

# Solow model

16MA4a

*Kenji Sato mail@kenjisato.jp*

```
library(ggplot2)
```

Model parameters:

```
alpha = 0.3  
delta = 0.05  
g = 0.02  
n = 0.01  
s = 0.4
```

and initial capital per unit of effective labor:

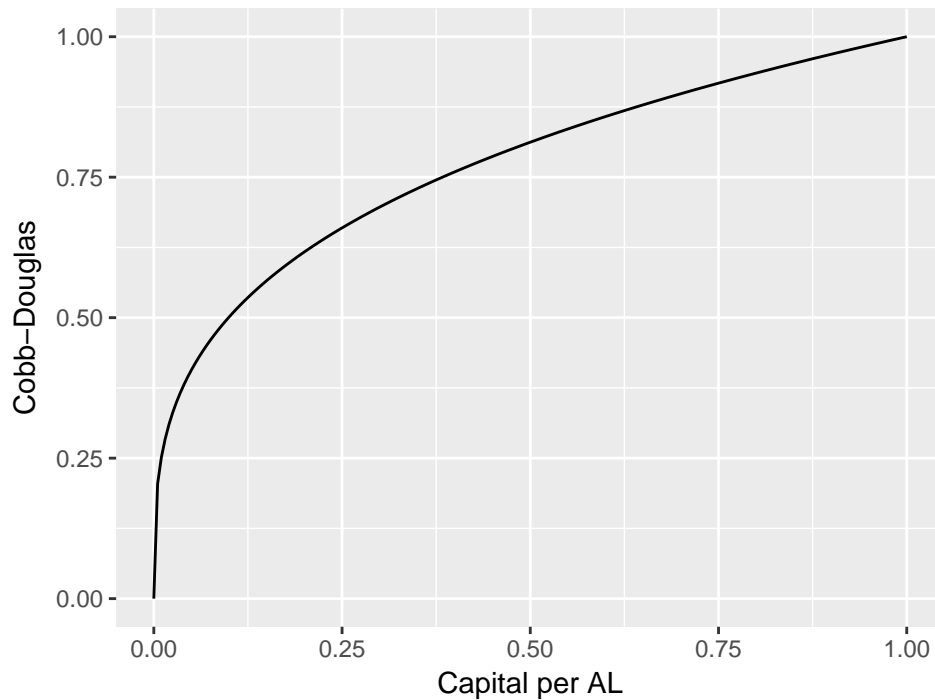
```
k0 = 0.1
```

Cobb–Douglas production function in its intensive form:

```
prod_func = function(k){  
  return(k^alpha)  
}
```

How to visualize?

```
karr = seq(from=0, to=1.0, length.out=200)  
  
qplot(x=karr, y=prod_func(karr), geom='line') +  
  labs(x="Capital per AL", y="Cobb–Douglas")
```



Simulation parameter:

```
dt = 0.01 # controls precision of approximation
t_max = 100 # simulation for t_max years
```

The master differential equation:

$$\dot{k}(t) = sf(k(t)) - (\delta + g + n)k(t)$$

... and its discrete-time approximation:

$$\frac{k(t + \Delta t) - k(t)}{\Delta t} = sf(k(t)) - (\delta + g + n)k(t)$$

or equivalently,

$$k(t + \Delta t) = k(t) + \Delta t[sf(k(t)) - (\delta + g + n)k(t)]$$

Now the code.

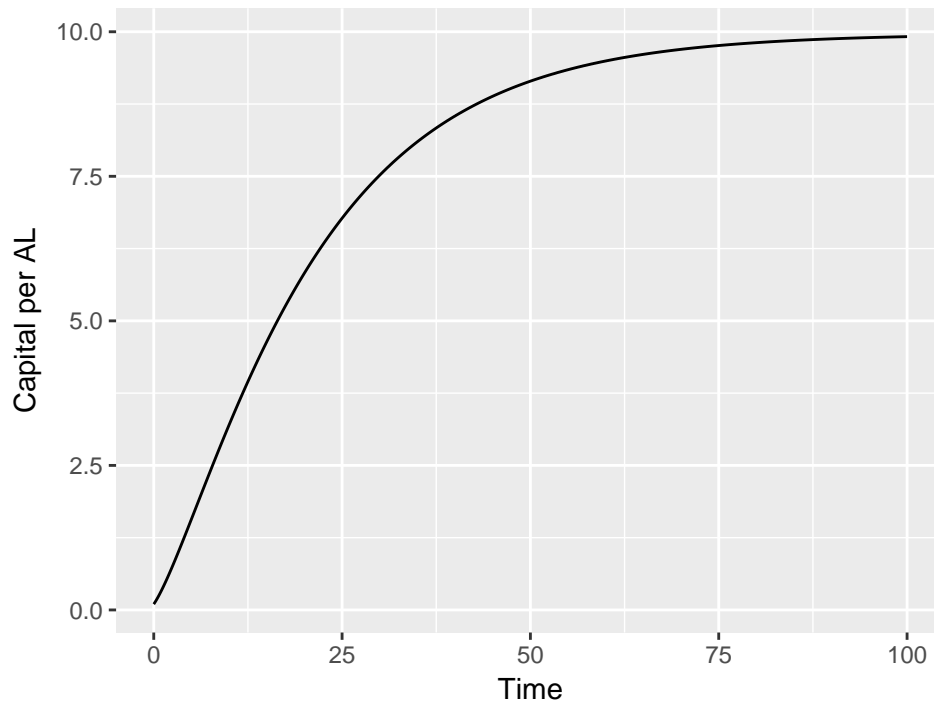
```
solow_update = function(k){
  k + dt * (s * prod_func(k) - (delta + g + n) * k)
}
```

This naive implementation does work.

```
t = seq(from=0, to=t_max, by=dt)
simulation = as.data.frame(t)

simulation[1, "k"] = k0
for (i in 2:nrow(simulation)){
  simulation[i, "k"] = solow_update(simulation[i-1, "k"])
}

ggplot(simulation, aes(x=t, y=k)) +
  geom_line() + labs(x="Time", y="Capital per AL")
```



It seems like the capital per unit of effective labor will eventually converge to a fixed value. You can find this “fixed point” by solving ...

$$0 = sf(k) - (g + n + \delta)k$$

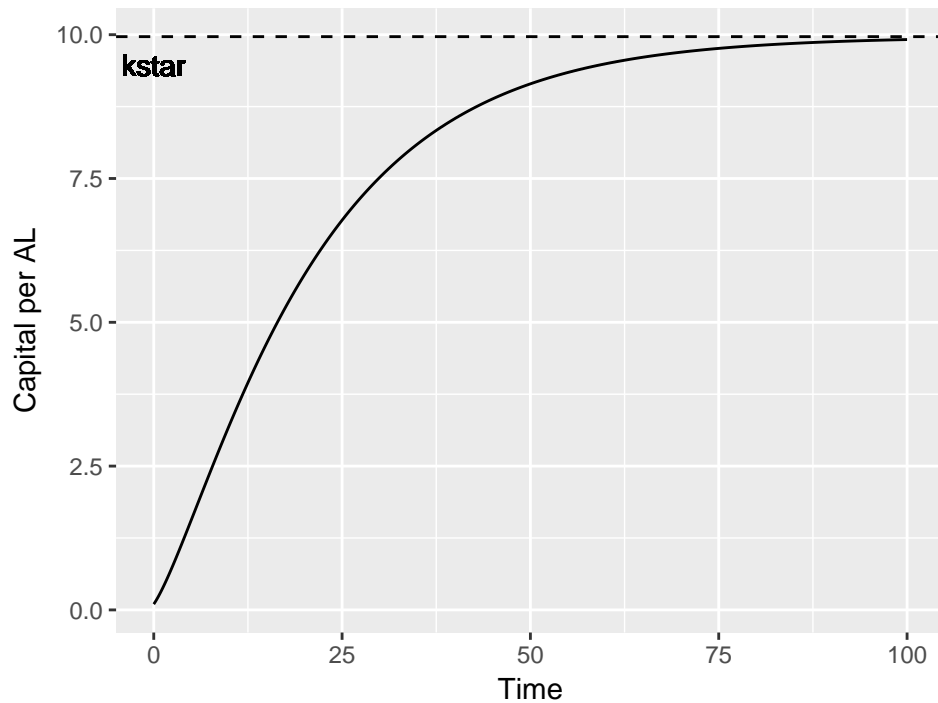
We get

$$k^* = \left( \frac{s}{g + n + \delta} \right)^{\frac{1}{1-\alpha}}$$

Let's check if this is the limit.

```
kstar = (s / (g + n + delta))^(1 / (1 - alpha))
```

```
ggplot(simulation, aes(x=t, y=k)) +
  geom_line() + labs(x="Time", y="Capital per AL") +
  geom_hline(yintercept=kstar, linetype="dashed") +
  geom_text(x=0, y=0.95*kstar, label="kstar")
```



## Exercise

Pick any initial  $k(0)$  above  $k^*$ . Verify that the level of capital per unit of effective labor falls down to  $k^*$ .

## R Notebooks

I wrote this handout with the help of R Notebooks, which is included in a recent update of RStudio. To reproduce this handout yourself, install the latest RStudio and open the source file distributed on my course website.

With R Notebooks (or more generally tools that support literate programming), you can integrate documents, codes and output including figures in one place. You can drastically reduce probability of making such mistakes as

- forgetting to update the graph after making a slight modification to the code, or vice versa,
- forgetting to update the code after changing column names in the data.

After a modification to any part of data or code, you must test whether the integrity of your codes you had is maintained. R Notebooks make it straightforward.