# Estrutura a termo da taxa de juros

March 1, 2021

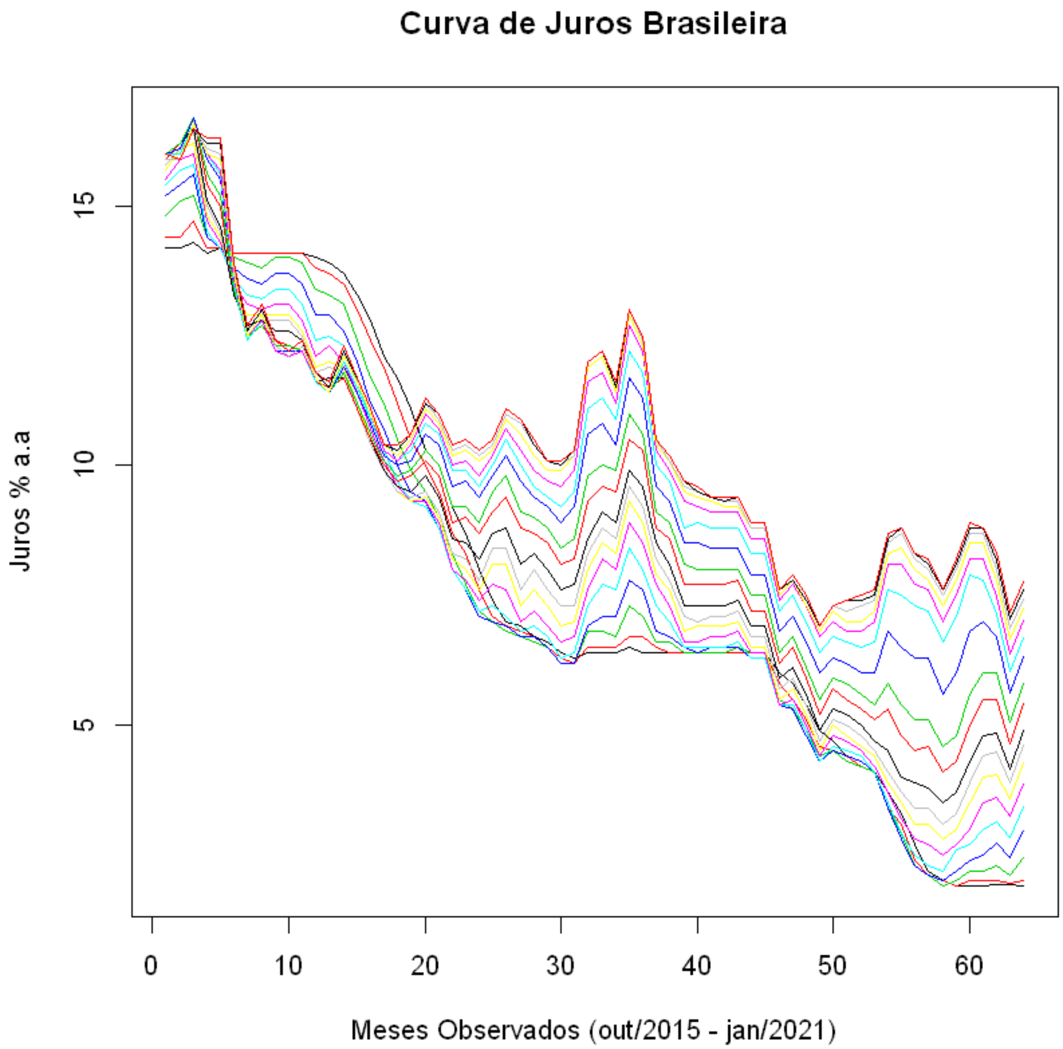
Neste exercício iremos fazer a previsão da curva de juros brasileira utilizando os dados históricos do Di-pré entre os meses de outubro de 2015 e janeiro de 2020. Será analisado a capacidade de previsão do modelo dentro e fora da amostra. Para tal andaremos 12 passos para trás, faremos a previsão e compararemos com o valor real. O modelo segue uma estrutra de analise dos componentes principais baseada em Nelson-Siegel.

## 1 Leitura dos dados

[1]:
```
jurosm1=read.table("Di.txt",header=T)
dim(jurosm1)
```

1. 64 2. 19

[2]:
```
# Plotando

yields=as.matrix(jurosm1[1:dim(jurosm1)[1],2:19])
datas=jurosm1[1:dim(jurosm1)[1],1]
vert=c(1,3, 6, 9, 12, 15, 18, 21, 24, 30, 36, 48, 60, 72, 84, 96, 108,120)

myield = apply(yields, 2, mean)
ts.plot(yields,col = 1:18, xlab = "Meses Observados (out/2015 - jan/2021)",
        ylab = "Juros % a.a",  main = 'Curva de Juros Brasileira')
```




## 1.1 Preparando os dados para analisar o PCA

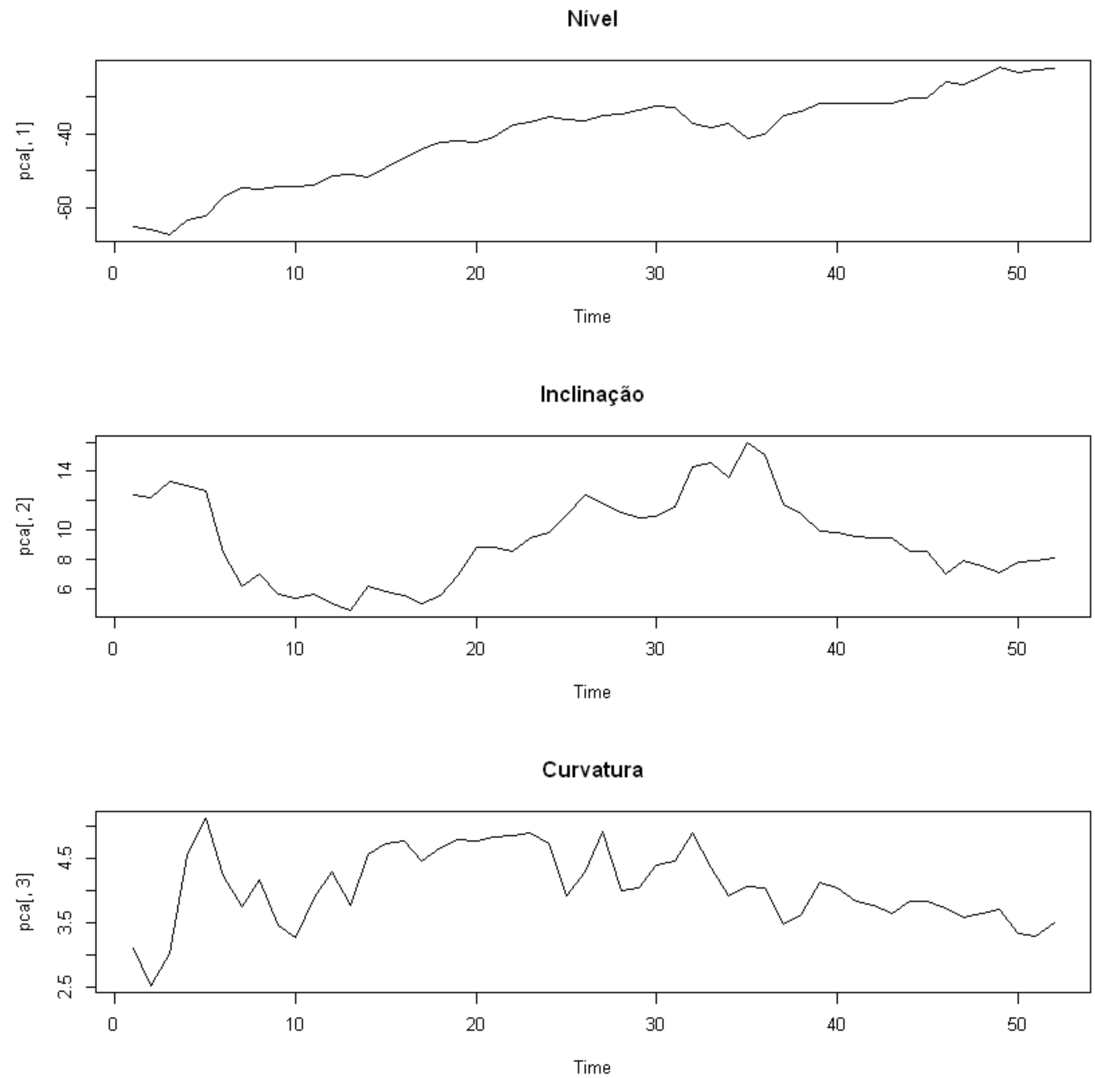
[3]:
```
K=12                    # 12 passos atrás
T=dim(yields)[1]
yieldsm=yields[1:(T-K),]

covy=cov(yieldsm)  # matrix de covariância amostral

eg=eigen(covy)   # decomposição autovalor e autovetor

cumsum(eg$values)/sum(eg$values)  # Proporção acumulativa  (2 ou 3 fatores,

# já explicam a dinâmica da curva)
```

1. 0.941598872263433 2. 0.997139069178269 3. 0.999258122636374 4. 0.999760948443083 5. 0.999841256121635 6. 0.999898899668293 7. 0.99992408087912 8. 0.999937489437781 9. 0.99995065072051 10. 0.999960335475431 11. 0.999968446781131 12. 0.999975742215549 13. 0.999981859866767 14. 0.999986794779225 15. 0.999991074087654 16. 0.999994733031077 17. 0.999997457256175 18. 1

Como visto, três componentes já são responsáveis pela explicação de 99,9% da curva de juros

### 1.2 Construção do PCA

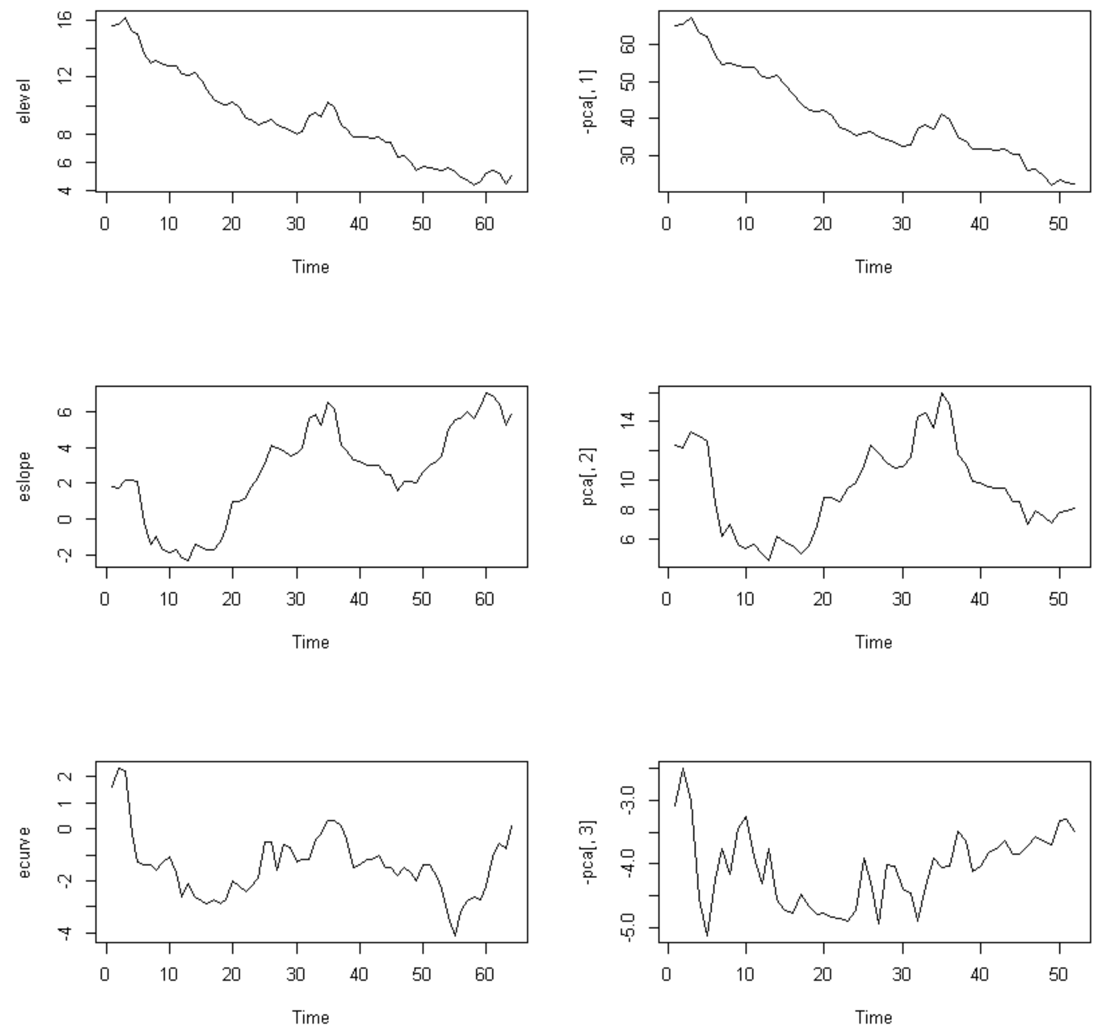
[4]:
```
pca=yieldsm%*%eg$vectors

par(mfrow=c(3,1))
ts.plot(pca[,1], main = "Nível")       # Nível é a média da curva
ts.plot(pca[,2], main = "Inclinação") # Curva longa - Curva curta
ts.plot(pca[,3], main = "Curvatura")  # Parâmetro de curvatura
```




### 1.2.1 Comparação entre a rotação dos componentes principais

[5]:
```
par(mfrow=c(3,2))
elevel=apply(yields,1,mean)
ts.plot(elevel)
ts.plot(-pca[,1])

eslope=yields[,18]-yields[,1]
ts.plot(eslope)
ts.plot(pca[,2])

ecurve=2*yields[,9]-(yields[,1]+yields[,18])
ts.plot(ecurve)
```

```r
ts.plot(-pca[,3])
```




# 2 Previsão

Será realizado uma estimação do Modelo Nelson-Siegel Dinâmico em dois passos (2 - STEP DNS)

## 2.1 Análise dentro da amostra

Ao realizar a análise dos dados dentro da amostra será feito a análise dos erros das estimativas como medida de fit

## 2.2 Análise fora da amostra

Para realizar a análise fora da amostra, foi analisada um conjunto de maturidades de taxa de juros, sendo de 1, 9, 15, 36, 120 meses de maturidade e seguimos o procedimento abaixo para a estimação:

1) Foi retirado da amostra original os ultimos cinco meses observados.
2) Montamos a matrix de covariância amostral e a decomposição dos autovalores e autovetores
3) Construimos o PCA
4) Fazemos a previsão da curva de juros via dois passos:

4.1) Prever o PCA

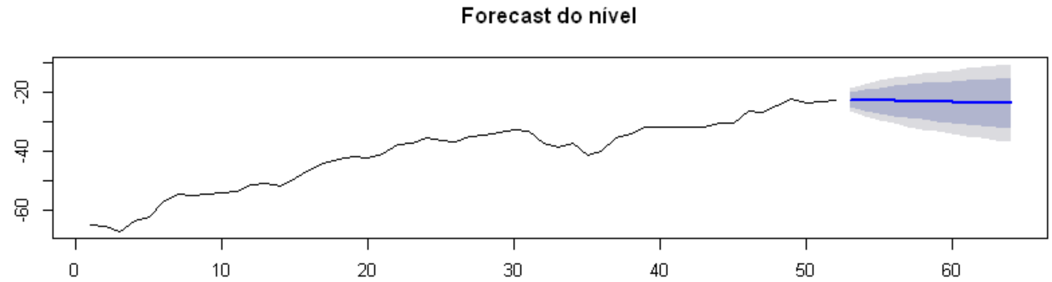
4.2) Construir a curva

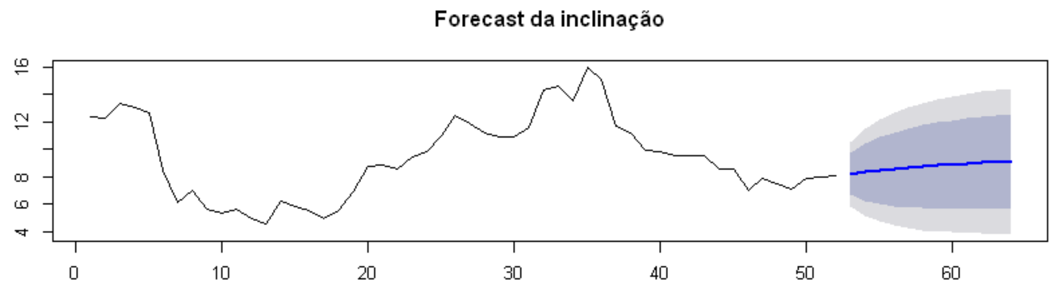
5) As estimativas das taxas de juros para cada vértice foram confrontadas com os valores observados nos ultimos doze meses

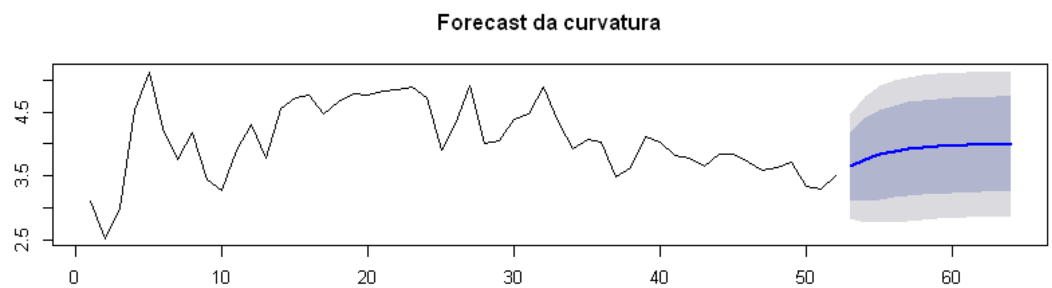
[6]:
```
suppressMessages(library(forecast))
```

```
Warning message:
"package 'forecast' was built under R version 3.6.3"
```

[7]:
```
pc1=pca[,1]
pc2=pca[,2]
pc3=pca[,3]

fpc1=arima(pc1,order=c(1,0,0))
fpc2=arima(pc2,order=c(1,0,0))
fpc3=arima(pc3,order=c(1,0,0))

ppc1=forecast(fpc1,h=K)$mean
ppc2=forecast(fpc2,h=K)$mean      # média pontual de 12 passos a frente
ppc3=forecast(fpc3,h=K)$mean

par(mfrow=c(3,1))
plot(forecast(fpc1,h=12), main = "Forecast do nível")
plot(forecast(fpc2,h=12), main = "Forecast da inclinação")
plot(forecast(fpc3,h=12), main = "Forecast da curvatura")
```

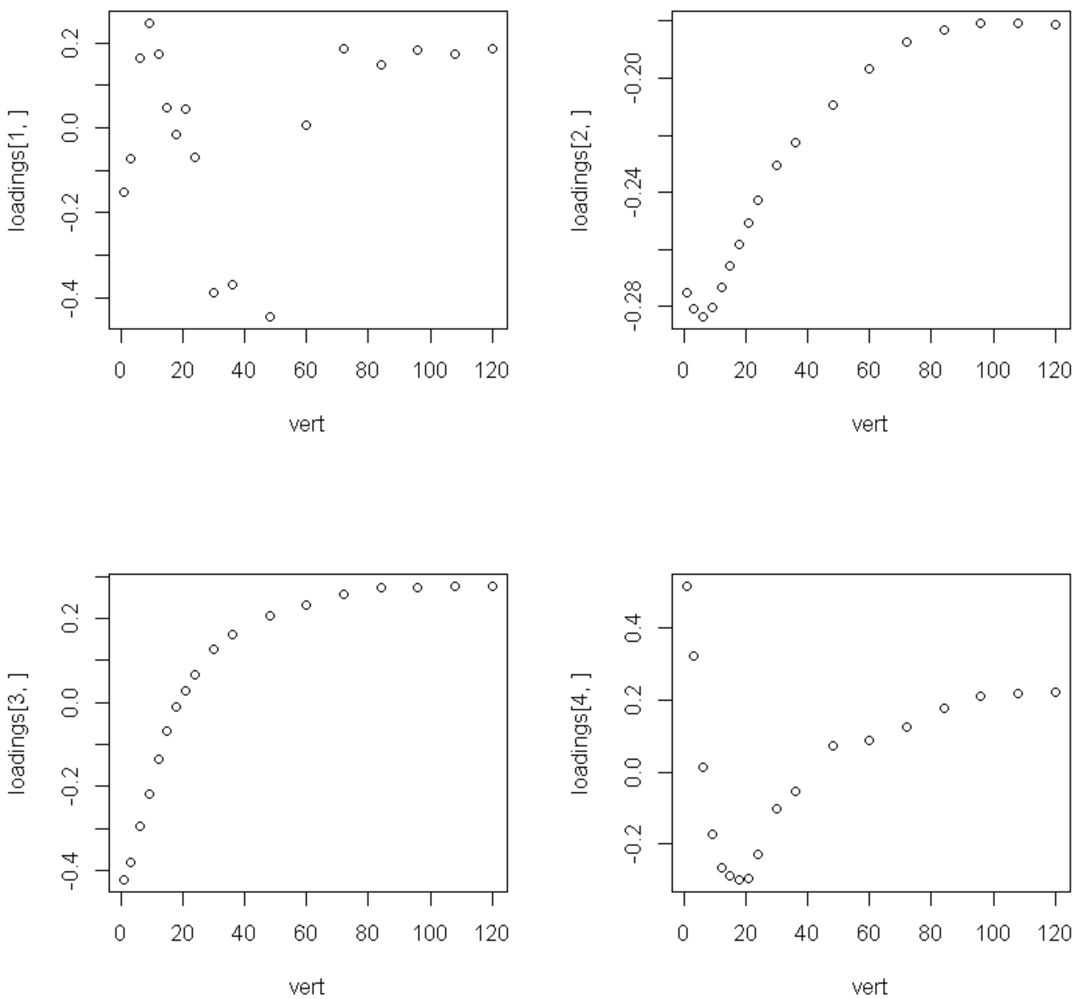


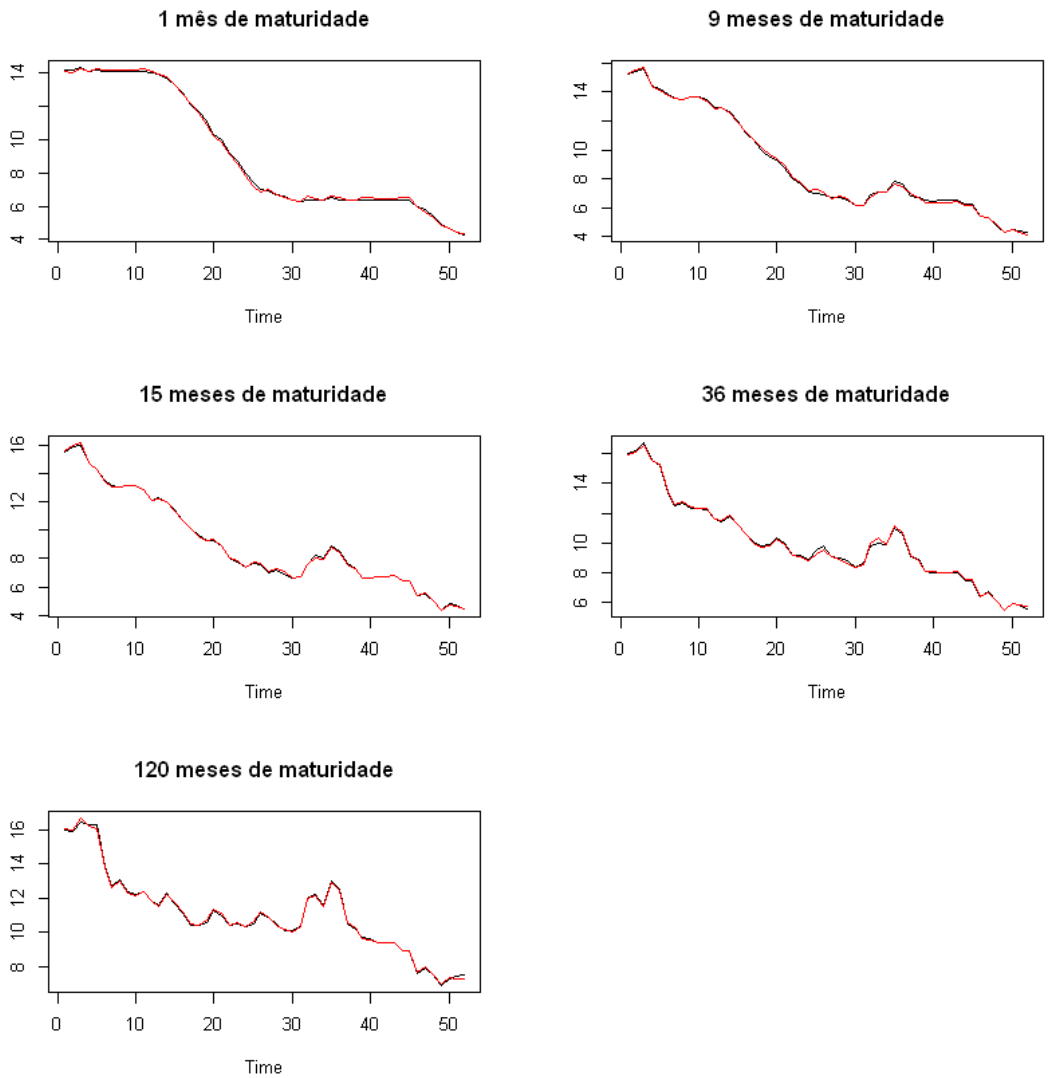

### 2.3 Previsão da curva

[8]:
```
loadings=coef(lm(yieldsm~pc1+pc2+pc3)) ## Qual a dependência de cada yield com os componentes
loadings=as.matrix(coef(lm(yieldsm~pc1+pc2+pc3)))

par(mfrow=c(2,2))
plot(vert,loadings[1,])
plot(vert,loadings[2,])
plot(vert,loadings[3,])
plot(vert,loadings[4,])
fitcurva=cbind(rep(1,length(pc1)),pc1,pc2,pc3)%*%loadings
```




### 2.3.1 Dentro da amostra

[9]:
```
par(mfrow = c(3,2))
ts.plot(cbind(yieldsm[,1],fitcurva[,1]),col=1:2, main = "1 mês de maturidade")
ts.plot(cbind(yieldsm[,4],fitcurva[,4]),col=1:2, main = "9 meses de maturidade")
ts.plot(cbind(yieldsm[,6],fitcurva[,6]),col=1:2, main = "15 meses de maturidade")
ts.plot(cbind(yieldsm[,11],fitcurva[,11]),col=1:2, main = "36 meses de maturidade")
ts.plot(cbind(yieldsm[,18],fitcurva[,18]),col=1:2, main = "120 meses de maturidade")
```




**Medidas de Fit**

[10]:
```
accuracy(yieldsm[,1],fitcurva[,1])
accuracy(yieldsm[,4],fitcurva[,4])
accuracy(yieldsm[,6],fitcurva[,6])
accuracy(yieldsm[,11],fitcurva[,11])
accuracy(yieldsm[,18],fitcurva[,18])
```

| | ME | RMSE | MAE | MPE | MAPE |
|---|---|---|---|---|---|
| Test set | -1.400594e-15 | 0.1028933 | 0.08685193 | 0.03244489 | 1.070679 |

| | ME | RMSE | MAE | MPE | MAPE |
|---|---|---|---|---|---|
| Test set | 1.16147e-15 | 0.0997892 | 0.08177266 | -0.03330599 | 1.068725 |

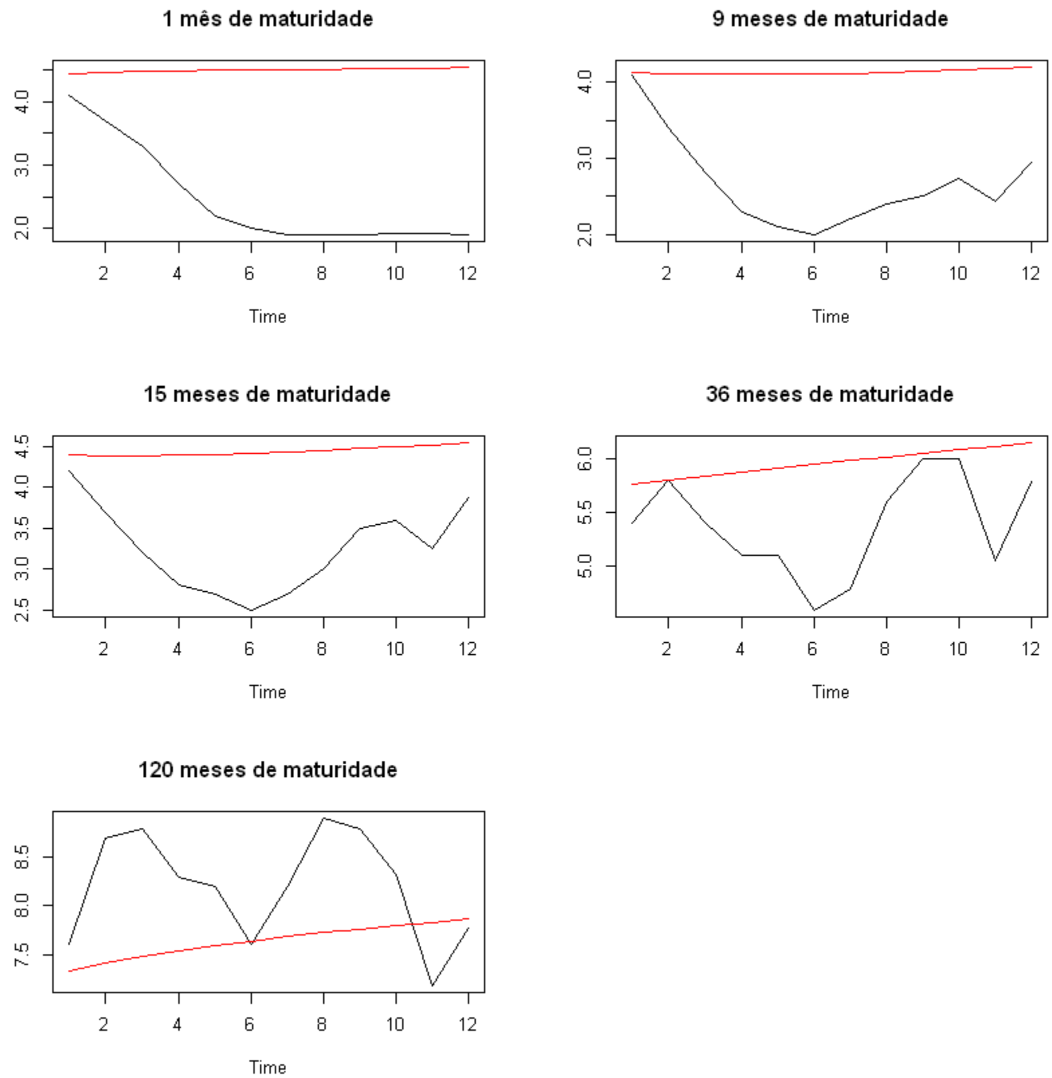
| | ME | RMSE | MAE | MPE | MAPE |
|---|---|---|---|---|---|
| Test set | 2.049645e-16 | 0.06203427 | 0.04484175 | -0.0006551058 | 0.5692642 |

| | ME | RMSE | MAE | MPE | MAPE |
|---|---|---|---|---|---|
| Test set | 2.28877e-15 | 0.1059366 | 0.08248054 | 0.0008229272 | 0.8311413 |

| | ME | RMSE | MAE | MPE | MAPE |
|---|---|---|---|---|---|
| Test set | -1.263949e-15 | 0.08382391 | 0.06837553 | -0.005290842 | 0.6480908 |

### 2.3.2 Fora da amostra

[11]:
```
prevcurva=cbind(rep(1,length(ppc1)),ppc1,ppc2,ppc3)%*%loadings

par(mfrow = c(3,2))
ts.plot(cbind(yields[(T-K+1):T,1],prevcurva[,1]),col=1:2,main = "1 mês de␣
↪maturidade")
ts.plot(cbind(yields[(T-K+1):T,4],prevcurva[,4]),col=1:2,main = "9 meses de␣
↪maturidade")
ts.plot(cbind(yields[(T-K+1):T,6],prevcurva[,6]),col=1:2,main = "15 meses de␣
↪maturidade")
ts.plot(cbind(yields[(T-K+1):T,11],prevcurva[,11]),col=1:2,main = "36 meses de␣
↪maturidade")
ts.plot(cbind(yields[(T-K+1):T,18],prevcurva[,18]),col=1:2,main = "120 meses de␣
↪maturidade")
```




**Medidas de fit**

[12]:
```
accuracy(yields[(T-K+1):T,1],prevcurva[,1])
accuracy(yields[(T-K+1):T,4],prevcurva[,4])
accuracy(yields[(T-K+1):T,6],prevcurva[,6])
accuracy(yields[(T-K+1):T,11],prevcurva[,11])
accuracy(yields[(T-K+1):T,18],prevcurva[,18])
```

| | ME | RMSE | MAE | MPE | MAPE |
|---|---|---|---|---|---|
| Test set | 2.04908 | 2.198518 | 2.04908 | 45.40486 | 45.40486 |

| | ME | RMSE | MAE | MPE | MAPE |
|---|---|---|---|---|---|
| Test set | 1.471025 | 1.577821 | 1.471025 | 35.61504 | 35.61504 |

| | ME | RMSE | MAE | MPE | MAPE |
|---|---|---|---|---|---|
| Test set | 1.193604 | 1.29295 | 1.193604 | 26.87794 | 26.87794 |

| | ME | RMSE | MAE | MPE | MAPE |
|---|---|---|---|---|---|
| Test set | 0.5730352 | 0.7206131 | 0.5731322 | 9.607145 | 9.608819 |

| | ME | RMSE | MAE | MPE | MAPE |
|---|---|---|---|---|---|
| Test set | -0.5609468 | 0.8107577 | 0.6919512 | -7.419292 | 9.093092 |

# 3 Referência.

Esse trabalho usou como base as notas de aula do professor Márcio P. Laurini.