Reproduction of Shaikh's (2016) capacity utilization calculations

Estimating corporate capacity for the US 1947 - 2011

Patrick Mokre

March 15 2019

1 Introduction and Procedure

Shaikh (2016) provides a framework for estimating capacity as a function of the impact of lagged output and capital stock on output in a log-linear dynamic linear regression.

The intuition is that output has two components, one capacity-determined and one path-dependent. In the calculation one seeks to estimate a counterfactual output under full capacity utilization. This amounts to an estimation of the capacity-determined part corrected for the path-dependent component.

In a dynamic linear regression, one treats lagged output and some yearly dummies as the path-dependent impact of capacity over- or under-utilization, and the impact of capital stock and as capacity-determining.

1.1 ADL estimation

The technique follows Pesaran, Shin and Smith's (2001) contribution to the estimation of long run equilibrium estimates with autoregressive distributed lag (ADL) equations, which is used in combination with, and as a substitute for established methods of error correction model (ECM) estimation.

Let an ECM for output Y and capital K be a model of the form:

$$\Delta lnY_t = \alpha + \beta t + \phi_1 lnY_{t-1} + \phi_2 lnK_{t-1} + \sum_{i=1}^{m-1} \gamma_i \Delta lnY_{t-i} + \sum_{j=1}^{n-1} \psi_j \Delta lnK_{t-j} + \epsilon_t$$

The ADL procedure of testing bounds is applicable regardless of the underlying relationship of the variables of interest, the crux however is to establish co-integration in the underlying ECM. The ECM is furthermore used te establish the best fit year dummies. If co-integration is not given, Engle-Granger two-step estimation is proposed.

Let $Y*$ be output at normal capacity utilization, i.e. normal capacity, tt a time trend, Y observed output (corporate value added) and K the estimated gross capital stock, all at constant prices. Furthermore, let DM be a vector with h entries of dummy variables for years in which one assumes a shock in the capital-capacity relationship.

One estimates the logarithmic relationship between output and the covariates.

$$lnY_t = \alpha + \beta t + \sum_h \delta_h DM_h + \sum_m^M \gamma_m lnY_{t-m} + \sum_{n=0}^{N-1} \phi_n lnK_{t-n} + \epsilon_t$$

For an ADL(M, N) model formulation, one extracts coefficients α, β, δ, and ϕ and corrects them for the lagged impact of output, γ to derive long-run parameters a, b, c, and d.

$$a = \frac{\alpha}{1 - \sum_m^M \gamma_m}$$
$$b = \frac{\beta}{1 - \sum_m^M \gamma_m}$$
$$c_h = \frac{\delta_h}{1 - \sum_m^M \gamma_m} \forall h \in H$$
$$d = \frac{sum_n^N \phi_n}{1 - \sum_m^M \gamma_m}$$

And calculate estimated capacity Y^*.

$$lnY_t^* = a + b * t + \sum_h^H c_h DM_h + \sum_{n=0}^{N-1} d_n lnK_{t-n} + \epsilon_t$$

1.2 Steps

The estimation consists of 7 steps:

1. Formulate an error correction model including a large number of lags in both output and capital stock to determine if there is a significant time trend. If the coefficient is not different form zero at the 5 % level, the time trend is not included in subsequent steps.
2. One transforms the residuals of the estimation in Step 1 by dividing their absolute value by their overall standard deviation (call that indicator d), and includes year dummies for years in which d exceeds 2.
3. One runs ECMS including the dummies as well as some range of lags in output and capital stock and records the Akaike Information Criterion (AIC). The model with the lowest AIC is selected.
4. One runs a Breusch-Godfrey Test for serial correlation on the AIC minimizing ECMl from Step 3. Only if the Null Hypothesis of no serial correlation cannot be rejected, is Pesaran-Shin-Smith bounds testing applicable.
5. One compares the Step 3 ECM's F-statistic with the critical values from Pesaran, Shin and Smith (PSS) with a Null hypothesis of no joint long-run relationship between output and capital stock. If the test rejects at the 5 % level, one follows the ADL estimation of long run parameters. If the test fails to reject, the Engle-Granger two-step procedure is suggested. Note that the PSS bounds test can be run for cases with and without a time trend.
6. In a similar procedure like in Step 3, one chooses a lag order for an ADL model by AIC minimization, including the previously chosen year dummies, and including the time trend only if significant in Step 1.
7. On retrieves the parameters from the Step 6 ADL estimation and calculates long-run coefficient point estimates. These are plugged into an estimation of capacity. In contrast to the third equation in Shaikh 2016 (Appendix 6.7.VI, page 854) the year dummies are not included in the calculation.

1.3 Data and Comparison

To run the estimation, the estimates for value added and capital stock need to be normalized by the same price deflator. Intuitively, this would be the price deflator of capital stock (Shaikh 2016, Appendix Table 6.8.I). However, for the calculations in Appendix Table 6.7.14 in Shaikh 2016 (Appendix 6.7.VI, p854) some other price deflator has been used.

I will first reproduce the estimates in Table 6.7.14 in the book, and derive capacity utilization. Then, I apply the same method to calculate capacity using the data on value added adjusted for imputed rents, gross corporate capital stock and the implicit price deflator for BEA capital stock (Appendix Tables 6.8.II.7, 6.8.II.5 and 6.8.II.3 respectively.

For the estimation, I use the Open Source software R and some extensions - `readxl` for reading Microsoft Excel coded data, `dynlm` for time series regressions, `lmtest` for deriving the Akaike Information Criterions in model selection, `nardl` for calculating the Pesaran-Shin-Smith critical values for bounds testing, `dplyr` as well as `reshape2` f

2 Reproduction of Appendix Table 6.7.14

```
rm(list=ls())
setwd("C:/Users/M/Google Drive/NSSR/Shaikh/Capacity Utilization/Reproduction")


library(readxl) #To read Excel files
library(WriteXLS) #To save Excel files
library(lmtest) # For BIC
```

```
## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(dynlm) #For ECM and ARDL
```

```
## Warning: package 'dynlm' was built under R version 3.5.2
```

```
library(nardl) #For PSS Bounds values
```

```
## Warning: package 'nardl' was built under R version 3.5.2
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(reshape2)
library(kableExtra)
```

```
dkeil <- read_excel("__Utilization Output Keil January 2013.xlsx",
                    sheet=3,
                    range="A2:AB67")
tkeil <- ts(dkeil)
```

2.1 Time Trend

```
dynlm(d(l_ys) ~ time + L(l_ys, 1) + L(l_ks, 1) +
                d(L(l_ys, 1)) + d(L(l_ys, 2)) + d(L(l_ys, 3)) + d(L(l_ys, 4)) +
                d(L(l_ks, 1)) + d(L(l_ks, 2)) + d(L(l_ks, 3)) + d(L(l_ks, 4)),
              data=tkeil) %>%
  summary()
```

```
##
## Time series regression with "ts" data:
## Start = 6, End = 65
##
## Call:
## dynlm(formula = d(l_ys) ~ time + L(l_ys, 1) + L(l_ks, 1) + d(L(l_ys,
##     1)) + d(L(l_ys, 2)) + d(L(l_ys, 3)) + d(L(l_ys, 4)) + d(L(l_ks,
##     1)) + d(L(l_ks, 2)) + d(L(l_ks, 3)) + d(L(l_ks, 4)), data = tkeil)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.08706 -0.01892 -0.00044  0.01926  0.04821
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    12.569090   7.609355   1.652  0.10510
## time           -0.006904   0.004287  -1.610  0.11386
## L(l_ys, 1)      0.034305   0.093922   0.365  0.71653
## L(l_ks, 1)      0.108567   0.071886   1.510  0.13753
## d(L(l_ys, 1))   0.602234   0.186677   3.226  0.00226 **
## d(L(l_ys, 2))   0.229344   0.198265   1.157  0.25310
## d(L(l_ys, 3))  -0.218048   0.183806  -1.186  0.24134
## d(L(l_ys, 4))   0.062128   0.125606   0.495  0.62312
## d(L(l_ks, 1))  -6.561585   1.367917  -4.797  1.6e-05 ***
## d(L(l_ks, 2))   5.669199   2.415067   2.347  0.02307 *
## d(L(l_ks, 3))   0.531702   2.431858   0.219  0.82786
## d(L(l_ks, 4))  -2.375803   1.499224  -1.585  0.11960
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02929 on 48 degrees of freedom
## Multiple R-squared:  0.4802, Adjusted R-squared:  0.3611
## F-statistic: 4.032 on 11 and 48 DF,  p-value: 0.000344
```

The time trend is not significantly different from zero at a 10 % level, and is thus excluded from future estimations.

2.2 DUMMY SELECTION

```
ecm0 <- dynlm(d(l_ys) ~ time + L(l_ys, 1) + L(l_ys, 2) + L(l_ys, 3) + L(l_ys, 4) +
                d(L(l_ys, 4)) +
                l_ks + L(l_ks, 1) + L(l_ks, 2) + L(l_ks, 3) + L(l_ks, 3) + L(l_ks, 4) +
                d(L(l_ks, 4)),
              data=tkeil)
ecm0$residuals %>%
```

```r
  as.data.frame() %>%
  mutate(year=seq(1952, 2011, 1),
         d=abs(x)/summary(ecm0)$sigma) %>%
  filter(d>2)
```

```
##              x year        d
## 1 -0.05215828 1956 2.466041
## 2 -0.05612190 1974 2.653441
## 3 -0.04682567 1980 2.213915
```

The normalized residual d exceeds 2 in three years, 1956, 1974 and 1980. I include dummies for these years in the dataset.

```r
dkeil2 <- dkeil %>%
  mutate(d56=ifelse(time==1956, 1, 0),
         d74=ifelse(time==1974, 1, 0),
         d80=ifelse(time==1980, 1, 0))
tkeil2 <- ts(dkeil2, start=1947, end=2011, frequency=1)
```

2.3 ECM Model Selection

```r
aics1 <- matrix(NA, ncol=5, nrow=5)
bics1 <- matrix(NA, ncol=5, nrow=5)
formulas1 <- matrix(NA, ncol=5, nrow=5)
for(i in 1:5){
  for(j in 1:5){
    formula <- c()
    formula[1] <- "d(l_ys) ~ d56 + d74 + d80 + L(l_ys, 1) + L(l_ks, 1) "
    for(k in 1:i){
      formula[1+k] <- paste0(" + L(d(l_ys), ",
                             k,
                             ")")
    }
    rm(k)

    formula[i+2] <- ""
    for(l in 1:j){
      formula[i+2+l] <- paste0(" + L(d(l_ks), ",
                               l,
                               ")")
    }
    rm(l)

    formulas1[i,j] <- paste(formula, collapse="")
    formula2 <- as.formula(formulas1[i,j])
    aics1[i,j] <- dynlm(formula=formula2,
                        data=tkeil2) %>%
      AIC()
    bics1[i,j] <- dynlm(formula=formula2,
                        data=tkeil2) %>%
      BIC()
    rm(formula, formula2)
  }
```

```
}

ecm1 <- dynlm(formula=as.formula(formulas1[which(aics1==min(aics1))]),
      data=tkeil2)
summary(ecm1)
```

```
##
## Time series regression with "ts" data:
## Start = 1950, End = 2011
##
## Call:
## dynlm(formula = as.formula(formulas1[which(aics1 == min(aics1))]),
##     data = tkeil2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.06375 -0.01628  0.00000  0.02071  0.04522
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.23346    0.10225   2.283 0.026526 *
## d56           -0.05464    0.02855  -1.914 0.061134 .
## d74           -0.10185    0.02794  -3.645 0.000617 ***
## d80           -0.03807    0.02865  -1.329 0.189789
## L(l_ys, 1)    -0.05451    0.06060  -0.900 0.372510
## L(l_ks, 1)     0.02596    0.04670   0.556 0.580674
## L(d(l_ys), 1)  0.58170    0.13418   4.335 6.70e-05 ***
## L(d(l_ys), 2)  0.14665    0.11140   1.316 0.193803
## L(d(l_ks), 1) -4.49107    1.05034  -4.276 8.16e-05 ***
## L(d(l_ks), 2)  3.97633    1.04590   3.802 0.000379 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02688 on 52 degrees of freedom
## Multiple R-squared:  0.5486, Adjusted R-squared:  0.4704
## F-statistic: 7.021 on 9 and 52 DF,  p-value: 1.465e-06
```

I run ECM(p, q) estimations with $p, q \in 1, 5$, and find that an ECM including the first lags of Y and K as well as the differences of the first two lags of each variable minimizes AIC.

2.4 Breusch-Godfrey Test for Serial Correlation

```
lmtest::bgtest(ecm1)
```

```
##
##  Breusch-Godfrey test for serial correlation of order up to 1
##
## data:  ecm1
## LM test = 0.021134, df = 1, p-value = 0.8844
```

A Breusch-Godfrey Test for serial correlation fails to reject the null hypothesis of no serial correlation. This means the ECM is fit for the Pesaran-Shin-Smith procedure.

2.5 Pesaran-Shin-Smith Bounds Testing

```
nardl::pssbounds(obs=65, #rejected at 5 % level
                 fstat=6.256,
                 case=3,
                 k=2)
```

```
##
##  PESARAN, SHIN AND SMITH (2001) COINTEGRATION TEST
##
##  Observations: 65
##  Number of Regressors (k): 2
##  Case: 3
##
##  ------------------------------------------------------
##  -                       F-test                       -
##  ------------------------------------------------------
##                 <------- I(0) ------------ I(1) ----->
##  10% critical value       3.3            4.25
##  5% critical value        4.01            5.08
##  1% critical value        5.583            6.853
##
##
##  F-statistic = 6.256
##
##  ------------------------------------------------------
##
##
```

```
nardl::pssbounds(obs=65, #rejected at 5 % level
                 fstat=6.256,
                 case=5,
                 k=2)
```

```
##
##  PESARAN, SHIN AND SMITH (2001) COINTEGRATION TEST
##
##  Observations: 65
##  Number of Regressors (k): 2
##  Case: 5
##
##  ------------------------------------------------------
##  -                       F-test                       -
##  ------------------------------------------------------
##                 <------- I(0) ------------ I(1) ----->
##  10% critical value       4.353            5.257
##  5% critical value        5.137            6.173
##  1% critical value        7.013            8.23
##
##
##  F-statistic = 6.256
##
##  ------------------------------------------------------
##
##
```

The Pesaran-Shin-Smith bounds testing procedure, both for a combination of an unbounded intercept and an unbounded time trend, and for an unbounded intercept term alone, reject I(1) integration in Y and K, and thus suggest co-integration.

2.6 ADL Model Selection

In the same fashion as for the ECM, I run ADL regressions with any combination of p lags on Y and q lags on K, including the year dummies, and choose the AIC minimizing model.

```
aics <- matrix(NA, ncol=5, nrow=5)
bics <- matrix(NA, ncol=5, nrow=5)
formulas <- matrix(NA, ncol=5, nrow=5)

for(i in 1:5){
  for(j in 1:5){
    formula <- c()
    formula[1] <- "l_ys ~ d56 + d74 + d80"
    for(k in 1:i){
      formula[1+k] <- paste0(" + L(l_ys, ",
                             k,
                             ")")
    }
    rm(k)

    formula[i+2] <- "+ l_ks "
    for(l in 1:j){
      formula[i+2+l] <- paste0(" + L(l_ks, ",
                               l,
                               ")")
    }
    rm(l)

    formulas[i,j] <- paste(formula, collapse="")
    formula2 <- as.formula(formulas[i,j])
    aics[i,j] <- dynlm(formula=formula2,
                       data = tkeil2) %>%
      AIC()
    bics[i,j] <- dynlm(formula=formula2,
                       data = tkeil2) %>%
      BIC()
  }
}

adl1 <- dynlm(formula=as.formula(formulas[which(aics==min(aics))]),
              data=tkeil2)
summary(adl1)
```

```
##
## Time series regression with "ts" data:
## Start = 1951, End = 2011
##
## Call:
## dynlm(formula = as.formula(formulas[which(aics == min(aics))]),
##     data = tkeil2)
```

```
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.033271 -0.009384  0.000000  0.009297  0.032664
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.20722    0.06388   3.244 0.002104 **
## d56          -0.07094    0.01707  -4.156 0.000126 ***
## d74          -0.08115    0.01709  -4.750 1.75e-05 ***
## d80          -0.04547    0.01740  -2.614 0.011800 *
## L(l_ys, 1)    1.29869    0.09676  13.422  < 2e-16 ***
## L(l_ys, 2)   -0.39369    0.09616  -4.094 0.000155 ***
## l_ks          5.06995    0.53706   9.440 1.08e-12 ***
## L(l_ks, 1)  -14.24513    1.21213 -11.752 5.35e-16 ***
## L(l_ks, 2)   15.66560    1.73074   9.051 4.12e-12 ***
## L(l_ks, 3)   -8.27488    1.47491  -5.610 8.77e-07 ***
## L(l_ks, 4)    1.84723    0.55162   3.349 0.001550 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01628 on 50 degrees of freedom
## Multiple R-squared:  0.9992, Adjusted R-squared:  0.999
## F-statistic:  6255 on 10 and 50 DF,  p-value: < 2.2e-16
```

The AIC minimizing model includes two lags of Y and four lags of K, as well as year dummies and the present value of K. For all coefficients in the estimation, they are significantly different from zero at the 5 % level.

2.7 Long-Run Estimators

I calculate the long run coefficients, and plug them into the simplified estimation of the logarithm of capacity, and not including the year dummies.

```
denominator <- 1 - sum(adl1$coefficients[c(5,6)])
a <- adl1$coefficients[1]/denominator
b <- 0
c56 <- adl1$coefficients[2]/denominator
c74 <- adl1$coefficients[3]/denominator
c80 <- adl1$coefficients[4]/denominator
d <- sum(adl1$coefficients[c(7:11)])/denominator

dkeil3 <- dkeil2 %>%
  mutate(lnystar = a + b*time + d*l_ks,
         ucap=exp(l_ys - lnystar))
```

```
dkeil3 %>%
  select(time, l_ys, lnystar) %>%
  melt(id="time") %>%
  ggplot() +
  geom_line(aes(x=time,
                y=value,
                group=variable,
                color=variable),
            size=1.2) +
```

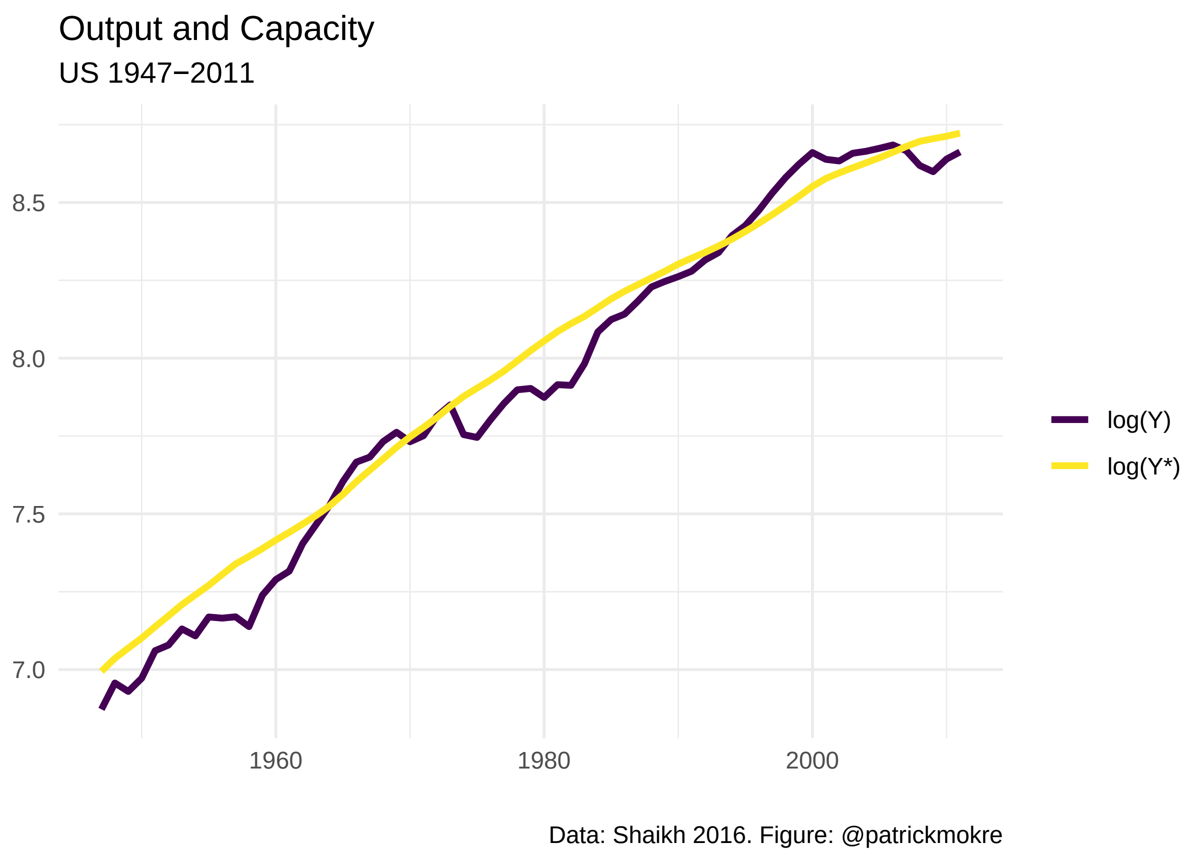
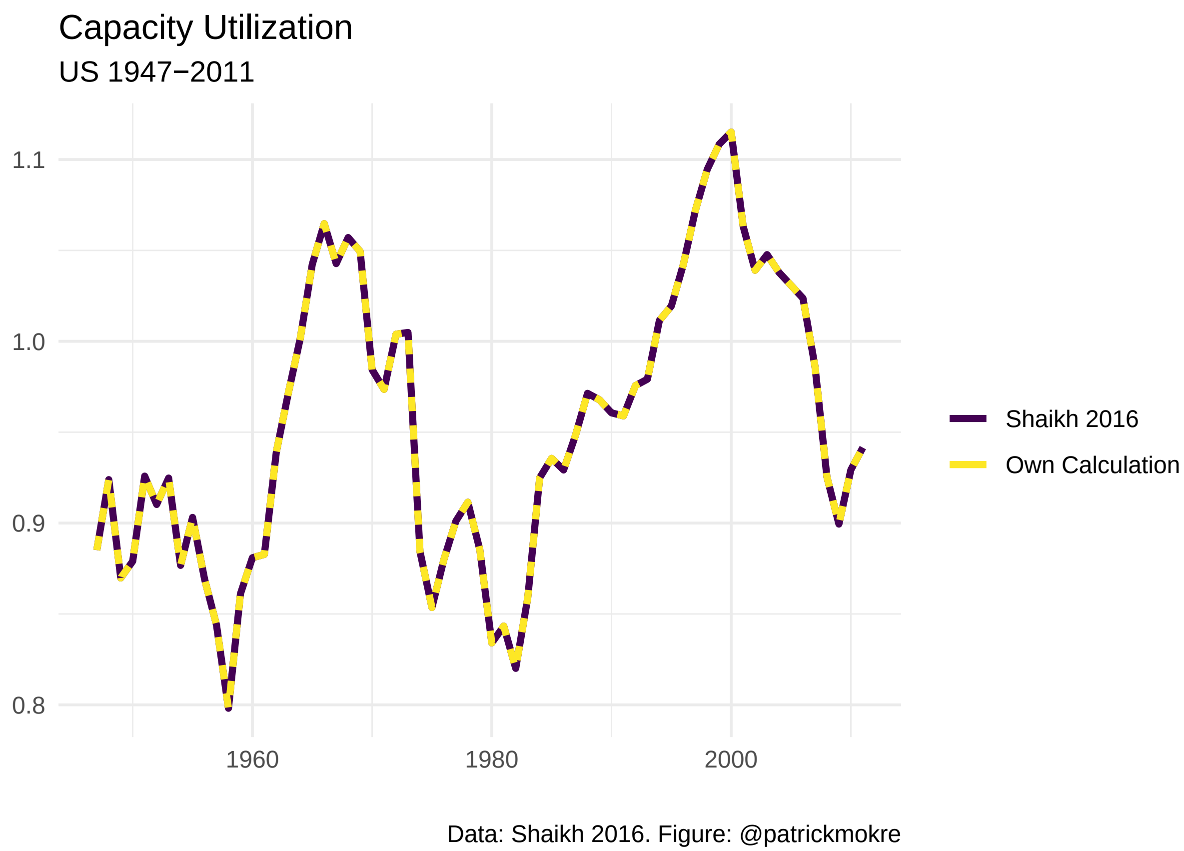


Figure 1: Output and Capacity

```
  scale_color_viridis_d(name="",
                        labels=c("log(Y)", "log(Y*)")) +
  theme_minimal() +
  labs(title="Output and Capacity",
       subtitle="US 1947-2011",
       caption="Data: Shaikh 2016. Figure: @patrickmokre",
       x="",
       y="")
```

```
dkeil3 %>%
  select(time, ucap, us) %>%
  melt(id="time") %>%
  ggplot() +
  geom_line(aes(x=time,
                y=value,
                group=variable,
                color=variable,
                linetype=variable),
            size=1.25) +
  scale_color_viridis_d(name="",
                        labels=c("Shaikh 2016", "Own Calculation")) +
  guides(linetype=F) +
```

```
  theme_minimal() +
  labs(title="Capacity Utilization",
       subtitle="US 1947-2011",
       caption="Data: Shaikh 2016. Figure: @patrickmokre",
       x="",
       y="")
```



3 Data from Shaikh 2016, Online Appendix

I showed above that I could replicate the methodology used for the results derived, when using the same input data. In this section I will apply the ECM-ALD-mixture method to estimate capacity utilization from the data in Shaikh 2016's online appendix.

3.1 Data

```
rm(list=ls())
setwd("C:/Users/M/Google Drive/NSSR/Shaikh/Capacity Utilization/Reproduction")

require(dplyr)
require(readxl)
require(dynlm)
require(lmtest)
```

```r
require(ggplot2)
require(reshape2)

options(scipen=999)

## IMPORT DATA
dkeil <- read_excel("__Utilization Output Keil January 2013.xlsx",
                    sheet=3,
                    range="A2:AB67")


data_pkn <- read_excel("_Appendix 6.8 Data Tables Corrected.xlsx",
                      sheet="Appndx 6.8.II.1",
                      range="Z22:CL22",
                      col_names=FALSE) %>%
  t() %>%
  as.data.frame() %>%
  rename(pKn=V1)

data_kgc <- read_excel("_Appendix 6.8 Data Tables Corrected.xlsx",
                       sheet="Appndx 6.8.II.5",
                       range="Z17:CL17",
                       col_names=FALSE) %>%
  t() %>%
  as.data.frame() %>%
  rename(KGCcorp=V1)
data_vac <- read_excel("_Appendix 6.8 Data Tables Corrected.xlsx",
                       sheet="Appndx 6.8.II.7",
                       range="G22:BS22",
                       col_names=FALSE) %>%
  t() %>%
  as.data.frame() %>%
  rename(VAcorp=V1)

data <- data.frame(time=seq(1947, 2011, 1),
                   VAcorp=data_vac$VAcorp,
                   KGCcorp=data_kgc$KGCcorp,
                   pKn=data_pkn$pKn,
                   lys_keil=dkeil$l_ys,
                   lks_keil=dkeil$l_ks,
                   us_keil=dkeil$us) %>%
  mutate(y_a_n = VAcorp*100/pKn,
         k_n = KGCcorp*100/pKn,
         lny=log(y_a_n),
         lnk=log(k_n))
tdata=ts(data, start=1947, end=2011, frequency=1)

rm(data_kgc, data_pkn, data_vac)
```

3.2 Time Trend

```
dynlm(d(lny) ~ time + L(lny, 1) + L(lnk, 1) +
        d(L(lny, 1)) + d(L(lny, 2)) + d(L(lny, 3)) + d(L(lny, 4)) +
        d(L(lnk, 1)) + d(L(lnk, 2)) + d(L(lnk, 3)) + d(L(lnk, 4)),
      data=tdata) %>%
  summary()
```

```
##
## Time series regression with "ts" data:
## Start = 1952, End = 2011
##
## Call:
## dynlm(formula = d(lny) ~ time + L(lny, 1) + L(lnk, 1) + d(L(lny,
##     1)) + d(L(lny, 2)) + d(L(lny, 3)) + d(L(lny, 4)) + d(L(lnk,
##     1)) + d(L(lnk, 2)) + d(L(lnk, 3)) + d(L(lnk, 4)), data = tdata)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.087927 -0.017496  0.002652  0.017783  0.046740
##
## Coefficients:
##               Estimate Std. Error t value  Pr(>|t|)
## (Intercept)  12.207082   7.647575   1.596    0.1170
## time         -0.006699   0.004301  -1.557    0.1259
## L(lny, 1)     0.025510   0.091094   0.280    0.7807
## L(lnk, 1)     0.110382   0.073584   1.500    0.1401
## d(L(lny, 1))  0.561929   0.180401   3.115    0.0031 **
## d(L(lny, 2))  0.193090   0.190856   1.012    0.3168
## d(L(lny, 3)) -0.206493   0.187343  -1.102    0.2759
## d(L(lny, 4))  0.061280   0.129858   0.472    0.6391
## d(L(lnk, 1)) -5.946323   1.318213  -4.511 0.0000417 ***
## d(L(lnk, 2))  5.126154   2.265323   2.263    0.0282 *
## d(L(lnk, 3))  0.436734   2.262952   0.193    0.8478
## d(L(lnk, 4)) -2.148242   1.442982  -1.489    0.1431
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02964 on 48 degrees of freedom
## Multiple R-squared:  0.4502, Adjusted R-squared:  0.3243
## F-statistic: 3.574 on 11 and 48 DF,  p-value: 0.001016
```

3.3 DUMMY SELECTION

```
ecm0 <- dynlm(d(lny) ~ time + L(lny, 1) + L(lny, 2) + L(lny, 3) + L(lny, 4) +
                d(L(lny, 4)) +
                lnk + L(lnk, 1) + L(lnk, 2) + L(lnk, 3) + L(lnk, 3) + L(lnk, 4) +
                d(L(lnk, 4)),
              data=tdata)
ecm0$residuals %>%
  as.data.frame() %>%
  mutate(year=seq(1952, 2011, 1),
```

```r
        d=abs(x)/summary(ecm0)$sigma) %>%
  filter(d>2)
```

```
##             x year        d
## 1 -0.05140855 1956 2.280199
## 2 -0.05779711 1974 2.563560
## 3 -0.04862635 1980 2.156796
## 4 -0.05198578 2008 2.305802
## 5  0.04650857 2009 2.062863
```

3.4 INTRODUCE DUMMIES

```r
data2 <- data %>%
  mutate(d56=ifelse(time==1956, 1, 0),
         d74=ifelse(time==1974, 1, 0),
         d80=ifelse(time==1980, 1, 0),
         d08=ifelse(time==2008, 1, 0),
         d09=ifelse(time==2009, 1, 0))
tdata2 <- ts(data2, start=1947, end=2011, frequency=1)
```

3.5 ECM: AIC MINIMIZATION MODEL SELECTION

```r
aics1 <- matrix(NA, ncol=5, nrow=5)
bics1 <- matrix(NA, ncol=5, nrow=5)
formulas1 <- matrix(NA, ncol=5, nrow=5)
for(i in 1:5){
  for(j in 1:5){
    formula <- c()
    formula[1] <- "d(lny) ~ d56 + d74 + d80 + d08 + d09 + L(lny, 1) + L(lnk, 1) "
    for(k in 1:i){
      formula[1+k] <- paste0(" + L(d(lny), ",
                             k,
                             ")")
    }
    rm(k)

    formula[i+2] <- ""
    for(l in 1:j){
      formula[i+2+l] <- paste0(" + L(d(lnk), ",
                               l,
                               ")")
    }
    rm(l)

    formulas1[i,j] <- paste(formula, collapse="")
    formula2 <- as.formula(formulas1[i,j])
    aics1[i,j] <- dynlm(formula=formula2,
                        data=tdata2) %>%
      AIC()
    bics1[i,j] <- dynlm(formula=formula2,
                        data=tdata2) %>%
```

```
      BIC()
    rm(formula, formula2)
  }
}

ecm1 <- dynlm(formula=as.formula(formulas1[which(aics1==min(aics1))]),
              data=tdata2)
summary(ecm1)
```

```
##
## Time series regression with "ts" data:
## Start = 1950, End = 2011
##
## Call:
## dynlm(formula = as.formula(formulas1[which(aics1 == min(aics1))]),
##     data = tdata2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.06460 -0.01457  0.00000  0.01701  0.04060
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.21419    0.09199   2.328 0.023891 *
## d56          -0.06005    0.02663  -2.255 0.028460 *
## d74          -0.10340    0.02673  -3.868 0.000312 ***
## d80          -0.04118    0.02747  -1.499 0.140101
## d08          -0.06823    0.02755  -2.476 0.016633 *
## d09           0.01059    0.02923   0.362 0.718715
## L(lny, 1)    -0.03860    0.05487  -0.704 0.484898
## L(lnk, 1)     0.01530    0.04386   0.349 0.728734
## L(d(lny), 1)  0.47780    0.13976   3.419 0.001245 **
## L(d(lnk), 1) -3.30603    0.95164  -3.474 0.001055 **
## L(d(lnk), 2)  2.78864    0.99700   2.797 0.007257 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02573 on 51 degrees of freedom
## Multiple R-squared:  0.5797, Adjusted R-squared:  0.4973
## F-statistic: 7.033 on 10 and 51 DF,  p-value: 0.0000008127
```

The AIC minimizing ECM includes one lag of differences for output and two lags of differences for capital stock. This setup I test for serial correlation in a Breusch-Godfrey setup.

3.6 BREUSCH-GODFREY TEST FOR SERIAL CORRELATION

```
lmtest::bgtest(ecm1)
```

```
##
##  Breusch-Godfrey test for serial correlation of order up to 1
##
## data:  ecm1
## LM test = 0.080224, df = 1, p-value = 0.777
```

The Breusch-Godfrey Test for no serial correlation fails to reject the NUll hypothesis at the 10 % level.

3.7 PESARAN-SHIN-SMITH

```
nardl::pssbounds(obs=65, #rejected at 1 % level
                 fstat=7.033,
                 case=3,
                 k=2)
```

```
##
##  PESARAN, SHIN AND SMITH (2001) COINTEGRATION TEST
##
##  Observations: 65
##  Number of Regressors (k): 2
##  Case: 3
##
##  ------------------------------------------------------
##  -                       F-test                       -
##  ------------------------------------------------------
##                  <------- I(0) ------------ I(1) ----->
##  10% critical value       3.3            4.25
##  5% critical value        4.01            5.08
##  1% critical value        5.583            6.853
##
##
##  F-statistic = 7.033
##
##  ------------------------------------------------------
##
##
```

The results of the Pesaran-Shin-Smith bounds test without a time trend, but including an intercept, suggest cointegration at the 1 % level.

```
nardl::pssbounds(obs=65, #rejected at 5 % level
                 fstat=7.033,
                 case=5,
                 k=2)
```

```
##
##  PESARAN, SHIN AND SMITH (2001) COINTEGRATION TEST
##
##  Observations: 65
##  Number of Regressors (k): 2
##  Case: 5
##
##  ------------------------------------------------------
##  -                       F-test                       -
##  ------------------------------------------------------
##                  <------- I(0) ------------ I(1) ----->
##  10% critical value       4.353            5.257
##  5% critical value        5.137            6.173
##  1% critical value        7.013            8.23
##
```

```
##
##  F-statistic = 7.033
##
##  ------------------------------------------------------
##
##
```

For the case of both intercept and time trend, cointegration is suggested at the 5 % level. Note that neither the ECM model, nor the ADL setup include a significant time trend.

3.8 ADL Model Selection

I run ADL models for all combinations of p lags of output and q lags of capital stock where $p, q \in (1, ..., 5)$ and choose the model that shows the lowest Akaike Information Criterion.

```
aics <- matrix(NA, ncol=5, nrow=5)
bics <- matrix(NA, ncol=5, nrow=5)
formulas <- matrix(NA, ncol=5, nrow=5)

for(i in 1:5){
  for(j in 1:5){
    formula <- c()
    formula[1] <- "lny ~ d56 + d74 + d80 + d08 + d09"
    for(k in 1:i){
      formula[1+k] <- paste0(" + L(lny, ",
                              k,
                              ")")
    }
    rm(k)

    formula[i+2] <- "+ lnk "
    for(l in 1:j){
      formula[i+2+l] <- paste0(" + L(lnk, ",
                                l,
                                ")")
    }
    rm(l)

    formulas[i,j] <- paste(formula, collapse="")
    formula2 <- as.formula(formulas[i,j])
    aics[i,j] <- dynlm(formula=formula2,
                       data = tdata2) %>%
      AIC()
    bics[i,j] <- dynlm(formula=formula2,
                       data = tdata2) %>%
      BIC()
  }
}
adl1 <- dynlm(formula=as.formula(formulas[which(aics==min(aics))]),
              data=tdata2)
summary(adl1)
```

```
##
## Time series regression with "ts" data:
```

```
## Start = 1951, End = 2011
##
## Call:
## dynlm(formula = as.formula(formulas[which(aics == min(aics))]),
##     data = tdata2)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.032587 -0.008909  0.000000  0.007785  0.032758
##
## Coefficients:
##              Estimate Std. Error t value            Pr(>|t|)
## (Intercept)   0.18039    0.05578   3.234             0.00221 **
## d56          -0.07064    0.01560  -4.529 0.000039315093952406 ***
## d74          -0.08132    0.01558  -5.221 0.000003784554363428 ***
## d80          -0.04612    0.01589  -2.903             0.00556 **
## d08          -0.05480    0.01597  -3.433             0.00124 **
## d09           0.05007    0.01730   2.894             0.00570 **
## L(lny, 1)     1.30224    0.09135  14.256 < 0.0000000000000002 ***
## L(lny, 2)    -0.39184    0.09025  -4.342 0.000072705362250449 ***
## lnk           4.99223    0.50138   9.957 0.000000000000291643 ***
## L(lnk, 1)   -13.70051    1.16067 -11.804 0.000000000000000846 ***
## L(lnk, 2)    14.66923    1.60006   9.168 0.000000000004020610 ***
## L(lnk, 3)    -7.60748    1.31976  -5.764 0.000000573273765468 ***
## L(lnk, 4)     1.70854    0.49422   3.457             0.00115 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01484 on 48 degrees of freedom
## Multiple R-squared:  0.9994, Adjusted R-squared:  0.9993
## F-statistic:  6856 on 12 and 48 DF,  p-value: < 0.00000000000000022
```

In contrast to the calculations in Section 2, for the updated data the AIC minimizing ADL model includes more lags than the AIC minimizing ECM. Inspecting the Table of AICs, where columns give the number of lags in K and rows give lags in Y, the difference between the ADL (2,4) setup and a (1, 2) setup as suggested in the ECM is substantial.

```r
aics %>%
  as.data.frame() %>%
  kable(digits=2,
        col.names=c(1,2,3,4,5),
        row.names=c(1,2,3,4,5),
        caption="AICs for different lag combinations in ADL models",
        booktabs=T) %>%
  kable_styling()
```

```
## Warning in if (is.na(row.names)) row.names = has_rownames(x): Bedingung hat
## Länge > 1 und nur das erste Element wird benutzt
```

```
## Warning in if (row.names) {: Bedingung hat Länge > 1 und nur das erste
## Element wird benutzt
```

On a sidenote, the Breusch-Godfrey test suggests the same result regarding serial correlation for the ADL model used, ex-post allowing for the Pesaran-Shin-Smith procedure.

Table 1: AICs for different lag combinations in ADL models

	1	2	3	4	5
1	-260.85	-303.26	-311.30	-308.99	-302.84
2	-256.55	-301.49	-319.27	-327.19	-321.76
3	-255.63	-293.72	-322.65	-326.52	-322.68
4	-249.42	-293.06	-325.62	-326.63	-320.68
5	-243.78	-290.99	-320.78	-325.11	-323.19

```
lmtest::bgtest(adl1)
```

```
##
##  Breusch-Godfrey test for serial correlation of order up to 1
##
## data:  adl1
## LM test = 0.001632, df = 1, p-value = 0.9678
```

3.9 Long Run Estimation

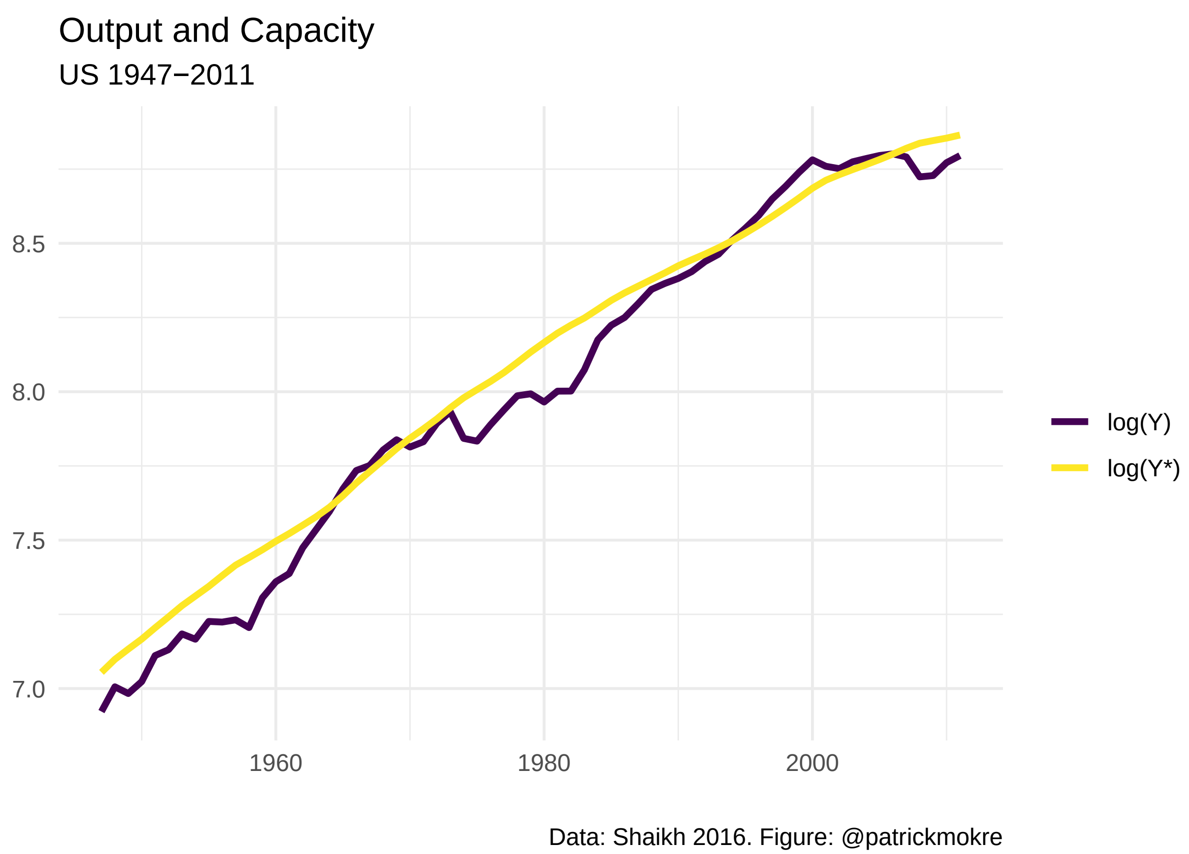
```
denominator <- 1 - sum(adl1$coefficients[c(7,8)])
a <- adl1$coefficients[1]/denominator
b <- 0
c56 <- adl1$coefficients[2]/denominator
c74 <- adl1$coefficients[3]/denominator
c80 <- adl1$coefficients[4]/denominator
c08 <- adl1$coefficients[5]/denominator
c09 <- adl1$coefficients[6]/denominator
d <- sum(adl1$coefficients[c(9:13)])/denominator

data3 <- data2 %>%
  mutate(lnystar = a + b*time + d*lnk,
         ucap=exp(lny - lnystar))
```

I retrieve the estimation coefficients and calculate intercept term a and current capital stock coefficient d to find $log(Y^*)$, and capacity utilization $u = exp(log(Y) - log(Y))$ for each year. Figure 3 shows both observed output and capacity. Figure 4 plots capacity utilization retrieved in Section 3 with the earlier results from Section 2.

While the time series differ due to different data input, the trends are similar.

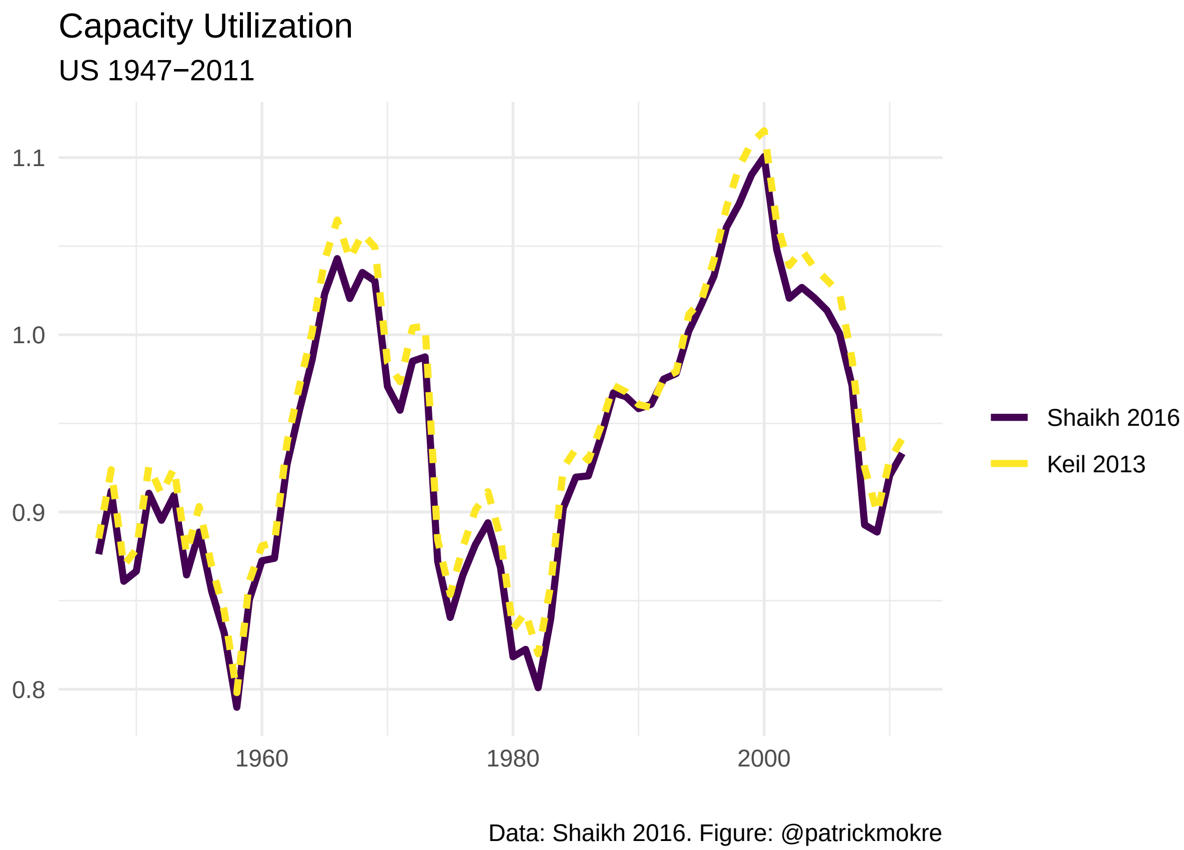
```
data3 %>%
  select(time, lny, lnystar) %>%
  melt(id="time") %>%
  ggplot() +
  geom_line(aes(x=time,
                y=value,
                group=variable,
                color=variable),
            size=1.2) +
  scale_color_viridis_d(name="",
                        labels=c("log(Y)",
```

```
                                         "log(Y*)")) +
  theme_minimal() +
  labs(title="Output and Capacity",
       subtitle="US 1947-2011",
       caption="Data: Shaikh 2016. Figure: @patrickmokre",
       x="",
       y="")
```




```
data3 %>%
  select(time, ucap, us_keil) %>%
  melt(id="time") %>%
  ggplot() +
  geom_line(aes(x=time,
                y=value,
                group=variable,
                color=variable,
                linetype=variable),
            size=1.25) +
  scale_color_viridis_d(name="",
                        labels=c("Shaikh 2016", "Keil 2013")) +
  guides(linetype=F) +
  theme_minimal() +
  labs(title="Capacity Utilization",
       subtitle="US 1947-2011",
       caption="Data: Shaikh 2016. Figure: @patrickmokre",
       x="",
```

```
y="")
```


Capacity Utilization
US 1947−2011
1.1
1.0
0.9
0.8
1960
1980
2000
Shaikh 2016
Keil 2013
Data: Shaikh 2016. Figure: @patrickmokre