Facit Datorlabb 5, SDA2

Mona Sfaxi

Set-up

Storlek på figurer

```
knitr::opts_chunk$set(fig.width = 5, fig.height = 3)
```

Ladda paket:

```
library(forecast)
library(fpp3)
library(cowplot) # för att ha plottar bredvid varandra
```

Del 1 - Visualisering av tidsserie

Vi börjar med att läsa in datasetet och välja ut staten *Victoria*. Tänk på att datasetet redan är definierad som en `tstibble` så definiera den *inte* som tibble när du läser in datasetet, då kommer funktionen autoplot att driva med dig.

Varje stat som finns med i datasetet består alltså av 74 observationer kvartalsdata mellan åren 1998-2016.

```
# A tsibble: 592 x 5 [1Q]
# Key:       State [8]
      Date State                      Takings Occupancy   CPI
     <qtr> <chr>                        <dbl>     <dbl> <dbl>
 1 1998 Q1 Australian Capital Territory    24.3      65    67
 2 1998 Q2 Australian Capital Territory    22.3      59    67.4
 3 1998 Q3 Australian Capital Territory    22.5      58    67.5
 4 1998 Q4 Australian Capital Territory    24.4      59    67.8
```

```
 5 1999 Q1 Australian Capital Territory    23.7      58    67.8
 6 1999 Q2 Australian Capital Territory    25.4      61    68.1
 7 1999 Q3 Australian Capital Territory    28.2      66    68.7
 8 1999 Q4 Australian Capital Territory    25.8      60    69.1
 9 2000 Q1 Australian Capital Territory    27.3      60.9  69.7
10 2000 Q2 Australian Capital Territory    30.1      64.7  70.2
# ... with 582 more rows
```

Vi väljer sedan ut staten Victoria och får

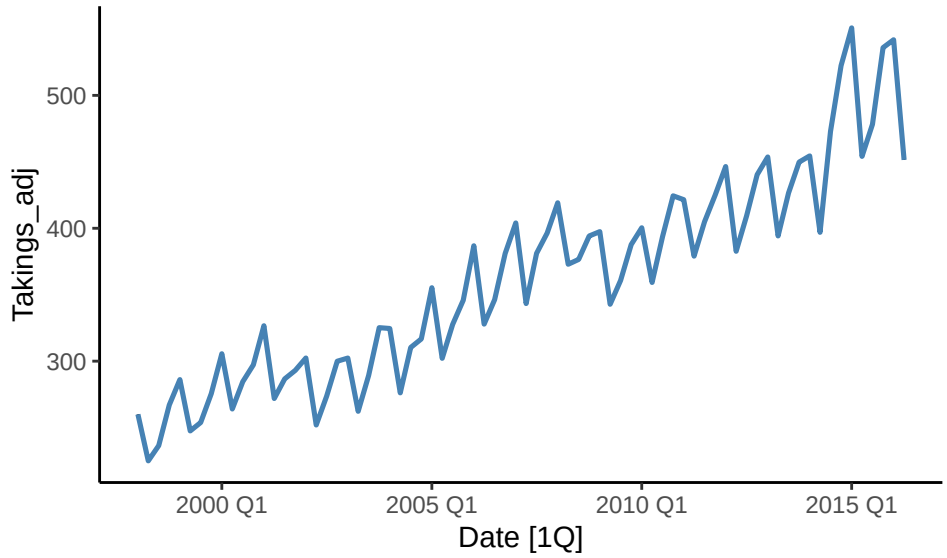
```
# A tsibble: 74 x 5 [1Q]
# Key:       State [1]
      Date State    Takings Occupancy   CPI
     <qtr> <chr>      <dbl>     <dbl> <dbl>
 1 1998 Q1 Victoria    174.      62    67
 2 1998 Q2 Victoria    152.      55    67.4
 3 1998 Q3 Victoria    160.      54    67.5
 4 1998 Q4 Victoria    181.      60    67.8
 5 1999 Q1 Victoria    194.      64    67.8
 6 1999 Q2 Victoria    169.      57    68.1
 7 1999 Q3 Victoria    174.      56    68.7
 8 1999 Q4 Victoria    190.      60    69.1
 9 2000 Q1 Victoria    213.      62.3  69.7
10 2000 Q2 Victoria    185.      57.1  70.2
# ... with 64 more rows
```

Därefter kan vi transformera variabeln med hjälp av funktionen `mutate()`

```
# A tsibble: 74 x 6 [1Q]
# Key:       State [1]
      Date State    Takings Occupancy   CPI Takings_adj
     <qtr> <chr>      <dbl>     <dbl> <dbl>       <dbl>
 1 1998 Q1 Victoria    174.      62    67          260.
 2 1998 Q2 Victoria    152.      55    67.4        225.
 3 1998 Q3 Victoria    160.      54    67.5        236.
 4 1998 Q4 Victoria    181.      60    67.8        267.
 5 1999 Q1 Victoria    194.      64    67.8        286.
 6 1999 Q2 Victoria    169.      57    68.1        248.
 7 1999 Q3 Victoria    174.      56    68.7        254.
 8 1999 Q4 Victoria    190.      60    69.1        276.
 9 2000 Q1 Victoria    213.      62.3  69.7        306.
```

```
10 2000 Q2 Victoria    185.      57.1  70.2        264.
# ... with 64 more rows
```

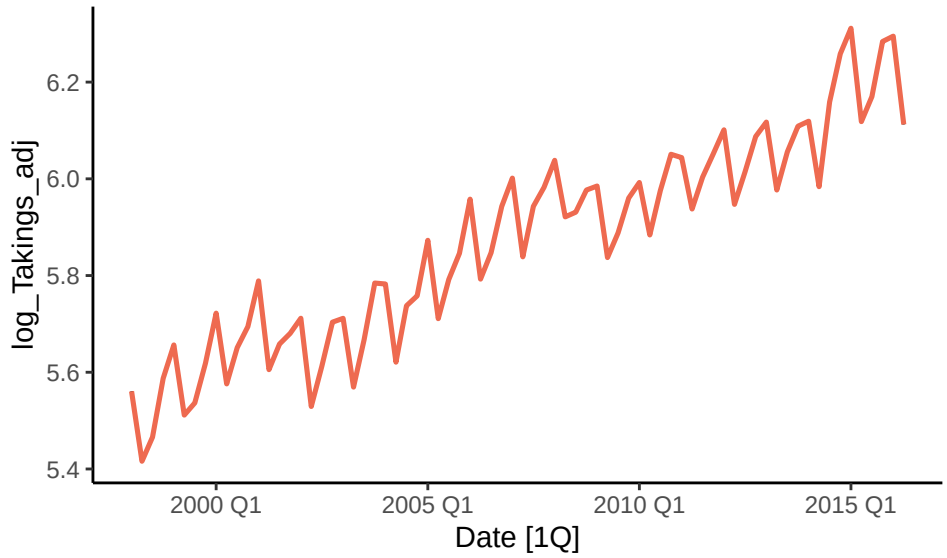
Låt oss plotta tidsserien:



Figur 1: Justerad-Takings (Logaritmerad)

Det är en tydlig trend med ett tydligt säsongsmönster. Vi ser exempelvis att Q1 kännetecknas av höga värden följt av skarpa nedgångar vid Q2. Det ser också ut som om vi har en additiv modell förutom i slutet på tidsserien ungefär vid 2014 och framåt där man kan ana att det börjar likna en multiplikativ modell då säsongsmönstret blir mer kraftigt med tiden. Men detta kan också bero på slumpen.

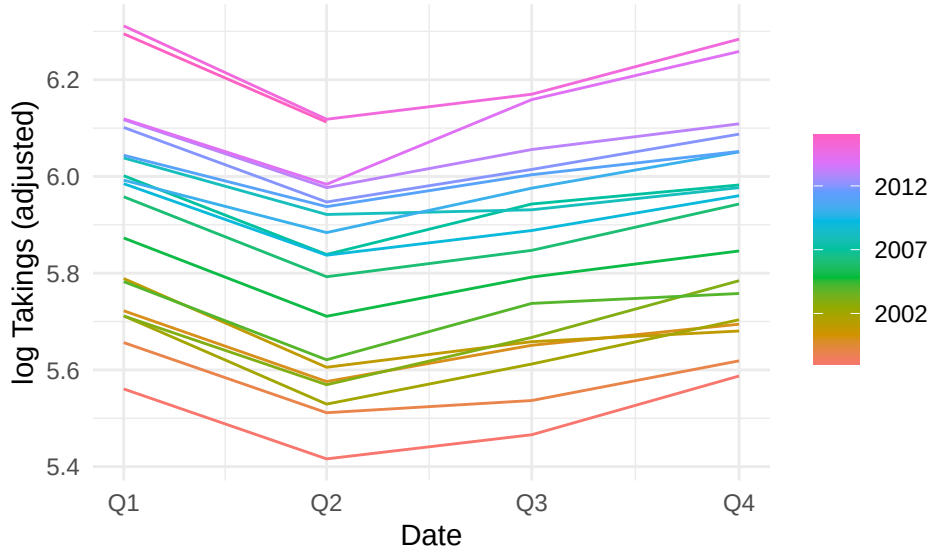
Vi logaritmerar `Takings_adj` och plottar den:



Figur 2: Justerad-Takings (Logaritmerad)

Inte mycket förändring syns i tidserien förutom kanske i slutet då svängningarna i säsongerna ser ut att ha blivit lite mindre och liknar åren dessförinnan.

Låt oss också skapa en säsongsplot för vår tidsserie.



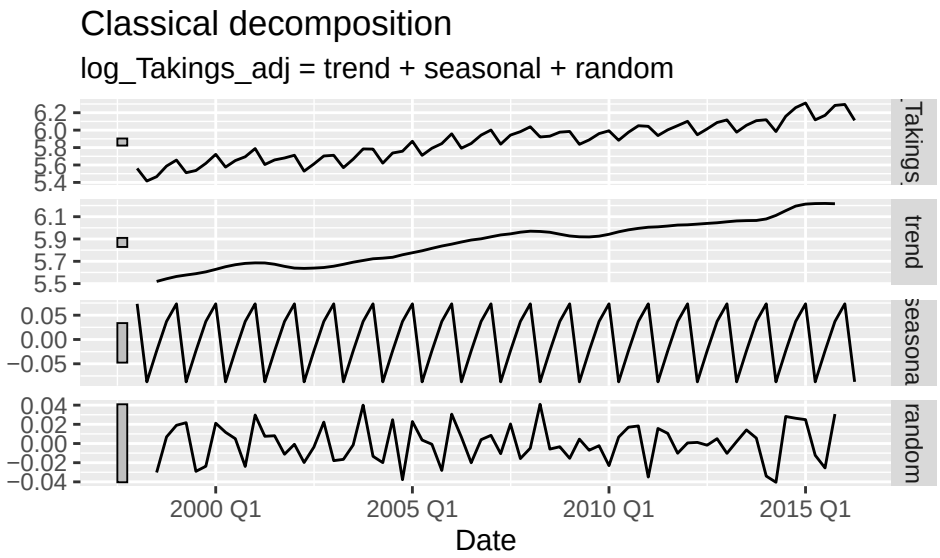
Figur 3: Säsongsplot över log_Takings

Vi ser att det finns ett tydligt säsongsmönster då de flesta linjerna är parallella med varandra. Det går även att se att kvartal 2 är det kvartal med de lägsta (justerade logaritmerade) intäkterna.

Del 2 - Klassisk dekomponering

Låt oss göra en dekomponering, dvs separera de olika beståndsdelarna. Observera att något av paketen fpp3 eller forecast verkar ha en konflikt med mosaic paketet. Det kan vara så att man inte kommer kunna använda sig av *model()* funktionen ifall man har mosaicpaketet igång samtidigt.

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

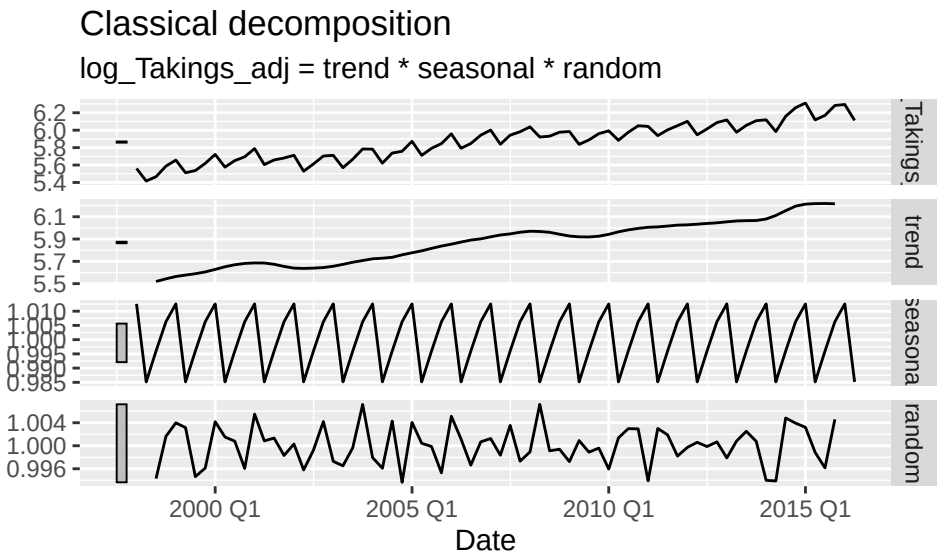


Figur 4: Klassisk dekomponering med additiv modell

De gråa lådorna visar på de relativa storlekarna av varje komponent. Ju mindre den gråa lådan är i relation till de andra lådorna, desto större dominans har den komponenten på variationen i vår tidsserie. Vi ser exempelvis att den gråa lådan för trenden är väldigt liten och nästan i samma storlek som originaldata. Det innebär att det här datasetet kännetecknas av en väldgt stark trend. Därefter kommer säsongskomponenten vars låda är lite större, så säsongsmönstret är inte lika framträdande som trenden här. Slutligen har vi den största gråa lådan hos random, som visar på att slumptermen står för en väldigt liten del av variationen av tidsserien.

Vi gör på samma sätt som ovan fast använder nu en multiplikativ modell:

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

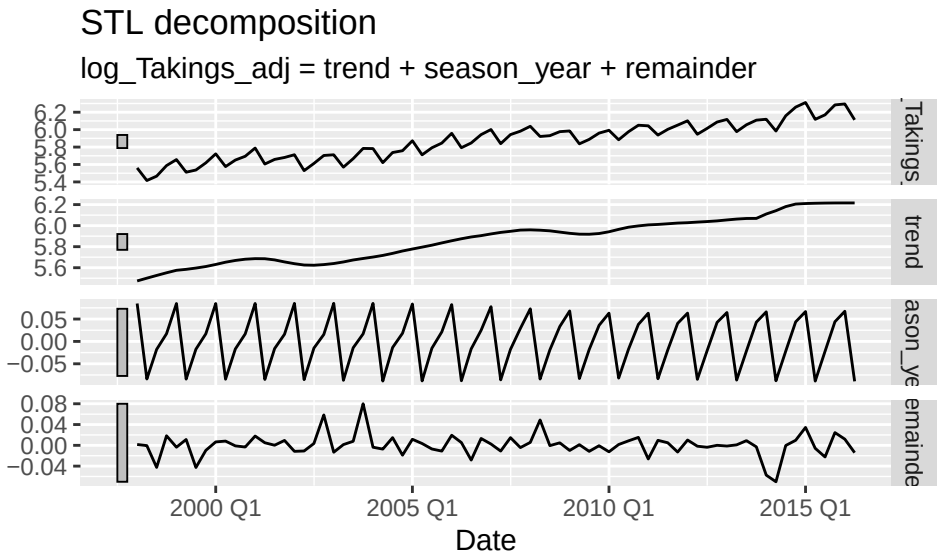


Figur 5: Klassisk dekomponering med multiplikativ modell

Modellen ser nästan exakt likadan ut så det är väldigt svårt att säga varför den multiplikativa skulle vara så dålig utifrån plottarna, allmänt kanske man kan tycka att det är konstigt att använda en multiplikatv modell på en tidsserie som redan dessförinnan blivit logaritmerad.

2b

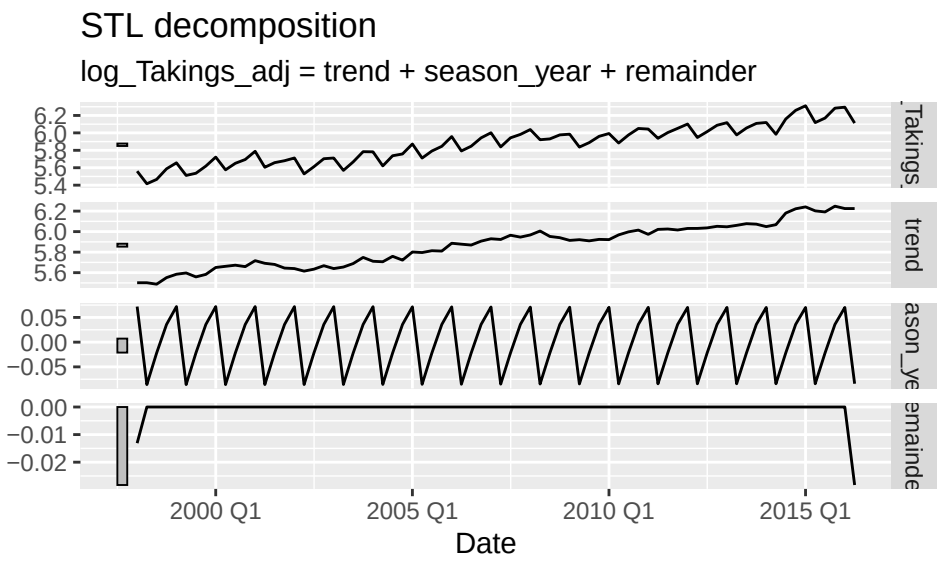
Vi gör en STL-dekomponering och plottar resultatet i Figur:



Figur 6: STL- dekomponering av log_Takings_adj

Vi testar därefter vad som händer ifall vi skriver trend(window = 1) och trend(window = 50). Först noterar vi återigen att vi endast har 74 observationer. När vi sätter trend(window = 1) så innebär det att vi endast använder 1 observation för att skatta trenden för varje tidpunkt, som resultat blir trendskattningen väldigt hackig, istället för att få en mer utjämnad (smooth) linje. Trendskattningen följer nog den faktiska observerade tidsserien lite för noga och slumptermen blir så gott som 0.

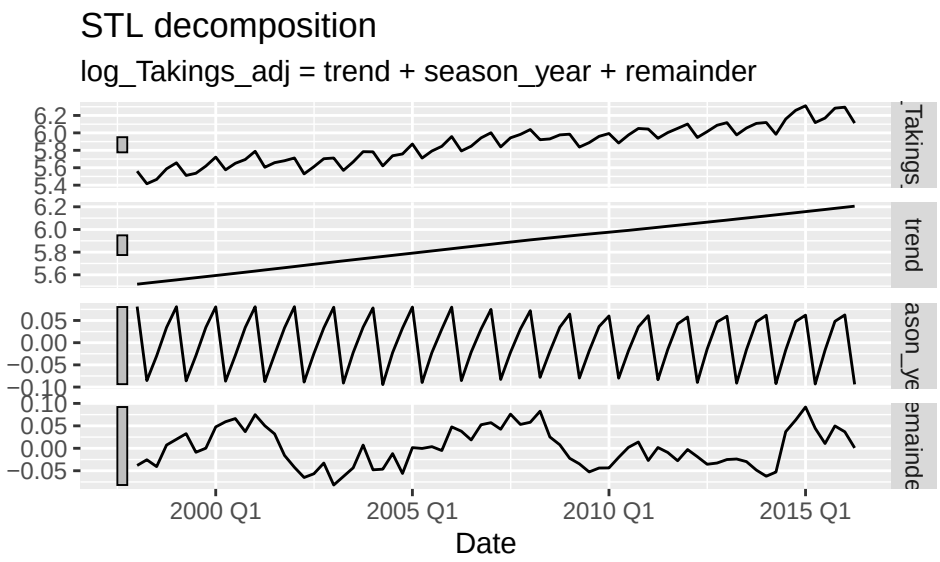
```
df %>%
  model(STL(log_Takings_adj ~ trend(window = 1) + season(), robust = TRUE)) %>%
    components() %>%
    autoplot()
```



Figur 7: STL- dekomponering, med window = 1 för trenden.

När vi skriver trend(window = 50) som visas i Figur 8, så innebär det att vi använder 50 observationer för att skatta trenden för en tidpunkt, vilket är varför linjen har blivit så rak. Här gör alltså funktionen tvärtemot från det vi gjorde i Figur 7. Den skattade trenden blir alltså alltför utjämnad för det här datasetet (som endast består av kvartalsdata med 74 observationer).

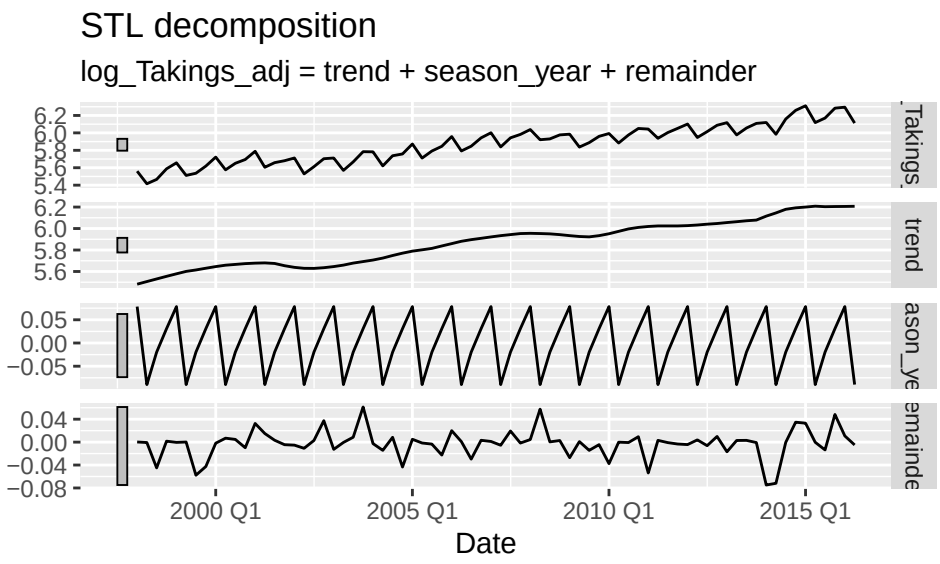
```
df %>%
  model(STL(log_Takings_adj ~ trend(window = 50) + season(), robust = TRUE)) %>%
    components() %>%
    autoplot()
```



Figur 8: STL- dekomponering, med window = 50 för trenden.

Skriver vi season(window = "periodic") får vi istället i Figur 9 plottar som liknar de i Figur 6 väldigt mycket (dvs där R själv har valt hur många observationer som ska användas vid säsongsskattningarna). Genom att använda season(window = "periodic") tvingar vi modellen att skatta säsongseffekten med hjälp av samtliga observationer. Vi jämför plotten med den första i 2b (Figur 6) och ser att det verkar inte göra så mycket skillnad för just det här datasetet genom att använda window = "periodic".

```
df %>%
  model(STL(log_Takings_adj ~ trend() + season(window = "periodic"),
            robust = TRUE)) %>%
  components() %>%
  autoplot()
```



Figur 9: STL- dekomponering, med season = "periodic"

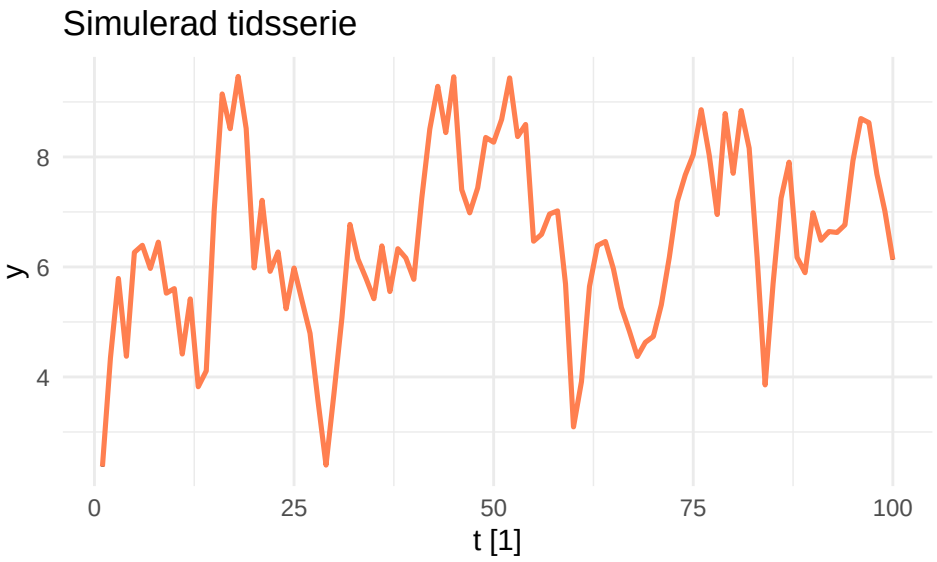
Del 3 - Autoregressiva processer

Skapa en funktion som simulerar från en Ar(1) modell.

```
sim_AR1 <- function(n, beta0, beta1) {
  y <- rep(NA, n)
  y[1] <- beta0 + beta1 * 0 + rnorm(1)
  for (i in 2:n) {
    y[i] <- beta0 + beta1 * y[i - 1] + rnorm(1)
  }
  dfy <- tsibble(
    y = y,
    t = 1:n,
    index = t
  )
  return(dfy)
}
```

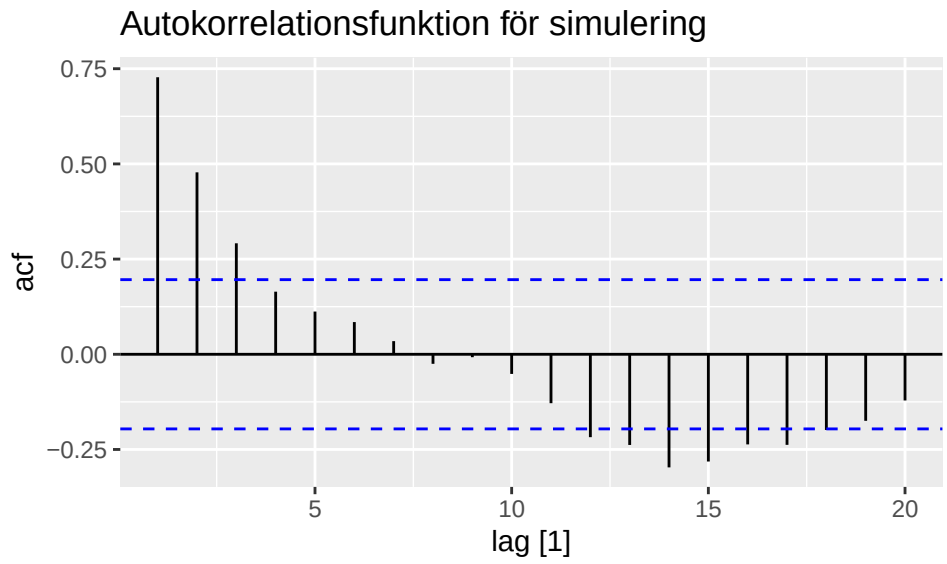
Vi testar resultatet från vår simulerade serie:

```r
sim_tidsserie <- sim_AR1(n = 100, beta0 = 2, beta1 = 0.7)
sim_tidsserie %>%
  autoplot(y) +
  labs(title = "Simulerad tidsserie") +
  theme_minimal() +
  geom_line(color = "coral", linewidth = 0.9)
```



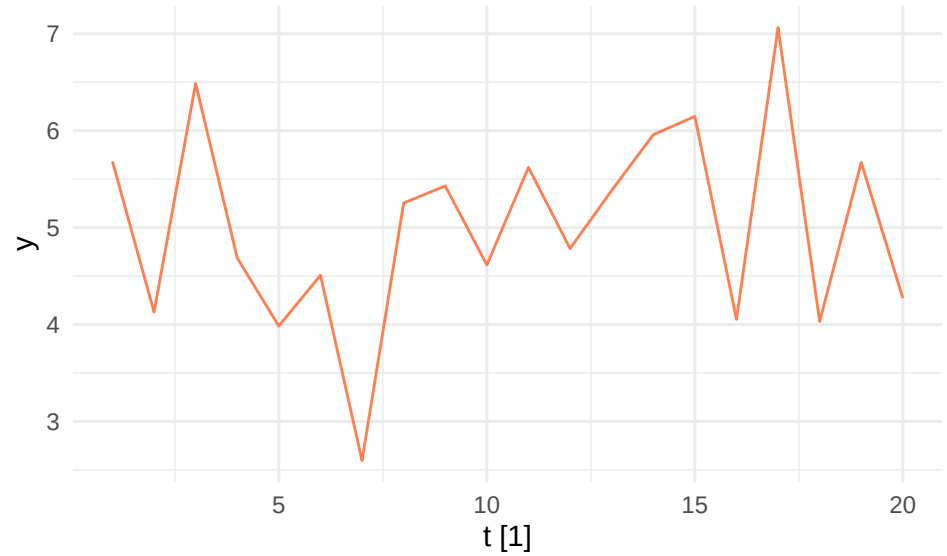
Skapa en ACF-plot:

```r
sim_tidsserie %>%
  ACF(y, lag_max = 20) %>%
  autoplot() +
  labs(title = "Autokorrelationsfunktion för simulering")
```



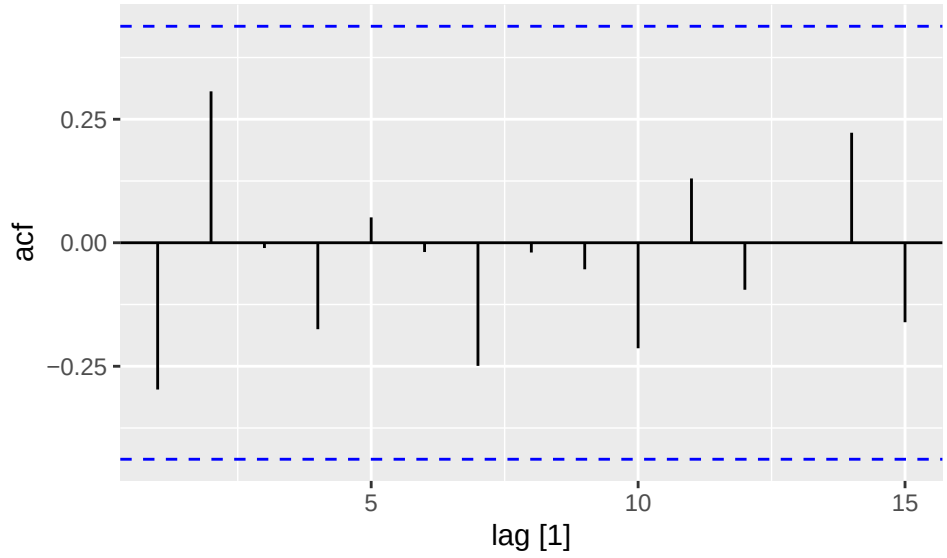
3a - Vitt brus

Vi ska simulera fram en tidsserie med vitt brus med n = 20 följt av n = 1000. Jag sätter seed till 621.



Figur 10: Tiddsserie som kännetecknas av Vitt brus

Tidsserien ser inte ut att ha någon trend för just denna simulering. Vi tittar på ACF-plotten



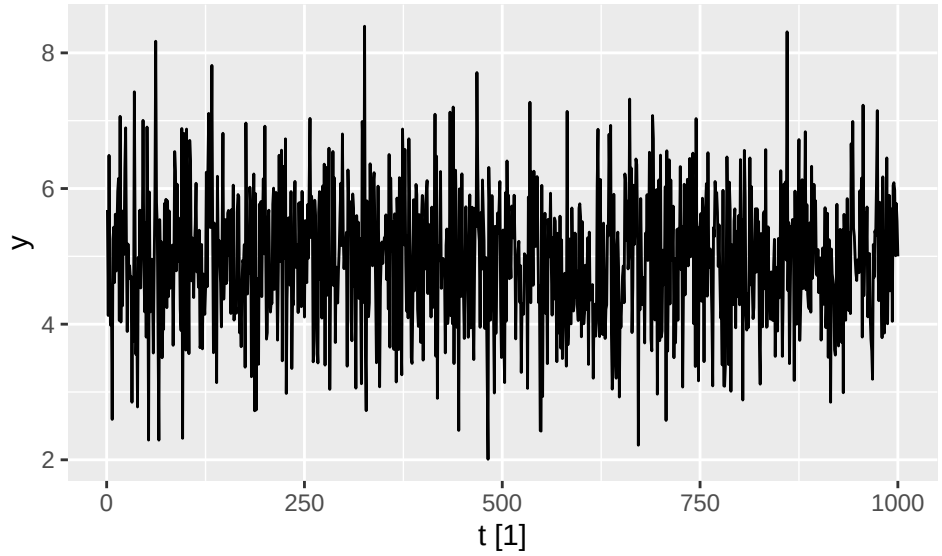
Figur 11: ACF för simulerad AR(1) med c = 5 och phi = 0

Vi ser att staplarna växlar mellan positiva och negativa värden men ingen av staplarna är signifikant. Vi verkar alltså inte ha någon korrelation alls mellan y och dess olika laggar, vilket kanske inte är så konstigt eftersom för en AR(1) ges den teoretiska korrelationen av:

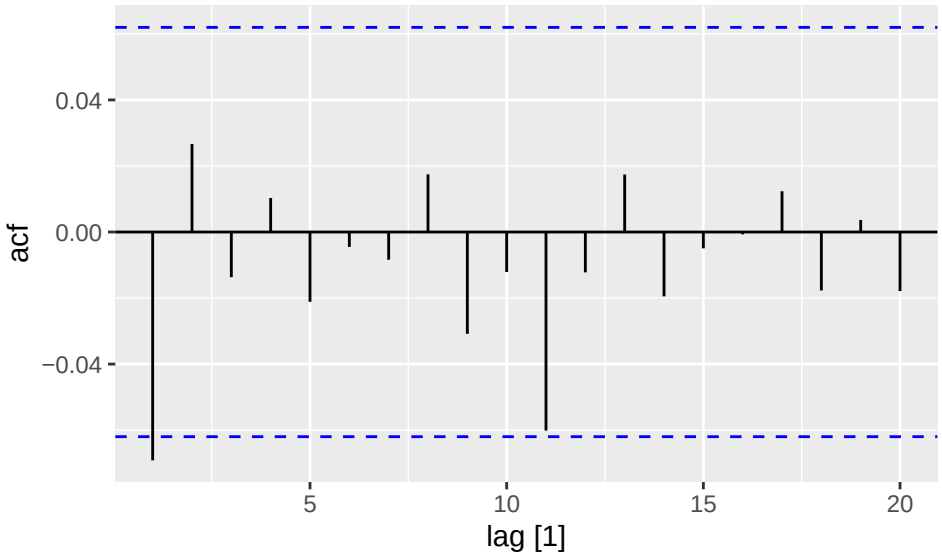
$$\rho_k = \phi_1^k$$

Så om $\phi_1 = 0$ så bör vi alltså inte ha någon korrelation alls.

Vi fortsätter med n = 1000 istället:



Figur 12: AR(1) med c = 5 och phi = 0

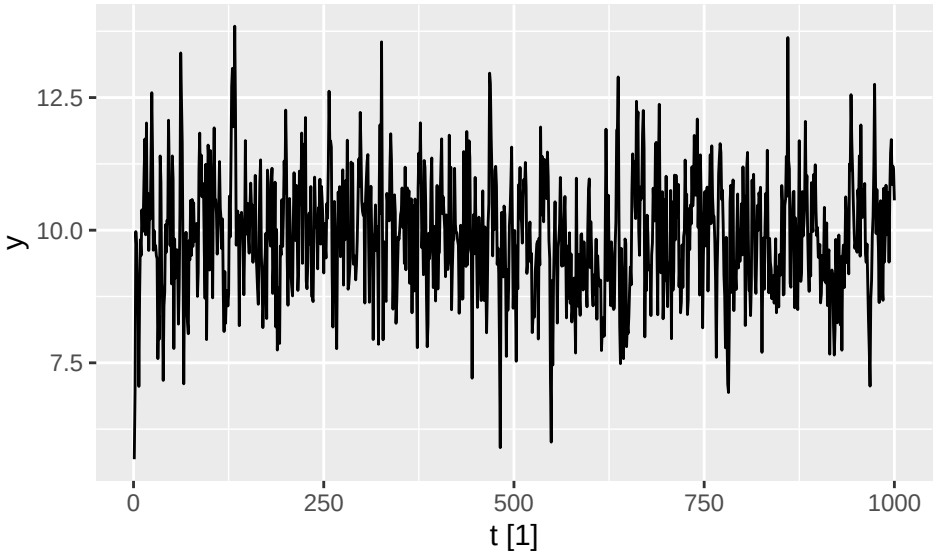


Figur 13: ACF för simulerad AR(1) med c = 5 och phi = 0

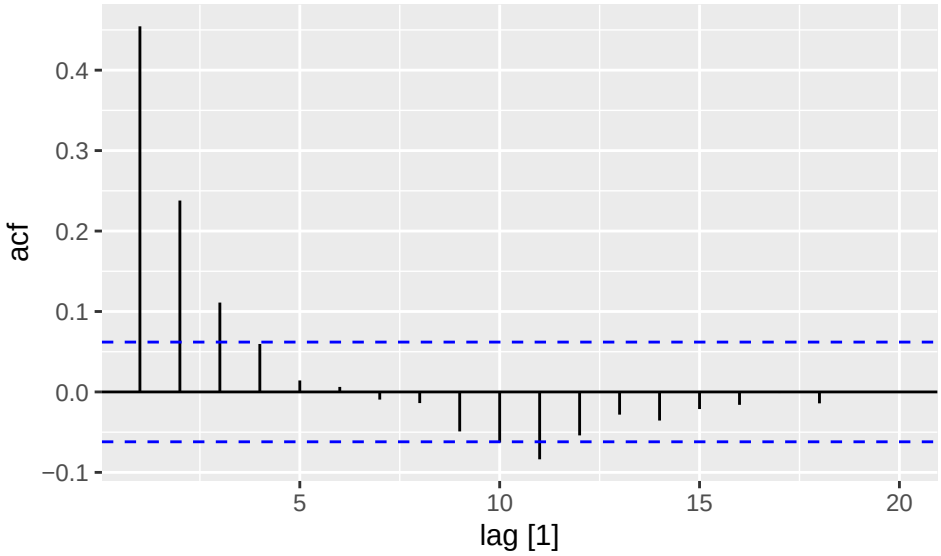
Denna gång ser det verkligen ut som om vi har vitt brus i den första tidsserieplotten. I ACF plotten ser vi att vi har ett värde som är signifikant, men den skattade korrelationen är ändå ganska låg, kanske runt 0.065.

3b - Stationära AR(1)-processer

Vi simulerar återigen, denna gång med $c = 0.5$ följt av $\phi_1 = -0.5$

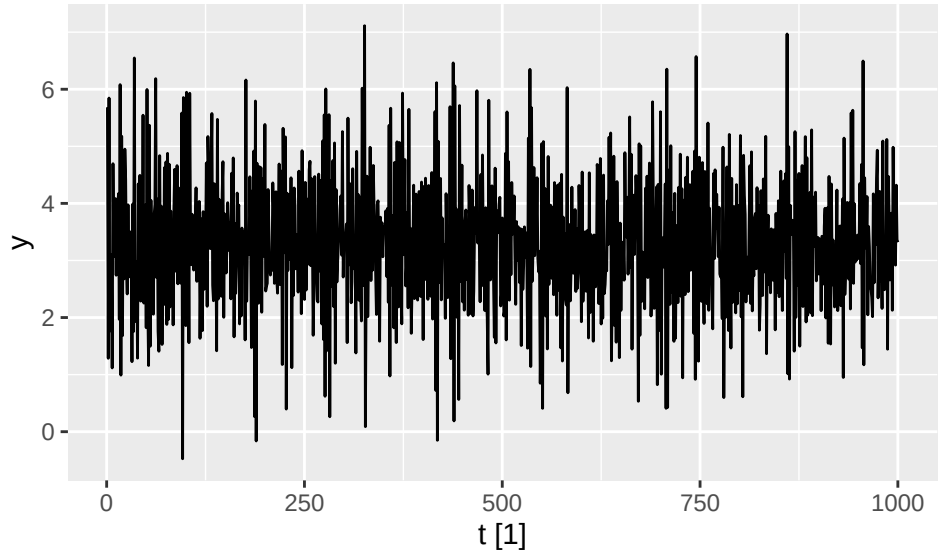


Figur 14: Simulerad AR(1) med c = 5 och phi = 0.5

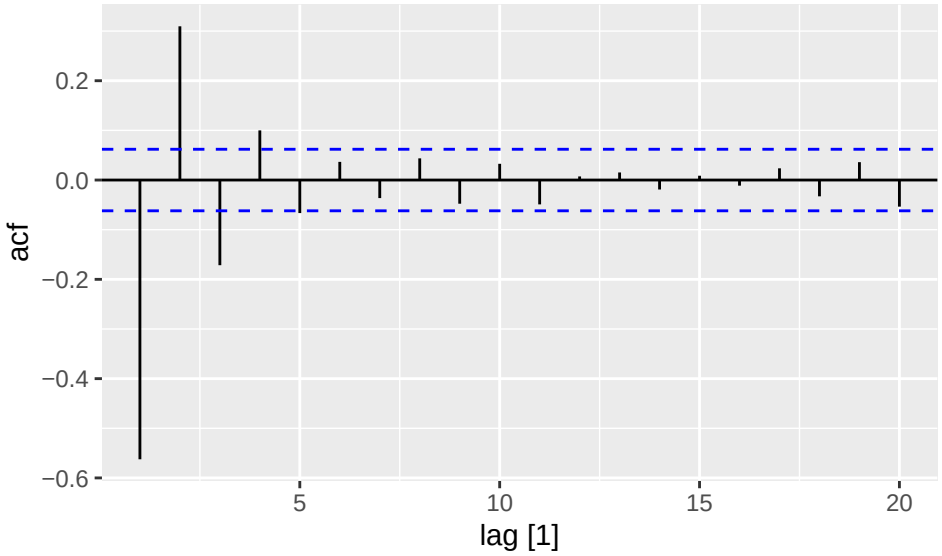


Figur 15: ACF för simulerad AR(1) med c = 5 och phi = 0.5

Vi kan inte se någon trend i den simulerade tidsserien. Medelvärdet ser ut att vara kring 10, det teretiska medelvärdet är $c/(1-\phi_1) = 5/(1-0.5) = 10$ vilket alltså bör stämma. Tittar vi på ACF:en så ser vi att de första 3 spikara är signifikanta, de är över 0.06 Och de skattade värdena ser ut att stämma med de teoretiska, även om vi ser en spik som är signifikant vid lagg 11 så kan detta var pga slumpen.



Figur 16: Simulerad AR(1) med c = 5 och phi = -0.5

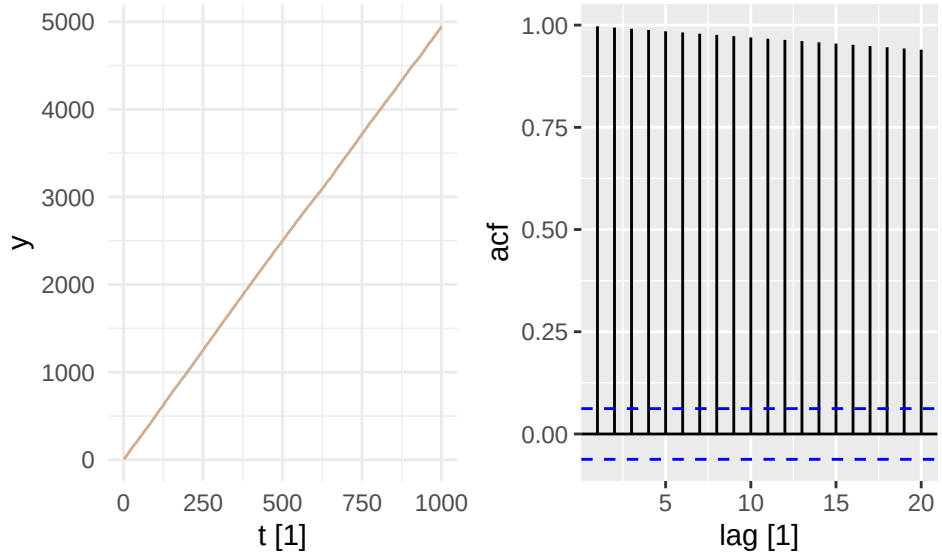


Figur 17: ACF för simulerad AR(1) med c = 5 och phi = -0.5

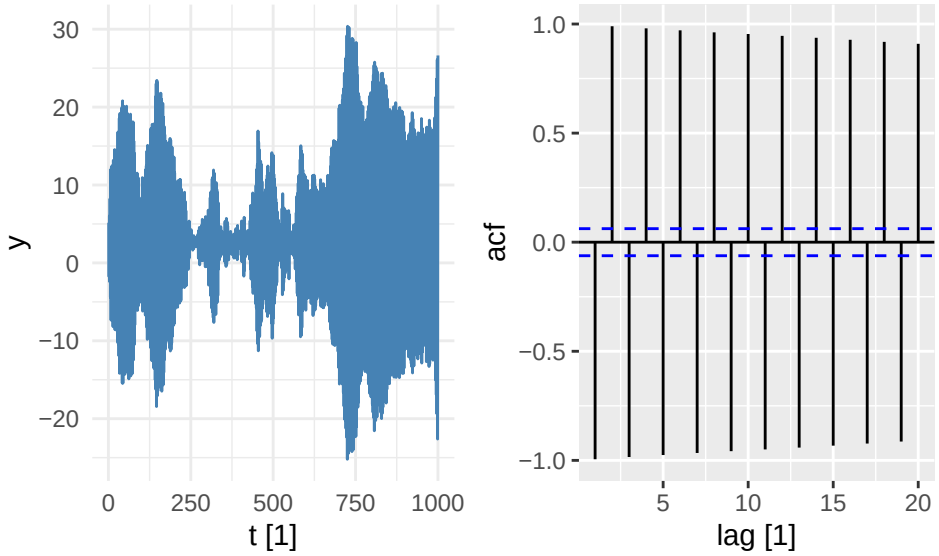
Denna gång ser vi heller ingen trend i serien som ser ut som brus. Medelvärdet ser ut att röra sig runt 3. Det teoretiska medelvärdet ges av $c/(1-\phi_1) = 5/(1--0.5) = 5/1.5 = 3.33$. Tittar vi på ACF:en ser vi att varannan stapel är negativ och varannan är positiv, vilket det bör vara eftersom ϕ_1 är negativ, så alla positiva laggar bör få en positiv autokorrelation. Värdena på de första spikarna ser också ut att stämma ganska bra med de faktiska teoretiska värdena.

3c - Slumpvandring

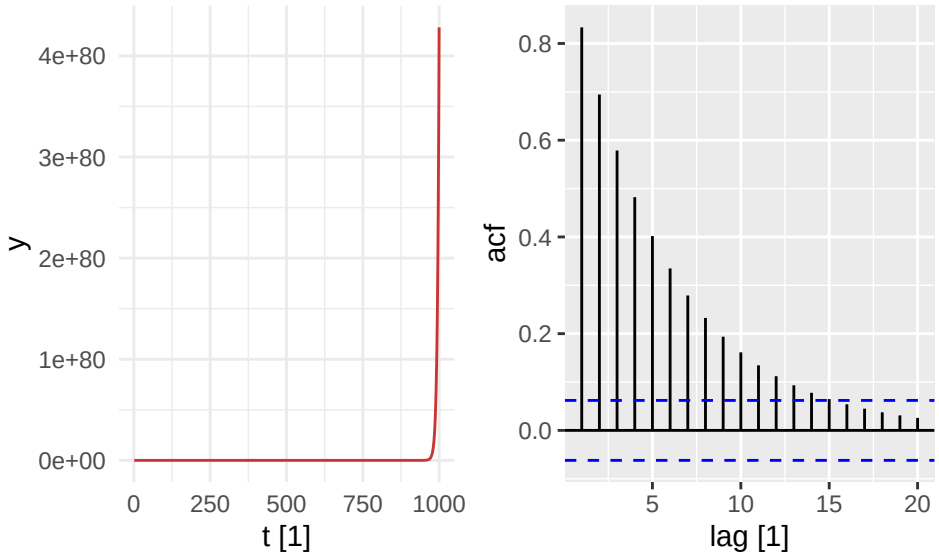
Vi simulerar 3 stycken radom walks med olika värden på ϕ_1 låt oss ta värdena 1, -1 och 1.2 och låta c fortsatt vara 5:



Figur 18: Slumpvandring för simulerad AR(1) med c = 5 och phi = 1



Figur 19: Slumpvandring för simulerad AR(1) med c = 5 och phi = -1



Figur 20: Slumpvandring för simulerad AR(1) med c = 5 och phi = 1.2

När vi använder $\phi = 1$ får vi en klar trend och ACF visar att alla 20 laggar är signifikanta.

När vi använder $\phi = -1$ får vi ett jätte kostigt kluddigt mönster i tidsserieplotten och ACF:en är vartannat negativ och vartannat positiv, men alla staplar (av 20) är signifikanta.

När vi använder $\phi = 1.2$ får vi något som ser ut som en möjligtvis exponentiell ökning mot oändligheten och vi ser att laggarna fram till lagg 14 är signifikanta.