

Nowcasting do PIB: um modelo de fatores.

<http://rleripio.com.br/nowcasting-do-pib-um-modelo-de-fatores/>

8 de novembro de 2018

Introdução

Em termos bem gerais, Nowcasting é a previsão do presente – ou do futuro/passado muito recente. O objetivo, neste caso, é encontrar um conjunto de variáveis capaz de capturar a dinâmica da série de interesse no mesmo período em que esta ocorre.

Entretanto, este conjunto pode envolver um número arbitrariamente grande de variáveis: dezenas, centenas ou até milhares. Se o número de variáveis for superior ao número de observações, não é possível estimar coeficientes em modelos paramétricos tradicionais (OLS, por exemplo). Por outro lado, mesmo que seja possível estimar os coeficientes, não é apropriado introduzir um número muito elevado de variáveis nos modelos: isto pode gerar overfitting, o que tende a produzir previsões bastante imprecisas.

Uma solução comumente adotada é utilizar um modelo com fatores. Mais especificamente, um fator busca reduzir o conjunto de informações (variáveis) a uma fonte de variação comum entre elas. Em geral, essa fonte de variação comum é não-observável. A ideia, portanto, é utilizar um conjunto de variáveis relacionadas a essa fonte e, ao extrair o componente de variação comum entre elas, obter uma aproximação desta fonte não-observável.

Para ficar mais claro, imagine que estejamos interessados em acompanhar em “tempo real” a evolução da atividade econômica. Sabemos que uma série de variáveis são afetados ou afetam a atividade econômica, embora em direções e magnitudes possivelmente distintas. Por exemplo, a utilização de energia elétrica, a produção da indústria, a confiança dos consumidores, etc. Ao extrair o componente de variação comum entre estas variáveis, poderíamos ter uma medida (indireta) da atividade econômica. E se tivermos estas informações disponíveis no instante t , poderemos ter informação sobre a atividade econômica também em t – um Nowcast.

Estimulado por essa ideia, o exercício busca realizar um Nowcast do PIB. O IBGE divulga os resultados do PIB cerca de 2 meses depois do fim de cada trimestre. Porém, uma variedade de informações sobre a atividade torna-se disponível no decorrer do próprio trimestre. É possível, então, reunir essas informações, extrair fatores e obter previsões. De início, serão utilizadas apenas 10 variáveis: medidas de inflação, atividade da indústria e confiança dos agentes. Após realizar o tratamento nos dados, serão construídos os fatores através de Componentes Principais (PCA). Em seguida, será ajustado um modelo OLS com a série de PIB como variável dependente e os quatro primeiros fatores como covariáveis.

Códigos para replicar (acompanha .RData)

Passo 1: carregar pacotes e importar dados

```
library(tidyverse)
library(rbc)
library(seasonal)
library(timetk)
library(readxl)
library(forecast)

series_bcb <- list("PMC" = 1455,
                  "Inf_ex3" = 27839,
                  "Energia" = 1404,
```

```

      "Industria" = 21862,
      "Capacidade" = 24352,
      "Consumidor" = 4393)

dados <- rbcB::get_series(code = series_bcb) %>% purrr::reduce(inner_join, by = "date")

caged <- readxl::read_excel("caged_jul18.xlsx") %>%

  dplyr::mutate(date = as.Date(Data)) %>%

  dplyr::select(date, caged = Dentro)

confianca <- readxl::read_excel("confiança.xlsx") %>%

  dplyr::mutate(date = as.Date(Data)) %>%

  dplyr::select(-Data)

dados_join <- dplyr::inner_join(dados, confianca, by = "date") %>%

  dplyr::mutate_at(vars(-date), funs(as.numeric))

```

Passo 2: Aplicar tratamento sazonal, transformação logarítmica, calcular os componentes principais e gerar as variáveis-índice

```

fun_ts <- function(x){timetk::tk_ts(x, start = c(2006,7), frequency = 12)}

dados_ts <- dados_join %>% dplyr::mutate_at(vars(-date), funs(fun_ts))

fun_sa <- function(x){final(seas(x))}

dados_sa <- dados_ts %>% dplyr::mutate_at(vars(-date, -Consumidor, -Termos, -Cons_ISA,
                                           Cons_EXP, Ind_ISA, Ind_EXP), funs(fun_sa))

log_1 <- function(x){log(1+x)}

dados_aux <- dados_sa %>% dplyr::mutate_at(vars(-date), funs(log_1)) %>% tidyr::drop_na()

dados_pca <- prcomp(dados_aux[, -1], center = TRUE, scale. = TRUE)

var_index <- as.matrix(dados_aux[, -1]) %*% dados_pca$rotation %>% as.tibble()

```

Passo 4: colocar as variáveis-índice em frequência trimestral (média), reunir com os dados do PIB e definir amostras de treino e de teste

```

dados_3m <- var_index %>%

  dplyr::mutate(Data = dados_aux$date) %>%

  dplyr::mutate(quarter = lubridate::quarter(Data, with_year = T)) %>%

  dplyr::group_by(quarter) %>%

  dplyr::summarise_at(vars(-c(Data, quarter)), funs(mean))

```

```

pib <- rbcdb::get_series(code = list("pib" = 22109)) %>%

  dplyr::mutate(quarter = lubridate::quarter(date, with_year = TRUE)) %>%

  dplyr::select(quarter, pib)

dados_reg <- dplyr::inner_join(pib, dados_3m) %>%

  dplyr::mutate(tri = sub('.*\\.', '', quarter) %>% as.factor())

reg_treino <- dados_reg %>% dplyr::filter(quarter < "2016.2")

reg_teste <- dados_reg %>% dplyr::filter(quarter >= "2016.2")

```

Passo 5: ajustar o modelo e observar o ajuste

```

modelo <- lm(pib ~ PC1 + PC2 + PC3 + PC4, data = reg_treino)

summary(modelo)

##
## Call:
## lm(formula = pib ~ PC1 + PC2 + PC3 + PC4, data = reg_treino)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.7338 -1.1591 -0.2311  1.2971  2.3923
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -539.163      26.144  -20.623  < 2e-16 ***
## PC1          -11.038       1.589   -6.945 5.25e-08 ***
## PC2           64.384       1.366   47.150 < 2e-16 ***
## PC3           36.665       4.666    7.859 3.77e-09 ***
## PC4          -24.735       3.045   -8.124 1.79e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.545 on 34 degrees of freedom
## Multiple R-squared:  0.9874, Adjusted R-squared:  0.9859
## F-statistic: 665.6 on 4 and 34 DF,  p-value: < 2.2e-16

modelo_tbl <- sweep::sw_augment(modelo) %>%

  dplyr::select(`PIB Observado` = pib, `Modelo` = .fitted) %>%

  dplyr::mutate(quarter = reg_treino$quarter)

modelo_tbl %>% tidyr::gather(key = var, value = valor, -quarter) %>%

  dplyr::mutate(var = factor(var, levels = c("PIB Observado", "Modelo"))) %>%

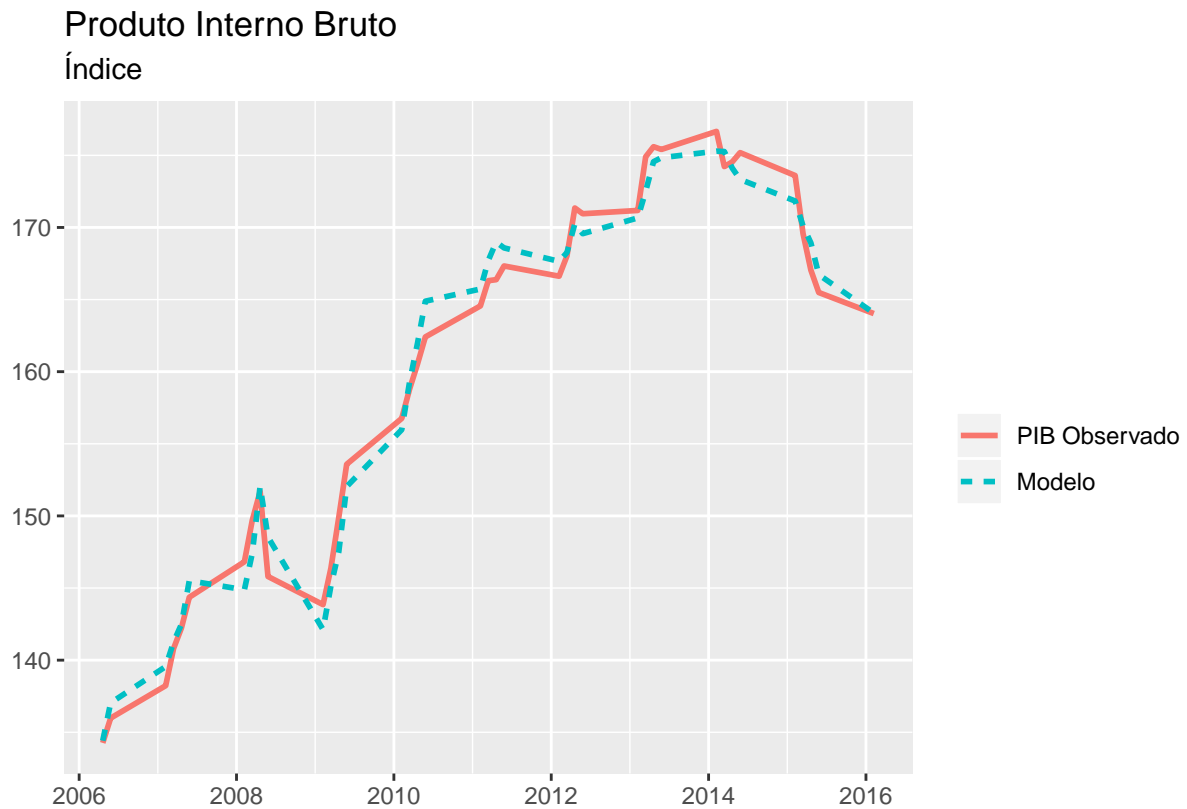
  ggplot(aes(x = quarter, y = valor, linetype = var, color = var), group = 1) +

  geom_line(lwd = 1) +

```

```
labs(title = "Produto Interno Bruto",
      subtitle = "Índice",
      color = "",
      linetype = "",
      x = "",
      y = "") +
```

```
theme_get()
```



Passo 6: calcular as previsões, observar a acurácia fora da amostra e gerar o gráfico

```
modelo_fc <- forecast(modelo, newdata = reg_teste)
```

```
accuracy(modelo_fc, pib$pib[82:90])
```

```
##               ME      RMSE      MAE      MPE      MAPE
## Training set  2.186322e-15  1.442473  1.247065 -0.007783022  0.7832457
## Test set     -2.363734e+00  2.503990  2.363734 -1.440537481  1.4405375
##               MASE
## Training set  0.1098876
## Test set     0.2082851
```

```
dados_fora <- reg_teste %>%
```

```
  dplyr::mutate(Nowcast = round(modelo_fc$mean,2)) %>%
```

```
  dplyr::mutate(quarter = gsub("\\.", " Q", quarter)) %>%
```

```
dplyr::select(quarter, PIB = pib, Nowcast)

dados_fora %>% tidyr::gather(key = var, value = valor, -quarter) %>%

ggplot(aes(x = quarter, y = valor, color = var, group = var)) +

geom_line(lwd = 1) +

labs(x = "", y = "",
      color = "",
      title = "Nowcast do Produto Interno Bruto",
      subtitle = "Índice")
```

