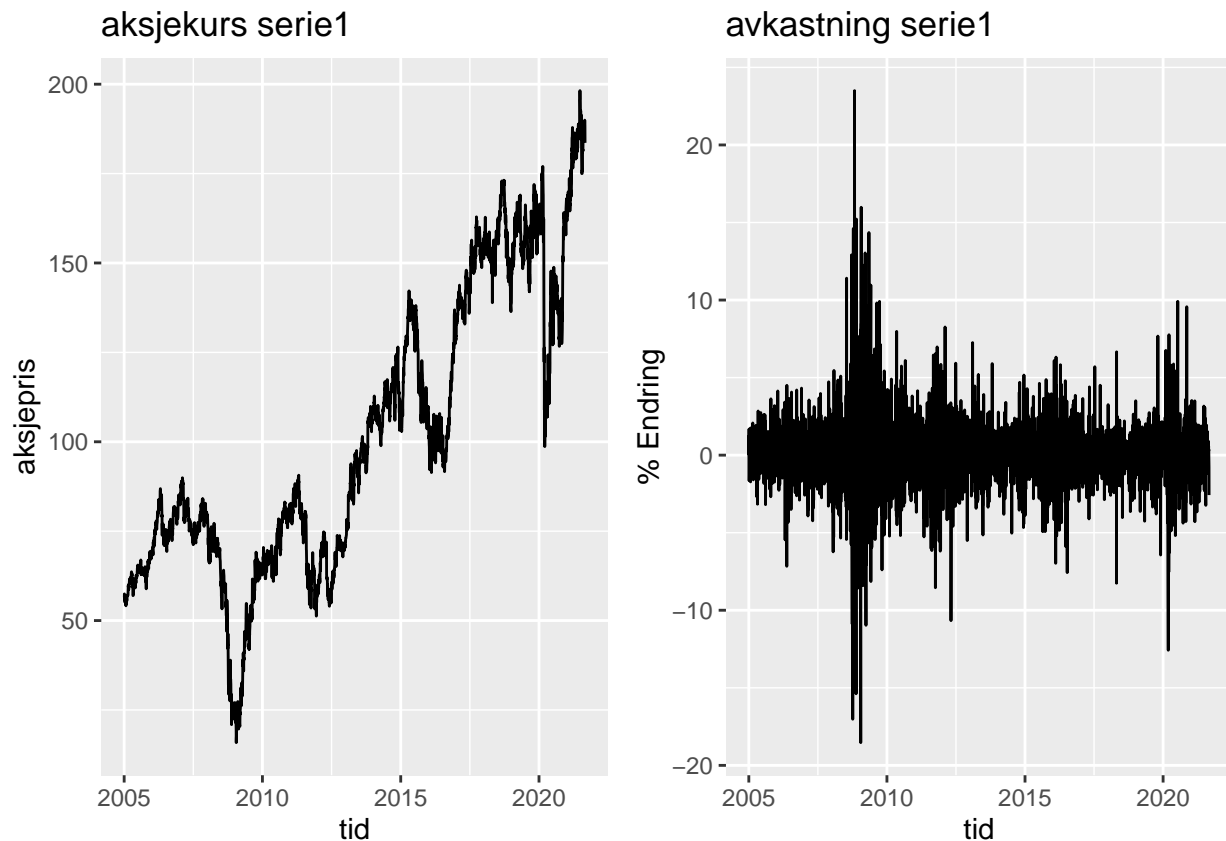
  if(is.na(tail(date,1))){
    end <- min(which(is.na(date)))-1
    df <- data.frame(date = date[1:end], value = value[1:end] )
  }
  else {
    df <- data.frame(date = date, value = value)
  }
  return(df)
}
```

```
serie1 <- manipulateSeries(mydata[1:2])
serie2 <- manipulateSeries(mydata[3:4])
serie3 <- manipulateSeries(mydata[5:6])
serie4 <- manipulateSeries(mydata[7:8])
serie5 <- manipulateSeries(mydata[9:10])
serie6 <- manipulateSeries(mydata[11:12])
serie7 <- manipulateSeries(mydata[13:14])
serie8 <- manipulateSeries(mydata[15:16])
volatilitets.index <- manipulateSeries(mydata[17:18])
```

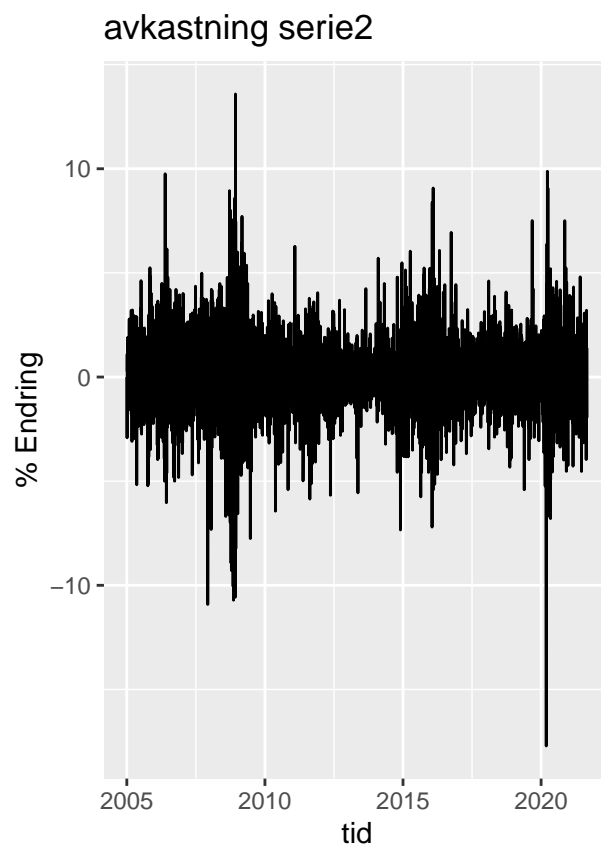
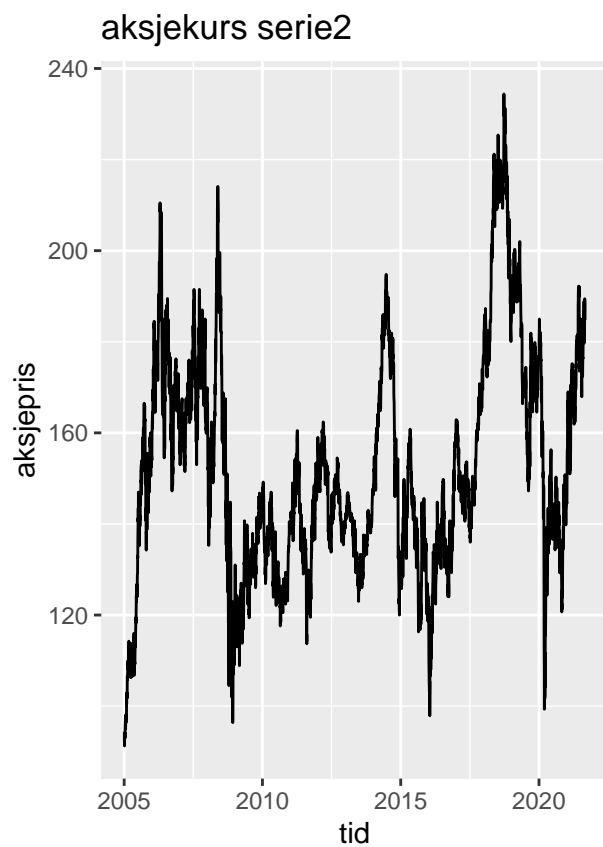
```
series <- list(serie1, serie2, serie3, serie4, serie5, serie6, serie7, serie8, volatilitets.index)
```

```
findReturn <- function(serie){
  prices <- serie[,2]
  returns = rep(0, length(prices))
  returns[2:length(prices)] = (prices[-1]-prices[-length(prices)])/prices[-length(prices)]*100
  return(returns)
}
returns <- lapply(series, findReturn)
```

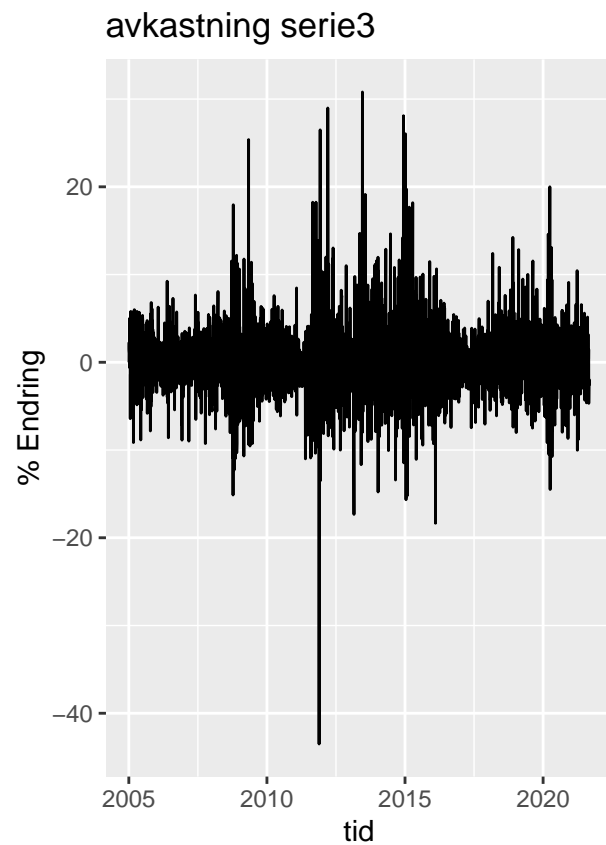
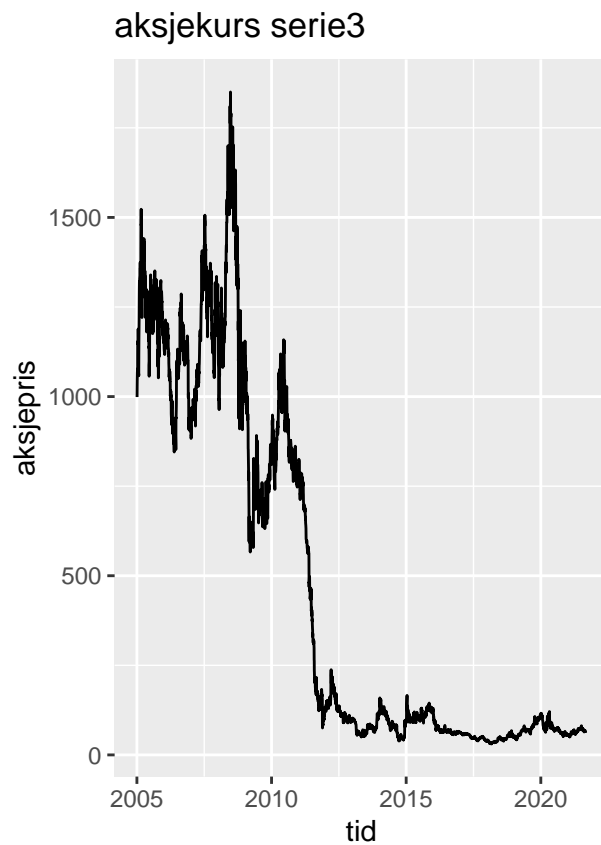
```
g11 <- ggplot(data = serie1, mapping = aes(x=date , y=value )) + geom_line() + ggtitle("aksjekurs serie1")
g12 <- ggplot(data = serie1, mapping = aes(x=date, y=returns[[1]])) + geom_line() + ggtitle("avkastning serie1")
ggarrange(g11,g12)
```



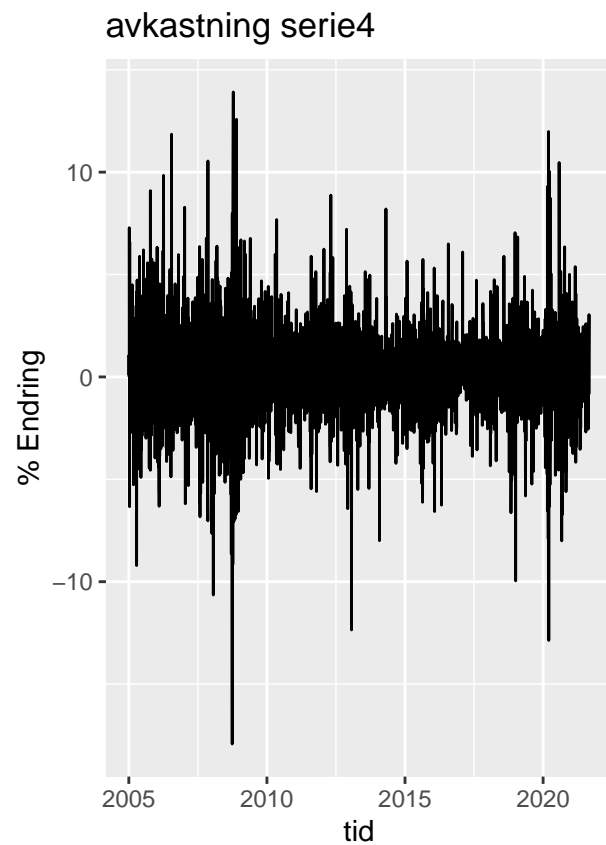
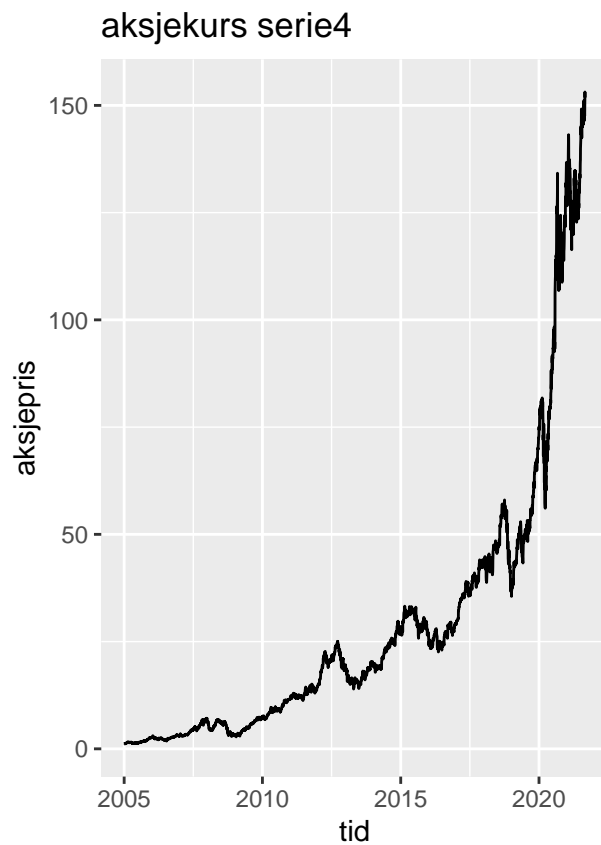
```
g21 <- ggplot(data = serie2, mapping = aes(x=date , y=value )) + geom_line() + ggtitle("aksjekurs serie2")
g22 <- ggplot(data = serie2, mapping = aes(x=date, y=returns[[2]])) + geom_line() + ggtitle("avkastning serie2")
ggarrange(g21,g22)
```



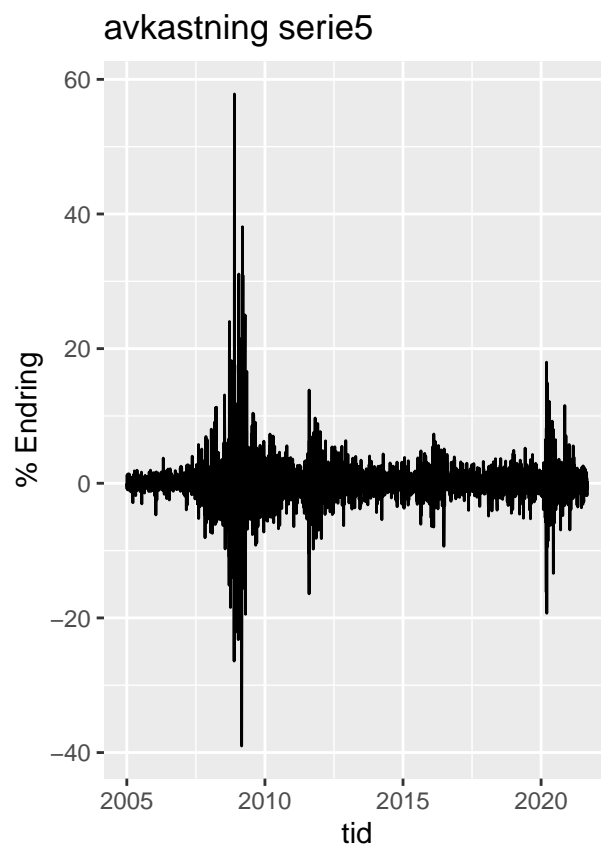
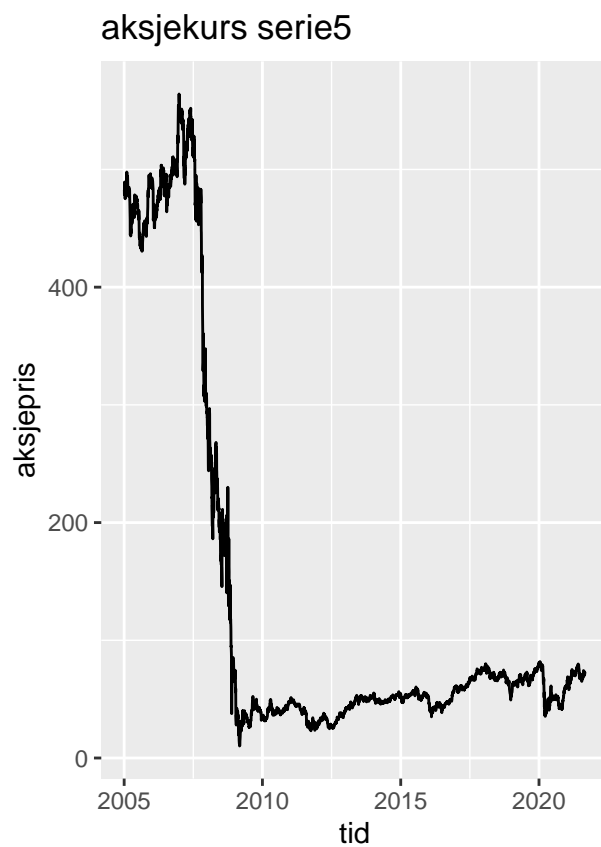
```
g31 <- ggplot(data = serie3, mapping = aes(x=date , y=value )) + geom_line() + ggtitle("aksjekurs serie2")
g32 <- ggplot(data = serie3, mapping = aes(x=date, y=returns[[3]])) + geom_line() + ggtitle("avkastning serie2")
ggarrange(g31,g32)
```



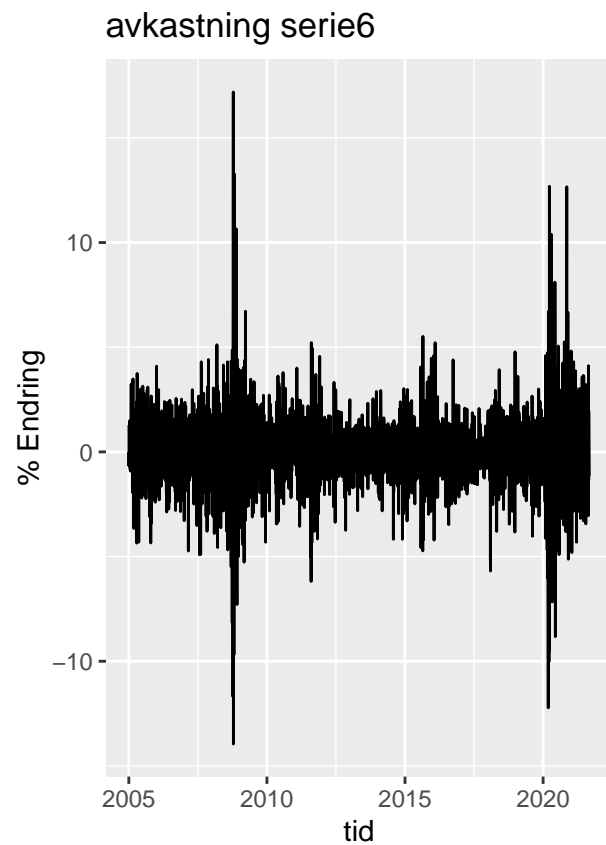
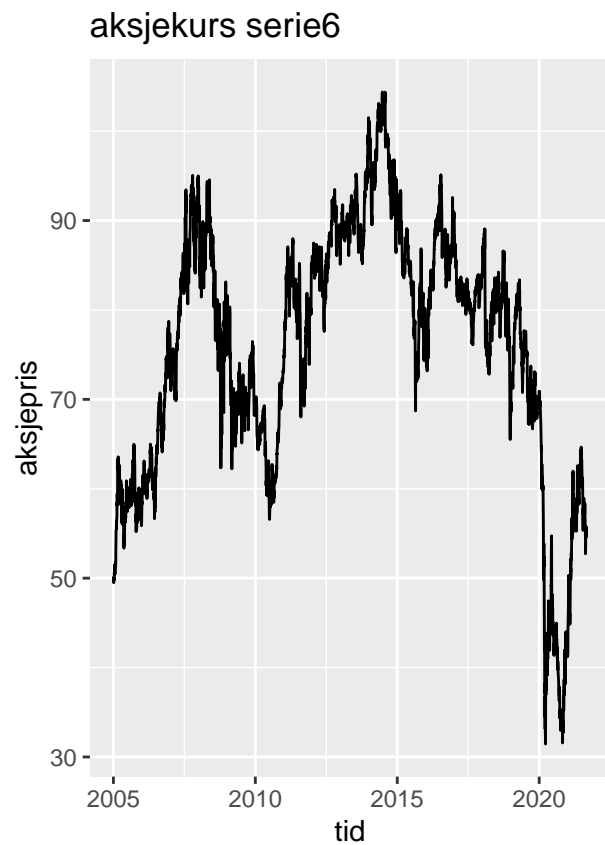
```
g41 <- ggplot(data = serie4, mapping = aes(x=date , y=value )) + geom_line() + ggtitle("aksjekurs serie3")
g42 <- ggplot(data = serie4, mapping = aes(x=date, y=returns[[4]])) + geom_line() + ggtitle("avkastning serie3")
ggarrange(g41,g42)
```



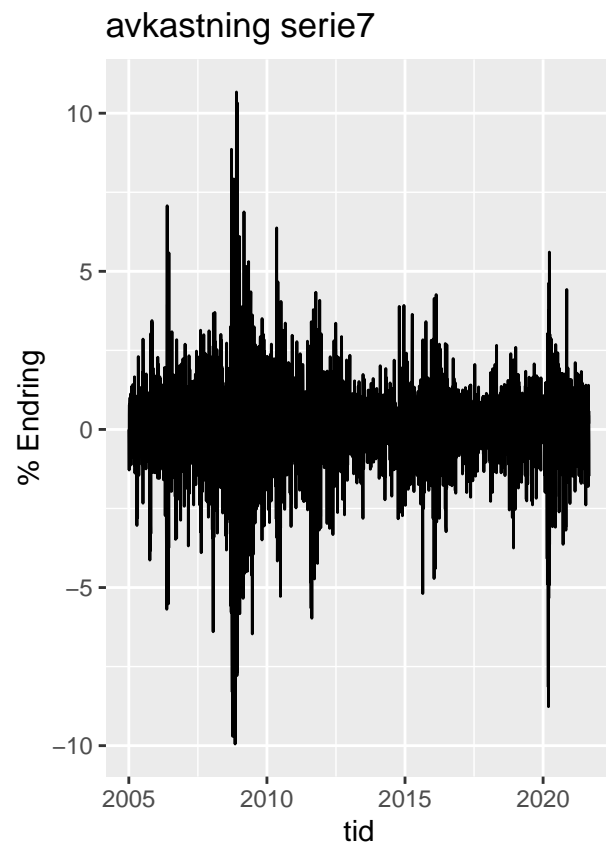
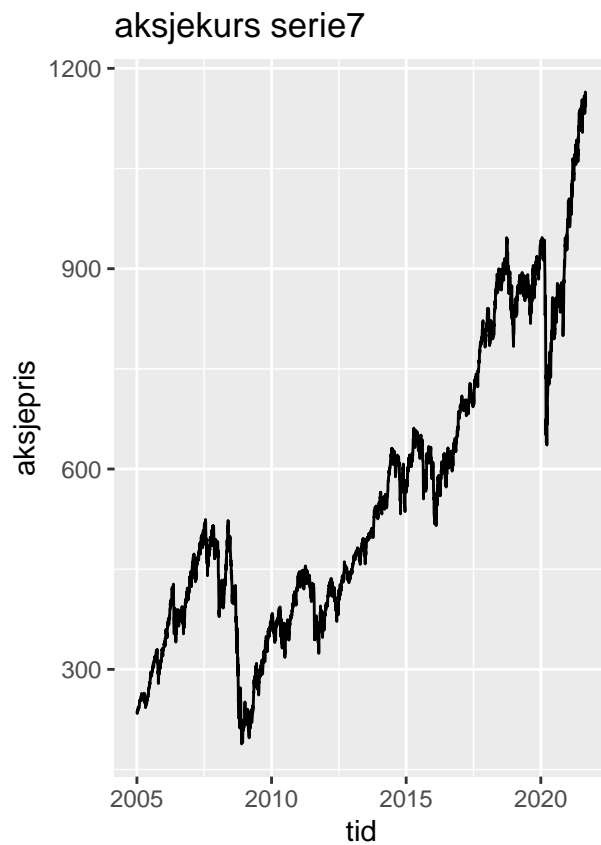
```
g51 <- ggplot(data = serie5, mapping = aes(x=date , y=value )) + geom_line() + ggtitle("aksjekurs serie4")
g52 <- ggplot(data = serie5, mapping = aes(x=date, y=returns[[5]])) + geom_line() + ggtitle("avkastning serie4")
ggarrange(g51,g52)
```



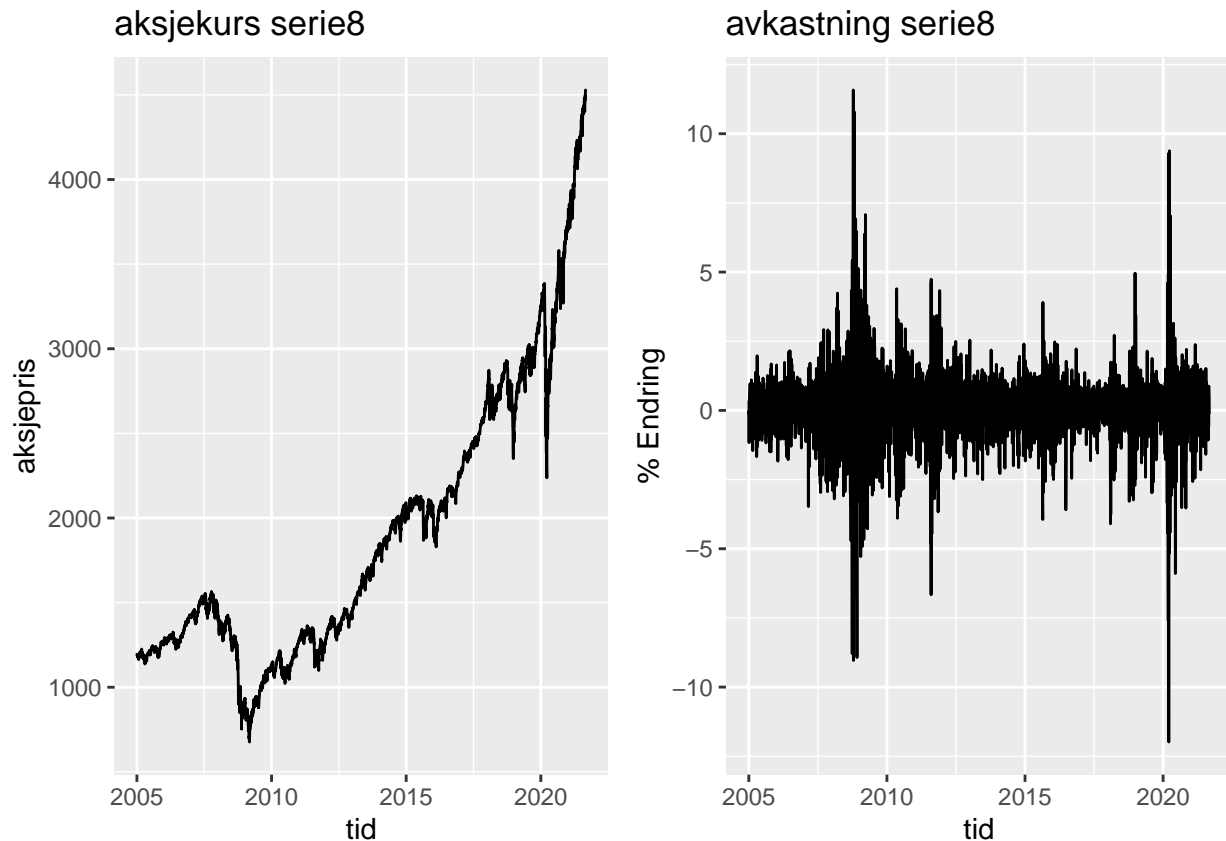
```
g61 <- ggplot(data = serie6, mapping = aes(x=date , y=value )) + geom_line() + ggtitle("aksjekurs serie5")
g62 <- ggplot(data = serie6, mapping = aes(x=date, y=returns[[6]])) + geom_line() + ggtitle("avkastning serie5")
ggarrange(g61,g62)
```



```
g71 <- ggplot(data = serie7, mapping = aes(x=date , y=value )) + geom_line() + ggtitle("aksjekurs serie6")
g72 <- ggplot(data = serie7, mapping = aes(x=date, y=returns[[7]] ) ) + geom_line() + ggtitle("avkastning serie6")
ggarrange(g71,g72)
```



```
g81 <- ggplot(data = serie8, mapping = aes(x=date , y=value )) + geom_line() + ggtitle("aksjekurs serie7")
g82 <- ggplot(data = serie8, mapping = aes(x=date, y=returns[[8]] ) ) + geom_line() + ggtitle("avkastning serie7")
ggarrange(g81,g82)
```

serie1:
serie2:
serie3:
serie4:
serie5:
serie6:
serie7:
serie8:

Knytt gjerne skifter i variasjonsmønster til kjente datoer/perioder:

2008-2009: finanskrisen

2011-2012: Statsgjeldskrisen som rammet Europa og USA?. Asia, Afrika, Latinamerika klarte seg godt.

2020-2021: korona

effekt av utbytte: kanskje den synker litt i verdi, brått en eller to ganger hvert år (systematisk fall)

oppgave 2

Velg ut to eller flere av de seks tidsseriene til venstre og velg en måte å implementere en random walk modell for disse med utgangspunkt i dataseriene. Kommenter hvordan modellen passer med tidsseriene ved å se på faktiske inkremerter vs. modellfordelingen, qq-plot el.l. Gjør det samme for en av de neste to tidsseriene mot høyre (serie 7 og 8 fra venstre).

```

#random walk
set.seed(123)

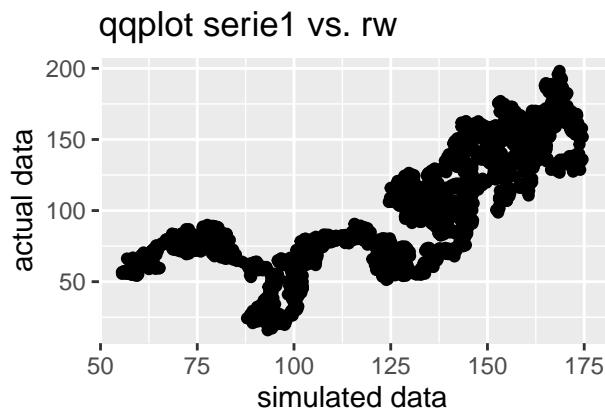
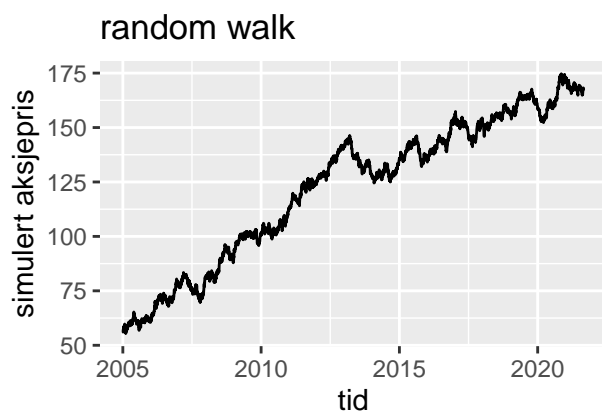
random.walk <- function(serie){
  days = length(serie[,1])
  mu = mean(serie[,2])/days
  sigma = sd(serie[,2])*1/sqrt(days)
  p0 = serie[1,2]

  r = rnorm(days, mu, sigma)
  price <- p0 + cumsum(r)

  return (price)
}

rw.price1 <- random.walk(serie1)
rw.series1 <- ggplot(data=serie1, aes(x=date,y=rw.price1)) + geom_line() + ggtitle("random walk") + labs
qq1 <- ggplot(mapping = aes(x = rw.price1, y = serie1$value)) + geom_point() + ggtitle("qqplot serie1 v
ggarrange(g11, rw.series1,qq1)

```



```

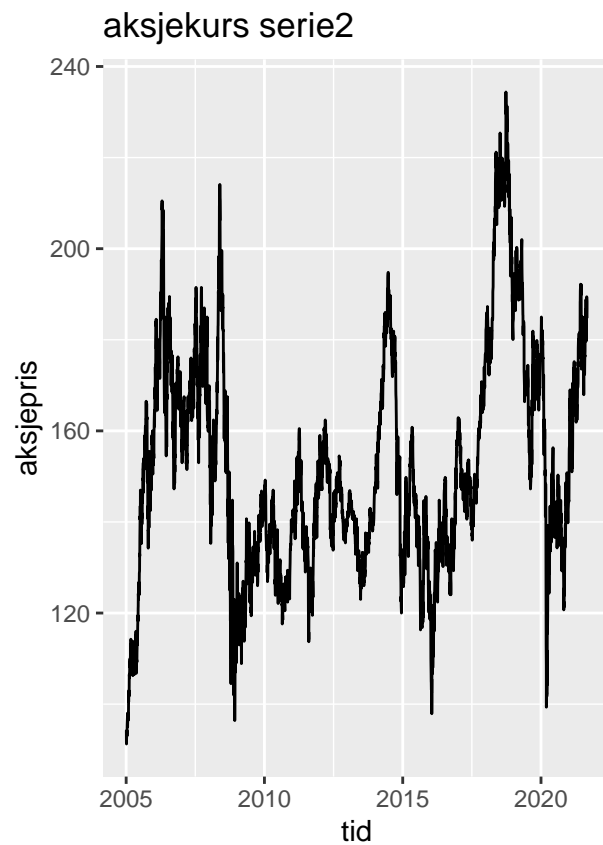
rw.series2 <- random.walk(serie2)
ggarrange(g21, rw.series2)

```

```

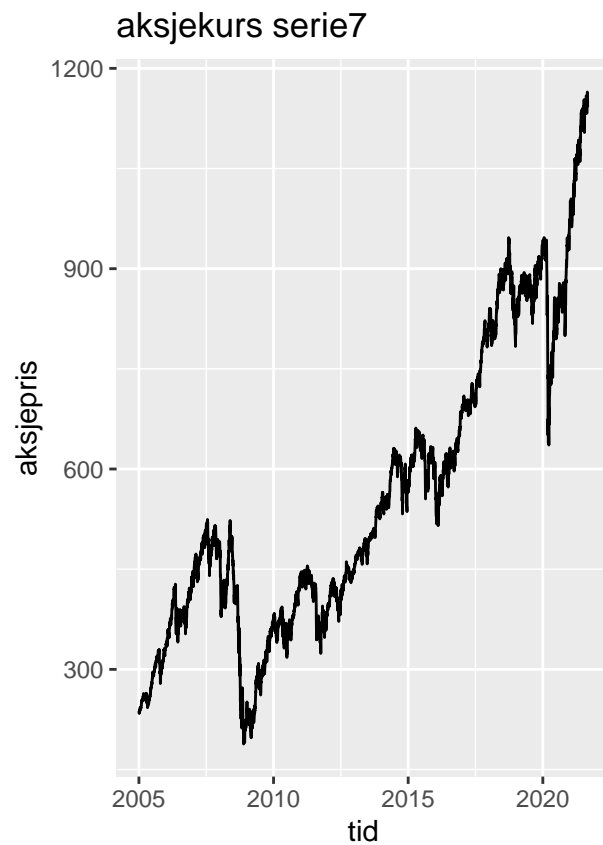
## Warning in as_grob.default(plot): Cannot convert object of class numeric into a
## grob.

```



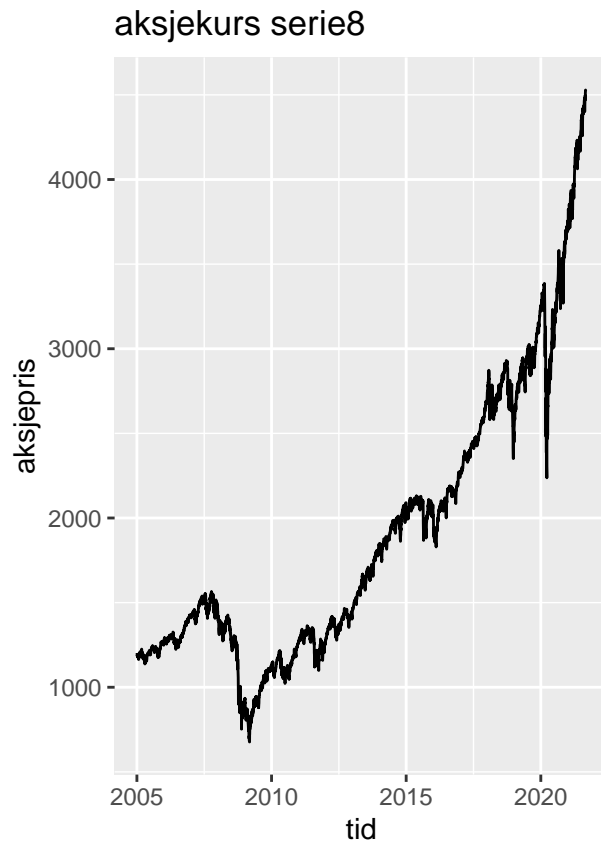
```
rw.serier7 <- random.walk(serie7)
ggarrange(g71, rw.serier7)
```

```
## Warning in as_grob.default(plot): Cannot convert object of class numeric into a
## grob.
```



```
rw.serie8 <- random.walk(serie8)
ggarrange(g81, rw.serie8)
```

```
## Warning in as_grob.default(plot): Cannot convert object of class numeric into a
## grob.
```



oppgave3)

Definer to enkle opsjoner på en eller flere av seriene og pris dem med både Black-Scholes formelen og ved å simulere fra en av modellene definert i 2) Velg en som er at-the-money og en som er et stykke out-of-the money. Verifiser at begge måter gir samme svar. Velg så et eller flere slutt punkt som ikke er enden av tidsserien (med nok tid igjen i data til at opsjonen kommer til forfall) og vis pris og fordeling ved forfall på et par tidspunkt i opsjonens levetid - samt hvor faktisk sluttverdi faller i disse.

Regn ut call-price:

$$C(S, t) = SN(d_1) - Ee^{-rT} - tN(d_2)$$

```
callPrice <- function(S, E, r, t, sigma){
  d_1 = (log(S/E) + (r+ 1/2*sigma^2 ) * t ) / (sigma*sqrt(t))
  d_2 = (log(S/E) + ( r- 1/2*sigma^2 ) * t ) / (sigma*sqrt(t))
  return (S*pnorm(d_1) - E*exp(-r*t)*pnorm(d_2))}
```

La oss se på serie 1. Start-dato er 5.februar 2019. Exercise price er 153,55. Forfallsdato er 5.august 2019 og da er faktisk aksjepris lik 151,95.

definere opsjoner for serie 1 “at the money” at the money = prisen for opsjonen skal være den samme som nåværende verdi / markedspris.

```
t = 1/2
S = E = 153.55
r = 0.05
sigma = sd(returns[[1]])/100

print("at the money:")
```

```
## [1] "at the money:"  
callPrice(S, E, r, t, sigma)
```

```
## [1] 3.852103
```

definere opsjoner for serie 1 “out of the money” out of the money = strike price/exercise price er høyere enn markedspris.

```
t = 1/2  
S = 153.55  
E = 160  
r = 0.05  
sigma = sd(returns[[1]])/100  
  
print("out of the money:")
```

```
## [1] "out of the money:"  
callPrice(S, E, r, t, sigma)
```

```
## [1] 0.2014362
```

Vil nå simulere fra random walk modellen i oppgave 2 og sjekke hva prisen er den 5.august 2019.

```
sim <- rw.price1[which(serie1$date == "2019-08-05")]  
ex.price.at <- 153.55  
ex.price.out <- 160  
  
print("simulert aksjepris ved forfall er:")
```

```
## [1] "simulert aksjepris ved forfall er:"  
sim
```

```
## [1] 163.0682  
print("at the money:")
```

```
## [1] "at the money:"  
sim - ex.price.at
```

```
## [1] 9.518225  
print("out of the money:")
```

```
## [1] "out of the money:"  
sim - ex.price.out
```

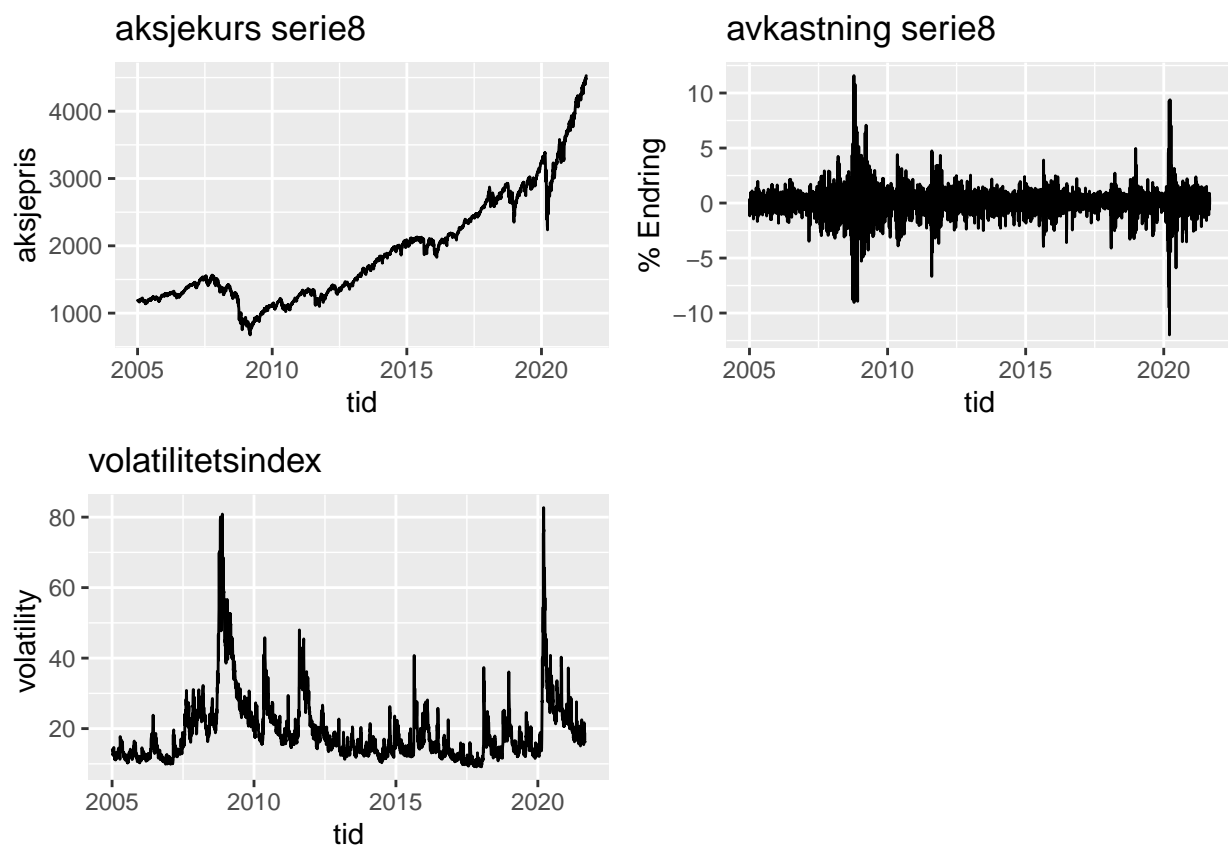
```
## [1] 3.068225
```

Call price antas å være differansen mellom simulert verdi og exercise price. For “at the money” vil man da sette den lik 2.7 og for “out of the money” lik 0?

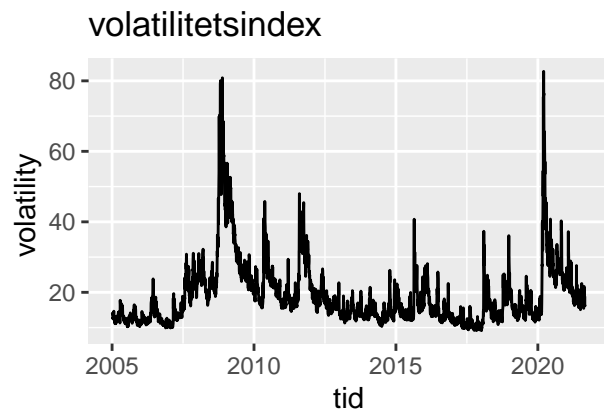
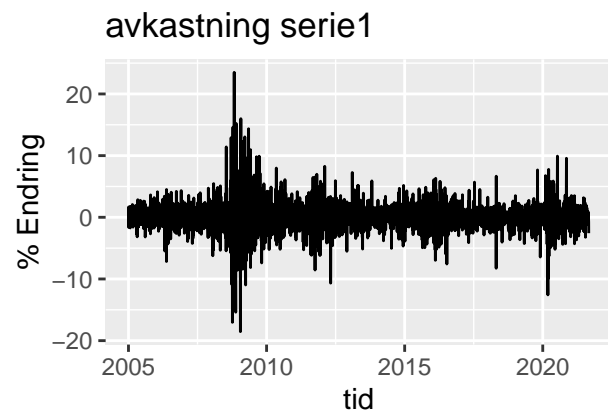
Oppgave 4

Ta til sist for deg volatilitetsindeksen til høyre og sammenhold den mot de seriene du valgte under 2. Kommenter på det du måtte finne interessant. Volatiliteten hører til serien til venstre for den, så til slutt kan du sammenligne nivået på den faktiske volatiliteten på serien og selve indeksverdien over tid. Er det noen systematiske forskjeller?

```
vol <- ggplot(data = volatilitets.index, mapping = aes(x=date , y=value )) + geom_line() + ggtitle("vol")
ggarrange(g81,g82,vol)
```



```
ggarrange(g11, g12,vol)
```



```
ggarrange(g21, g22, vol)
```