

# Lista para Modelo Dinâmico

Thiago Mendes Rosa

13/06/2019

## Teoria

### Modelo Econômico

Harold Zurcher gerencia uma frota de ônibus que é sujeita a todo tipo de problema quando esta na rua. A milhagem (quilometragem) acumulada de um ônibus  $x_t$  é a variável de estado do problema. O desgaste do ônibus afeta o custo operacional esperado  $c(x_t; \theta_1)$  que depende da milhagem e um vetor de parâmetros não conhecido  $\theta_1 = \{\theta_{11}, \dots, \theta_{1n}\}$ .

Assuma que os custos dos ônibus vem de dois componentes: manutenção regular e despesas operacionais  $m(\cdot)$  e o custo  $f(\cdot)$  de substituir o motor no caso de falha (que é um evento estocástico que ocorre com alguma probabilidade).

a) Escreva o custo como uma combinação destes dois componentes. Indique claramente quais elementos são função de  $x_t$ . Argumente informalmente que  $\frac{\partial c}{\partial x_t} > 0$ . Esta hipótese é necessária para a solução do modelo. Ela é uma boa hipótese? Você pode fazer um argumento para explicar por que  $\frac{\partial c}{\partial x_t}$  poderia ser negativa (pelo menos para alguns valores de  $x_t$ )?

Considere que o custo  $m(\cdot)$  pode ser decomposto em  $m(m_r(x_t), m_o(\cdot))$ , em que  $m_r$  denota as despesas com manutenção regular e  $m_o$  as despesas operacionais. A função custo pode ser escrita como  $c = (m(m_r(x_t), m_o(\cdot)), f(x_t, \cdot))$ . Espera-se que, quanto maior for a utilização de um ônibus da frota, maiores serão tanto os custos de manutenção regular quanto os custos para substituição do motor em caso de falha. Com isso, quanto mais o ônibus roda, maior é o desgaste de suas peças e, portanto, maiores serão os custos. O custo poderia diminuir com  $x_t$  em seus valores iniciais se, por exemplo, houvesse uma garantia do fornecedor para qualquer tipo de problema nas milhas iniciais.

```
# Carregar a base de dados

#### Ler o dicionário de variáveis
# Extrair tabela do PDF com a descrição das variáveis
dic <- tabulizer::extract_tables("lista2019.pdf",
                                output = "data.frame")[[1]]

# Listar arquivos com as bases de dados
bases<- data.frame(arquivo=list.files("rust_data"),
                   nome=gsub(".asc","", list.files("rust_data")),
                   linhas=c(rep(137,4),128,36,60,81),
                   stringsAsFactors = F)

# Looping para carregar as bases
for(b in bases$arquivo){

  # Ler a base de dados
  base <- data.table::fread(paste0("rust_data/",b))

  # Definir o número de linhas da matriz
  nl<-bases[bases$arquivo==b,]$linhas
```

```

# Definir número de meses existentes na base
nm<-nl-12+1

# Criar objeto para receber os dados fixos de cada ônibus
dados <- c()

# Criar um objeto para receber as informações mensais
t<-c()

# Criar um objeto para receber as referências
ref<-c()

# Iniciar looping para carregar as informações fixas,
# repetindo para o número de meses
for(i in 1:11){

  # Capturar as informações de cada ônibus
  assign(paste0("V",i), unlist(rep(base[seq(i,nrow(base),nl),],nm)))

  # Juntar resultados
  dados <- cbind(dados,get(paste0("V",i)))
}

# Retirar as informações mensais (odômetros)

for(j in 12:nl){

  # Retirar para cada ônibus
  V12 <- base[seq(j,nrow(base),nl)]

  # Juntar resultados
  t <- rbind(t,V12)
}

# Criar um objeto com a referência,
# tendo como base o mês e ano inicial do odômetro

for(i in seq(10,nrow(base),nl)){

  # Capturar o mês inicial, o ano inicial, definir sempre o primeiro
  # dia de cada mês para transformar em data e criar a sequência de
  # meses conforme o número de meses(nm) disponíveis na base

  r <- data.frame(ref=seq(
    # Define a data
    lubridate::dmy(paste("01",base[i],
                        base[i+1],sep = "/")),
    # Sequência mensal
    by="month",
    # Pelo número de meses

```

```

length.out = nm))

# Juntar resultados
ref<- rbind(ref,r)
}

# Juntar a base final
assign(bases[bases$arquivo==b,]$nome,
cbind(dados,V12=t) %>%
# Organizar por ônibus
dplyr::arrange(V1) %>%
# Trazer referência
dplyr::bind_cols(ref) %>%
# Ajustar odômetros para ocasião da troca
dplyr::mutate(V12_adj=case_when(V12.V1>V6&V6>0~V12.V1-V6,
                                TRUE~V12.V1), # Primeira troca
              V12_adj=case_when(V12.V1>V9&V9>0~V12.V1-V9,
                                TRUE~V12_adj))) # Segunda troca

# Remover objetos desnecessários
rm(base,dados,r,ref,t,list=ls(pattern = "V\\d"),b,i,j,nl,nm)
}

```