figure_models

```
set.seed(931229)

# Parameter values and settings
NumObsSim     = 5000 # Simulated Sample size
numObsToPlot  = 500
ACFLagstoPlot = 20
sigma2        = 1    # variance of noise term
```

Simulate MA(1)

Model 1

```
pdf("figure/1.pdf", width = 10, height = 3)
par(mfrow = c(1,3)) # resets to default
# MA(1) -1
theta_1      = -1
# Simulate MA(1) - using built in arima.sim()
y_MA_1 <- arima.sim(model = list(ma = c(theta_1)), n = NumObsSim, sd = sqrt(sigma2))
# Time series plot, ACF and PACF in one
ts.plot(y_MA_1[1:numObsToPlot], ylab = "Y", main = "Time series plot of MA(1)")
acf(y_MA_1[1:NumObsSim], lag.max = ACFLagstoPlot,
    type = "correlation", plot = T, main = "ACF of MA(1)",
    xlim=c(1,ACFLagstoPlot))
acf(y_MA_1[1:NumObsSim], lag.max = ACFLagstoPlot, type = "partial", plot = T, main = "PACF of MA(1)")
dev.off()
```

```
## pdf
##   2
```

Model 2

```
pdf("figure/2.pdf", width = 10, height = 3)
par(mfrow = c(1,3)) # resets to default
# MA(1) -0.45
theta_1      = -0.45
# Simulate MA(1) - using built in arima.sim()
y_MA_1 <- arima.sim(model = list(ma = c(theta_1)), n = NumObsSim, sd = sqrt(sigma2))
# Time series plot, ACF and PACF in one


ts.plot(y_MA_1[1:numObsToPlot], ylab = "Y", main = "Time series plot of MA(1)")
acf(y_MA_1[1:NumObsSim], lag.max = ACFLagstoPlot,
    type = "correlation", plot = T, main = "ACF of MA(1)",
    xlim=c(1,ACFLagstoPlot))
acf(y_MA_1[1:NumObsSim], lag.max = ACFLagstoPlot, type = "partial", plot = T, main = "PACF of MA(1)")
par(mfrow = c(1,1)) # resets to default
dev.off()
```

```
## pdf
##   2
```

Model 3

```
pdf("figure/3.pdf", width = 10, height = 3)
par(mfrow = c(1,3)) # resets to default
# Simulate MA(1) 0
theta_1      = 0
# Simulate MA(1) - using built in arima.sim()
y_MA_1<- arima.sim(model = list(ma = c(theta_1)), n = NumObsSim, sd = sqrt(sigma2))
# Time series plot, ACF and PACF in one
ts.plot(y_MA_1[1:numObsToPlot], ylab = "Y", main = "Time series plot of MA(1)")
acf(y_MA_1[1:NumObsSim], lag.max = ACFLagstoPlot,
    type = "correlation", plot = T, main = "ACF of MA(1)",
    xlim=c(1,ACFLagstoPlot))
acf(y_MA_1[1:NumObsSim], lag.max = ACFLagstoPlot, type = "partial", plot = T, main = "PACF of MA(1)")
par(mfrow = c(1,1)) # resets to default
dev.off()
```

```
## pdf
##   2
```

Model 4

```
pdf("figure/4.pdf", width = 10, height = 3)
par(mfrow = c(1,3)) # resets to default
# Simulate MA(1) 0.45
theta_1      = 0.45
# Simulate MA(1) - using built in arima.sim()
y_MA_1 <- arima.sim(model = list(ma = c(theta_1)), n = NumObsSim, sd = sqrt(sigma2))
# Time series plot, ACF and PACF in one
ts.plot(y_MA_1[1:numObsToPlot], ylab = "Y", main = "Time series plot of MA(1)")
acf(y_MA_1[1:NumObsSim], lag.max = ACFLagstoPlot,
    type = "correlation", plot = T, main = "ACF of MA(1)",
    xlim=c(1,ACFLagstoPlot))
acf(y_MA_1[1:NumObsSim], lag.max = ACFLagstoPlot, type = "partial", plot = T, main = "PACF of MA(1)")
par(mfrow = c(1,1)) # resets to default
dev.off()
```

```
## pdf
##   2
```

Model 5

```
pdf("figure/5.pdf", width = 10, height = 3)
par(mfrow = c(1,3)) # resets to default
# Simulate MA(1) 1
theta_1      = 1
# Simulate MA(1) - using built in arima.sim()
y_MA_1<- arima.sim(model = list(ma = c(theta_1)), n = NumObsSim, sd = sqrt(sigma2))
```

```r
# Time series plot, ACF and PACF in one


ts.plot(y_MA_1[1:numObsToPlot], ylab = "Y", main = "Time series plot of MA(1)")
acf(y_MA_1[1:NumObsSim], lag.max = ACFLagstoPlot,
    type = "correlation", plot = T, main = "ACF of MA(1)",
    xlim=c(1,ACFLagstoPlot))
acf(y_MA_1[1:NumObsSim], lag.max = ACFLagstoPlot, type = "partial", plot = T, main = "PACF of MA(1)")
par(mfrow = c(1,1)) # resets to default
dev.off()
```

```
## pdf
##   2
```

Model 6

```r
pdf("figure/6.pdf", width = 10, height = 3)
par(mfrow = c(1,3)) # resets to default
# Simulate MA(1) 2
theta_1      = 2
# Simulate MA(1) - using built in arima.sim()
y_MA<- arima.sim(model = list(ma = c(theta_1)), n = NumObsSim, sd = sqrt(sigma2))
# Time series plot, ACF and PACF in one


ts.plot(y_MA_1[1:numObsToPlot], ylab = "Y", main = "Time series plot of MA(1)")
acf(y_MA_1[1:NumObsSim], lag.max = ACFLagstoPlot,
    type = "correlation", plot = T, main = "ACF of MA(1)",
    xlim=c(1,ACFLagstoPlot))
acf(y_MA_1[1:NumObsSim], lag.max = ACFLagstoPlot, type = "partial", plot = T, main = "PACF of MA(1)")
par(mfrow = c(1,1)) # resets to default
dev.off()
```

```
## pdf
##   2
```

Simulate MA(2)

Model 7

```r
pdf("figure/7.pdf", width = 10, height = 3)
par(mfrow = c(1,3)) # resets to default
# Simulate MA(2)
theta_1      = -0.8
theta_2      = 0
y_MA_2 <- arima.sim(model = list(ma = c(theta_1, theta_2)), n = NumObsSim, sd = sqrt(sigma2))


ts.plot(y_MA_2[1:numObsToPlot], ylab = "Y", main = "Time series plot of MA(2)")
acf(y_MA_2[1:NumObsSim], lag.max = ACFLagstoPlot,
    type = "correlation", plot = T, main = "ACF of MA(2)",
    xlim=c(1,ACFLagstoPlot))
```

```
acf(y_MA_2[1:NumObsSim], lag.max = ACFLagstoPlot, type = "partial", plot = T, main = "PACF of MA(2)")
par(mfrow = c(1,1)) # resets to default
dev.off()
```

```
## pdf
##   2
```

Model 8

```
pdf("figure/8.pdf", width = 10, height = 3)
par(mfrow = c(1,3)) # resets to default
# Simulate MA(2)
theta_1      = -0.8
theta_2      = 0.7
y_MA_2 <- arima.sim(model = list(ma = c(theta_1, theta_2)), n = NumObsSim, sd = sqrt(sigma2))


ts.plot(y_MA_2[1:numObsToPlot], ylab = "Y", main = "Time series plot of MA(2)")
acf(y_MA_2[1:NumObsSim], lag.max = ACFLagstoPlot,
    type = "correlation", plot = T, main = "ACF of MA(2)",
    xlim=c(1,ACFLagstoPlot))
acf(y_MA_2[1:NumObsSim], lag.max = ACFLagstoPlot, type = "partial", plot = T, main = "PACF of MA(2)")
par(mfrow = c(1,1)) # resets to default
dev.off()
```

```
## pdf
##   2
```

Model 9

```
pdf("figure/9.pdf", width = 10, height = 3)
par(mfrow = c(1,3)) # resets to default
# Simulate MA(2)
theta_1      = -0.8
theta_2      = 1
y_MA_2 <- arima.sim(model = list(ma = c(theta_1, theta_2)), n = NumObsSim, sd = sqrt(sigma2))


ts.plot(y_MA_2[1:numObsToPlot], ylab = "Y", main = "Time series plot of MA(2)")
acf(y_MA_2[1:NumObsSim], lag.max = ACFLagstoPlot,
    type = "correlation", plot = T, main = "ACF of MA(2)",
    xlim=c(1,ACFLagstoPlot))
acf(y_MA_2[1:NumObsSim], lag.max = ACFLagstoPlot, type = "partial", plot = T, main = "PACF of MA(2)")
par(mfrow = c(1,1)) # resets to default
dev.off()
```

```
## pdf
##   2
```

Model 10

```
pdf("figure/10.pdf", width = 10, height = 3)
par(mfrow = c(1,3)) # resets to default
# Simulate MA(2)
theta_1      = 0
theta_2      = 0
y_MA_2 <- arima.sim(model = list(ma = c(theta_1, theta_2)), n = NumObsSim, sd = sqrt(sigma2))


ts.plot(y_MA_2[1:numObsToPlot], ylab = "Y", main = "Time series plot of MA(2)")
acf(y_MA_2[1:NumObsSim], lag.max = ACFLagstoPlot,
    type = "correlation", plot = T, main = "ACF of MA(2)",
    xlim=c(1,ACFLagstoPlot))
acf(y_MA_2[1:NumObsSim], lag.max = ACFLagstoPlot, type = "partial", plot = T, main = "PACF of MA(2)")
par(mfrow = c(1,1)) # resets to default
dev.off()
```

```
## pdf
##   2
```

Model 11

```
pdf("figure/11.pdf", width = 10, height = 3)
par(mfrow = c(1,3)) # resets to default
# Simulate MA(2)
theta_1      = 0
theta_2      = 0.7
y_MA_2 <- arima.sim(model = list(ma = c(theta_1, theta_2)), n = NumObsSim, sd = sqrt(sigma2))


ts.plot(y_MA_2[1:numObsToPlot], ylab = "Y", main = "Time series plot of MA(2)")
acf(y_MA_2[1:NumObsSim], lag.max = ACFLagstoPlot,
    type = "correlation", plot = T, main = "ACF of MA(2)",
    xlim=c(1,ACFLagstoPlot))
acf(y_MA_2[1:NumObsSim], lag.max = ACFLagstoPlot, type = "partial", plot = T, main = "PACF of MA(2)")
par(mfrow = c(1,1)) # resets to default
dev.off()
```

```
## pdf
##   2
```

Model 12

```
pdf("figure/12.pdf", width = 10, height = 3)
par(mfrow = c(1,3)) # resets to default
# Simulate MA(2)
theta_1      = 0
theta_2      = 1
y_MA_2 <- arima.sim(model = list(ma = c(theta_1, theta_2)), n = NumObsSim, sd = sqrt(sigma2))
```

```r
ts.plot(y_MA_2[1:numObsToPlot], ylab = "Y", main = "Time series plot of MA(2)")
acf(y_MA_2[1:NumObsSim], lag.max = ACFLagstoPlot,
    type = "correlation", plot = T, main = "ACF of MA(2)",
    xlim=c(1,ACFLagstoPlot))
acf(y_MA_2[1:NumObsSim], lag.max = ACFLagstoPlot, type = "partial", plot = T, main = "PACF of MA(2)")
par(mfrow = c(1,1)) # resets to default
dev.off()
```

```
## pdf
##   2
```

Model 13

```r
pdf("figure/13.pdf", width = 10, height = 3)
par(mfrow = c(1,3)) # resets to default
# Simulate MA(2)
theta_1      = 0.8
theta_2      = 0
y_MA_2 <- arima.sim(model = list(ma = c(theta_1, theta_2)), n = NumObsSim, sd = sqrt(sigma2))



ts.plot(y_MA_2[1:numObsToPlot], ylab = "Y", main = "Time series plot of MA(2)")
acf(y_MA_2[1:NumObsSim], lag.max = ACFLagstoPlot,
    type = "correlation", plot = T, main = "ACF of MA(2)",
    xlim=c(1,ACFLagstoPlot))
acf(y_MA_2[1:NumObsSim], lag.max = ACFLagstoPlot, type = "partial", plot = T, main = "PACF of MA(2)")
par(mfrow = c(1,1)) # resets to default
dev.off()
```

```
## pdf
##   2
```

Model 14

```r
pdf("figure/14.pdf", width = 10, height = 3)
par(mfrow = c(1,3)) # resets to default
# Simulate MA(2)
theta_1      = 0.8
theta_2      = 0.7
y_MA_2 <- arima.sim(model = list(ma = c(theta_1, theta_2)), n = NumObsSim, sd = sqrt(sigma2))



ts.plot(y_MA_2[1:numObsToPlot], ylab = "Y", main = "Time series plot of MA(2)")
acf(y_MA_2[1:NumObsSim], lag.max = ACFLagstoPlot,
    type = "correlation", plot = T, main = "ACF of MA(2)",
    xlim=c(1,ACFLagstoPlot))
acf(y_MA_2[1:NumObsSim], lag.max = ACFLagstoPlot, type = "partial", plot = T, main = "PACF of MA(2)")
par(mfrow = c(1,1)) # resets to default
dev.off()
```

```
## pdf
##   2
```

Model 15

```
pdf("figure/15.pdf", width = 10, height = 3)
par(mfrow = c(1,3)) # resets to default
# Simulate MA(2)
theta_1      = 0.8
theta_2      = 1
y_MA_2 <- arima.sim(model = list(ma = c(theta_1, theta_2)), n = NumObsSim, sd = sqrt(sigma2))


ts.plot(y_MA_2[1:numObsToPlot], ylab = "Y", main = "Time series plot of MA(2)")
acf(y_MA_2[1:NumObsSim], lag.max = ACFLagstoPlot,
    type = "correlation", plot = T, main = "ACF of MA(2)",
    xlim=c(1,ACFLagstoPlot))
acf(y_MA_2[1:NumObsSim], lag.max = ACFLagstoPlot, type = "partial", plot = T, main = "PACF of MA(2)")
par(mfrow = c(1,1)) # resets to default
dev.off()
```

```
## pdf
##   2
```

Simulate AR(1)

```
# initiate series. Let the first two elements of y_AR be innovations
y_AR1 <- c(rnorm(1)*sqrt(sigma2), rnorm(1)*sqrt(sigma2), rep(0, NumObsSim-2))
y_AR2 <- c(rnorm(1)*sqrt(sigma2), rnorm(1)*sqrt(sigma2), rep(0, NumObsSim-2))
e    <- rnorm(NumObsSim)*sqrt(sigma2) # residual
```

Model 16

```
pdf("figure/16.pdf", width = 10, height = 3)
par(mfrow = c(1,3)) # resets to default
# AR(1)
phi_1        = -1
# Simulate/generate realization of AR
y_AR1 <- c(rnorm(1)*sqrt(sigma2), rnorm(1)*sqrt(sigma2), rep(0, NumObsSim-2))
e    <- rnorm(NumObsSim)*sqrt(sigma2) # residual

for(i in 3:NumObsSim){
  y_AR1[i] <- phi_1*y_AR1[i-1] + e[i]
}
# should be stored as time series object, makes for easier ploting.
y_AR1 <- as.ts(y_AR1)

# Time series plot, ACF and PACF in one
```

```r
ts.plot(y_AR1[1:numObsToPlot], ylab = "Y", main = "Time series plot of AR(1)")
acf(y_AR1[1:NumObsSim], lag.max = ACFLagstoPlot,
    type = "correlation", plot = T, main = "ACF of AR(1)",
    xlim=c(1,ACFLagstoPlot))
acf(y_AR1[1:NumObsSim], lag.max = ACFLagstoPlot, type = "partial", plot = T, main = "PACF of AR(1)")
par(mfrow = c(1,1)) # resets to default
dev.off()
```

```
## pdf
##   2
```

Model 17

```r
pdf("figure/17.pdf", width = 10, height = 3)
par(mfrow = c(1,3)) # resets to default
# AR(1)
phi_1        = -0.95
# Simulate/generate realization of AR
y_AR1 <- c(rnorm(1)*sqrt(sigma2), rnorm(1)*sqrt(sigma2), rep(0, NumObsSim-2))
e    <- rnorm(NumObsSim)*sqrt(sigma2) # residual

for(i in 3:NumObsSim){
  y_AR1[i] <- phi_1*y_AR1[i-1] + e[i]
}
# should be stored as time series object, makes for easier ploting.
y_AR1 <- as.ts(y_AR1)

# Time series plot, ACF and PACF in one


ts.plot(y_AR1[1:numObsToPlot], ylab = "Y", main = "Time series plot of AR(1)")
acf(y_AR1[1:NumObsSim], lag.max = ACFLagstoPlot,
    type = "correlation", plot = T, main = "ACF of AR(1)",
    xlim=c(1,ACFLagstoPlot))
acf(y_AR1[1:NumObsSim], lag.max = ACFLagstoPlot, type = "partial", plot = T, main = "PACF of AR(1)")
par(mfrow = c(1,1)) # resets to default
dev.off()
```

```
## pdf
##   2
```

Model 18

```r
pdf("figure/18.pdf", width = 10, height = 3)
par(mfrow = c(1,3)) # resets to default
# AR(1)
phi_1        = -0.75
# Simulate/generate realization of AR
y_AR1 <- c(rnorm(1)*sqrt(sigma2), rnorm(1)*sqrt(sigma2), rep(0, NumObsSim-2))
e    <- rnorm(NumObsSim)*sqrt(sigma2) # residual
```

```r
for(i in 3:NumObsSim){
  y_AR1[i] <- phi_1*y_AR1[i-1] + e[i]
}
# should be stored as time series object, makes for easier ploting.
y_AR1 <- as.ts(y_AR1)

# Time series plot, ACF and PACF in one


ts.plot(y_AR1[1:numObsToPlot], ylab = "Y", main = "Time series plot of AR(1)")
acf(y_AR1[1:NumObsSim], lag.max = ACFLagstoPlot,
    type = "correlation", plot = T, main = "ACF of AR(1)",
    xlim=c(1,ACFLagstoPlot))
acf(y_AR1[1:NumObsSim], lag.max = ACFLagstoPlot, type = "partial", plot = T, main = "PACF of AR(1)")
par(mfrow = c(1,1)) # resets to default
dev.off()
```

```
## pdf
##   2
```

Model 19

```r
pdf("figure/19.pdf", width = 10, height = 3)
par(mfrow = c(1,3)) # resets to default
# AR(1)
phi_1        = 0
# Simulate/generate realization of AR
y_AR1 <- c(rnorm(1)*sqrt(sigma2), rnorm(1)*sqrt(sigma2), rep(0, NumObsSim-2))
e    <- rnorm(NumObsSim)*sqrt(sigma2) # residual

for(i in 3:NumObsSim){
  y_AR1[i] <- phi_1*y_AR1[i-1] + e[i]
}
# should be stored as time series object, makes for easier ploting.
y_AR1 <- as.ts(y_AR1)

# Time series plot, ACF and PACF in one


ts.plot(y_AR1[1:numObsToPlot], ylab = "Y", main = "Time series plot of AR(1)")
acf(y_AR1[1:NumObsSim], lag.max = ACFLagstoPlot,
    type = "correlation", plot = T, main = "ACF of AR(1)",
    xlim=c(1,ACFLagstoPlot))
acf(y_AR1[1:NumObsSim], lag.max = ACFLagstoPlot, type = "partial", plot = T, main = "PACF of AR(1)")
par(mfrow = c(1,1)) # resets to default
dev.off()
```

```
## pdf
##   2
```

Model 20

```
pdf("figure/20.pdf", width = 10, height = 3)
par(mfrow = c(1,3)) # resets to default
# AR(1)
phi_1        = 0.75
# Simulate/generate realization of AR
y_AR1 <- c(rnorm(1)*sqrt(sigma2), rnorm(1)*sqrt(sigma2), rep(0, NumObsSim-2))
e    <- rnorm(NumObsSim)*sqrt(sigma2) # residual

for(i in 3:NumObsSim){
  y_AR1[i] <- phi_1*y_AR1[i-1] + e[i]
}
# should be stored as time series object, makes for easier ploting.
y_AR1 <- as.ts(y_AR1)
# Time series plot, ACF and PACF in one


ts.plot(y_AR1[1:numObsToPlot], ylab = "Y", main = "Time series plot of AR(1)")
acf(y_AR1[1:NumObsSim], lag.max = ACFLagstoPlot,
    type = "correlation", plot = T, main = "ACF of AR(1)",
    xlim=c(1,ACFLagstoPlot))
acf(y_AR1[1:NumObsSim], lag.max = ACFLagstoPlot, type = "partial", plot = T, main = "PACF of AR(1)")
par(mfrow = c(1,1)) # resets to default
dev.off()
```

```
## pdf
##   2
```

Model 21

```
pdf("figure/21.pdf", width = 10, height = 3)
par(mfrow = c(1,3)) # resets to default
# AR(1)
phi_1        = 0.95
# Simulate/generate realization of AR
y_AR1 <- c(rnorm(1)*sqrt(sigma2), rnorm(1)*sqrt(sigma2), rep(0, NumObsSim-2))
e    <- rnorm(NumObsSim)*sqrt(sigma2) # residual

for(i in 3:NumObsSim){
  y_AR1[i] <- phi_1*y_AR1[i-1] + e[i]
}
# should be stored as time series object, makes for easier ploting.
y_AR1 <- as.ts(y_AR1)
# Time series plot, ACF and PACF in one


ts.plot(y_AR1[1:numObsToPlot], ylab = "Y", main = "Time series plot of AR(1)")
acf(y_AR1[1:NumObsSim], lag.max = ACFLagstoPlot,
    type = "correlation", plot = T, main = "ACF of AR(1)",
    xlim=c(1,ACFLagstoPlot))
acf(y_AR1[1:NumObsSim], lag.max = ACFLagstoPlot, type = "partial", plot = T, main = "PACF of AR(1)")
```

```r
par(mfrow = c(1,1)) # resets to default
dev.off()
```

```
## pdf
##   2
```

Model 22

```r
pdf("figure/22.pdf", width = 10, height = 3)
par(mfrow = c(1,3)) # resets to default
# AR(1)
phi_1        = 1
# Simulate/generate realization of AR
y_AR1 <- c(rnorm(1)*sqrt(sigma2), rnorm(1)*sqrt(sigma2), rep(0, NumObsSim-2))
e    <- rnorm(NumObsSim)*sqrt(sigma2) # residual

for(i in 3:NumObsSim){
  y_AR1[i] <- phi_1*y_AR1[i-1] + e[i]
}
# should be stored as time series object, makes for easier ploting.
y_AR1 <- as.ts(y_AR1)
# Time series plot, ACF and PACF in one


ts.plot(y_AR1[1:numObsToPlot], ylab = "Y", main = "Time series plot of AR(1)")
acf(y_AR1[1:NumObsSim], lag.max = ACFLagstoPlot,
    type = "correlation", plot = T, main = "ACF of AR(1)",
    xlim=c(1,ACFLagstoPlot))
acf(y_AR1[1:NumObsSim], lag.max = ACFLagstoPlot, type = "partial", plot = T, main = "PACF of AR(1)")
par(mfrow = c(1,1)) # resets to default
dev.off()
```

```
## pdf
##   2
```

Simulate AR(2)

Model 23

```r
pdf("figure/23.pdf", width = 10, height = 3)
par(mfrow = c(1,3)) # resets to default
phi_1        = -0.9  # AR(1) parameter
phi_2        = 0.1     # AR(2) parameter
# Simulate/generate realization of AR
y_AR2 <- c(rnorm(1)*sqrt(sigma2), rnorm(1)*sqrt(sigma2), rep(0, NumObsSim-2))
e    <- rnorm(NumObsSim)*sqrt(sigma2) # residual

for(i in 3:NumObsSim){
  y_AR2[i] <- phi_1*y_AR2[i-1] + phi_2*y_AR2[i-2] + e[i]
}
# should be stored as time series object, makes for easier ploting.
```

```
y_AR2 <- as.ts(y_AR2)

# Time series plot, ACF and PACF in one


ts.plot(y_AR2[1:numObsToPlot], ylab = "Y", main = "Time series plot of AR(2)")
acf(y_AR2[1:NumObsSim], lag.max = ACFLagstoPlot,
    type = "correlation", plot = T, main = "ACF of AR(2)",
    xlim=c(1,ACFLagstoPlot))
acf(y_AR2[1:NumObsSim], lag.max = ACFLagstoPlot, type = "partial", plot = T, main = "PACF of AR(2)")
par(mfrow = c(1,1)) # resets to default
dev.off()
```

```
## pdf
##   2
```

Model 24 - fixed

```
pdf("figure/24.pdf", width = 10, height = 3)
par(mfrow = c(1,3)) # resets to default
# AR(2)
# initiate series. Let the first two elements of y_AR be innovations
phi_1        = -0.9
phi_2        = 0.20
NumObsSim    = 500

# initiate series. Let the first two elements of y_AR be innovations
y_AR2 <- c(rnorm(1)*sqrt(sigma2), rnorm(1)*sqrt(sigma2), rep(0, NumObsSim-2))
e <- rnorm(NumObsSim)*sqrt(sigma2) # residual

# Simulate/generate realization of AR
for(i in 3:NumObsSim){
  y_AR2[i] <- phi_1*y_AR2[i-1] + phi_2*y_AR2[i-2] + e[i]
}
# should be stored as time series object, makes for easier ploting.
y_AR2 <- as.ts(y_AR2)

# Time series plot, ACF and PACF in one


ts.plot(y_AR2[1:numObsToPlot], ylab = "Y", main = "Time series plot of AR(2)")
acf(y_AR2[1:NumObsSim], lag.max = ACFLagstoPlot,
    type = "correlation", plot = T, main = "ACF of AR(2)",
    xlim=c(1,ACFLagstoPlot))
acf(y_AR2[1:NumObsSim], lag.max = ACFLagstoPlot, type = "partial", plot = T, main = "PACF of AR(2)")
par(mfrow = c(1,1)) # resets to default
dev.off()
```

```
## pdf
##   2
```

Model 25 - fixed

```
pdf("figure/25.pdf", width = 10, height = 3)
par(mfrow = c(1,3)) # resets to default
# AR(2)
phi_1        = -0.9
phi_2        = 0.8
NumObsSim    = 500

# Simulate/generate realization of AR

y_AR2 <- c(rnorm(1)*sqrt(sigma2), rnorm(1)*sqrt(sigma2), rep(0, NumObsSim-2))
e    <- rnorm(NumObsSim)*sqrt(sigma2) # residual

for(i in 3:NumObsSim){
  y_AR2[i] <- phi_1*y_AR2[i-1] + phi_2*y_AR2[i-2] + e[i]
}
# should be stored as time series object, makes for easier ploting.
y_AR2 <- as.ts(y_AR2)

# Time series plot, ACF and PACF in one


ts.plot(y_AR2[1:numObsToPlot], ylab = "Y", main = "Time series plot of AR(2)")
acf(y_AR2[1:NumObsSim], lag.max = ACFLagstoPlot,
    type = "correlation", plot = T, main = "ACF of AR(2)",
    xlim=c(1,ACFLagstoPlot))
acf(y_AR2[1:NumObsSim], lag.max = ACFLagstoPlot, type = "partial", plot = T, main = "PACF of AR(2)")
par(mfrow = c(1,1)) # resets to default
dev.off()
```

```
## pdf
##   2
```

Model 26

```
pdf("figure/26.pdf", width = 10, height = 3)
par(mfrow = c(1,3)) # resets to default
# AR(2)
phi_1        = 0
phi_2        = 0.1
# Simulate/generate realization of AR

y_AR2 <- c(rnorm(1)*sqrt(sigma2), rnorm(1)*sqrt(sigma2), rep(0, NumObsSim-2))
e    <- rnorm(NumObsSim)*sqrt(sigma2) # residual

for(i in 3:NumObsSim){
  y_AR2[i] <- phi_1*y_AR2[i-1] + phi_2*y_AR2[i-2] + e[i]
}
# should be stored as time series object, makes for easier ploting.
y_AR2 <- as.ts(y_AR2)
# Time series plot, ACF and PACF in one
```

```r
ts.plot(y_AR2[1:numObsToPlot], ylab = "Y", main = "Time series plot of AR(2)")
acf(y_AR2[1:NumObsSim], lag.max = ACFLagstoPlot,
    type = "correlation", plot = T, main = "ACF of AR(2)",
    xlim=c(1,ACFLagstoPlot))
acf(y_AR2[1:NumObsSim], lag.max = ACFLagstoPlot, type = "partial", plot = T, main = "PACF of AR(2)")
par(mfrow = c(1,1)) # resets to default
dev.off()
```

```
## pdf
##   2
```

Model 27

```r
pdf("figure/27.pdf", width = 10, height = 3)
par(mfrow = c(1,3)) # resets to default
# AR(2)
phi_1        = 0
phi_2        = 0.2
# Simulate/generate realization of AR
y_AR2 <- c(rnorm(1)*sqrt(sigma2), rnorm(1)*sqrt(sigma2), rep(0, NumObsSim-2))
e     <- rnorm(NumObsSim)*sqrt(sigma2) # residual

for(i in 3:NumObsSim){
  y_AR2[i] <- phi_1*y_AR2[i-1] + phi_2*y_AR2[i-2] + e[i]
}
# should be stored as time series object, makes for easier ploting.
y_AR2 <- as.ts(y_AR2)

# Time series plot, ACF and PACF in one


ts.plot(y_AR2[1:numObsToPlot], ylab = "Y", main = "Time series plot of AR(2)")
acf(y_AR2[1:NumObsSim], lag.max = ACFLagstoPlot,
    type = "correlation", plot = T, main = "ACF of AR(2)",
    xlim=c(1,ACFLagstoPlot))
acf(y_AR2[1:NumObsSim], lag.max = ACFLagstoPlot, type = "partial", plot = T, main = "PACF of AR(2)")
par(mfrow = c(1,1)) # resets to default
dev.off()
```

```
## pdf
##   2
```

Model 28

```r
pdf("figure/28.pdf", width = 10, height = 3)
par(mfrow = c(1,3)) # resets to default
# AR(2)
phi_1        = 0
phi_2        = 0.8
# Simulate/generate realization of AR
```

```
y_AR2 <- c(rnorm(1)*sqrt(sigma2), rnorm(1)*sqrt(sigma2), rep(0, NumObsSim-2))
e    <- rnorm(NumObsSim)*sqrt(sigma2) # residual

for(i in 3:NumObsSim){
  y_AR2[i] <- phi_1*y_AR2[i-1] + phi_2*y_AR2[i-2] + e[i]
}
# should be stored as time series object, makes for easier ploting.
y_AR2 <- as.ts(y_AR2)
# Time series plot, ACF and PACF in one


ts.plot(y_AR2[1:numObsToPlot], ylab = "Y", main = "Time series plot of AR(2)")
acf(y_AR2[1:NumObsSim], lag.max = ACFLagstoPlot,
    type = "correlation", plot = T, main = "ACF of AR(2)",
    xlim=c(1,ACFLagstoPlot))
acf(y_AR2[1:NumObsSim], lag.max = ACFLagstoPlot, type = "partial", plot = T, main = "PACF of AR(2)")
par(mfrow = c(1,1)) # resets to default
dev.off()
```

```
## pdf
##   2
```

Model 29

```
pdf("figure/29.pdf", width = 10, height = 3)
par(mfrow = c(1,3)) # resets to default
# AR(2)
phi_1        = 0.7
phi_2        = 0.1
# Simulate/generate realization of AR
y_AR2 <- c(rnorm(1)*sqrt(sigma2), rnorm(1)*sqrt(sigma2), rep(0, NumObsSim-2))
e    <- rnorm(NumObsSim)*sqrt(sigma2) # residual

for(i in 3:NumObsSim){
  y_AR2[i] <- phi_1*y_AR2[i-1] + phi_2*y_AR2[i-2] + e[i]
}
# should be stored as time series object, makes for easier ploting.
y_AR2 <- as.ts(y_AR2)

# Time series plot, ACF and PACF in one


ts.plot(y_AR2[1:numObsToPlot], ylab = "Y", main = "Time series plot of AR(2)")
acf(y_AR2[1:NumObsSim], lag.max = ACFLagstoPlot,
    type = "correlation", plot = T, main = "ACF of AR(2)",
    xlim=c(1,ACFLagstoPlot))
acf(y_AR2[1:NumObsSim], lag.max = ACFLagstoPlot, type = "partial", plot = T, main = "PACF of AR(2)")
par(mfrow = c(1,1)) # resets to default
```

Model 30

```
pdf("figure/30.pdf", width = 10, height = 3)
par(mfrow = c(1,3)) # resets to default
# AR(2)
phi_1        = 0.7
phi_2        = 0.2
# Simulate/generate realization of AR

y_AR2 <- c(rnorm(1)*sqrt(sigma2), rnorm(1)*sqrt(sigma2), rep(0, NumObsSim-2))
e    <- rnorm(NumObsSim)*sqrt(sigma2) # residual

for(i in 3:NumObsSim){
  y_AR2[i] <- phi_1*y_AR2[i-1] + phi_2*y_AR2[i-2] + e[i]
}
# should be stored as time series object, makes for easier ploting.
y_AR2 <- as.ts(y_AR2)

# Time series plot, ACF and PACF in one


ts.plot(y_AR2[1:numObsToPlot], ylab = "Y", main = "Time series plot of AR(2)")
acf(y_AR2[1:NumObsSim], lag.max = ACFLagstoPlot,
    type = "correlation", plot = T, main = "ACF of AR(2)",
    xlim=c(1,ACFLagstoPlot))
acf(y_AR2[1:NumObsSim], lag.max = ACFLagstoPlot, type = "partial", plot = T, main = "PACF of AR(2)")
par(mfrow = c(1,1)) # resets to default
dev.off()
```

```
## pdf
##   2
```

Model 31 - fixed

```
pdf("figure/31.pdf", width = 10, height = 3)
par(mfrow = c(1,3)) # resets to default
# AR(2)
phi_1        = 0.7
phi_2        = 0.8
NumObsSim    = 500

# Simulate/generate realization of AR
y_AR2 <- c(rnorm(1)*sqrt(sigma2), rnorm(1)*sqrt(sigma2), rep(0, NumObsSim-2))
e    <- rnorm(NumObsSim)*sqrt(sigma2) # residual

for(i in 3:NumObsSim){
  y_AR2[i] <- phi_1*y_AR2[i-1] + phi_2*y_AR2[i-2] + e[i]
}
# should be stored as time series object, makes for easier ploting.
y_AR2 <- as.ts(y_AR2)

# Time series plot, ACF and PACF in one
```

```r
ts.plot(y_AR2[1:numObsToPlot], ylab = "Y", main = "Time series plot of AR(2)")
acf(y_AR2[1:NumObsSim], lag.max = ACFLagstoPlot,
    type = "correlation", plot = T, main = "ACF of AR(2)",
    xlim=c(1,ACFLagstoPlot))
acf(y_AR2[1:NumObsSim], lag.max = ACFLagstoPlot, type = "partial", plot = T, main = "PACF of AR(2)")
par(mfrow = c(1,1)) # resets to default
dev.off()
```

```
## pdf
##   2
```

Simulate ARMA (1,1) process

Model 32

```r
pdf("figure/32.pdf", width = 10, height = 3)
par(mfrow = c(1,3)) # resets to default
# Set parameters
phi_1   = -0.90
theta_1 = -0.4

# Simulate/generate realization of ARMA
y_ARMA <- arima.sim(model = list(ar = phi_1, ma = theta_1), n = NumObsSim, sd = sqrt(sigma2))

# should be stored as time series object, makes for easier ploting.
y_ARMA <- as.ts(y_ARMA)

# Time series plot, ACF and PACF in one


ts.plot(y_AR2[1:numObsToPlot], ylab = "Y", main = "Time series plot of ARMA(1,1))")
acf(y_ARMA[1:NumObsSim], lag.max = ACFLagstoPlot,
    type = "correlation", plot = T, main = "ACF of AR(2)",
    xlim=c(1,ACFLagstoPlot))
acf(y_ARMA[1:NumObsSim], lag.max = ACFLagstoPlot, type = "partial", plot = T, main = "PACF of AR(2)")
par(mfrow = c(1,1)) # resets to default
dev.off()
```

```
## pdf
##   2
```

Model 33

```r
pdf("figure/33.pdf", width = 10, height = 3)
par(mfrow = c(1,3)) # resets to default
# Set parameters
phi_1   = -0.90
theta_1 = 0.4

# Simulate/generate realization of ARMA
```

```r
y_ARMA <- arima.sim(model = list(ar = phi_1, ma = theta_1), n = NumObsSim, sd = sqrt(sigma2))

# should be stored as time series object, makes for easier ploting.
y_ARMA <- as.ts(y_ARMA)

# Time series plot, ACF and PACF in one


ts.plot(y_AR2[1:numObsToPlot], ylab = "Y", main = "Time series plot of ARMA(1,1))")
acf(y_ARMA[1:NumObsSim], lag.max = ACFLagstoPlot,
    type = "correlation", plot = T, main = "ACF of AR(2)",
    xlim=c(1,ACFLagstoPlot))
acf(y_ARMA[1:NumObsSim], lag.max = ACFLagstoPlot, type = "partial", plot = T, main = "PACF of AR(2)")
par(mfrow = c(1,1)) # resets to default
dev.off()
```

```
## pdf
##   2
```

Model 34

```r
pdf("figure/34.pdf", width = 10, height = 3)
par(mfrow = c(1,3)) # resets to default
# Set parameters
phi_1   = -0.80
theta_1 = 0.4

# Simulate/generate realization of ARMA
y_ARMA <- arima.sim(model = list(ar = phi_1, ma = theta_1), n = NumObsSim, sd = sqrt(sigma2))

# should be stored as time series object, makes for easier ploting.
y_ARMA <- as.ts(y_ARMA)

# Time series plot, ACF and PACF in one


ts.plot(y_AR2[1:numObsToPlot], ylab = "Y", main = "Time series plot of ARMA(1,1))")
acf(y_ARMA[1:NumObsSim], lag.max = ACFLagstoPlot,
    type = "correlation", plot = T, main = "ACF of AR(2)",
    xlim=c(1,ACFLagstoPlot))
acf(y_ARMA[1:NumObsSim], lag.max = ACFLagstoPlot, type = "partial", plot = T, main = "PACF of AR(2)")
par(mfrow = c(1,1)) # resets to default
dev.off()
```

```
## pdf
##   2
```

Model 35

```r
pdf("figure/35.pdf", width = 10, height = 3)
par(mfrow = c(1,3)) # resets to default
# Set parameters
```

```r
phi_1   = -0.80
theta_1 = -0.4

# Simulate/generate realization of ARMA
y_ARMA <- arima.sim(model = list(ar = phi_1, ma = theta_1), n = NumObsSim, sd = sqrt(sigma2))

# should be stored as time series object, makes for easier ploting.
y_ARMA <- as.ts(y_ARMA)

# Time series plot, ACF and PACF in one


ts.plot(y_AR2[1:numObsToPlot], ylab = "Y", main = "Time series plot of ARMA(1,1))")
acf(y_ARMA[1:NumObsSim], lag.max = ACFLagstoPlot,
    type = "correlation", plot = T, main = "ACF of AR(2)",
    xlim=c(1,ACFLagstoPlot))
acf(y_ARMA[1:NumObsSim], lag.max = ACFLagstoPlot, type = "partial", plot = T, main = "PACF of AR(2)")
par(mfrow = c(1,1)) # resets to default
dev.off()
```

```
## pdf
##   2
```

Model 36

```r
pdf("figure/36.pdf", width = 10, height = 3)
par(mfrow = c(1,3)) # resets to default
# Set parameters
phi_1   = 0.90
theta_1 = 0.4

# Simulate/generate realization of ARMA
y_ARMA <- arima.sim(model = list(ar = phi_1, ma = theta_1), n = NumObsSim, sd = sqrt(sigma2))

# should be stored as time series object, makes for easier ploting.
y_ARMA <- as.ts(y_ARMA)

# Time series plot, ACF and PACF in one


ts.plot(y_AR2[1:numObsToPlot], ylab = "Y", main = "Time series plot of ARMA(1,1))")
acf(y_ARMA[1:NumObsSim], lag.max = ACFLagstoPlot,
    type = "correlation", plot = T, main = "ACF of AR(2)",
    xlim=c(1,ACFLagstoPlot))
acf(y_ARMA[1:NumObsSim], lag.max = ACFLagstoPlot, type = "partial", plot = T, main = "PACF of AR(2)")
par(mfrow = c(1,1)) # resets to default
dev.off()
```

```
## pdf
##   2
```

Model 37

```
pdf("figure/37.pdf", width = 10, height = 3)
par(mfrow = c(1,3)) # resets to default
# Set parameters
phi_1   = 0.90
theta_1 = -0.4

# Simulate/generate realization of ARMA
y_ARMA <- arima.sim(model = list(ar = phi_1, ma = theta_1), n = NumObsSim, sd = sqrt(sigma2))

# should be stored as time series object, makes for easier ploting.
y_ARMA <- as.ts(y_ARMA)

# Time series plot, ACF and PACF in one


ts.plot(y_AR2[1:numObsToPlot], ylab = "Y", main = "Time series plot of ARMA(1,1))")
acf(y_ARMA[1:NumObsSim], lag.max = ACFLagstoPlot,
    type = "correlation", plot = T, main = "ACF of AR(2)",
    xlim=c(1,ACFLagstoPlot))
acf(y_ARMA[1:NumObsSim], lag.max = ACFLagstoPlot, type = "partial", plot = T, main = "PACF of AR(2)")
par(mfrow = c(1,1)) # resets to default
dev.off()
```

```
## pdf
##   2
```