Augmented Solow Model - Quantitative Similation

Tomas Oles

2025-03-02

```r
# Load necessary libraries
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.2.3
```

```r
library(gridExtra)
```

Parametrize the Solow model per Effective Unit of Labor

```r
# Parameters
alpha <- 0.33
s1 <- 0.2
s2 <- 0.3
A <- 1
delta <- 0.1
z <- 0.02
n <- 0.01

# Steady state capital per effective unit of labor
k_star <- (s1 * A / ((1 + z) * (1 + n) - (1 - delta))) ** (1 / (1 - alpha))

# Time periods
T <- 50
k_hat <- numeric(T)
k_hat[1] <- k_star

# Arrays to store the results
y_hat <- numeric(T)
c_hat <- numeric(T)
i_hat <- numeric(T)
R_t <- numeric(T)
w_t <- numeric(T)
```

Initial values for t=0

```r
y_hat[1] <- A * k_hat[1] ** alpha
c_hat[1] <- (1 - s1) * A * k_hat[1] ** alpha
i_hat[1] <- s1 * A * k_hat[1] ** alpha
R_t[1] <- alpha * A * k_hat[1] ** (alpha - 1)
w_t[1] <- (1 - alpha) * A * k_hat[1] ** alpha
```

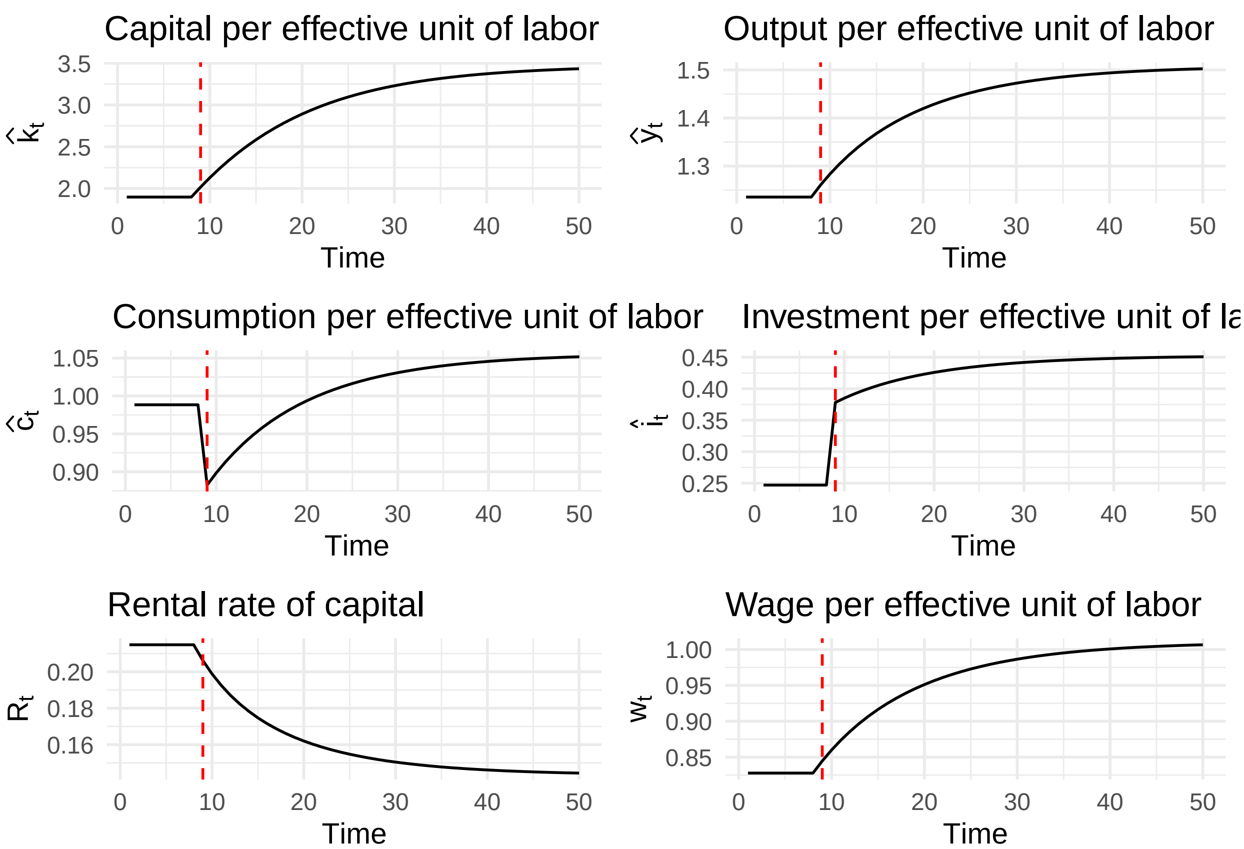
Iteratively calculate the values for each period

```
for (t in 2:T) {
  if (t < 9) {
    s <- s1
  } else {
    s <- s2
  }

  k_hat[t] <- (1 / ((1 + z) * (1 + n))) * (s * A * k_hat[t - 1] ** alpha + (1 - delta) * k_hat[t - 1])
  y_hat[t] <- A * k_hat[t] ** alpha
  c_hat[t] <- (1 - s) * A * k_hat[t] ** alpha
  i_hat[t] <- s * A * k_hat[t] ** alpha
  R_t[t] <- alpha * A * k_hat[t] ** (alpha - 1)
  w_t[t] <- (1 - alpha) * A * k_hat[t] ** alpha
}
```

Plot the results

```
time <- 1:T

p1 <- ggplot(data.frame(time, k_hat), aes(x = time, y = k_hat)) +
  geom_line() +
  geom_vline(xintercept = 9, color = "red", linetype = "dashed") +
  labs(title = "Capital per effective unit of labor", x = "Time", y = expression(widehat(k)[t])) +
  theme_minimal()

p2 <- ggplot(data.frame(time, y_hat), aes(x = time, y = y_hat)) +
  geom_line() +
  geom_vline(xintercept = 9, color = "red", linetype = "dashed") +
  labs(title = "Output per effective unit of labor", x = "Time", y = expression(widehat(y)[t])) +
  theme_minimal()

p3 <- ggplot(data.frame(time, c_hat), aes(x = time, y = c_hat)) +
  geom_line() +
  geom_vline(xintercept = 9, color = "red", linetype = "dashed") +
  labs(title = "Consumption per effective unit of labor", x = "Time", y = expression(widehat(c)[t])) +
  theme_minimal()

p4 <- ggplot(data.frame(time, i_hat), aes(x = time, y = i_hat)) +
  geom_line() +
  geom_vline(xintercept = 9, color = "red", linetype = "dashed") +
  labs(title = "Investment per effective unit of labor", x = "Time", y = expression(widehat(i)[t])) +
  theme_minimal()

p5 <- ggplot(data.frame(time, R_t), aes(x = time, y = R_t)) +
  geom_line() +
  geom_vline(xintercept = 9, color = "red", linetype = "dashed") +
  labs(title = "Rental rate of capital", x = "Time", y = expression(R[t])) +
  theme_minimal()

p6 <- ggplot(data.frame(time, w_t), aes(x = time, y = w_t)) +
  geom_line() +
```

```r
  geom_vline(xintercept = 9, color = "red", linetype = "dashed") +
  labs(title = "Wage per effective unit of labor", x = "Time", y = expression(w[t])) +
  theme_minimal()

# Arrange plots in a grid
grid.arrange(p1, p2, p3, p4, p5, p6, ncol = 2)
```



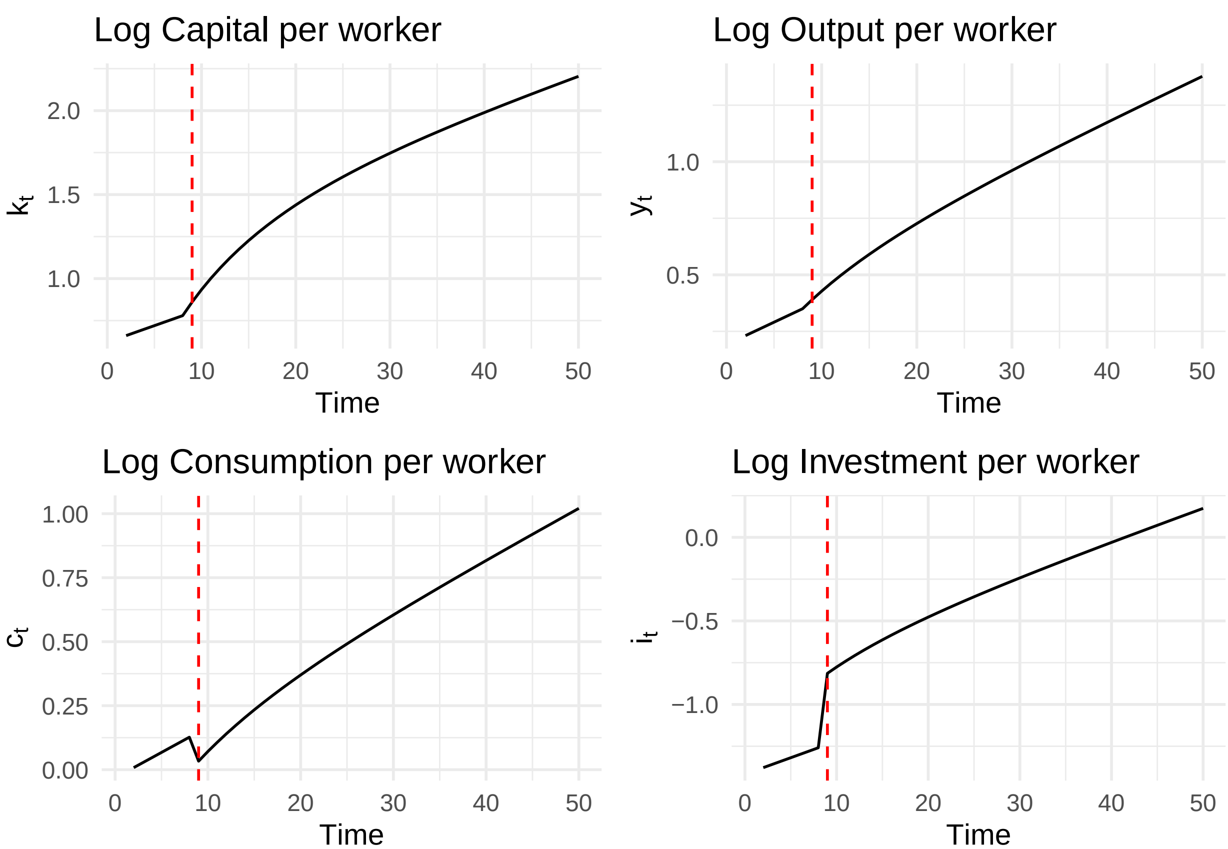
Parametrize the Solow Model per Worker

```r
rm(list = ls())
# Parameters
alpha <- 0.33
s1 <- 0.2
s2 <- 0.3
A <- 1
delta <- 0.1
z <- 0.02
n <- 0.01

# Steady state capital per effective unit of labor
k_star <- (s1 * A / ((1 + z) * (1 + n) - (1 - delta))) ** (1 / (1 - alpha))

# Time periods
T <- 50
k_hat <- numeric(T)
```

```r
k_hat[1] <- k_star

# Arrays to store the results
y_hat <- numeric(T)
c_hat <- numeric(T)
i_hat <- numeric(T)
R_t <- numeric(T)
w_t <- numeric(T)

# Arrays to store per worker variables
k <- numeric(T)
y <- numeric(T)
c <- numeric(T)
i <- numeric(T)

Z_t <- 1  # Initial level of technology

# Calculate per worker variables for t=0
k[1] <- k_hat[1] * Z_t
y[1] <- y_hat[1] * Z_t
c[1] <- c_hat[1] * Z_t
i[1] <- i_hat[1] * Z_t


# Iteratively calculate the values for each period
for (t in 2:T) {
  if (t < 9) {
    s <- s1
  } else {
    s <- s2
  }

  k_hat[t] <- (1 / ((1 + z) * (1 + n))) * (s * A * k_hat[t - 1] ** alpha + (1 - delta) * k_hat[t - 1])
  y_hat[t] <- A * k_hat[t] ** alpha
  c_hat[t] <- (1 - s) * A * k_hat[t] ** alpha
  i_hat[t] <- s * A * k_hat[t] ** alpha
  R_t[t] <- alpha * A * k_hat[t] ** (alpha - 1)
  w_t[t] <- (1 - alpha) * A * k_hat[t] ** alpha

  Z_t <- Z_t * (1 + z)  # Update the level of technology

  # Calculate per worker variables
  k[t] <- log(k_hat[t] * Z_t)
  y[t] <- log(y_hat[t] * Z_t)
  c[t] <- log(c_hat[t] * Z_t)
  i[t] <- log(i_hat[t] * Z_t)
}

# Replace first observation in each array with NA
k[1] <- NA
y[1] <- NA
c[1] <- NA
i[1] <- NA
```

Plot the dynamic paths per worker variables

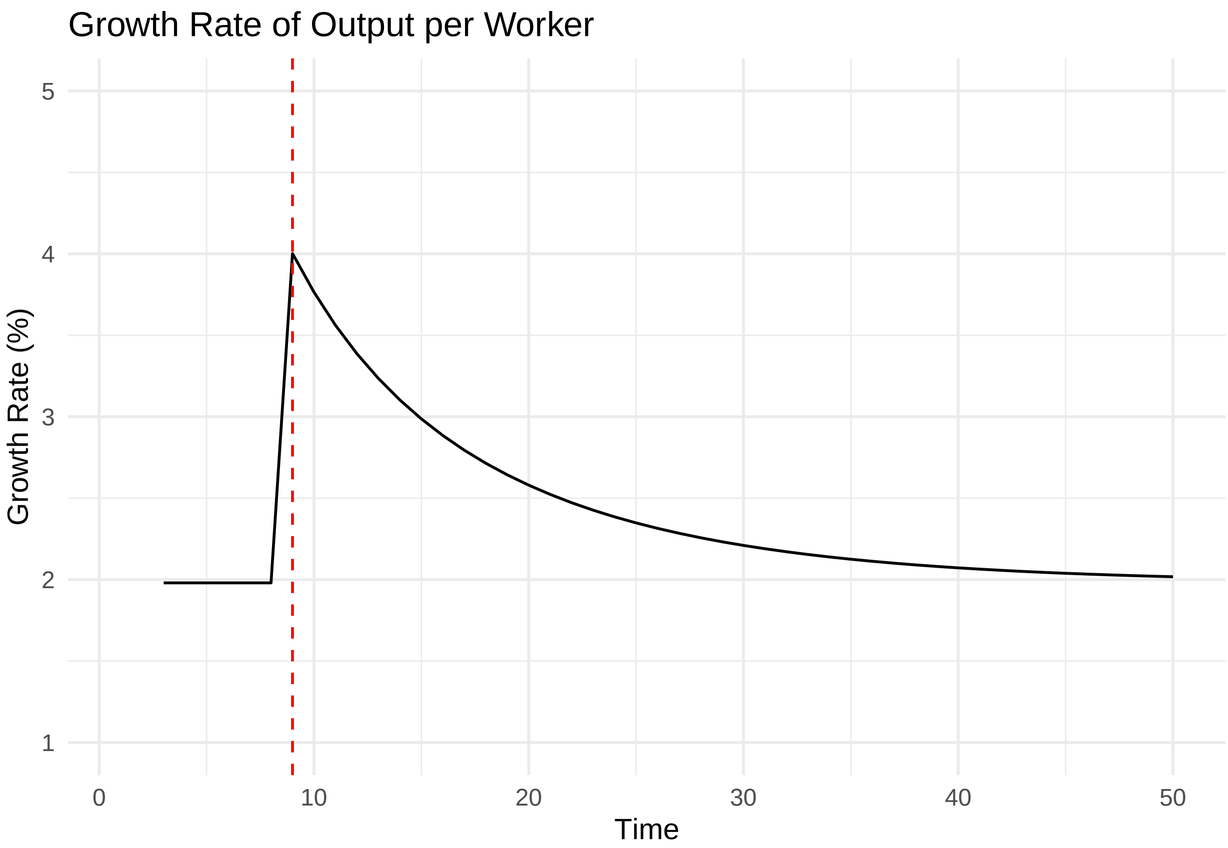
```
time <- 1:T

p1 <- ggplot(data.frame(time, k), aes(x = time, y = k)) +
  geom_line() +
  geom_vline(xintercept = 9, color = "red", linetype = "dashed") +
  labs(title = "Log Capital per worker", x = "Time", y = expression(k[t])) +
  theme_minimal()

p2 <- ggplot(data.frame(time, y), aes(x = time, y = y)) +
  geom_line() +
  geom_vline(xintercept = 9, color = "red", linetype = "dashed") +
  labs(title = "Log Output per worker", x = "Time", y = expression(y[t])) +
  theme_minimal()

p3 <- ggplot(data.frame(time, c), aes(x = time, y = c)) +
  geom_line() +
  geom_vline(xintercept = 9, color = "red", linetype = "dashed") +
  labs(title = "Log Consumption per worker", x = "Time", y = expression(c[t])) +
  theme_minimal()

p4 <- ggplot(data.frame(time, i), aes(x = time, y = i)) +
  geom_line() +
  geom_vline(xintercept = 9, color = "red", linetype = "dashed") +
  labs(title = "Log Investment per worker", x = "Time", y = expression(i[t])) +
  theme_minimal()

# Arrange plots in a grid
grid.arrange(p1, p2, p3, p4, ncol = 2)
```



Compute and plot the growth rate per worker output

```
# Array to store the growth rate of per worker output
y_growth <- numeric(T)

# Calculate growth rate of per worker output
for (t in 2:T) {
  y_growth[t] <- (y[t] - y[t-1]) * 100
}

y_growth[1] <- NA

# Plot the growth rate of per worker output
time <- 1:T

ggplot(data.frame(time, y_growth), aes(x = time, y = y_growth)) +
  geom_line() +
  geom_vline(xintercept = 9, color = "red", linetype = "dashed") +
  labs(title = "Growth Rate of Output per Worker", x = "Time", y = "Growth Rate (%)") +
  ylim(1,5)+
  theme_minimal()
```

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_line()`).
```


Growth Rate of Output per Worker
Growth Rate (%)
5
4
3
2
1
0
10
20
30
40
50
Time