Desafio Magazine Luiza

true

novembro 01, 2017

Abstract

Documento consiste em analisar e interpretar resultados e parâmetros estatísticos obtidos do teste. O teste consiste em classificar distintos produtos e, posteriormente, obter uma previsão da demanda desses produtos nos próximos meses

Códigos dos outputs do documento estarão disponíveis no GitHub. Documento apresentará apenas principais resultados. Tratamento de base, agrupamento, códigos, etc, não serão mostrados aqui

Etapa 1 - Classificação

Para classificação utilizaremos a metodologia de cluster, analisando o melhor valor para o parâmetro "K" para o agrupamento em questão.

Para isso, devemos interpretar como exetamente devemos particionar nossa base.

A base consiste em:

179149 observações distribúidas em 14 variáveis. Sendo as variáveis: order_id, code, quantity, price, pis_cofins, icms, tax_substitution, category, liquid_cost, order_status, capture_date, process_date, process_status, source_channel

Para a realização do cluster, podemos partir de 4 técnicnas distintas:

1. utilizar metodologia de Kmeans com distâncias euclidianas (bastante utilizada para Cluster, utiliza comente de dados quantitativos)
2. Utilizar metodologia de distância de Gower (que clusteriza com base em dados quantitativos e dados qualitativos)
3. Observar a performance de distâncias euclidianas utilizando variávies dummy para "burlar" os dados qualitativos
4. Rodar modelos K-means, Knn, Pam e clara utilizando de bootstrap, k-fold ou reamostragem para analisar o melhor modelo

O mais correto seria utilizar o último método, que tornaria nossa análise mais assertiva. Porém, escolhi o primeiro método por ser mais "ágil" e principalmente ser um método mais confiável "às escuras".

Fazendo o tratamento dos dados, ficamos com:

1. 7 variáveis, sendo elas: code, price, qtd, liquid, pis, icms, tax
2. Distribuídas em 131 observações

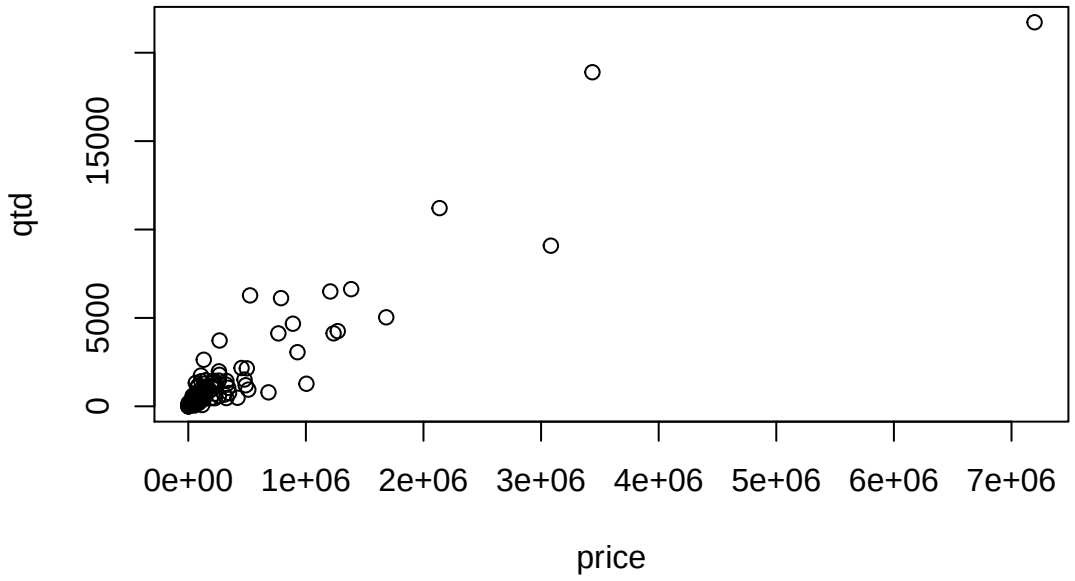
Como a ideia é clusterizar produtos, parti do princípio de que os produtos em si não precisariam ser correlacionados com as outras variáveis, como categoria, source_channel, order_status, etc...

Fizemos então o agrupamento considerando apenas as variáveis acima.

Foi mencionado no documento de teste que os produtos deveriam ser classificados de acordo com sua peculariedade. Considerei no agrupamento que as pecularidades de cada produto eram definidas pelo seu código, onde, caso existisse a peculariedade, o código do produto era alterado.

Caso isso não fosse verdade, poderíamos agrupar pelas taxas de cada produto, onde cada peculariedade do produto era definida de acordo com as taxas cobradas. Porém, ao realizar esse agrupamento e gerar as classificações, acabei me deparando com problemas de performance computacional, voltando então com a ideia citada no parágrafo acima

Fazendo-se o plot entre a quantidade e o preço, conseguimos observar alguns outliers



1. Eliminamos os outliers da base, para não influenciar na classificação. Posteriormente, testaremos o melhor valor de "K" para obtermos os "K" grupos e faremos posteriormente K+1 grupos, com os dados sem a exclusão de outliers.
2. Normalizamos os dados para que o modelo possa ser executado.

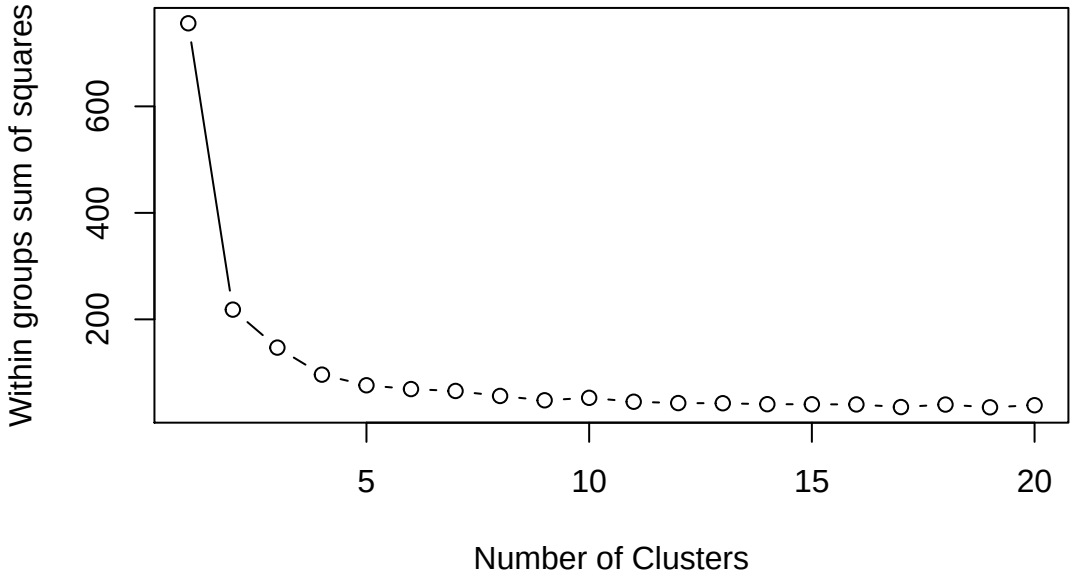
```
> Essa normalização dos dados é necessária para que os pesos das variáveis sejam consideradas ig
```

Fazemos isso por: 1) verificar a significância dos grupos formados sem outlier e dos grupos formados considerando os outliers. Caso a classificação seja boa para ambos, analisamos com K+1 para ver se é plausível a classificação (para que 1 cluster fique com os outliers caso haja necessidade). 2) Caso a estimação sem o outlier na base seja muito melhor, classificaremos outliers como novo grupo manualmente.

Definição do Número de Agrupamentos

Geramos agora a estimativa do parâmetro "K", para descobrir qual o melhor número de cluster dado nossa base. Utilizei, inicialmente, a metodologia "Elbow". Que gerará um plot com "cotovelos" para o número de cluster. Nessa metologia analisamos as distânias dos centros de cada classificação.

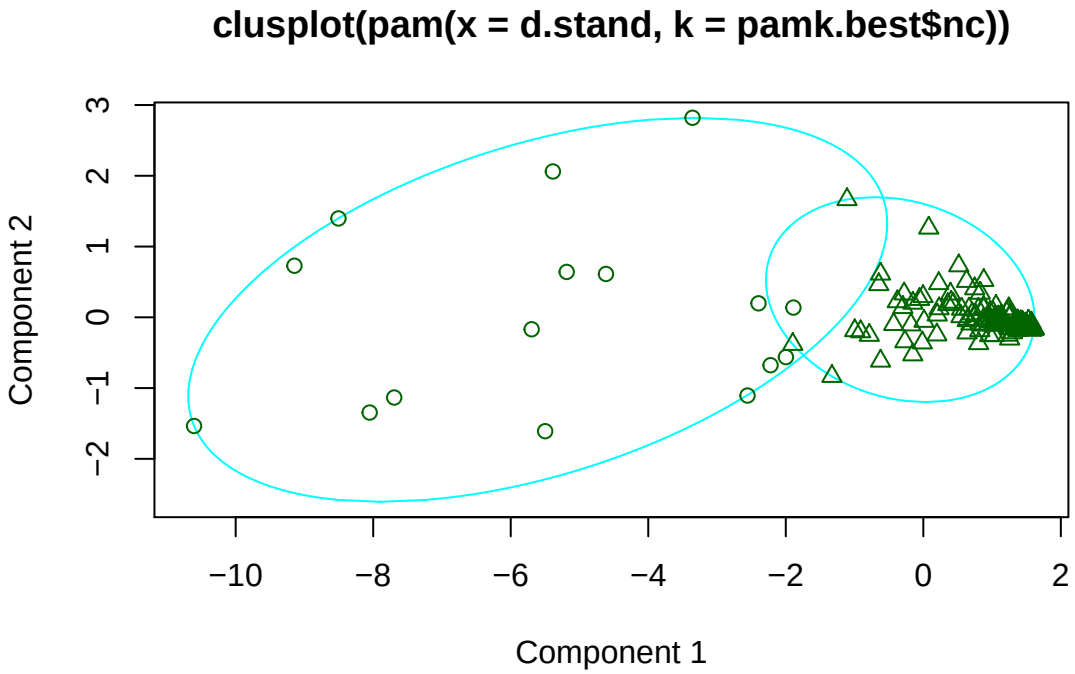
> *Para a análise (visual) do gráfico, vemos o ângulo das retas entre um ponto K e outro ponto K. Consideramos como melhor valor de "k", ângulos próximos a 90 graus (cotovelos), ou seja, quanto menos na vertical a reta estiver entre um ponto e outro, melhor. Isso nos diz o "ganho" que temos entre selecionar os valores de K.*

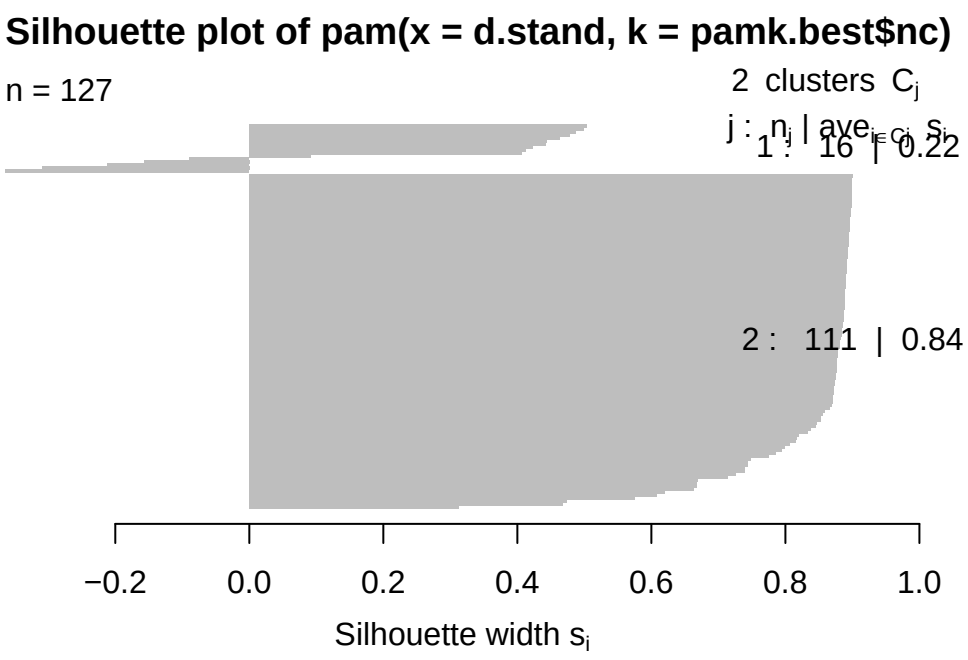


Vemos que o primeiro cotovelo é formado com K = 2, podendo considerar, também, K=3 ou K=5. Obtendo então um range de possibilidades. O modelo acima nos diz que temos um ganho muito bom entre K=1 e K=2, um ganho menor entre K=2 e K=3 e assim sucessivamente. Até que, a partir de K=5 não temos mais ganhos significativos no aumento de cluster.

Agora que temos um range de observações, utilizaremos o método de silhouette para afunilar o range obtido.

> *Silhouette é uma técnica, que, assim como a técnica utilizada acima, nos dá a melhor possibilidade de clusterização. Seu índice varia de 0 a 1, podendo ser considerado como bom agrupamento valores acima de 0.7.*



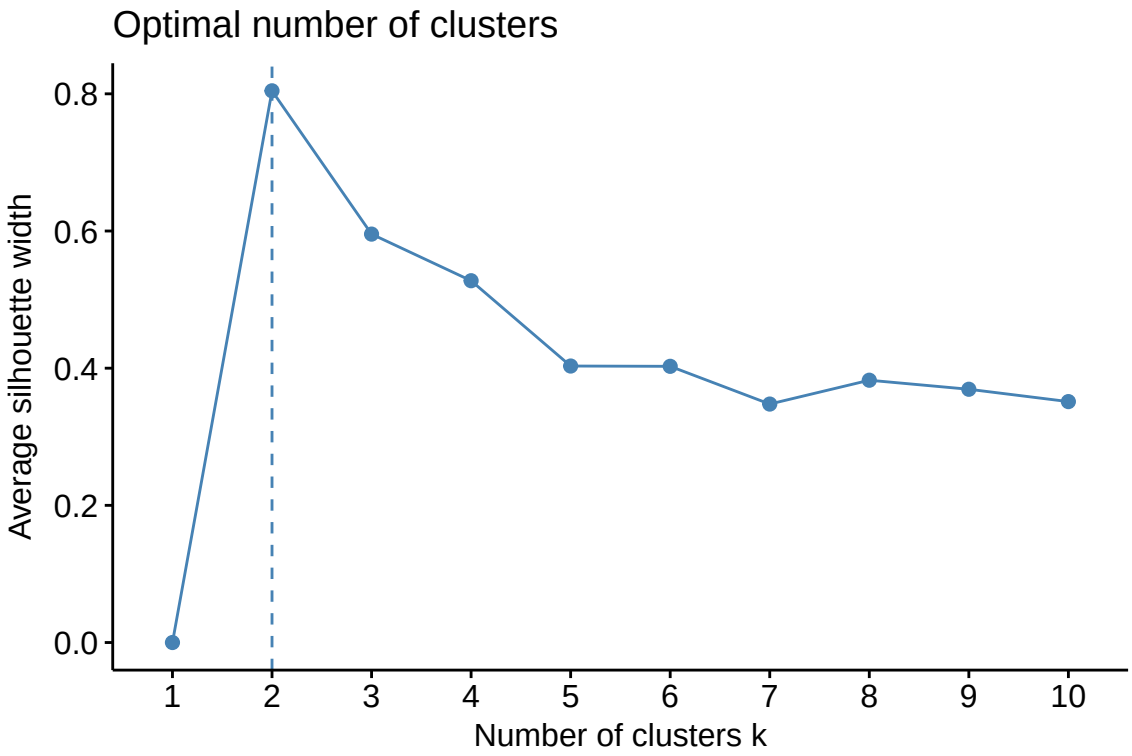



```
## number of clusters estimated by optimum average silhouette width: 2
```

Observamos no primeiro plot que o modelo consegue explicar 96% da variação dos dados. Gerando o plot de análise de componentes principais acima. Observando o segundo plot vemos o gráfico de silhouette, que nos indica também que a melhor clusterização é K=2 grupos. Onde obtemos

o maior ganho. Seu índice de qualidade do ajuste é de 0.76, indicando uma boa modelagem dos dados para k=2.

Utilizando mais metodologias para confirmação do método, veremos graficamente o silhouette e as diferenças entre os grupos. Ao contrário do método de Elbow, o gráfico de silhouette analisa o maior valor entre os agrupamentos.

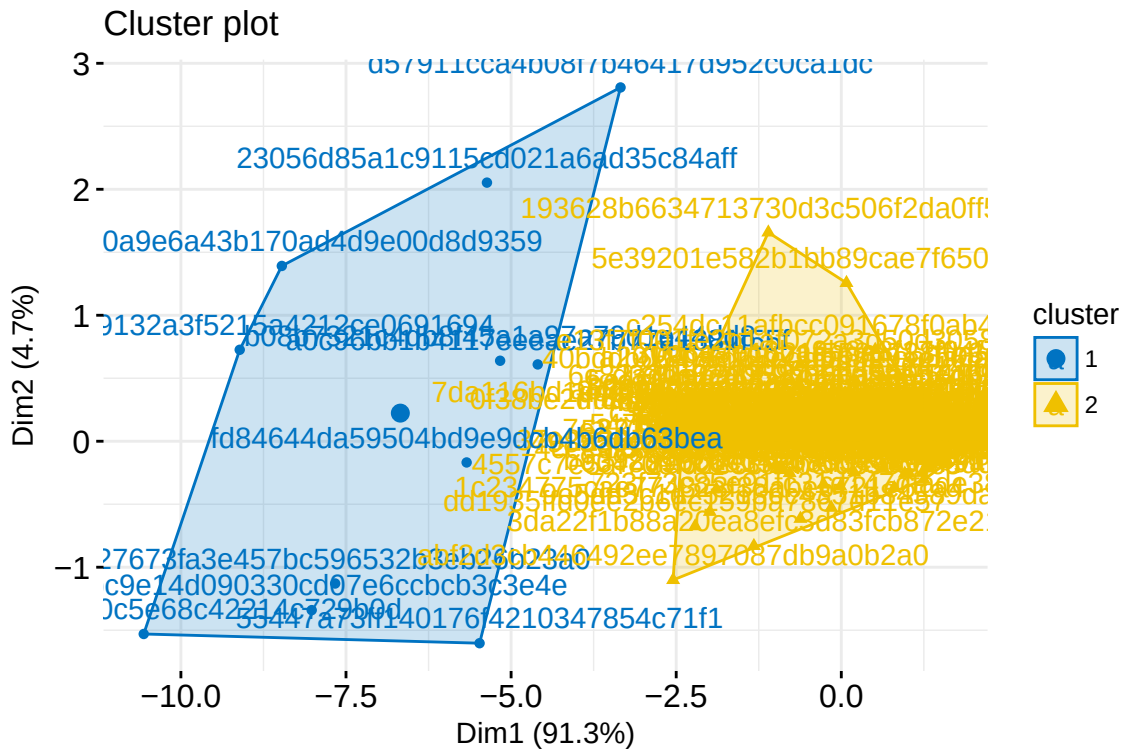


Novamente, vemos que k=2 nos gera o melhor agrupamento, seguido de K=3 e K=4. Por útlimo, geramos um data frame contendo os valores dos índices de silhouette, para compararmos numéricamente os melhores ajustes:

```
##           [,1]
## [1,]        NA
## [2,] 0.8321186
## [3,] 0.7953591
## [4,] 0.6874259
## [5,] 0.6645386
## [6,] 0.6556970
```

Segundo metodologia silhouette para validação de número de agrupamentos, encontramos K=2 e K=3 como sendo os melhores agrupamentos.

Dados todos os índicios acima, utilizaremos K=2 agrupamentos. Fazendo uma verificação de cluster com 2 agrupamentos, temos:



A maioria dos produtos acabam pertencendo ao cluster = 2. Temos: 8.6614% para cluster 1 e 91.3386% para custer 2.

Em resumo: >Retirando-se os outlier da base, temos que a melhor classificação é agruparmos as observações em 2 grupos distintos. >Como nesse caso não podemos deixar os outlier de fora, vamos, agora, voltar à base com os outliers e analisar, de maneira mais resumida, o melhor agrupamento. Onde, o ideal, é que possamos encontrar K=3 grupos, sendo os clusters 1 e 2 referente a não outliers e 1 agrupamento considerando apenas os outlier e valores próximos.

Utilizando apenas da último algoritmo silhouette citado (que nos geram as tabelas com os respectivos índices para cada agrupamento), temos:

```
## [1] 131
```

```
##           [,1]
## [1,]        NA
## [2,] 0.9356110
## [3,] 0.8795951
## [4,] 0.8489611
## [5,] 0.8156488
## [6,] 0.8089458
```

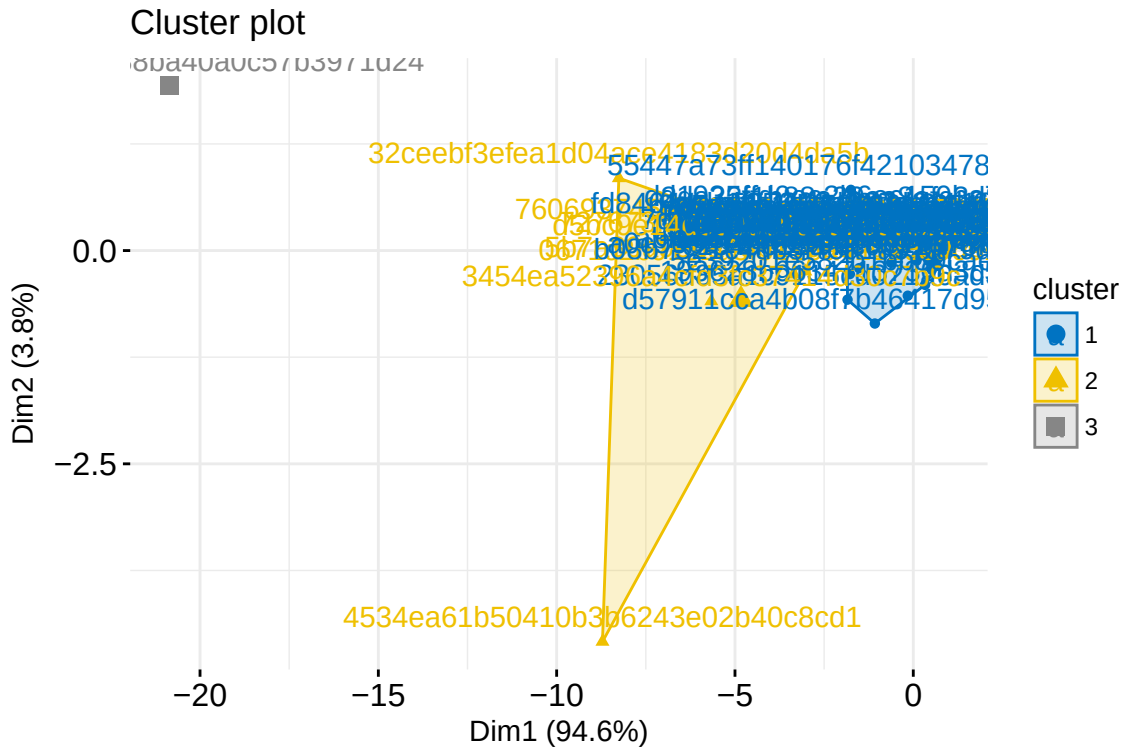
Onde temos uma boa classificação para K=2 novamente, mas k=3 com a base sem retirar os outliers é ainda melhor que k=2 sem os respectivos outliers.

Seguindo a estratégia inicial, utilizaremos K=3. Onde 2 desses cluster conseguem mensurar boa parte dos nossos dados e deixaremos 1 cluster para tratar dados que sejam mais discrepantes.

Observando a nova classificação e gerando um cluster por "kmeans", temos:

K-means, é uma técnica de cllusterização que leva como base os "vizinhos mais próximos", onde parte-se de um ponto médio e então, vai se agrupando item por item por localização mais próxima, até que nõ haja mais pontos próximos. Isso é feito simultâneamente para todos os K grupos. Nosso método iniciará de 25 grupos distintos e iremos agrupar até que sobrem apenas 3 grupos

Clusterização Final utilizando K-means.



Nossa nova classfiicação consegue explicar muito bem a variação dos dados (3,8%+94,6%=~98%). Temos atribuídos a cada grupo: 93.13% para o primeiro cluster, 6% para o segundo cluster e 1% para o terceiro cluster.

Agora que temos os clusters, fazemos as tranformações necessárias para que possamos dar match com a base original exsitente. Com isso, conseguimos separar os produtos por custerizações e fazer as previsões para cada cluster. Pois, independente da categoria do produto ou do produto em si, seu comportamento é semelhante, fazendo com que seja possível fazermos 3 diferentes previsões ao invés de uma previsão para cada produto distinto.

Após o match temos:

- Cluster 1 contém 95969 produtos distintos e 122 inserções distintas na base, contabilizando ao todo 101600 unidades vendidas.
- Cluster 2 contém 62236 produtos distintos e 8 inserções distintas na base, contabilizando ao todo 65729 unidades vendidas.
- Cluster 3 contém 20944 produtos distintos e 1 inserções distintas na base, contabilizando ao todo 21723 unidades vendidas.

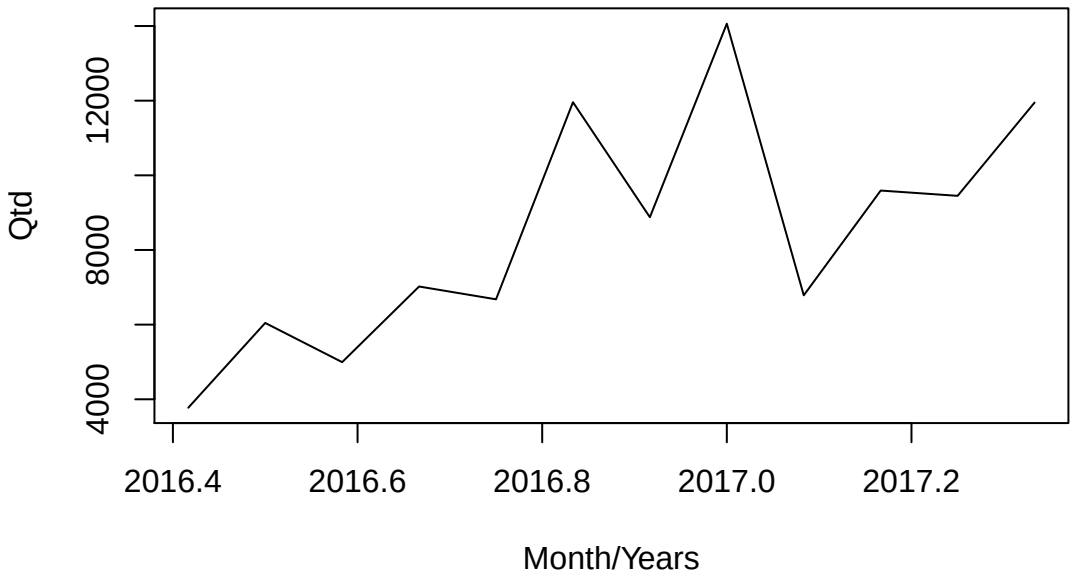
Etapa 2 - Previsões para os meses seguintes

Previsão Cluster 1

Utilizaremos a base gerada acima com suas respectivas variáveis e cluster para filtrar apenas cluster = 1. Posteriormente agruparemos os dados em meses e geraremos uma base de série temporal. Nos retornando:

```
##   month_year value2
## 1 2016-06-01   3774
## 2 2016-07-01   6045
## 3 2016-08-01   4995
## 4 2016-09-01   7021
## 5 2016-10-01   6678
## 6 2016-11-01  11959
```

Plotando a série temporal, obtemos o seguinte comportamento:

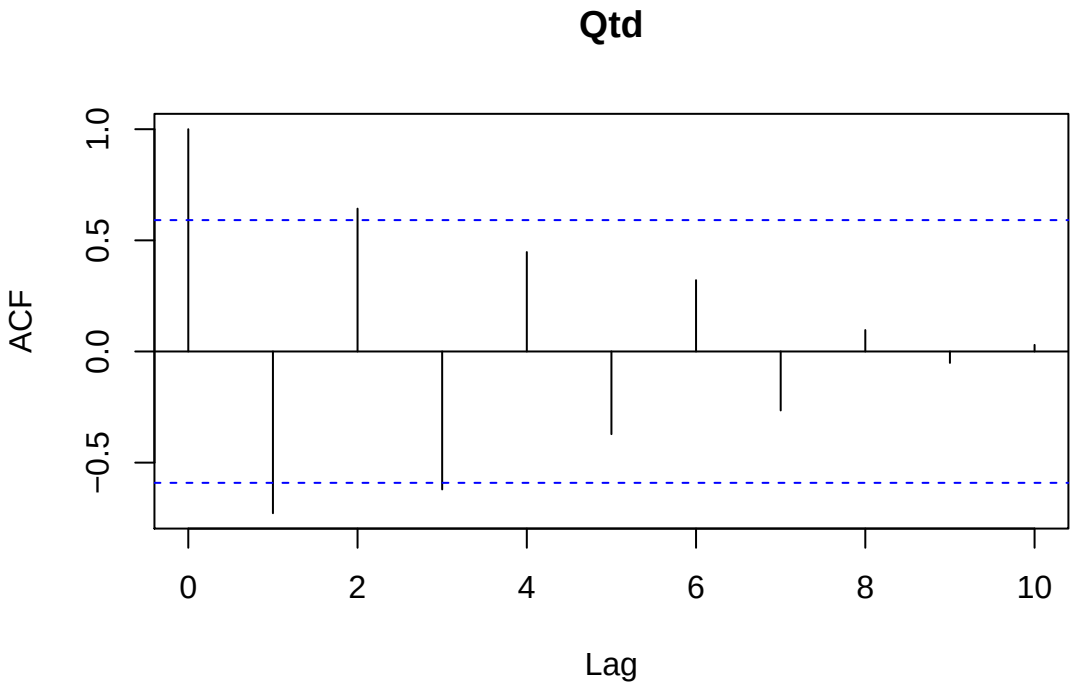


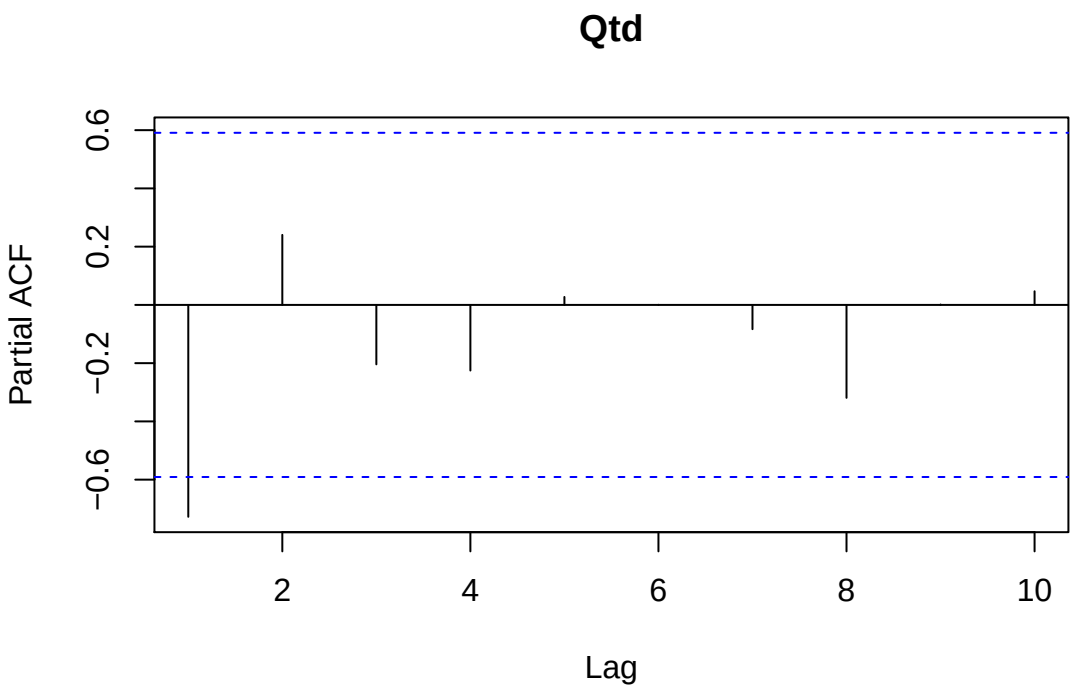
Notamos aqui já um empecílho: Teremos que validar o modelo através de índices estatísticos, ao invés de utilizarmos validações de treino e teste. Isso ocorre pois, se particionarmos a validação entre treino e teste, teríamos que retirar o mês de março/2017, estimar a série temporal e, posteriormente, plotar o comparativo. Porém, temos apenas uma obervação para o mês, não gerando o aprendizado necessário para o teste através de simulação por termos o período de 12 meses fechado. Caso tivéssemos o período de 18 meses por exemplo, conseguiríamos rodar essa forma de análise.

Analisando a ACF e PACF: > *Tanto a ACF (autocorrlation function) quanto a PACF () são utilizados para*

traçar autocorrelações. Esse gráfico nos permite entender se a série é aleatória ou possui alguma tendência ou sazonalidade.

> *A interpretação do gráfico é simples, as linhas tracejadas em azul indicam ruídos brancos/aleatoriedade. Quando todas ou apenas a primeira linha na vertical estiverem dentro desse range, teremos indícios de que a série foi bem modelada, restando apenas resíduos aleatórios que não podem ser modelados pelo algoritmo. Quando temos linhas verticais que excedem as linhas horizontais pontilhadas azul, temos indícios de que há sazonalidade ou tendência nos dados, que devem ser tratados antes da execução do algoritmo. Quanto mais a reta passar dessa linha tracejada, maior a sazonalidade ou tendência.* * A sazonalidade só é realmente significativa quando obervamos padrões ao longo das "lags" de tempo.*





Vemos pelo primeiro gráfico que temos uma sazonalidade ou tendência bastante pequena, pois, apesar de termos observações que extrapolam a linha tracejada, ela não possui um padrão específico de sazonalidade. Mas, por termos essas linhas extrapolando, sabemos que existe uma tendência. No segundo gráfico, observamos que essa tendência realmente existe e que será necessário retirar a tendência para análise do modelo. Como temos apeans 1 valor fora da linha tracejada, temos que apenas uma tendência foi identificada.

Rodaremos o modelo de ARIMA, passando uma transformação em log(10) para estimar os dados. Essa transformação é necessária para tornar os dados estacionários (sem a tendência).

> *ARIMA é um modelo auto regressivo (AR), com diferenciação (I) e médias móveis (MA) (AR) indica que que a variável de interesse é correlacionada com o tempo anterior (MA) indica que o erro de regressão é na verdade uma combinação dos termos dos erros*

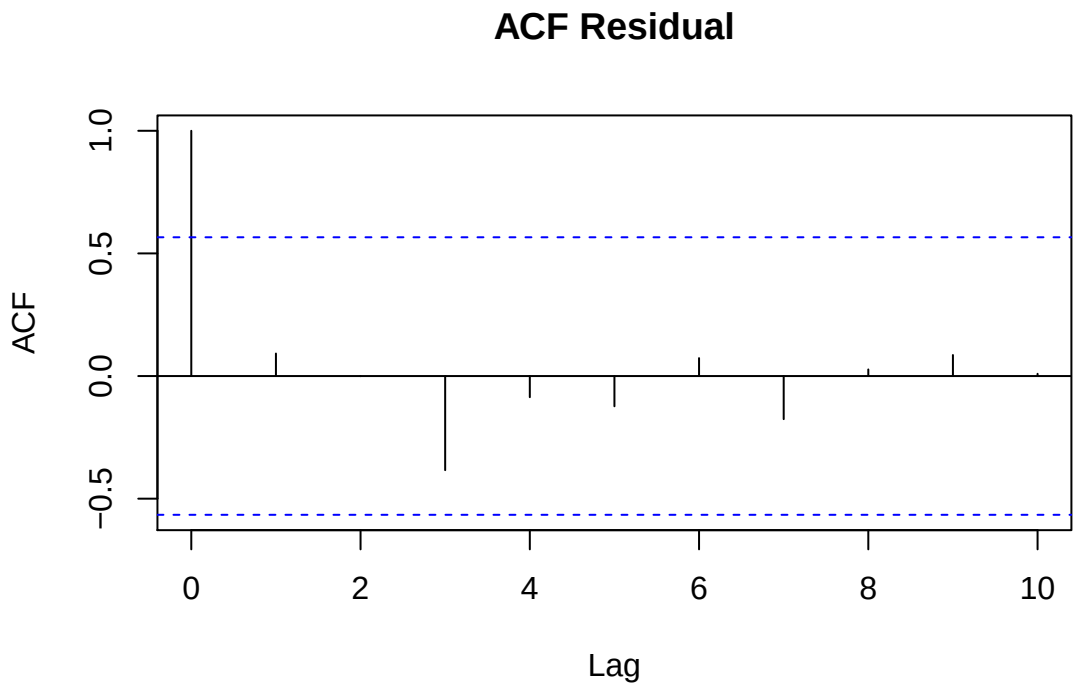
```
## Series: log10(data)
## ARIMA(1,1,0)
##
## Coefficients:
##           ar1
##       -0.6925
## s.e.   0.2078
##
## sigma^2 estimated as 0.01663:  log likelihood=7.12
## AIC=-10.24   AICc=-8.74   BIC=-9.44
##
## Training set error measures:
##                      ME      RMSE       MAE      MPE     MAPE MASE
```

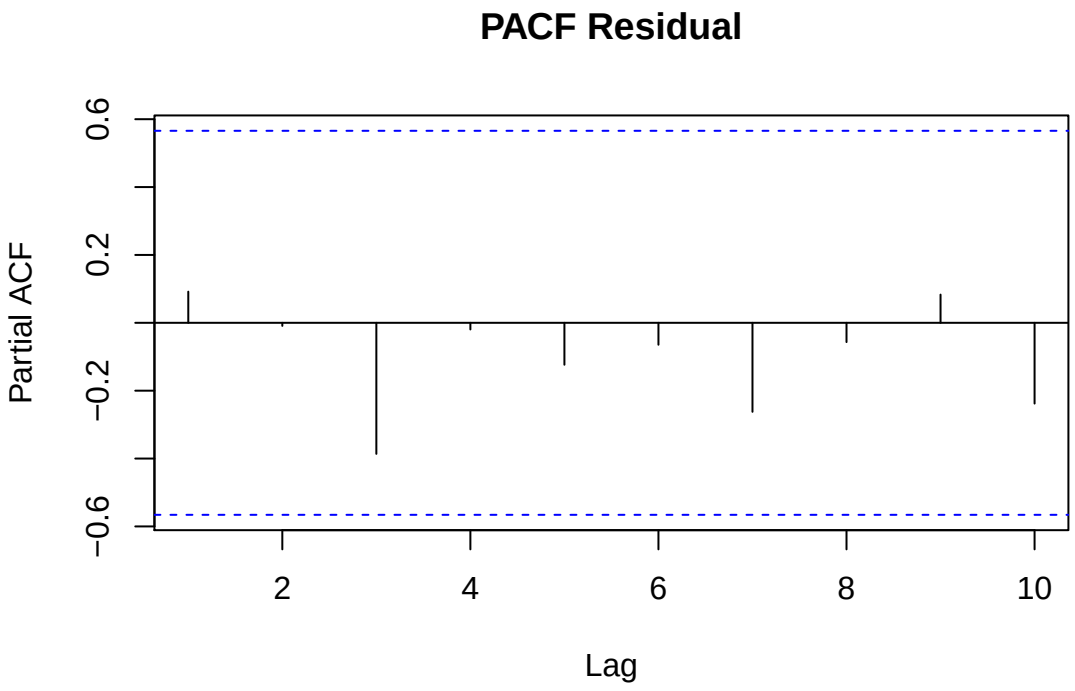
```
## Training set 0.0602641 0.1177263 0.1014252 1.514534 2.577497  NaN
##                     ACF1
## Training set 0.0918617
```

O modelo gerado nos retorna ARIMA(1,1,0), onde ARIMA(AR,I,MA). Temos, portanto, que a variável é correlacionada com o tempo anterior e possui uma tendência, utilizada para estimar o modelo.

> *A modelagem de ARIMA consiste em ajustar várias combinações de modelos ARIMA(1,1,1),ARIMA(1,1,0),ARIMA(1,0 etc... e nos retorna o melhor modelo segundo o critério AIC e BIC.*
>
> *AIC e BIC são medidas comparativas, seus valores sozinhos não possuem muita interpretação.*
> *AIC e BIC são, de forma mais generalizada, funções de custo (ganho e perda)*

Validando o modelo, plotaremos novamente os gráficos de ACF e PACF. Para indicativo de bom ajuste, devemos observar todos os valores dentro do range entre as linhas tracejadas em azul. (Para ACF a primeira linha vertical normalmente será maior, podemos desconsiderar a primeira lag)



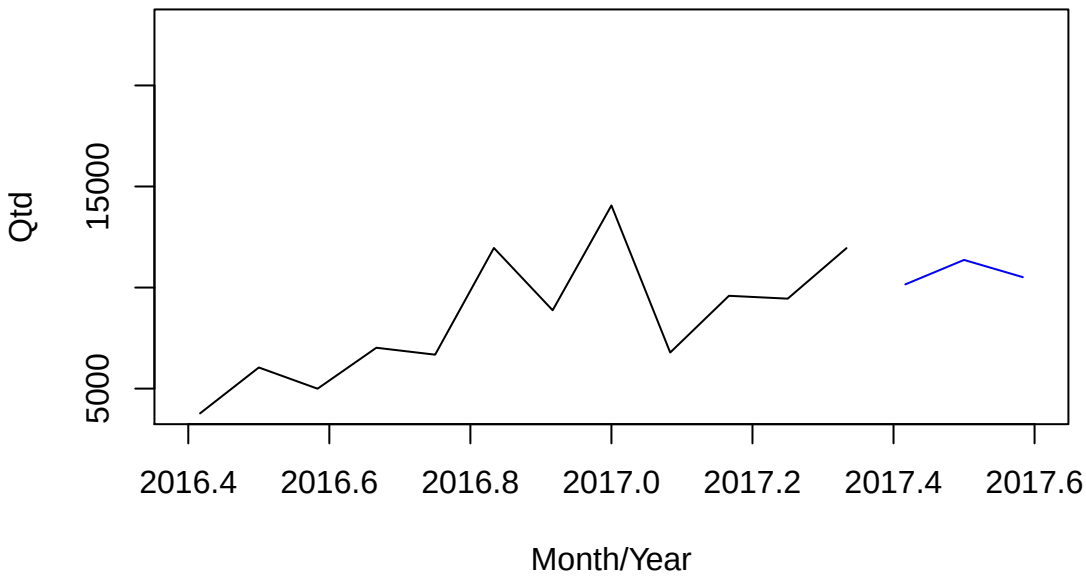


Temos indícios de que a modelagem foi boa. Fazendo um segundo teste, buscaremos os resíduos (parte dos dados que não foi possível modelar) e, estimaremos um novo modelo ARIMA. O resultado que queremos obter é ARIMA(0,0,0) que indica ruído branco e indica que os resíduos estão normalizados. Caso obtenhamos outro modelo que não ARIMA(0,0,0) podemos adidiconar esse modelo à previsão, fazendo um *boosting* do modelo.

```
## Series: pred2$residuals
## ARIMA(0,0,0) with non-zero mean
##
## Coefficients:
##         mean
##       0.0603
## s.e.  0.0292
##
## sigma^2 estimated as 0.01116:  log likelihood=10.47
## AIC=-16.94   AICc=-15.6   BIC=-15.97
```

Obtemos ARIMA(0,0,0), portanto, a modelagem respondeu bem aos dados.

Utilizaremos, portanto, ARIMA(1,1,0) para predizer os próximos 3 meses.



E seus respectivos valores:

```
##           Jun      Jul      Aug
## 2017 10156.96 11366.68 10514.54
```

Portanto, para os 3 meses seguintes para o cluster = 1, temos a previsão de demanda de:

```
## [1] "Junho:  10157 itens"
```

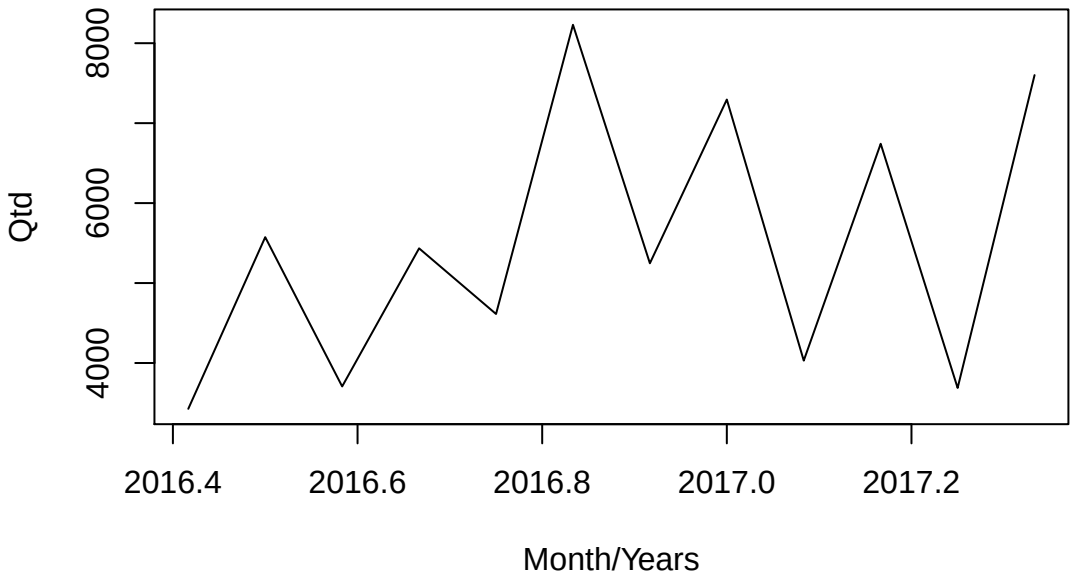
```
## [1] "Julho:  11367 itens"
```

```
## [1] "Agosto:  10515 itens"
```

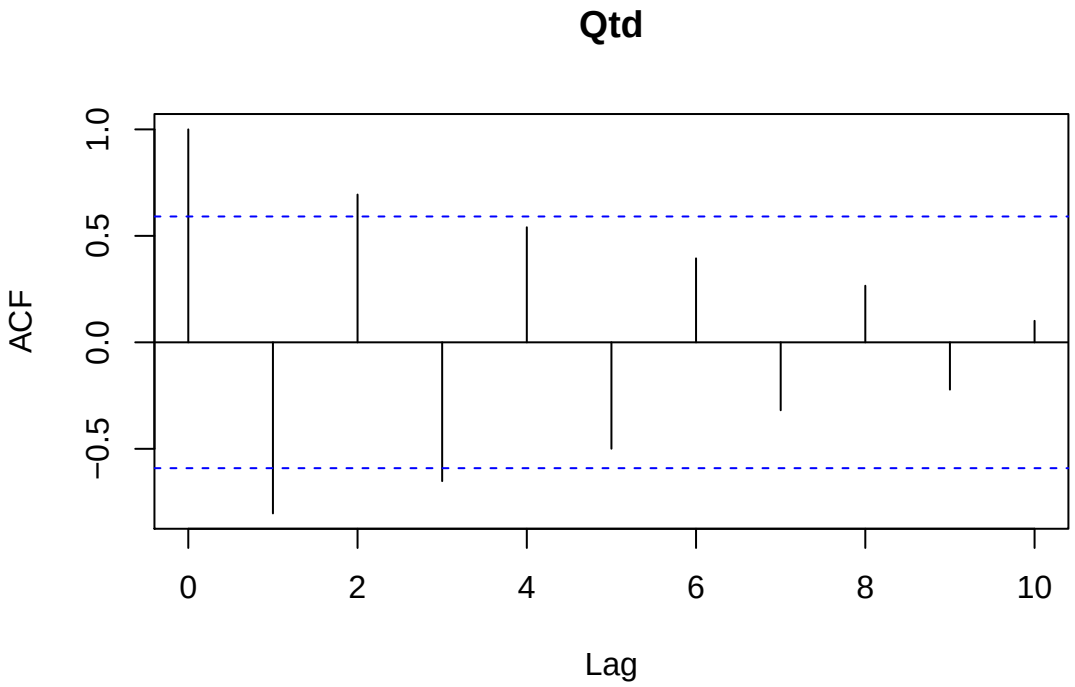
Repetiremos o mesmo processo para Cluster = 2 e Cluster = 3, porém, sem divagar sobre as metodologias utilizadas.

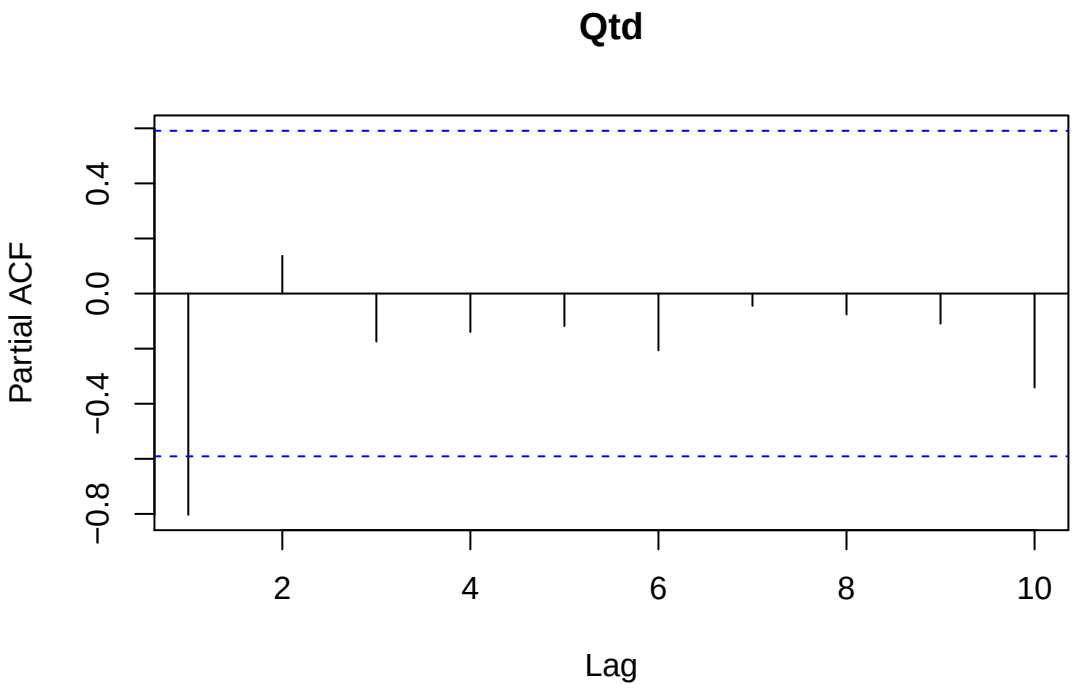
Cluster 2

Fazendo os mesmos tratamento e buscando na base apenas cluster = 2, temos o seguinte plot:



Onde observamos muitas oscilações entre os meses. Plotando ACF e PACf temos:





Em relação à série temporal de cluster = 1, observamos um comportamento parecido com tendências e sazonalidades ligeiramente maiores. Observando a PACF, vemos um comportamento semelhante à d e cluster = 1. Indicando uma tendência ao longo do tempo.

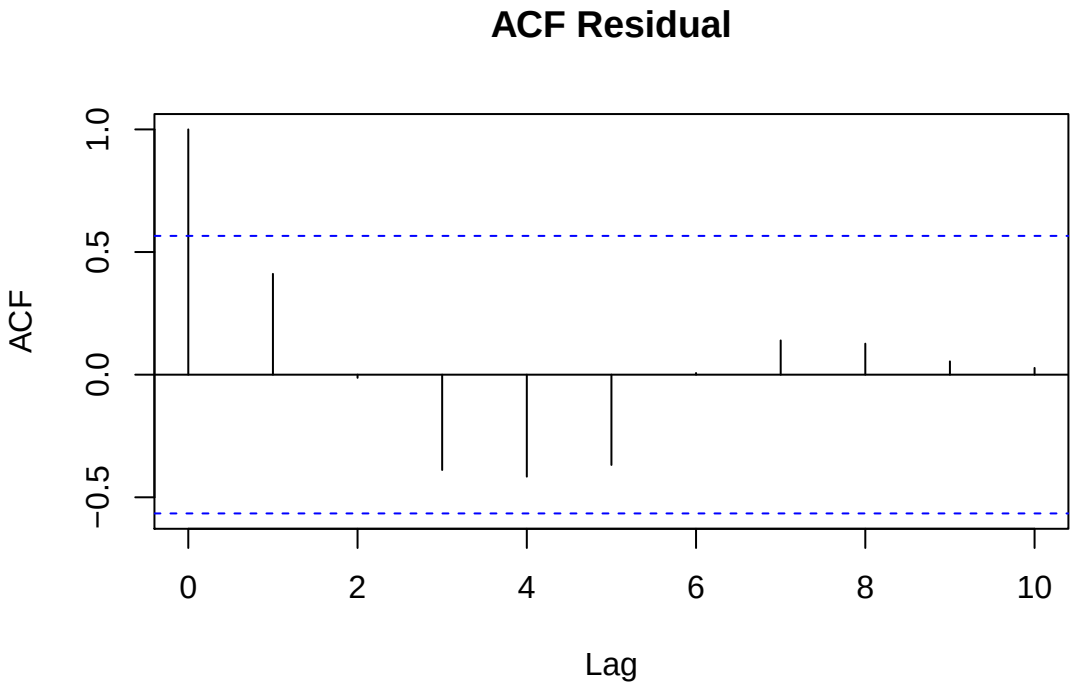
Modelando os dados temos:

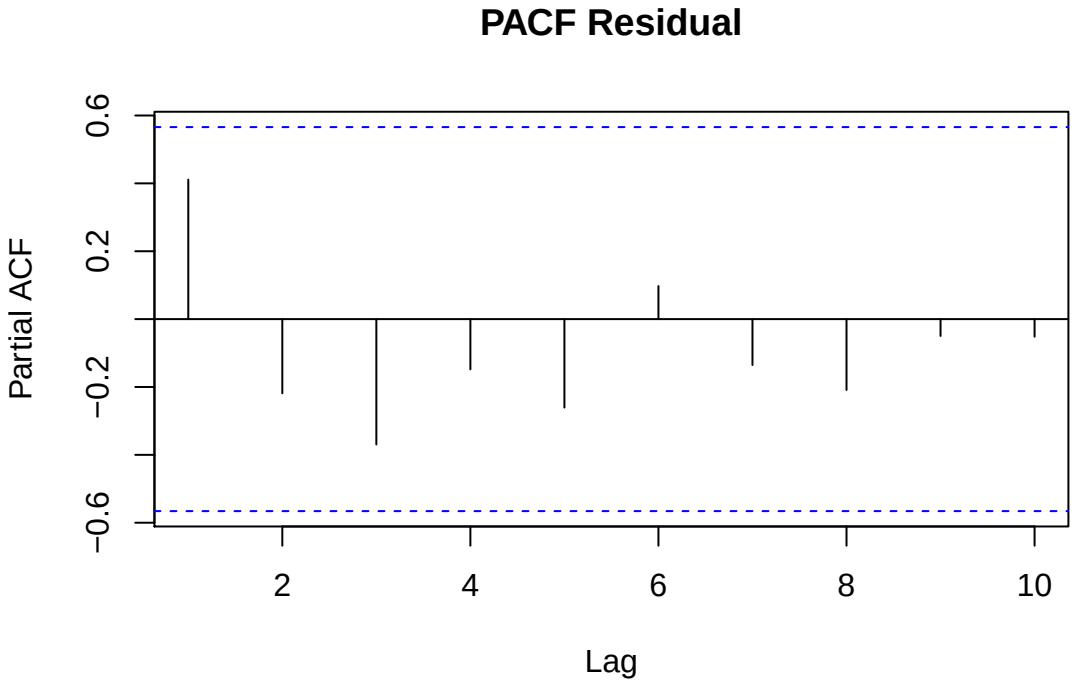
```
## Series: log10(data)
## ARIMA(2,0,0) with non-zero mean
##
## Coefficients:
##           ar1     ar2    mean
##       -0.2026  0.6949  3.7075
## s.e.   0.1917  0.2002  0.0372
##
## sigma^2 estimated as 0.007515:  log likelihood=13.09
## AIC=-18.19   AICc=-12.47   BIC=-16.25
##
## Training set error measures:
##                       ME       RMSE       MAE       MPE     MAPE MASE
## Training set 0.006633391 0.07507485 0.0595423 0.1331442 1.596059  NaN
##                    ACF1
## Training set 0.4109744
```

Foi identificado pela função que o melhor modelo dentre as combinações possíveis é ARIMA(2,0,0). Dando indício de que o tempo atual possui correlação com o tempo anterior. Ou seja, foi modelado, identificado 1 correlação com o tempo anterior, retirada essa correlação, modelado novamente e identificado uma nova correlação. Essas 2 correlações podem ser: 1 correlação do tempo atual com

o tempo anterior dado mês a mês e 1 correlação do tempo atual com o tempo anterior dado a cada 3 meses (trimestral).

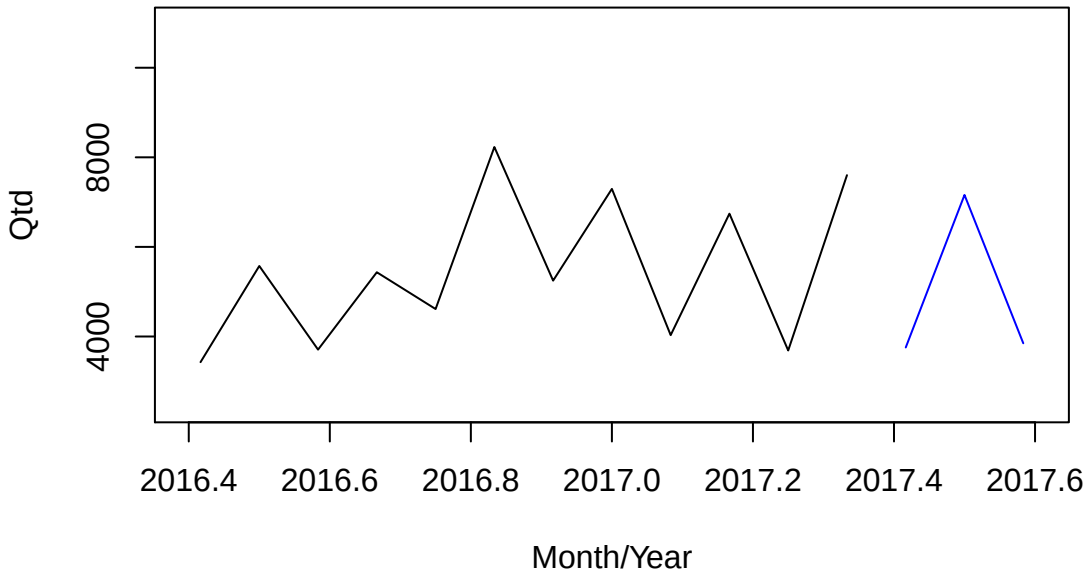
Analisando a qualidade do ajuste do modelo através do plot de ACF, PACF e ARIMA através dos resíduos. Para identificar algum novo padrão.





```
## Series: pred2$residuals
## ARIMA(0,0,0) with zero mean
##
## sigma^2 estimated as 0.005636:  log likelihood=14.04
## AIC=-26.09   AICc=-25.69   BIC=-25.6
```

Obtemos ruído branco, aleatoriedade dos dados. Todos os gráficos se mostram OK nas suas interpretações e o modelo gerado com os resíduos ARIMA(0,0,0) reforçam que a modelagem responde bem aos dados.



Fazendo as predições:

E seus respectivos valores:

```
##           Jun      Jul      Aug
## 2017 3754.964 7160.044 3848.535
```

Portanto, para os 3 meses seguintes para o cluster = 1, temos a previsão de demanda de:

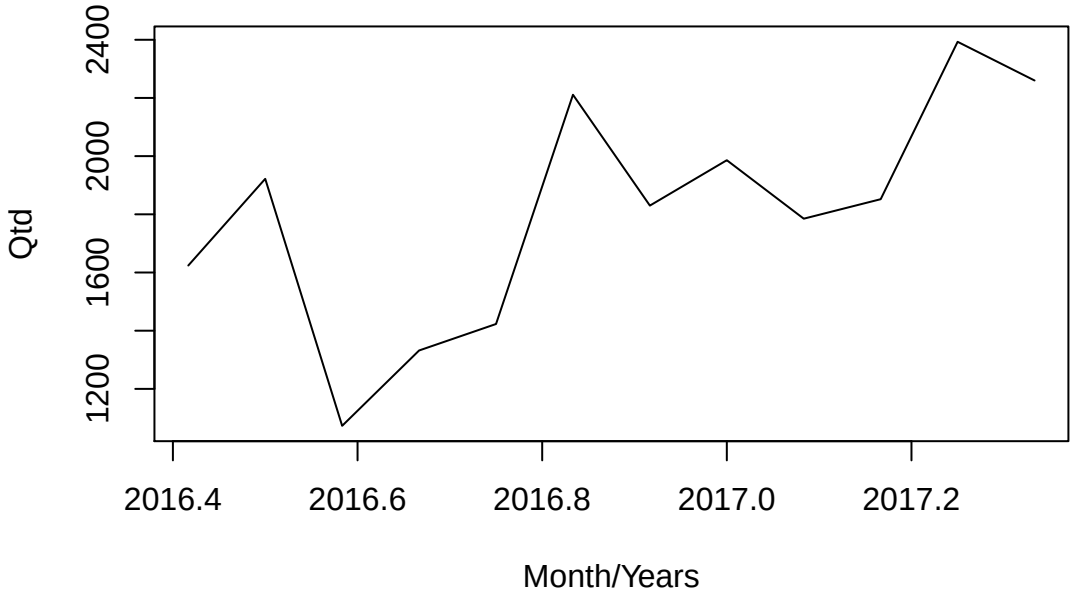
```
## [1] "Junho:  3755 itens"
```

```
## [1] "Julho:  7160 itens"
```

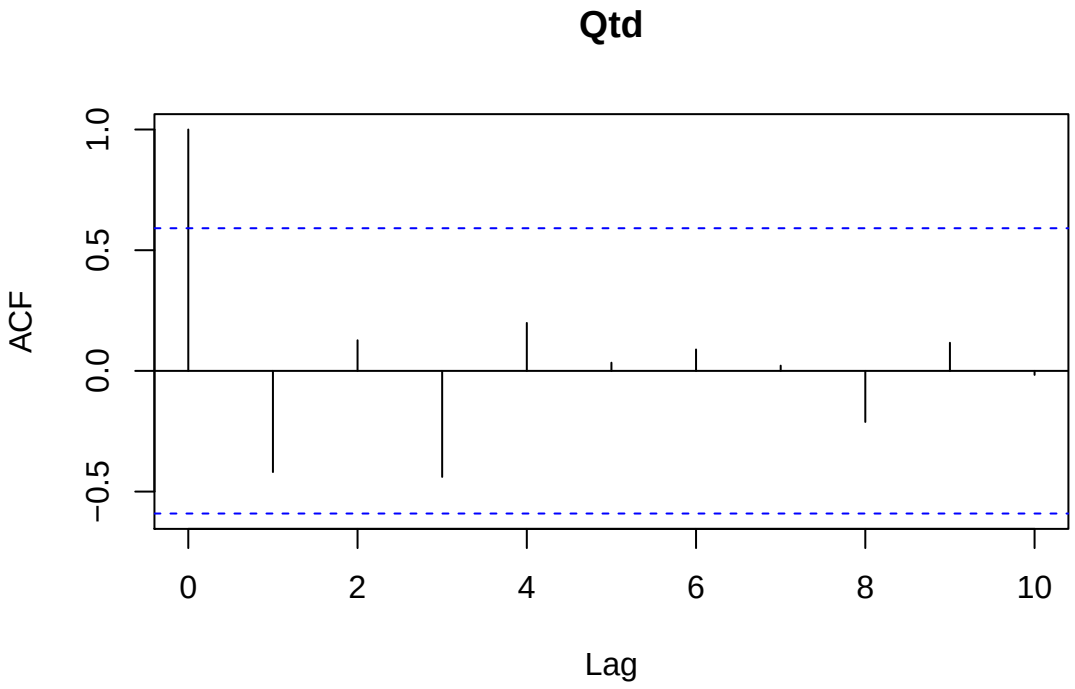
```
## [1] "Agosto:  3849 itens"
```

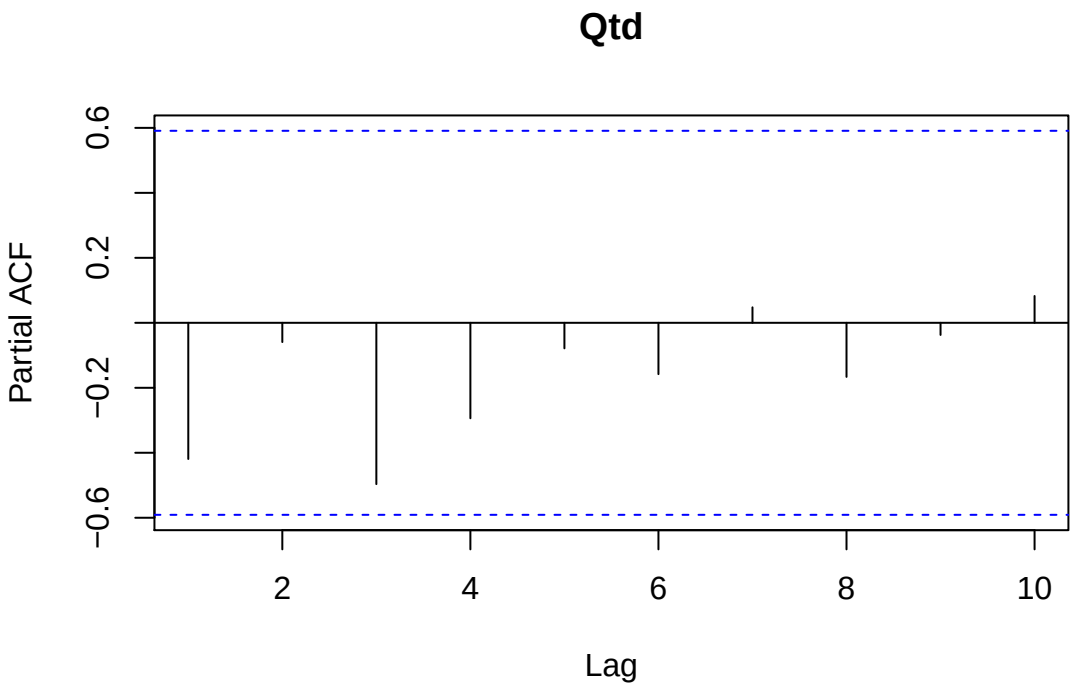
Cluster 3

Fazendo os mesmos tratamento e buscando na base apenas cluster = 3, temos o seguinte plot:



Onde observamos tendência crescente ao longo dos meses com alguns vales. Plotando ACF e PACf temos:





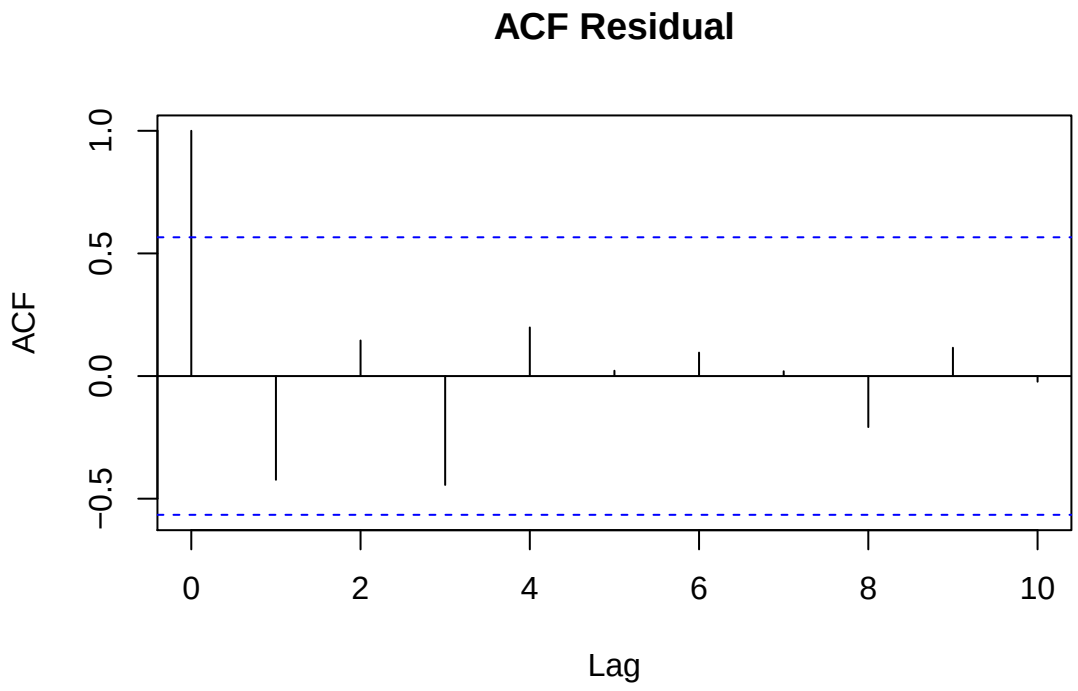
Porém, a falta de "lags" ultrapassando a linha tracejada pode dar indício de que possua mais aleatoriedade do que padrões na série.

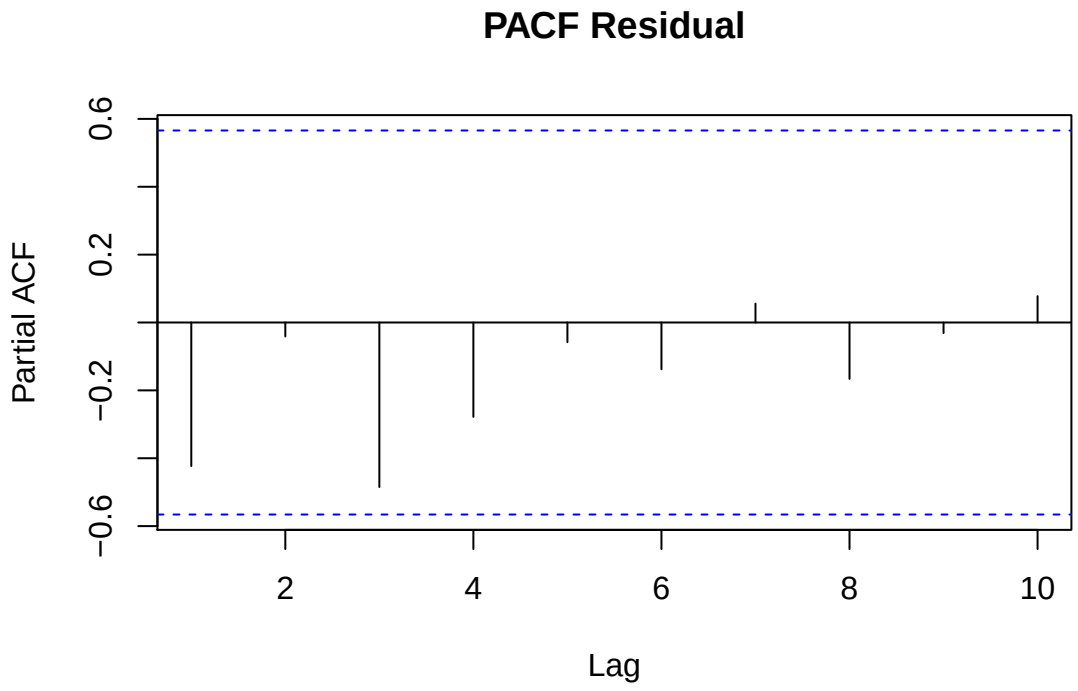
Modelando os dados temos:

```
## Series: log10(data)
## ARIMA(0,1,0)
##
## sigma^2 estimated as 0.01265:  log likelihood=8.43
## AIC=-14.86   AICc=-14.41   BIC=-14.46
##
## Training set error measures:
##                       ME      RMSE        MAE      MPE     MAPE MASE
## Training set 0.01222775 0.1076791 0.07997218 0.315873 2.488622  NaN
##                     ACF1
## Training set -0.4229487
```

Para o cluster = 3, obtemos ARIMA(0,1,0). Esse modelo em particular recebe o nome de "passeio aleatório com deriva", ou seja, a quantidade ao longo dos meses é aleatória segundo o modelo, com parâmetro não aleatório igual a sua tendência. Ou seja, basciamente é a tendência quem dita o forecast.

Analisando a qualidade do ajuste do modelo através do plot de ACF, PACF e ARIMA através dos resíduos. Para identificar algum novo padrão.

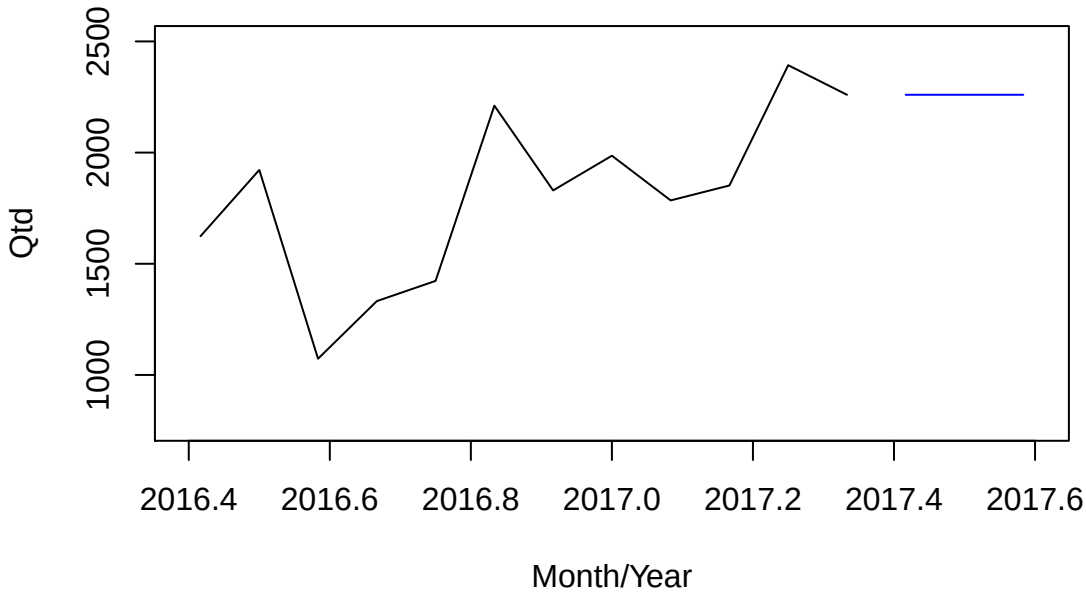





```
## Series: pred2$residuals
## ARIMA(0,0,0) with zero mean
##
## sigma^2 estimated as 0.01159:  log likelihood=9.72
```

```
## AIC=-17.43   AICc=-17.03   BIC=-16.95
```

Obtemos ruído branco, aleatoriedade dos dados. Todos os gráficos se mostram OK nas suas interpretações e o modelo gerado com os resíduos ARIMA(0,0,0) reforçam que a modelagem responde bem aos dados.



Fazendo as predições:

E seus respectivos valores:

```
##       Jun  Jul  Aug
## 2017 2260 2260 2260
```

Portanto, para os 3 meses seguintes para o cluster = 1, temos a previsão de demanda de:

```
## [1] "Junho:  2260 itens"
```

```
## [1] "Julho:  2260 itens"
```

```
## [1] "Agosto:  2260 itens"
```