Continuation of Lecture 1, September 8, 2025

Stationary series

Marta Boczon

Department of Economics

Copnehagen Business School

mbo.eco@cbs.dk

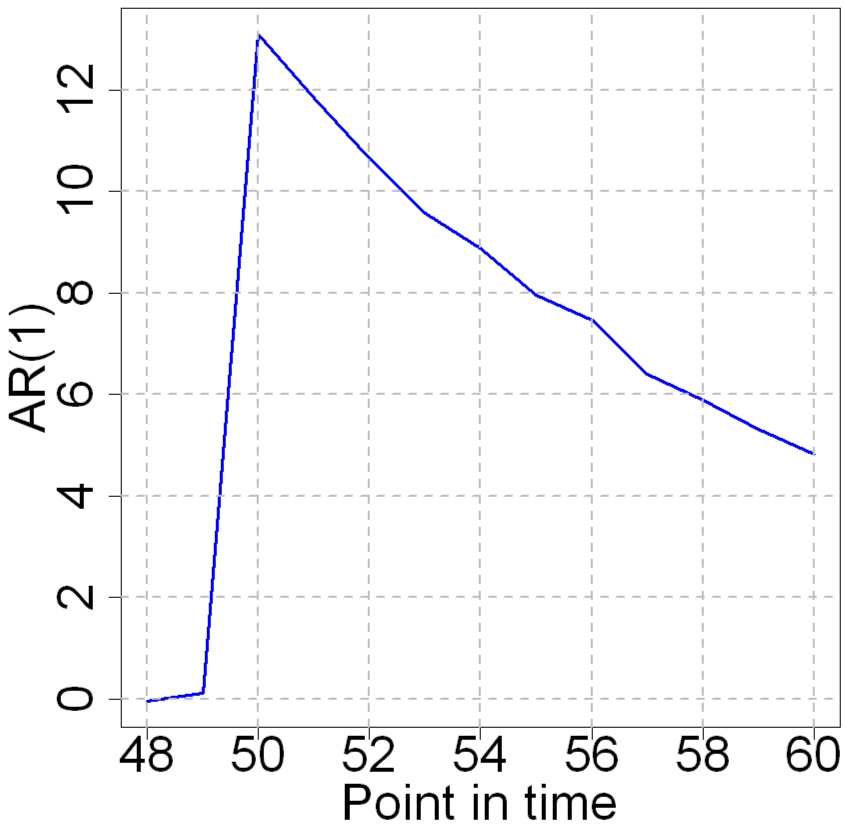
Effect of a shock on AR process

In an AR(p) process, the impact of a shock at time t persists and gradually decays but never fully disappears.

This means that the effect of a shock is felt indefinitely, although its influence becomes weaker over time. In other words, an AR(p) model incorporates the history of all prior shocks, with their impact fading as they move further into the past. This makes AR processes well-suited for modeling phenomena that evolve smoothly rather than jumping abruptly. For example, body temperature does not fluctuate wildly from one moment to the next — instead, it adjusts gradually, and current measurements remain close to recent ones.

In [36]:
```
# Define the number of time periods (observations) to simulate
T = 200

# Initialize a vector of length T with zeros
# to store the AR(1) process values
ar1h = rep(0, T)

# Set the coefficient (AR(1) parameter)
beta = 0.9

# Generate shocks (innovations) from a Normal distribution
set.seed(123)               # set seed for reproducibility
e = rnorm(T, mean = 0, sd = 0.2)

# Generate the AR(1) process with a large shock at t = 50
for (t in 2:T) {
  if (t == 50) {
    e[t] = e[t] + 13 # insert a big positive shock at time t = 50
  }
  # AR(1) recursion
  ar1h[t] = beta * ar1h[t-1] + e[t]
}
```

```
# Set plotting options
options(repr.plot.width = 8, repr.plot.height = 8)
par(mar = c(5, 5, 5, 5))

# Plot a zoomed-in window around the shock (t = 48 to 60)
plot(
  c(48:60), ar1h[48:60],
  type = "l", col = "blue", lwd = 3,            # line plot
  xlab = "Point in time", ylab = "AR(1)",       # axis labels
  cex.lab = 2.5, cex.axis = 2.5                 # scaling factors
)

# Add gridlines for readability
grid(nx = NULL, ny = NULL,
     lty = 2,       # dashed lines
     col = "gray", # grid color
     lwd = 2)       # grid width
```



Effect of a shock on MA process

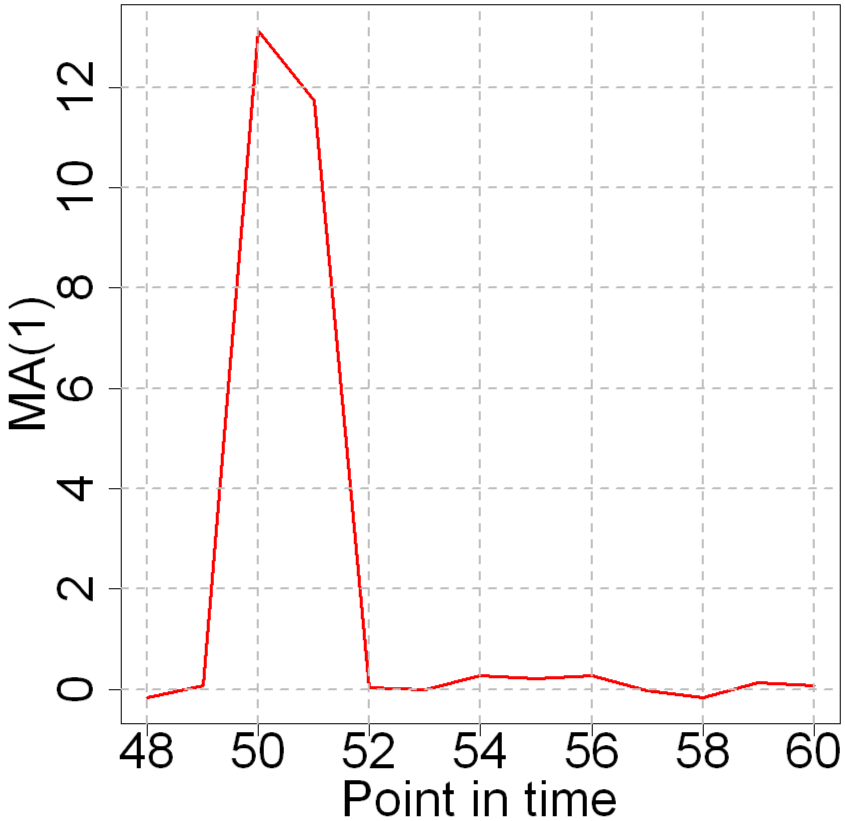
For MA processes, the effect of a shock fully disappears after q periods.

For example, in an MA(1) process, a shock at time t influences y_t (through the current shock) and y_{t+1} (through the lagged shock). By $t + 2$, its effect is gone.

In general, for an MA(q) process, each shock affects the series for at most $q + 1$ periods before vanishing completely. The magnitude of its impact during those periods depends on the size of the coefficients β_i.

This is in sharp contrast to AR(p) processes, where the effect of a single shock persists indefinitely — fading gradually but never truly disappearing. Thus, AR processes have infinite memory, while MA processes have finite memory.

In [37]:
```
# Define the number of time periods (observations) to simulate
T = 200

# Initialize a vector of length T with zeros
# to store the MA(1) process values
ma1h = rep(0, T)

# Set the coefficient (MA(1) parameter)
beta = 0.9

# Generate shocks (innovations) from a Normal distribution
set.seed(123)               # set seed for reproducibility
e = rnorm(T, mean = 0, sd = 0.2)

# Generate the MA(1) process with a big shock at t = 50
for (t in 2:T) {
  if (t == 50) {
    e[t] = e[t] + 13 # insert a big positive shock at time t = 50
  }
  # MA(1) recursion
  ma1h[t] = beta * e[t-1] + e[t]
}

# Set plotting options
options(repr.plot.width = 8, repr.plot.height = 8)
par(mar = c(5, 5, 5, 5))

# Plot a zoomed-in window around the shock (t = 48 to 60)
plot(
  c(48:60), ma1h[48:60],
  type = "l", col = "red", lwd = 3,             # line plot
  xlab = "Point in time", ylab = "MA(1)",       # axis labels
  cex.lab = 2.5, cex.axis = 2.5                 # scaling factors
)

# Add gridlines for readability
grid(nx = NULL, ny = NULL,
     lty = 2,       # dashed lines
     col = "gray", # grid color
     lwd = 2)       # grid width
```



Exercise:

Simulate 1,000 observations from an AR(3) process, and an MA(3) process, using the coefficients:

$\beta_1 = 0.2$

$\beta_2 = 0.1$

$\beta_3 = 0.2$

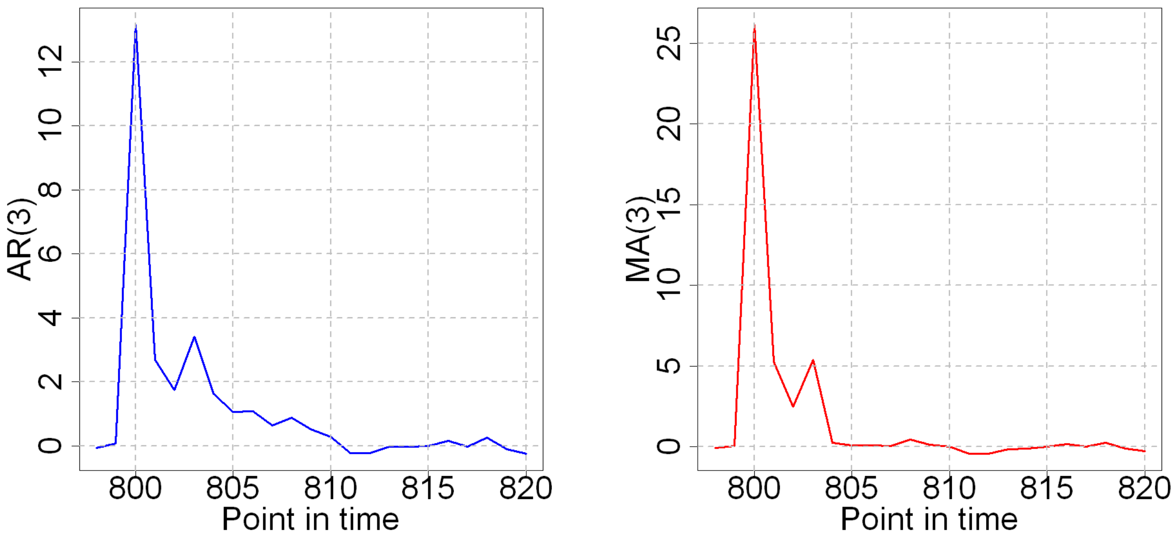
Introduce a large random shock at time $t = 800$, and plot the series to compare how the effect of the shock evolves in the AR(3) versus the MA(3) process.

Solution:

In [40]:
```
# Define the number of time periods (observations) to simulate
T = 1000

# Initialize vectors with zeros to store the AR(3) and MA(3) process values
ar3 = rep(0, T) # AR(3) series
ma3 = rep(0, T) # MA(3) series
```

```r
# Set the coefficients for the AR(3) and MA(3) models
beta1 = 0.2
beta2 = 0.1
beta3 = 0.2

# Generate shocks (innovations) from a Normal distribution
set.seed(123)                # set seed for reproducibility
e = rnorm(T, mean = 0, sd = 0.2)

# Generate the AR(3) process with a large shock at t = 800
for (t in 4:T) {
  if (t == 800) {
    e[t] = e[t] + 13         # insert a big positive shock at time t = 800
  }
  # AR(3) recursion
  # Start at t = 4 because the AR(3) recursion needs the three previous values:
  # ar3[t-1], ar3[t-2], and ar3[t-3].
  # For t < 4, not all these lags exist.
  ar3[t] = beta1 * ar3[t-1] + beta2 * ar3[t-2] + beta3 * ar3[t-3] + e[t]
}

# Generate the MA(3) process with a big shock at t = 800
for (t in 4:T) {
  if (t == 800) {
    e[t] = e[t] + 13         # insert a big positive shock at time t = 800
  }
  # MA(3) recursion
  # Start at t = 4 because the MA(3) recursion needs the three previous shocks:
  # e[t-1], e[t-2], and e[t-3].
  # For t < 4, not all these lags exist.
  ma3[t] = beta1 * e[t-1] + beta2 * e[t-2] + beta3 * e[t-3] + e[t]
}

# Set plot dimensions for Jupyter/IRkernel (ignored in base R)
options(repr.plot.width = 16, repr.plot.height = 8)

# Adjust plot margins (bottom, left, top, right) in lines of text
par(mar = c(5, 5, 5, 5))

# Arrange plots in a 1x2 grid (1 row, 2 columns)
par(mfrow = c(1, 2))

# Plot a zoomed-in window around the shock
plot(
  c(798:820), ar3[798:820],
  type = "l", col = "blue", lwd = 3,             # line plot
  xlab = "Point in time", ylab = "AR(3)",        # axis labels
  cex.lab = 2.5, cex.axis = 2.5                  # scaling factors
)

# Add gridlines for readability
grid(nx = NULL, ny = NULL,
     lty = 2,      # dashed lines
     col = "gray", # grid color
     lwd = 2)      # grid width

# Plot a zoomed-in window around the shock
plot(
  c(798:820), ma3[798:820],
```

```r
  type = "l", col = "red", lwd = 3,            # line plot
  xlab = "Point in time", ylab = "MA(3)",       # axis labels
  cex.lab = 2.5, cex.axis = 2.5                 # scaling factors
)

# Add gridlines for readability
grid(nx = NULL, ny = NULL,
     lty = 2,       # dashed lines
     col = "gray", # grid color
     lwd = 2)       # grid width
```



Lecture 2, September 8, 2025

Non-stationary series

Marta Boczon

Department of Economics

Copnehagen Business School

mbo.eco@cbs.dk

Introduction

Stationarity is needed for statistical inference because of the difficulties in modeling data whose structure changes over time.

Strict stationarity

1. A time series is strictly stationary if the joint distribution of $(y_t, y_{t+1}, \ldots, y_{t+k})$ is the same for all t.

2. In practice: all statistical properties (mean, variance, covariance, higher moments) are time-invariant.

3. Very strong condition — rarely holds in real data.

Weak (or covariance) stationarity

1. A weaker form of stationarity — often sufficient for econometrics.
2. Most time series models (ARMA, VAR, etc.) assume weak stationarity.
3. Requires:

    3.1 Constant mean: $E(y_t) = \mu$ for all t

    3.2 Constant variance: $Var(y_t) = \sigma^2$ for all t

    3.3 Covariance depends only on the lag between observations, not on time itself: $Cov(y_t, y_{t+h}) = \gamma(h)$ for all t

Examples of non-stationary processes:

- a) $y_t = \beta_0 + \beta_1 t + \epsilon_t$
- b) $y_t = \beta_0 + \beta_1 y_{t-1} + \beta_2 t + \epsilon_t, \quad |\beta_1| < 1$
- c) $y_t = y_{t-1} + \epsilon_t$
- d) $y_t = \beta_0 + y_{t-1} + \epsilon_t$
- e) $y_t = \beta_0 + y_{t-1} + \beta_1 t + \epsilon_t$

In [3]:
```
#install.packages('pracma')
#install.packages('urca')
#install.packages('stargazer')
```

Example a)

Constant and deterministic trend

Example a: y_t as a function of time t

$$y_t = \beta_0 + \beta_1 t + \epsilon_t,$$

where

- y_t: the value of the time series at time t
- t: time (e.g., day, month, year)
- β_0: intercept, drift (it adds a constant expected change each period, regardless of t)
- β_1: slope on deterministic trend (time-driven component)

- ϵ_t: random error term

This process is *non-stationary* because the mean of y_t changes over time due to the trend $\beta_1 t$.

Expected value of y_t:

$$\mathbb{E}(y_t) = \beta_0 + \beta_1 t$$

Hence,

$$\mathbb{E}(y_1) = \beta_0 + \beta_1, \quad \mathbb{E}(y_2) = \beta_0 + 2\beta_1, \quad \mathbb{E}(y_3) = \beta_0 + 3\beta_1, \ldots$$

The expected value is not constant over time.

Variance of y_t:

$$\mathbb{V}\mathrm{ar}(y_t) = \sigma_\epsilon^2$$

However, the variance is constant over time.

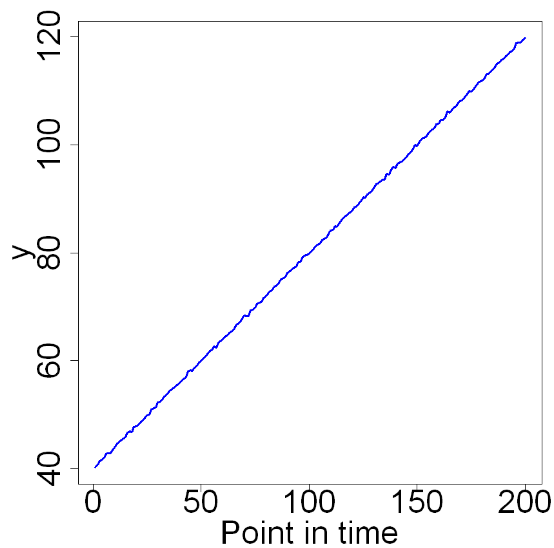
Proof: $\mathbb{E}(y_t) = \mathbb{E}(\beta_0 + \beta_1 t + \epsilon_t) = \underbrace{\mathbb{E}(\beta_0)}_{=\beta_0} + \underbrace{\mathbb{E}(\beta_1 t)}_{=\beta_1 t} + \underbrace{\mathbb{E}(\epsilon_t)}_{=0} = \beta_0 + \beta_1 t$

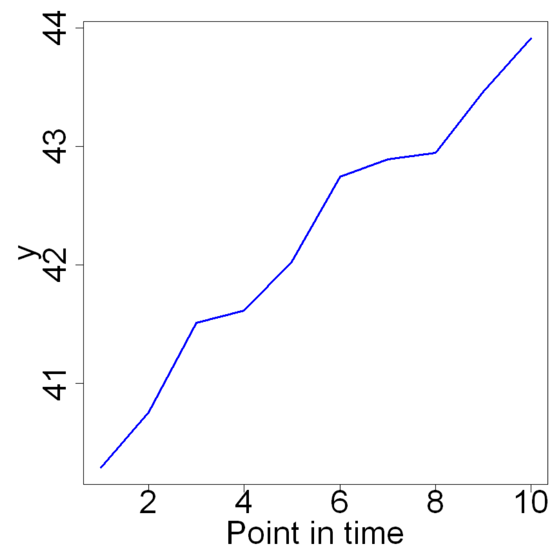
$\mathbb{V}\mathrm{ar}(y_t) = \mathbb{V}\mathrm{ar}(\beta_0 + \beta_1 t + \epsilon_t) = \underbrace{\mathbb{V}\mathrm{ar}(\beta_0)}_{=0} + \underbrace{\mathbb{V}\mathrm{ar}(\beta_1 t)}_{=0} + \underbrace{\mathbb{V}\mathrm{ar}(\epsilon_t)}_{=\sigma_\epsilon^2} = \sigma_\epsilon^2$

In [35]:

```
T = 200                          # Set the number of time periods (length of the series)
ex.a = rep(0, T)                 # Create an empty vector of length T to store the simulate
set.seed(123)                    # Fix the random seed for reproducibility
e = rnorm(T, 0, 0.2)             # Generate T random errors from N(0, 0.2^2)
beta0 = 40                       # Set the intercept (starting level of the series)
beta1 = 0.4                      # Set the slope (deterministic trend)
t = 1:T                          # Create a time index from 1 to T

# Loop over each time period and compute the value of y_t
for (i in 1:T) {
  ex.a[i] = beta0 + beta1 * t[i] + e[i]
}

# Set plotting options: figure size
options(repr.plot.width=16, repr.plot.height=8)

# Set margins and layout: larger margins, 1 row and 2 plots side by side
par(mar = c(5, 5, 5, 5), mfrow = c(1, 2))

# Plot the entire time series
plot(ex.a, type="l", col="blue",
     cex.lab=2.5, cex.axis=2.5,
     xlab= "Point in time", ylab = "y", lwd=3)

# Plot only the first 10 observations (zoom in to see short-run behavior)
plot(c(1:10), ex.a[1:10], type="l", col="blue",
```

```
    cex.lab=2.5, cex.axis=2.5,
    xlab= "Point in time", ylab = "y", lwd=3)
```





Example b)

Constant, deterministic time trend, and past value of y

Example b: y_t as a function of time t and y_{t-1} (i.e., yesterday's value of y_t)

$$y_t = \beta_0 + \beta_1 y_{t-1} + \beta_2 t + \epsilon_t, \quad |\beta_1| < 1$$

- y_t: the value of the time series at time t
- y_{t-1}: the lagged value of the series (yesterday's y_t)
- t: time (e.g., day, month, year)
- β_0: intercept, drift (it adds a constant expected change each period, regardless of t)
- β_1: autoregressive parameter (how strongly today's value depends on yesterday's value)
- β_2: slope on deterministic trend (time-driven component)
- ϵ_t: random error term

This process is *non-stationary* because the deterministic trend term $\beta_2 t$ makes the mean of y_t change over time. If $\beta_2 = 0$, the process could be stationary (as long as $|\beta_1| < 1$).

Expected value of y_t:

$$\mathbb{E}(y_t) = \frac{\beta_0}{1-\beta_1} + \frac{\beta_2 t}{1-\beta_2} - \frac{\beta_2\beta_1}{(1-\beta_1)^2}$$

We see that the expected value is not constant over time.

Variance of y_t:

$$\mathbb{V}\text{ar}(y_t) = \frac{\sigma_\epsilon^2}{1-\beta_1^2}$$

However, the variance is constant over time.

Proof: 1. Find the general recursive form of y_t: Begin by expanding (y_t recursively by substituting $y_{t-1}, y_{t-2}, \ldots$) which gives the general recursive form:

$$y_t = \sum_{j=0}^{\infty} \beta_1^j \left(\beta_0 + \beta_2(t-j) + \epsilon_{t-j}\right)$$

2. Simplify the general recursive form using the properties of the infinite sum of a geometric series: Since $|\beta_1| < 1$, the powers of β_1 form a geometric series. Use the geometric series sum formula to simplify the expression:

$$\sum_{j=0}^{\infty} \beta_1^j = \frac{1}{1-\beta_1}$$

Additionally, apply the known result for the sum of weighted terms $j\beta_1^j$:

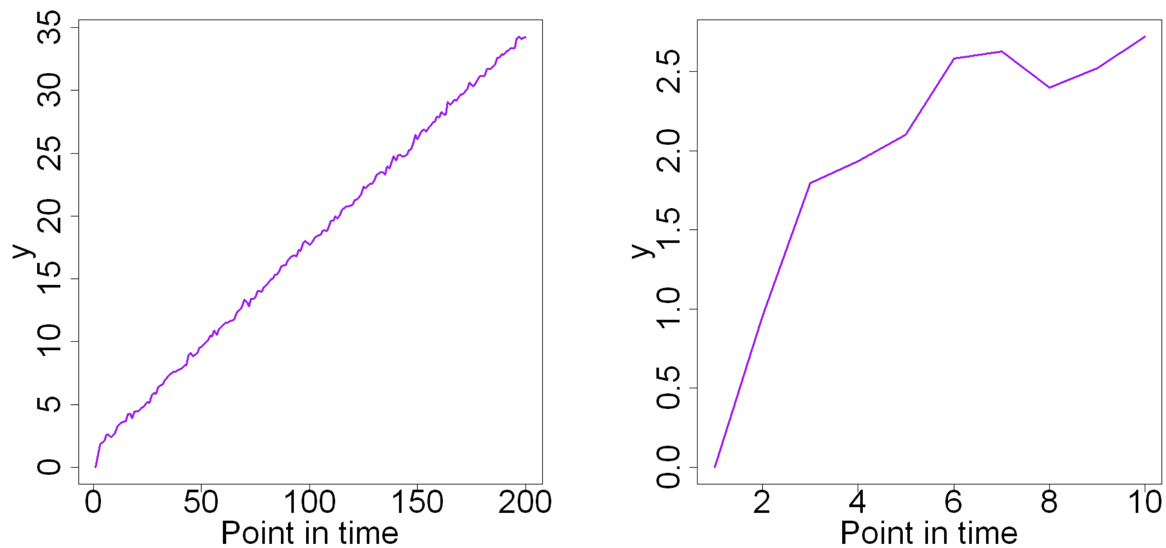
$$\sum_{j=0}^{\infty} j\beta_1^j = \frac{\beta_1}{(1-\beta_1)^2}$$

These will help reduce the infinite sum to a manageable closed-form expression. **3. Compute the expected value of y_t:** Taking the expectation (variance) of both sides and using $E[\epsilon_t] = 0$ ($E[\epsilon_t] = \sigma_\epsilon^2$), compute the expected value (variance).

In [38]:

```
T = 200                         # Set the number of time periods (length of the series)
ex.b = rep(0, T)                # Create an empty vector of length T to store the simulate
set.seed(123)                   # Fix the random seed for reproducibility
e = rnorm(T, 0, 0.2)            # Generate T random errors from N(0, 0.2^2)
beta0 = 0.8                     # Constant term (drift component)
beta1 = 0.4                     # Autoregressive coefficient (effect of past y on current
beta2 = 0.1                     # Coefficient on time trend (deterministic component)
t = 1:T                         # Create a time index from 1 to T

# Loop from the 2nd observation onwards
# Each value depends on:
#   - a constant (beta0)
#   - the lagged value (beta1 * ex.b[i-1]) → autoregressive component
#   - a deterministic time trend (beta2 * t[i])
#   - a random shock (e[i])
for (i in 2:T) {
  ex.b[i] = beta0 + beta1 * ex.b[i-1] + beta2 * t[i] + e[i]
}

# Set plotting options: figure size
options(repr.plot.width=16, repr.plot.height=8)
```

```r
# Set margins and layout: larger margins, 1 row and 2 plots side by side
par(mar = c(5, 5, 5, 5), mfrow = c(1, 2))

# Plot the entire time series
plot(ex.b, type="l", col="purple",
     cex.lab=2.5, cex.axis=2.5,
     xlab= "Point in time", ylab = "y", lwd=3)

# Plot only the first 10 observations (zoom in to see short-run behavior)
plot(c(1:10), ex.b[1:10], type="l", col="purple",
     cex.lab=2.5, cex.axis=2.5,
     xlab= "Point in time", ylab = "y", lwd=3)
```



Example c)

Stochastic trend (unit root) and no constant

Example c: Random walk without drift

$$y_t = y_{t-1} + \epsilon_t$$

- y_t: the value of the time series at time t
- y_{t-1}: the lagged value (yesterday's y_t)
- ϵ_t: random error term (white noise, mean zero, constant variance)

This is a random walk: today's value equals yesterday's value plus a random shock. It is *non-stationary* because the variance depends on time.

Expected value of y_t:

$$\mathbb{E}(y_t) = y_0,$$

where y_0 is the starting value.

We see that the expected value is constant over time.

Variance of y_t:

$$\color{blue}{\mathbb{V}\text{ar}(y_t) =} \color{red}{t}\color{blue}{\sigma_\epsilon^2}$$

Hence,

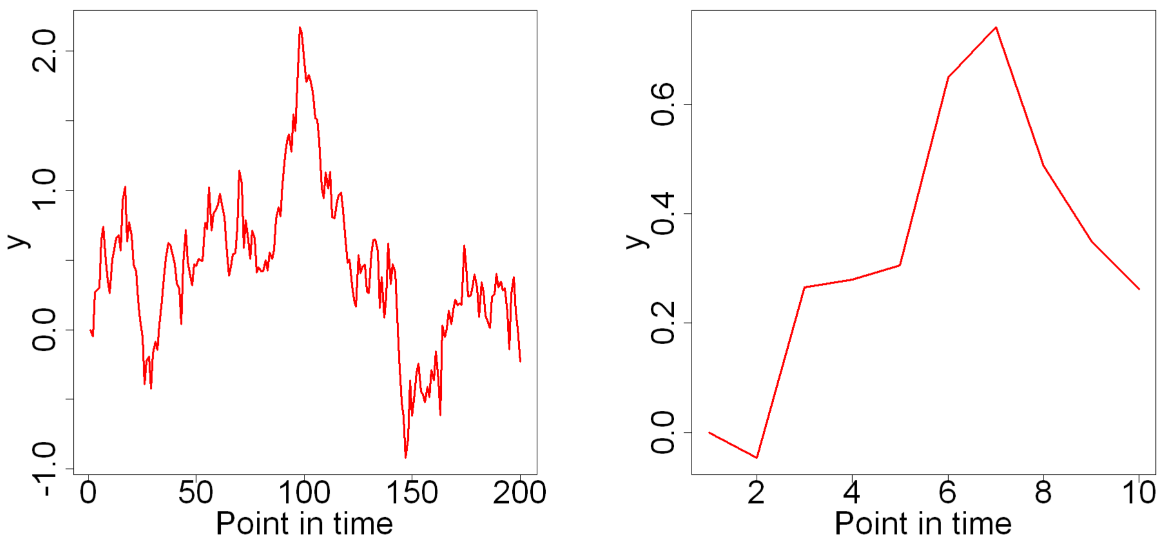
$\color{blue}{\mathbb{V}\text{ar}(y_1) = \sigma_\epsilon^2, \mathbb{V}\text{ar}(y_2) =} \color{red}{2}\color{blue}{\sigma_\epsilon^2, \mathbb{V}\text{ar}(y_3) =} \color{red}{3}\color{blue}{\sigma_\epsilon^2}, \ldots$

We see that the variance is not constant over time.

Proof: Random walk without constant can be rewritten as:
$y_t = y_0 + \sum_{i=1}^t \epsilon_i, \quad \epsilon_t \sim (0, \sigma_\epsilon^2)$ Then, it follows that:
$\mathbb{E}(y_t) = \mathbb{E}(y_0 + \sum_{i=1}^t \epsilon_i) = \underbrace{\mathbb{E}(y_0)}_{=y_0} + \mathbb{E}(\sum_{i=1}^t \epsilon_i) = y_0 + \sum_{i=1}^t \underbrace{\mathbb{E}(\epsilon_i)}_{=0} = y_0$
$\mathbb{V}\text{ar}(y_t) = \mathbb{V}\text{ar}(y_0 + \sum_{i=1}^t \epsilon_i) = \underbrace{\mathbb{V}\text{ar}(y_0)}_{=0} + \mathbb{V}\text{ar}(\sum_{i=1}^t \epsilon_i) = \sum_{i=1}^t \underbrace{\mathbb{V}\text{ar}(\epsilon_i)}_{=\sigma_t^2} = t\sigma_2$

In [39]:

```
T = 200                           # Set the number of time periods (length of the series)
ex.c = rep(0, T)                  # Create an empty vector of length T to store the simulate
set.seed(123)                     # Fix the random seed for reproducibility
e = rnorm(T, 0, 0.2)              # Generate T random errors from N(0, 0.2^2)

# Loop from the 2nd observation onwards
# Each value is yesterday’s value plus a random shock (random walk)
for (t in 2:T) {
  ex.c[t] = ex.c[t-1] + e[t]
}

# Set plotting options: figure size
options(repr.plot.width=16, repr.plot.height=8)

# Set margins and layout: larger margins, 1 row and 2 plots side by side
par(mar = c(5, 5, 5, 5), mfrow = c(1, 2))

# Plot the entire random walk
plot(ex.c, type="l",
     col="red", cex = 2.5, cex.lab=2.5, cex.axis=2.5, cex.main=2.5, cex.sub=2.5,
     xlab="Point in time", ylab="y", lwd=3)

# Plot only the first 10 observations (zoom in to see short-run fluctuations)
plot(c(1:10), ex.c[1:10], type="l",
     col="red", cex = 2.5, cex.lab=2.5, cex.axis=2.5, cex.main=2.5, cex.sub=2.5,
     xlab= "Point in time", ylab="y", lwd=3)
```



Example d)

Stochastic trend (unit root) and constant

Example d: Random walk with drift

$$y_t = \beta_0 + y_{t-1} + \epsilon_t$$

- y_t: the value of the time series at time t
- y_{t-1}: the lagged value (yesterday's y_t)
- β_0: intercept, drift (it adds a constant expected change each period, regardless of t)
- ϵ_t: random error term

This is a random walk with drift: each step is yesterday's value plus a constant shift and a random shock.
It is non-stationary because both the mean and variance depend on time.

Expected value of y_t:

$$\mathbb{E}(y_t) = y_0 + t\beta_0$$

Hence,

$$\mathbb{E}(y_1) = y_0 + \beta_0, \quad \mathbb{E}(y_2) = y_0 + 2\beta_0, \quad \mathbb{E}(y_3) = y_0 + 3\beta_0, \ldots$$

We see that the expected value is not constant over time.

Variance of y_t:

$$\mathbb{V}\text{ar}(y_t) = t\sigma_\epsilon^2$$

Hence,

$\mathbb{V}\mathrm{ar}(y_1) = \sigma_\epsilon^2, \mathbb{V}\mathrm{ar}(y_2) = 2\sigma_\epsilon^2, \mathbb{V}\mathrm{ar}(y_3) = 3\sigma_\epsilon^2, \ldots$

We see that the variance is not constant over time.

Proof: Random walk with constant can be rewritten as:
$y_t = y_0 + t\beta_0 + \sum_{i=1}^t \epsilon_i, \quad \epsilon_t \sim (0, \sigma_\epsilon^2)$ Then, it follows that:

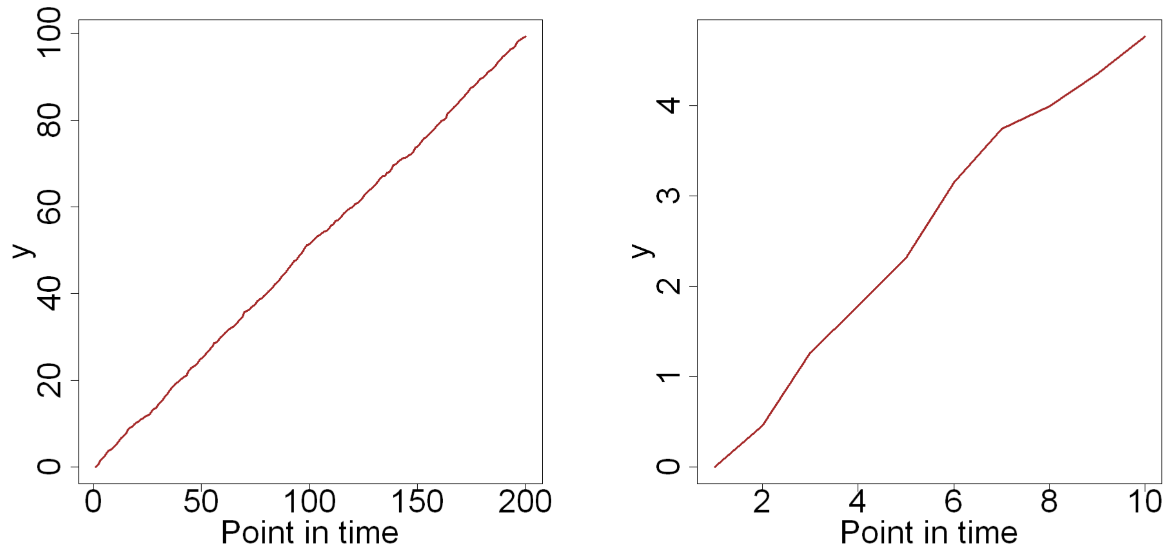
$$\mathbb{E}(y_t) = \mathbb{E}(y_0 + t\beta_0 + \sum_{i=1}^t \epsilon_i) = \underbrace{\mathbb{E}(y_0 + t\beta_0)}_{=y_0+t\beta_0} + \mathbb{E}(\sum_{i=1}^t \epsilon_i)$$

$$= y_0 + t\beta_0 + \sum_{i=1}^t \underbrace{\mathbb{E}(\epsilon_i)}_{=0} = y_0 + t\beta_0$$

$$\mathbb{V}\mathrm{ar}(y_t) = \mathbb{V}\mathrm{ar}(y_0 + t\beta_0 + \sum_{i=1}^t \epsilon_i) = \underbrace{\mathbb{V}\mathrm{ar}(y_0 + t\beta_0)}_{=0} + \mathbb{V}\mathrm{ar}(\sum_{i=1}^t \epsilon_i)$$

$$= \sum_{i=1}^t \underbrace{\mathbb{V}\mathrm{ar}(\epsilon_i)}_{=\sigma_t^2} = t\sigma_2$$

In [10]:
```
T = 200                        # Set the number of time periods (length of the series)
ex.d = rep(0, T)               # Create an empty vector of length T to store the simulate
beta0 = 0.5                    # Constant term (drift) that shifts the process upward ove
set.seed(123)                  # Fix the random seed for reproducibility
e = rnorm(T, 0, 0.2)           # Generate T random errors from N(0, 0.2^2)

# Loop from the 2nd observation onwards
# Each value equals yesterday’s value + a constant drift + a random shock
for (t in 2:T) {
  ex.d[t] = beta0 + ex.d[t-1] + e[t]
}

# Set plotting options: figure size
options(repr.plot.width=16, repr.plot.height=8)

# Set margins and layout: larger margins, 1 row and 2 plots side by side
par(mar = c(5, 5, 5, 5), mfrow = c(1, 2))

# Plot the entire random walk with drift
plot(ex.d, type="l",
     col="brown", cex = 2.5, cex.lab=2.5, cex.axis=2.5, cex.main=2.5, cex.sub=2.5,
     xlab="Point in time", ylab="y", lwd=3)

# Plot only the first 10 observations (zoom in to see short-run fluctuations)
plot(c(1:10), ex.d[1:10], type="l",
     col="brown", cex = 2.5, cex.lab=2.5, cex.axis=2.5, cex.main=2.5, cex.sub=2.5,
     xlab="Point in time", ylab="y", lwd=3)
```



Example e)

Stochastic trend (unit root), deterministic trend, and constant

Example e: Random walk with drift and deterministic trend

$$y_t = \beta_0 + y_{t-1} + \beta_1 t + \epsilon_t$$

- y_t: the value of the time series at time t
- y_{t-1}: the lagged value (yesterday's y_t)
- β_0: intercept, drift (it adds a constant expected change each period, regardless of t)
- β_1: slope on deterministic trend (time-driven component)
- ϵ_t: random error term

This is a random walk with drift and deterministic trend: each step is yesterday's value plus a constant shift, a time-dependent trend, and a random shock. It is non-stationary because both the mean and variance depend on time.

Expected value of y_t:

$$\mathbb{E}(y_t) = y_0 + t\beta_0 + \beta_1 \frac{t(t+1)}{2}$$

Hence,

$$\mathbb{E}(y_1) = y_0 + \beta_0 + \beta_1, \quad \mathbb{E}(y_2) = y_0 + 2\beta_0 + 6\beta_1, \quad \mathbb{E}(y_3) = y_0 + 3\beta_0 + 12\beta_1, \ldots$$

We see that the expected value is not constant over time.

Variance of y_t:

$$\mathbb{V}\text{ar}(y_t) = t\sigma^2_\epsilon$$

Hence,

$\mathbb{V}\text{ar}(y_1) = \sigma^2_\epsilon, \mathbb{V}\text{ar}(y_2) = 2\sigma^2_\epsilon, \mathbb{V}\text{ar}(y_3) = 3\sigma^2_\epsilon, \ldots$

We see that the variance is not constant over time.

Proof: Random walk with constant can be rewritten as:
$y_t = y_0 + t\beta_0 + \beta_1 \frac{t(1+t)}{2} + \sum_{i=1}^t \epsilon_i, \quad \epsilon_t \sim (0, \sigma^2_\epsilon)$ Then, it follows that:

$$\mathbb{E}(y_t) = \mathbb{E}(y_0 + t\beta_0 + \beta_1 \frac{t(1+t)}{2} + \sum_{i=1}^t \epsilon_i) = \underbrace{\mathbb{E}(y_0 + t\beta_0) + \beta_1 \frac{t(1+t)}{2})}_{=y_0 + t\beta_0 + \beta_1 \frac{t(1+t)}{2}} + \mathbb{E}(\sum_{i=1}^t \epsilon_i)$$

$$= y_0 + t\beta_0 + \beta_1 \frac{t(1+t)}{2} + \sum_{i=1}^t \underbrace{\mathbb{E}(\epsilon_i)}_{=0} = y_0 + t\beta_0 + \beta_1 \frac{t(1+t)}{2}$$

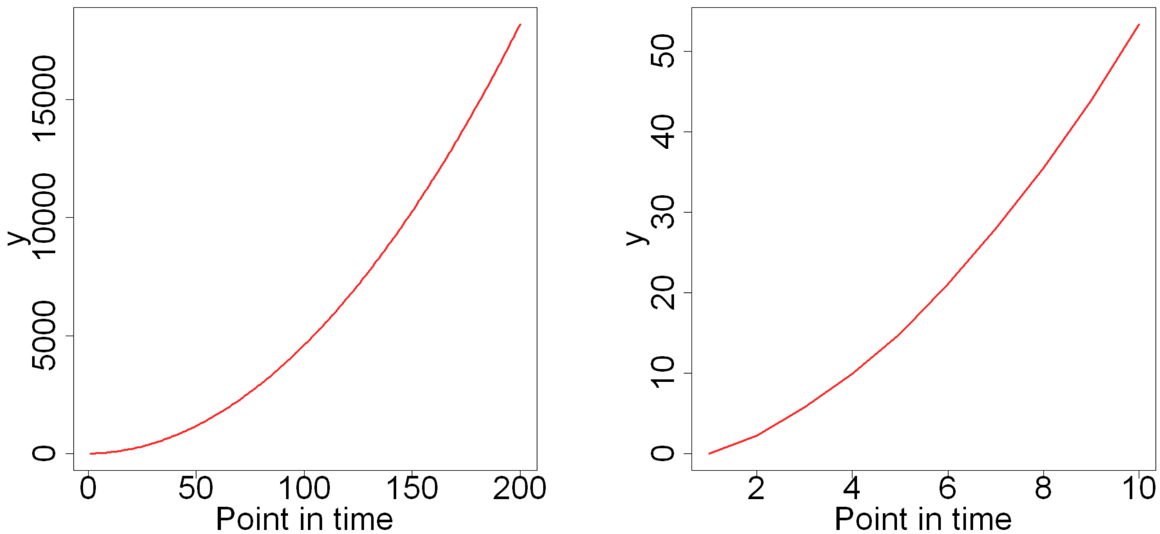
$$\mathbb{V}\text{ar}(y_t) = \mathbb{V}\text{ar}(y_0 + t\beta_0 + \beta_1 \frac{t(1+t)}{2} + \sum_{i=1}^t \epsilon_i)$$

$$= \underbrace{\mathbb{V}\text{ar}(y_0 + t\beta_0 + \beta_1 \frac{t(1+t)}{2})}_{=0} + \mathbb{V}\text{ar}(\sum_{i=1}^t \epsilon_i) = \sum_{i=1}^t \underbrace{\mathbb{V}\text{ar}(\epsilon_i)}_{=\sigma_t^2} = t\sigma_2$$

In [34]:

```
T = 200                           # Set the number of time periods (length of the series)
ex.e = rep(0, T)                  # Create an empty vector of length T to store the simul
beta0 = 0.5                       # Constant drift term (shifts the process upward each s
beta1 = 0.9                       # Coefficient on deterministic trend (linear trend over
set.seed(123)                     # Fix the random seed for reproducibility
e = rnorm(T, 0, 0.2)              # Generate T random errors from N(0, 0.2^2)
t = 1:T                           # Define time index from 1 to T

# Loop from the 2nd observation onwards
# Each value = yesterday's value + constant drift + time trend + random shock
for (t in 2:T) {
  ex.e[t] = beta0 + beta1 * t + ex.e[t-1] + e[t]
}

# Set plotting options: figure size
options(repr.plot.width=16, repr.plot.height=8)

# Set margins and layout: larger margins, 1 row and 2 plots side by side
par(mar = c(5, 5, 5, 5), mfrow = c(1, 2))

# Plot the entire random walk with drift and trend
plot(ex.e, type="l",
     col="firebrick1", cex = 2.5, cex.lab=2.5, cex.axis=2.5, cex.main=2.5, cex.sub=2.5
```

```
     xlab="Point in time", ylab="y", lwd=3)

# Plot only the first 10 observations (zoom in to see short-run fluctuations)
plot(c(1:10), ex.e[1:10], type="l",
     col="firebrick1", cex = 2.5, cex.lab=2.5, cex.axis=2.5, cex.main=2.5, cex.sub=2.5
     xlab="Point in time", ylab="y", lwd=3)
```



Monte-Carlo simulations

Comparison of processes with stochastic and deterministic trends

Example b: y_t as a function of time t and y_{t-1} (i.e., yesterday's value of y_t)

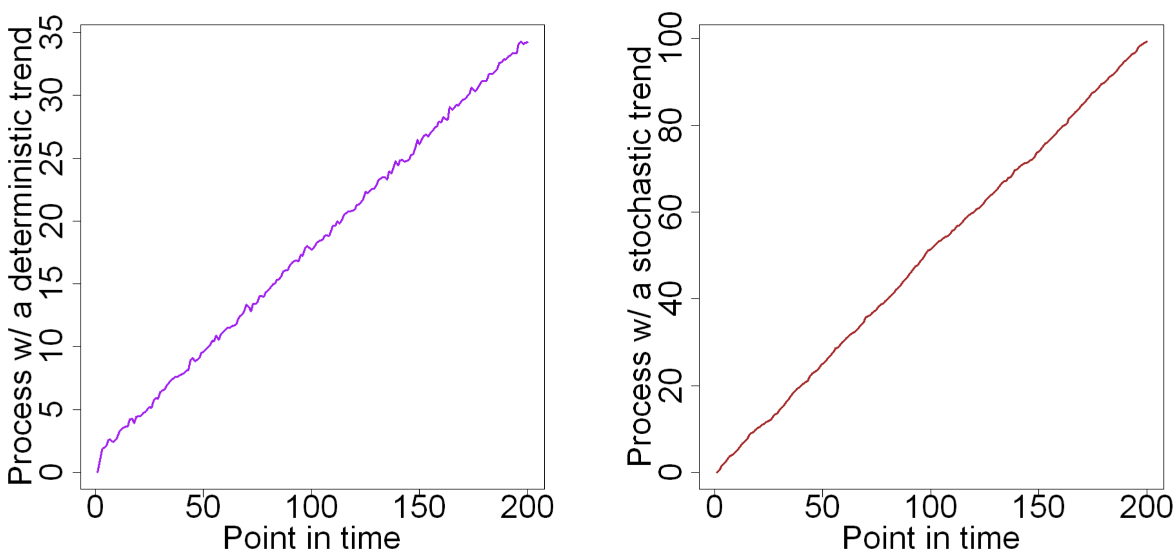
$$y_t = \beta_0 + \beta_1 y_{t-1} + \beta_2 t + \epsilon_t, \quad |\beta_1| < 1$$

Example d: Random walk with constant

$$y_t = \beta_0 + y_{t-1} + \epsilon_t$$

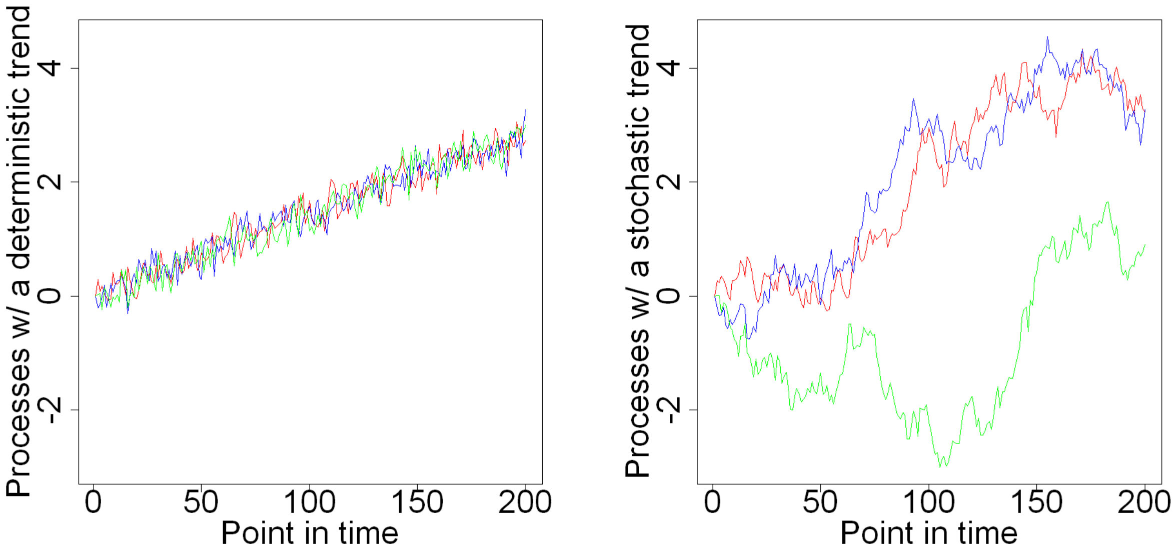
In [15]:
```
# Set plotting options: figure size for the output
options(repr.plot.width=16, repr.plot.height=8)

# Set margins and layout
# mar = margin sizes (bottom, left, top, right)
# mfrow = number of plots (1 row, 2 columns → two plots side by side)
par(mar = c(5, 5, 5, 5), mfrow = c(1, 2))

# ------------------------
# Plot process with a deterministic trend (Example b)
# ------------------------
```

```
plot(ex.b, type="l",                    # Plot as a line ("l")
     col="purple",                      # Line color = purple
     lwd=3,                             # Line width = 3
     cex.lab=2.5, cex.axis=2.5,         # Scale up axis labels and tick labels
     xlab="Point in time",              # X-axis label
     ylab="Process w/ a deterministic trend")   # Y-axis label

# ------------------------
# Plot process with a stochastic trend (Example d)
# ------------------------
plot(ex.d, type="l",                    # Plot as a line ("l")
     col="brown",                       # Line color = brown
     lwd=3,                             # Line width = 3
     cex.lab=2.5, cex.axis=2.5,         # Scale up axis labels and tick labels
     xlab="Point in time",              # X-axis label
     ylab="Process w/ a stochastic trend")  # Y-axis label
```



In [14]:
```
T = 200                             # Number of time periods

# Initialize empty vectors for three deterministic-trend processes
d1 = rep(0, T)
d2 = rep(0, T)
d3 = rep(0, T)

# Initialize empty vectors for three stochastic-trend processes
s1 = rep(0, T)
s2 = rep(0, T)
s3 = rep(0, T)

# Parameters
beta0 = 0.01                        # Constant drift term
beta1 = 0.3                         # Autoregressive coefficient (used in deterministic tre
beta2 = 0.01                        # Deterministic time trend slope

# Random error terms for each process (different shocks for each)
e1 = rnorm(T, 0, 0.2)
e2 = rnorm(T, 0, 0.2)
e3 = rnorm(T, 0, 0.2)

t = 1:T                             # Time index
```

```r
# Loop to generate the processes
for (i in 2:T) {
  # Deterministic trend processes: AR(1) with drift + time trend + noise
  d1[i] = beta0 + beta1 * d1[i-1] + beta2 * t[i] + e1[i]
  d2[i] = beta0 + beta1 * d2[i-1] + beta2 * t[i] + e2[i]
  d3[i] = beta0 + beta1 * d3[i-1] + beta2 * t[i] + e3[i]

  # Stochastic trend processes: random walk with drift + noise
  s1[i] = beta0 + s1[i-1] + e1[i]
  s2[i] = beta0 + s2[i-1] + e2[i]
  s3[i] = beta0 + s3[i-1] + e3[i]
}

# Set plotting options: figure size
options(repr.plot.width=16, repr.plot.height=8)

# Set margins and layout: 1 row, 2 plots side by side
par(mar = c(5, 5, 5, 5), mfrow = c(1, 2))

# Find min and max across all series to set same y-axis scale
minv = min(d1, d2, d3, s1, s2, s3)
maxv = max(d1, d2, d3, s1, s2, s3)

# ------------------------
# Plot deterministic trend processes
# ------------------------
plot(d1, type="l", col="red", lwd=1,
     cex = 2.5, cex.lab=2.5, cex.axis=2.5, cex.main=2.5, cex.sub=2.5,
     xlab= "Point in time",
     ylab="Processes w/ a deterministic trend",
     ylim=c(minv, maxv))   # Use consistent y-axis
lines(d2, col="blue", lwd=1)   # Add second deterministic trend process
lines(d3, col="green", lwd=1)  # Add third deterministic trend process

# ------------------------
# Plot stochastic trend processes
# ------------------------
plot(s1, type="l", col="red", lwd=1,
     cex = 2.5, cex.lab=2.5, cex.axis=2.5, cex.main=2.5, cex.sub=2.5,
     xlab= "Point in time",
     ylab="Processes w/ a stochastic trend",
     ylim=c(minv, maxv))       # Use consistent y-axis
lines(s2, col="blue", lwd=1)   # Add second stochastic trend process
lines(s3, col="green", lwd=1)  # Add third stochastic trend process
```



Remedy for non-stationarity caused by deterministic trend

Solution: Detrending

Run a regression of the data on the deterministic trend (e.g., a constant plus time index) to estimate the trend and remove it from the data.

Proof Example a):

$$\begin{aligned}\tilde{y}_t \equiv y_t - \hat{y} &= \beta_0 + \beta_1 t + \epsilon_t - \left(\hat{\beta}_0 + \hat{\beta}_1 t\right) \\ &= \beta_0 - \hat{\beta}_0 + t(\beta_1 - \hat{\beta}_1) + \epsilon_t\end{aligned}$$

Under standard OLS assumptions,

$$\hat{\beta} \xrightarrow{p} \beta,$$

so

$$(\hat{\beta}_0 - \beta_0) + (\hat{\beta}_1 - \beta_1)t \xrightarrow{p} 0 \quad \text{for each fixed } t.$$

Thus

$$\tilde{y}_t \xrightarrow{p} \epsilon_t,$$

and the detrended series converges to the stationary noise.

Detrending using estimated coefficients yields residuals that, in large samples, behave like the stationary error process ϵ_t.

Proof Example b):

$$\begin{aligned}\tilde{y}_t \equiv y_t - \hat{y} &= \beta_0 + \beta_1 t + \beta_2 y_{t-1} + \epsilon_t - \left(\hat{\beta}_0 + \hat{\beta}_1 t + \hat{\beta}_2 y_{t-1}\right) \\ &= \beta_0 - \hat{\beta}_0 + t(\beta_1 - \hat{\beta}_1) + y_{t-1}(\beta_2 - \hat{\beta}_2) + \epsilon_t, \quad |\beta_2| < 1\end{aligned}$$

Under standard OLS assumptions,

$$\hat{\beta} \xrightarrow{p} \beta,$$

so

$$(\hat{\beta}_0 - \beta_0) + (\hat{\beta}_1 - \beta_1)t + (\beta_2 - \hat{\beta}_2)y_{t-1} \xrightarrow{p} 0 \quad \text{for each fixed } t.$$

Thus

$$\tilde{y}_t \xrightarrow{p} \epsilon_t,$$

and the detrended series converges to the stationary noise.

Detrending using estimated coefficients yields residuals that, in large samples, behave like the stationary error process ϵ_t.

In [27]:
```
# Detrend using R's built-in detrending.
library(pracma)
ex.a.detrend = detrend(ex.a, 'linear')
```

Alternatively:

1. Regress y_t on time:

$$y_t = \alpha_0 + \alpha_1 t + u_t$$

2. Store estimated coefficients:

$$\{\tilde{\alpha}_0, \tilde{\alpha}_1\}$$

3. Compute predicated values of y_t:

$$\tilde{y}_t = \tilde{\alpha}_0 + \tilde{\alpha}_1 t$$

4. Compute model residulas:

$$\tilde{u}_t = y_t - \tilde{y}_t$$

The estimated model residuals are detrended values of y_t.

In [28]:

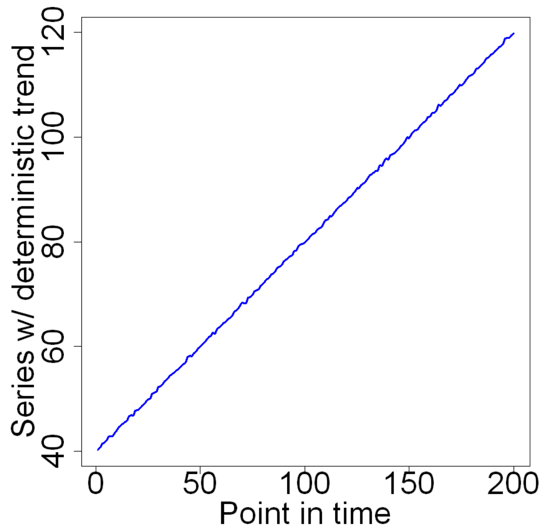
```
# Detrend manually
# Fit a linear regression of ex.a on time t
model = lm(ex.a ~ t)

# Extract estimated coefficients (intercept and slope)
coef = model$coefficients

# Manually detrend: subtract fitted intercept and slope*t from the series
ex.a.lm = ex.a - coef[1] - t * coef[2]
# Compare the first few values of the two approaches (should be identical)
data.frame(head(ex.a.detrend), head(ex.a.lm))
```

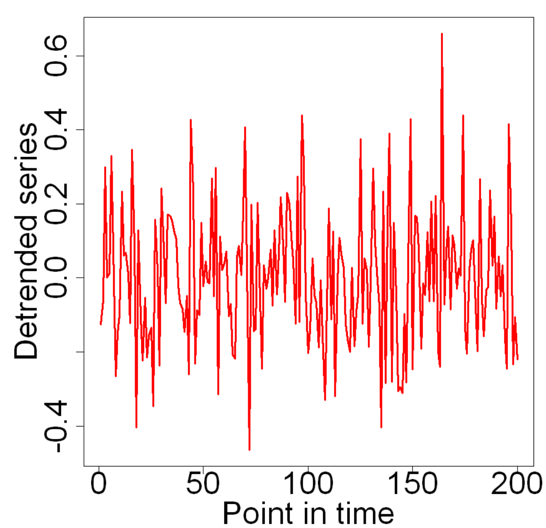
A data.frame: 6 × 2

head.ex.a.detrend.	head.ex.a.lm.
<dbl>	<dbl>
-0.1257143849	-0.1257143849
-0.0595006495	-0.0595006495
0.2984306152	0.2984306152
0.0009447346	0.0009447346
0.0128547073	0.0128547073
0.3301642616	0.3301642616

In [29]:

```
# Set plotting options: figure size for output
options(repr.plot.width=16, repr.plot.height=8)

# Set margins and layout:
# mar = margins (bottom, left, top, right)
# mfrow = number of plots (1 row, 2 columns → two plots side by side)
par(mar = c(5, 5, 5, 5), mfrow = c(1,2))

# ------------------------
# Plot the original series (with deterministic trend)
# ------------------------
plot(ex.a, type="l", col="blue", lwd=3,
     cex = 2.5, cex.lab=2.5, cex.axis=2.5,  # Scale labels and axes
     cex.main=2.5, cex.sub=2.5,             # Scale title/subtitle
     xlab= "Point in time",                 # X-axis label
     ylab="Series w/ deterministic trend")  # Y-axis label

# ------------------------
# Plot the detrended series (residuals after removing linear trend)
# ------------------------
plot(ex.a.detrend, type="l", col="red", lwd=3,
     cex = 2.5, cex.lab=2.5, cex.axis=2.5,
     cex.main=2.5, cex.sub=2.5,
     xlab= "Point in time",
     ylab="Detrended series")
```





Remedy for non-stationarity caused by stochastic trend

Solution: Differentiating

Transform the series into the differences between every two consecutive observations

$$\Delta y_t = y_t - y_{t-1}$$

t	yt	Lyt	Δ yt
1	y1		
2	y2	y1	y2-y1
3	y3	y2	y3-y2
4	y4	y3	y4-y3
5	y5	y4	y5-y4
6	y6	y5	y6-y5

Proof Example c): Random walk without constant can be rewritten as: $y_t = y_0 + \sum_{i=1}^{t} \epsilon_i, \quad \epsilon_t \sim (0, \sigma_\epsilon^2)$ The first difference is defined as: $\Delta y_t = y_t - y_{t-1} = y_0 + \sum_{i=1}^{t} \epsilon_i - \left(y_0 + \sum_{i=1}^{t-1} \epsilon_i\right) = \epsilon_t$ $\mathbb{E}(\Delta y_t) = \mathbb{E}(\epsilon_t) = 0$ $\mathbb{V}\mathrm{ar}(\Delta y_t) = \mathbb{V}\mathrm{ar}(\epsilon_t) = \sigma_\epsilon^2$ We see that the differenced series has constant mean and variance.

Proof Example d): Random walk with constant can be rewritten as: $y_t = y_0 + t\beta_0 + \sum_{i=1}^{t} \epsilon_i, \quad \epsilon_t \sim (0, \sigma_\epsilon^2)$ Then, it follows that:

$\Delta y_t = y_t - y_{t-1} = y_0 + t\beta_0 + \sum_{i=1}^t \epsilon_i - \left(y_0 + (t-1)\beta_0 + \sum_{i=1}^{t-1} \epsilon_i\right) = \beta + \epsilon_t$
$\mathbb{E}(\Delta y_t) = \mathbb{E}(\beta + \epsilon_t) = \beta$ $\mathbb{V}\mathrm{ar}(\Delta y_t) = \mathbb{V}\mathrm{ar}(\beta + \epsilon_t) = \sigma_\epsilon^2$ We see that the differenced series has constant mean and variance.

In [20]:

```
# Take first differences using R’s built-in differencing
ex.c.diff = diff(ex.c)
```

In [21]:

```
# Compute the differences manually
T = length(ex.c)  # Length of the original series

c1 = ex.c[1:T-1]          # Create a vector of the first T-1 observations (lagged
c2 = ex.c[2:T]            # Create a vector of the last T-1 observations (current

ex.c.alt = c2 - c1        # Manually compute differences: y_t - y_{t-1}

# Compare the first few values of the two approaches (should be identical)
data.frame(head(ex.c.diff), head(ex.c.alt))
```

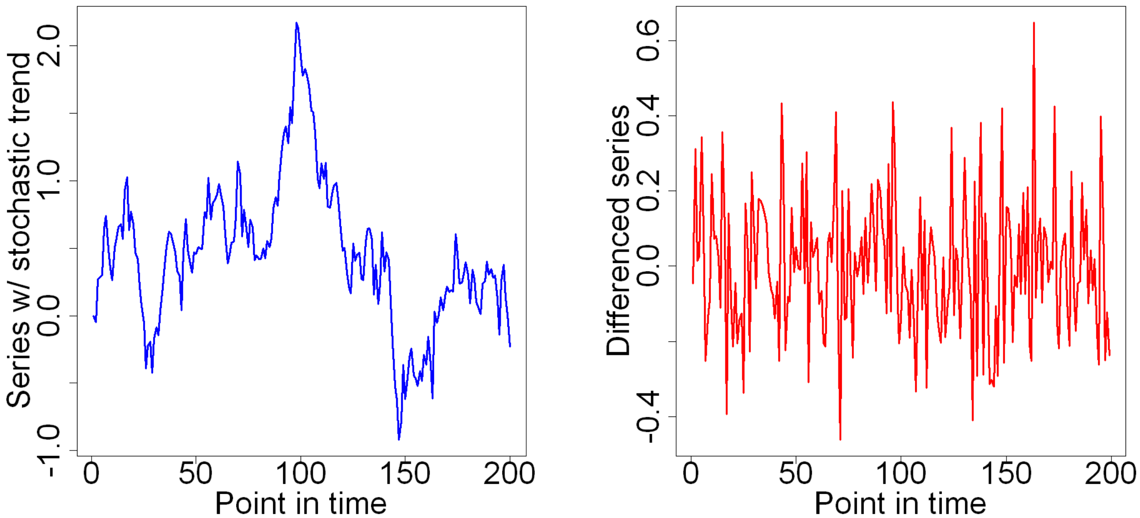
A data.frame: 6 × 2

head.ex.c.diff.	head.ex.c.alt.
<dbl>	<dbl>
-0.04603550	-0.04603550
0.31174166	0.31174166
0.01410168	0.01410168
0.02585755	0.02585755
0.34301300	0.34301300
0.09218324	0.09218324

In [22]:

```
# Set plotting options: figure size for output
options(repr.plot.width=16, repr.plot.height=8)

# Set margins and layout:
# mar = margins (bottom, left, top, right)
# mfrow = number of plots (1 row, 2 columns → two plots side by side)
par(mar = c(5, 5, 5, 5), mfrow = c(1,2))

# ------------------------
# Plot the original series (ex.c: random walk without constant)
# ------------------------
plot(ex.c, type="l",
     col="blue",                     # Line color = blue
     lwd=3,                          # Line width
     cex.lab=2.5, cex.axis=2.5,      # Scale up axis labels and tick labels
     xlab= "Point in time",          # X-axis label
     ylab="Series w/ stochastic trend")  # Y-axis label

# ------------------------
# Plot the differenced series (ex.c.diff = Δy_t)
# ------------------------
plot(ex.c.diff, type="l",
     col="red",                      # Line color = red
```

```
    lwd=3,                          # Line width
    cex.lab=2.5, cex.axis=2.5,      # Scale up axis labels and tick labels
    xlab= "Point in time",          # X-axis label
    ylab="Differenced series")      # Y-axis label
```



Can differencing fix the problem of a deterministic trend?

If the data generating process is of the form (example e),

$$y_t = \beta_0 + \beta_1 t + y_{t-1} + \epsilon_t,$$

i.e., contains both deterministic and stochastic trends, differentiating the series will not eliminate the deterministic trend component.

$\Delta y_t = y_t - y_{t-1}$

$\Delta y_t = \beta_0 + \beta_1 t + y_{t-1} + \epsilon_t - y_{t-1}$

$\Delta y_t = \beta_0 + \beta_1 \color{red}{t} + \epsilon_t$

If the data generating process is of the form (example a),

$$y_t = \beta_0 + \beta_1 t + \epsilon_t,$$

i.e., contains only deterministic trend, differentiating the series will eliminate the deterministic trend component.

$\Delta y_t = y_t - y_{t-1}$

$\Delta y_t = \beta_0 + \beta_1 t + \epsilon_t - (\beta_0 + \beta_1 (t-1) + \epsilon_{t-1})$

$\Delta y_t = \beta_1 + \epsilon_t + \epsilon_{t-1}$

However, differentiating a process that is only subject to a deterministic trend leads to **over-differentiation**: instead of the true shocks (ϵ_t), we now have $(\Delta\epsilon_t)$. This throws away information and distorts the dynamics.

Takeaways

- **Trend-stationary (only deterministic trend):**
  → Detrend. Differencing would destroy useful information.

- **Difference-stationary (stochastic trend, possibly with deterministic trend):**
  → Difference once to remove the unit root. If a deterministic trend remains, detrend after differencing.

- **Never difference twice unless the series is truly I(2).**
  Over-differencing throws away information and alters the dynamics.

Augmented Dickey–Fuller (ADF)

Testing for a unit root using ADF requires manowuverung between the following three regression equations:

(A) $\Delta y_t = \alpha + \beta t + \gamma y_{t-1} + \sum_{i=1}^{p} \delta_i \Delta y_{t-i} + \epsilon_t$

(B) $\Delta y_t = \alpha + \gamma y_{t-1} + \sum_{i=1}^{p} \delta_i \Delta y_{t-i} + \epsilon_t$

(C) $\Delta y_t = \gamma y_{t-1} + \sum_{i=1}^{p} \delta_i \Delta y_{t-i} + \epsilon_t$

Enders' Iterative Procedure

The **φ-tests** are *joint tests* that always include $\gamma = 0$ (the unit root null).
They are not "pure deterministic term tests."

- **φ_3:** $H_0 : \gamma = 0 \text{ and } \beta = 0$
- **φ_2:** $H_0 : \gamma = 0 \text{ and } \alpha = 0 \text{ and } \beta = 0$
- **φ_1:** $H_0 : \gamma = 0 \text{ and } \alpha = 0$

1. **Start with model (A)** (constant + trend):

$$\Delta y_t = \alpha + \beta t + \gamma y_{t-1} + \sum_{i=1}^{p} \delta_i \Delta y_{t-i} + \varepsilon_t$$

- Compute **τ_3** (tests $H_0 : \gamma = 0$).
- Compute **φ_2** and **φ_3**:
  - **φ_3:** if you **reject**, keep the trend.
  - If you **cannot reject φ_3**, test φ_2:
    - **φ_2 rejected** → drop trend, keep constant → go to model (B).
    - **φ_2 not rejected** → drop both trend and constant → go to model (C).

1. **In the chosen model (A, B, or C):**
   - Perform the **τ-test** on $H_0 : \gamma = 0$ using the correct critical values.
   - If τ rejects → the series is **stationary** (possibly around a mean or a trend).
   - If τ does not reject → the series has a **unit root** (stochastic trend).

ADF test in R

Syntax

```
library(urca)
summary(ur.df(ex.a, type='drift', lags=8, selectlags="AIC"))
```

type = "trend", includes $\alpha + \beta t + \gamma y_{t-1}$

```
tau3: test $H_0: \gamma = 0$

phi2: test $H_0: \gamma = \alpha = \beta = 0$

phi3: test $H_0: \gamma = \beta = 0$
```

type = "drift", includes $\alpha + \gamma y_{t-1}$

```
tau2: test $H_0: \gamma = 0$

phi1: test $H_0: \gamma = \alpha = 0$
```

type = "none", includes γy_{t-1}

```
tau1: test $H_0: \gamma = 0$
```

Lag length (lags)

```
lags = 1 → include up to 1 lag of $\Delta y_{t-i}$

lags = 8 → include up to 8 lags

lags = k → include up to $k$ lags
```

Lag selection (selectlags)

```
    selectlags = "AIC" → AIC chooses the optimal lag length between
0 and lags

    selectlags = "BIC" → BIC chooses the optimal lag length between
0 and lags
```

φ-tests in the ADF framework

- **φ₁** (only relevant in the **drift** model, i.e. `type = "drift"`):
  - Tests **H_0: α = 0** (no constant).
  - If rejected → keep the constant.
- **φ₂** (relevant in the **trend** model, i.e. `type = "trend"`):
  - Tests **H_0: α = 0 and β = 0** (no constant, no trend).
  - If rejected → you need at least a constant.
- **φ₃** (also relevant in the **trend** model):
  - Tests **H_0: β = 0** (no trend).
  - If rejected → you need the trend.

Testing for a unit root in R

Example a:

$$y_t = \beta_0 + \beta_1 y_{t-1} + \beta_2 t + \epsilon_t$$

In [33]:
```
library(urca)
summary(ur.df(ex.a, type='trend', lags=8, selectlags="AIC"))
```

```
###############################################
# Augmented Dickey-Fuller Test Unit Root Test #
###############################################

Test regression trend


Call:
lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.45413 -0.12172 -0.01179  0.12125  0.59979 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 46.73937    6.16594   7.580 1.61e-12 ***
z.lag.1     -1.16065    0.15648  -7.417 4.18e-12 ***
tt           0.46412    0.06259   7.416 4.22e-12 ***
z.diff.lag1  0.10125    0.13637   0.742   0.4587    
z.diff.lag2  0.02452    0.10662   0.230   0.8184    
z.diff.lag3  0.12064    0.07268   1.660   0.0986 .  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1875 on 185 degrees of freedom
Multiple R-squared:  0.5508,	Adjusted R-squared:  0.5387 
F-statistic: 45.37 on 5 and 185 DF,  p-value: < 2.2e-16


Value of test-statistic is: -7.4174 88.3098 27.612 

Critical values for test statistics: 
      1pct  5pct 10pct
tau3 -3.99 -3.43 -3.13
phi2  6.22  4.75  4.07
phi3  8.43  6.49  5.47
```

We started with the most general ADF regression (including constant and trend). The φ-tests reject the hypotheses that we can drop the constant or the trend, so both terms are required. The τ_3 test rejects the null of a unit root, meaning the process is not difference-stationary but instead trend-stationary. This implies that once we remove the deterministic linear trend, the remaining fluctuations are stationary. In other words, the series wanders around a predictable upward/downward sloping path, rather than following a stochastic trend.