
Objetivo

Apresentar estudo inicial realizado para domínio da plataforma de computação estatística "R". Os dados utilizados foram extraídos a partir de BI (Banco de Inteligência de Negócio) criado pela CTI em Fevereiro de 2017, sobre contas médicas e suas dimensões, com uma série temporal entre 2014 até a última referência fechada e atualizada, em março de 2017.

ATENÇÃO

O conteúdo deste documento é de natureza confidencial e não deverá ser circulado sem expressa autorização da Coordenação Geral do Planserv.

Não será objetivo deste relatório tratar em detalhes as técnicas estatísticas utilizadas, com exceção de pontos relevantes para demonstrar a flexibilidade da ferramenta, como por exemplo a abordagem para verificar problemas de autocorrelação em séries temporais. A teoria estatística envolvida, embora interessante para alguns públicos, tornaria o relato demasiado extenso.

Conteúdo

1. Descrição da Tecnologia
2. Introdução ao Estudo Proposto
3. Análise de Correlação
4. Gráficos para Comunicação
5. Direcionamento no Estudo
6. Análise Detalhada sobre Consulta e Exames
 - 6.1 Preparação
 - 6.2 Seleção de Escopo pelo Indicador de Internação
7. Regressão Linear
 - 7.1 Modelo Inicial
 - 7.2 Aspectos e Avaliação da Regressão
 - 7.3 Considerações sobre Séries Temporais
 - 7.4 Avaliação Específica para Objetos do Pacote "Orcutt"
 - 7.5 Alternativa de Transformação
8. Outros Experimentos

9. Visualizações de Dados

10. Conclusão

Descrição da Tecnologia

Retorno para Conteúdo

A ferramenta foi criada em 1992, por Ross Ihaka e Robert Gentleman, professores de estatística na Universidade de Auckland, Nova Zelândia, como um desenvolvimento comunitário de fonte livre (gratuito) sobre a linguagem “S”, núcleo fundamental da plataforma estatística S-Plus, comercializada.

“R” pode ser descrita como uma linguagem de programação interpretativa (não é necessário compilar seus scripts), possui integração com sistemas de gerenciamento de bancos de dados, outras linguagens (como Python), outras plataformas estatísticas (importações de SPSS, SAS, Stata, Minitab), MSEXcel e arquivos planos.

Desde sua criação, seu conteúdo básico foi extensamente ampliado pela comunidade por meio da construção de pacotes que podem facilmente ser anexados ao ambiente básico, e que ampliam as funcionalidades da ferramenta contemplando a maioria das práticas estatísticas conhecidas, estilos e ferramentas para apresentações gráficas, manipulação de dados, controles de qualidade, funções para facilitar e simplificar o desenvolvimento de seu código, entre outras.

Como exemplo do funcionamento desta estrutura modular, abaixo estão os pacotes que foram utilizados no estudo, contendo cada um várias das funções aplicadas na geração dos resultados demonstrados abaixo:

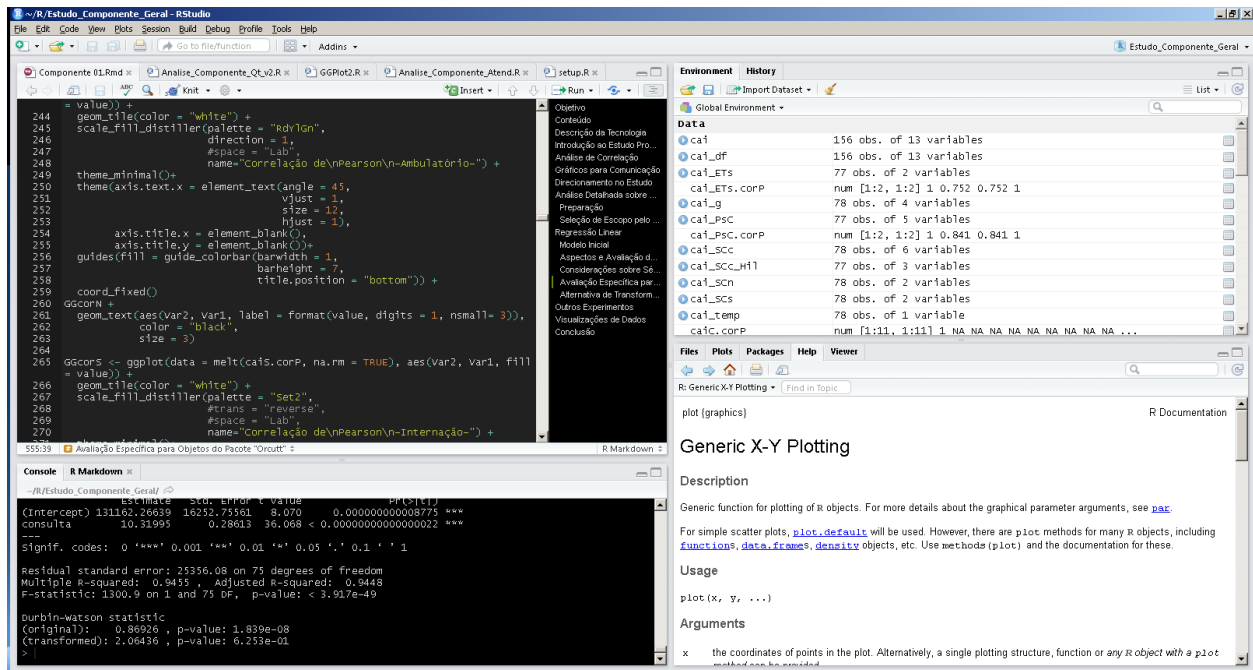
```
## Opções iniciais de configuração do ambiente -----
knitr::opts_chunk$set(echo = TRUE)
options(scipen=999)
#### -----
library("reshape2", lib.loc=~R/win-library/3.4")
library("readxl", lib.loc=~R/win-library/3.4")
library("qcc", lib.loc=~R/win-library/3.4")
library("modelr", lib.loc=~R/win-library/3.4")
library("dplyr", lib.loc=~R/win-library/3.4")
library("ggplot2", lib.loc=~R/win-library/3.4")
library("readr", lib.loc=~R/win-library/3.4")
library("tibble", lib.loc=~R/win-library/3.4")
library("tidyr", lib.loc=~R/win-library/3.4")
library("purrr", lib.loc=~R/win-library/3.4")
library("forcats", lib.loc=~R/win-library/3.4")
library("plotrix", lib.loc=~R/win-library/3.4")

## ** Pacotes básicos que compõe o Tidyverse poderiam ser ativados com uma unica linha:
#library("tidyverse", lib.loc=~R/win-library/3.4")
#Loading tidyverse: ggplot2
#Loading tidyverse: tibble
#Loading tidyverse: tidyr
#Loading tidyverse: readr
#Loading tidyverse: purrr
#Loading tidyverse: dplyr
```

Versão da plataforma “R”: 3.4.1 (2017-06-30) “Single Candle” <https://www.r-project.org/> Versão do ambiente de programação “RStudio”: 1.0.143 <https://www.rstudio.com/> Pacotes principais nesta apresentação:

Tidyverse 1.1.1 <http://tidyverse.org/>

A apresentação deste relatório foi integralmente concebida com a utilização do ambiente de programação RStudio, com a funcionalidade R-Markdown. <http://rmarkdown.rstudio.com/index.html>



Ambiente de desenvolvimento integrado, RStudio

Para efeito demonstrativo, a maior parte dos comandos e funções utilizados para geração de resultados na ferramenta foi mantido à vista, porém a supressão destes itens, mantendo apenas tabelas, comentários e gráficos é facilmente executada dentro da estrutura RStudio + R-Markdown.

Introdução ao Estudo Proposto

Retorno para Conteúdo

O trabalho teve início com dados gerados a partir da ferramenta Power BI e exportados para Excel, tratados e importados para o “R” de maneira não transformada (arquivo plano), e posteriormente também importados com transformações para processamento estatístico (onde colunas = variáveis). Os dados incluídos seguem as seguintes características:

- Série temporal pela data de atendimento do componente assistencial;
- Duas consolidações de quantidade por mês de referência, uma entre dias 1 a 15 e outra entre 16 a 31, para aumento da eficácia dos testes relacionados;
- Na transformação, foram filtrados dados de maior relevância, a partir de janeiro de 2014, gerando 78 observações por componente;
- Dados de Componente, Ano/mês/Quinzena e Indicador de Internação (CAI);

```
#### Formato original, trabalhado no MS Excel -----  
org <- read_excel('CAIv2.xlsx');org;
```

```
## # A tibble: 22,096 x 5  
##   componente internacao    amq valor quantidade
```

```
##      <chr>      <chr>      <chr> <dbl>      <dbl>
## 1 material      n 20121002  1.68      1
## 2 material      n 20130101  1.68      1
## 3 sadt          s 20130702  2.80      1
## 4 sadt          s 20130102  3.30      1
## 5 sadt          n 20130402  5.00      1
## 6 sadt          n 20130402  5.00      1
## 7 honorario     n 20130601  5.50      1
## 8 nulo          n 20161002  6.50      1
## 9 nulo          n 20161002  6.50      1
## 10 nulo         n 20161102  6.50      1
## # ... with 22,086 more rows

#### Formato para processamento estatístico -----
cai <- read_excel('CAIv2.xlsx') %>% # readxl: importação
  group_by(componente,internacao,amq) %>% # dplyr: agrupamento
  summarise(valor = sum(valor), qt = sum(quantidade)) %>% # dplyr: sumarização
  select(-valor) %>% # dplyr: retirada de uma coluna
  spread(componente,qt) %>% # tidyr: mudança de formato, de chave-valor para colunas
  arrange(amq, internacao) %>% # dplyr: reordenação
  filter(amq>="20140101"); cai; # dplyr: seleção;

## # A tibble: 156 x 13
## # Groups:   internacao [2]
##   internacao      amq consulta domiciliar emergencia honorario material
##   <chr>      <chr>      <dbl>      <dbl>      <dbl>      <dbl>      <dbl>
## 1 n 20140101  37776      NA      21558      40467      33578
## 2 s 20140101      NA      3195      NA      17868      176369
## 3 n 20140102  51028      NA      22816      58106      48888
## 4 s 20140102      NA      2905      NA      19907      196991
## 5 n 20140201  42239      NA      20012      51574      34024
## 6 s 20140201      NA      3366      NA      19231      164092
## 7 n 20140202  35062      NA      18037      43491      31142
## 8 s 20140202      NA      2046      NA      16628      154004
## 9 n 20140301  29567      NA      20011      34624      23456
## 10 s 20140301      NA      3530      NA      17787      174785
## # ... with 146 more rows, and 6 more variables: medicamento <dbl>,
## #   nulo <dbl>, pacote <dbl>, remocao <dbl>, sadt <dbl>, taxa <dbl>
```

Análise de Correlação

Retorno para Conteúdo

A partir da tabela transformada acima, foi criada uma matriz de correlação entre os componentes, para verificação de quais pares apresentavam influência sobre os demais. De modo a avaliar todas as relacionamentos, foram criadas três versões: completo (C), sem internação(N) e com internação (S):

```
#### Transformações sobre dados da "CAI" -----
caiC.precor <- cai %>%
  arrange(amq, internacao) %>%
  group_by(amq) %>%
  summarise(consulta = sum(consulta, na.rm = TRUE),
            domiciliar = sum(domiciliar, na.rm = TRUE),
```

```

    emergencia = sum(emergencia, na.rm = TRUE),
    honorario = sum(honorario, na.rm = TRUE),
    material = sum(material, na.rm = TRUE),
    medicamento = sum(medicamento, na.rm = TRUE),
    nulo = sum(nulo, na.rm = TRUE),
    pacote = sum(pacote, na.rm = TRUE),
    remocao = sum(remocao, na.rm = TRUE),
    sadt = sum(sadt, na.rm = TRUE),
    taxa = sum(taxa, na.rm = TRUE)) %>%
as.data.frame(.) %>%
select(-amq) %>%
as.tibble()

caiN.precor <- cai %>%
  filter(internacao == 'n') %>%
  arrange(amq) %>%
  group_by(amq) %>%
  summarise(consulta = sum(consulta, na.rm = TRUE),
            domiciliar = sum(domiciliar, na.rm = TRUE),
            emergencia = sum(emergencia, na.rm = TRUE),
            honorario = sum(honorario, na.rm = TRUE),
            material = sum(material, na.rm = TRUE),
            medicamento = sum(medicamento, na.rm = TRUE),
            nulo = sum(nulo, na.rm = TRUE),
            pacote = sum(pacote, na.rm = TRUE),
            remocao = sum(remocao, na.rm = TRUE),
            sadt = sum(sadt, na.rm = TRUE),
            taxa = sum(taxa, na.rm = TRUE)) %>%
as.data.frame(.) %>%
select(-amq) %>%
as.tibble()

caiS.precor <- cai %>%
  filter(internacao == 's') %>%
  arrange(amq) %>%
  group_by(amq) %>%
  summarise(consulta = sum(consulta, na.rm = TRUE),
            domiciliar = sum(domiciliar, na.rm = TRUE),
            emergencia = sum(emergencia, na.rm = TRUE),
            honorario = sum(honorario, na.rm = TRUE),
            material = sum(material, na.rm = TRUE),
            medicamento = sum(medicamento, na.rm = TRUE),
            nulo = sum(nulo, na.rm = TRUE),
            pacote = sum(pacote, na.rm = TRUE),
            remocao = sum(remocao, na.rm = TRUE),
            sadt = sum(sadt, na.rm = TRUE),
            taxa = sum(taxa, na.rm = TRUE)) %>%
as.data.frame(.) %>%
select(-amq) %>%
as.tibble()

#### Limpeza de objetos, caso necessário para recriação -----
rm(caiC.cor,caiS.cor,caiN.cor)

```

```
#### Criação de tabelas -----
caiC.corP <- caiC.precor %>%
  cor(., method = 'pearson');as.tibble(caiC.corP)

## # A tibble: 11 x 11
##   consulta domiciliar emergencia honorario material medicamento
##   <dbl>      <dbl>      <dbl>      <dbl>      <dbl>      <dbl>
## 1 1.00000000 0.247697684 0.49182483 0.89854026 0.116323687 0.3728395
## 2 0.24769768 1.000000000 -0.02912586 0.21398499 -0.008553451 0.4547423
## 3 0.49182483 -0.02912586 1.00000000 0.67206922 0.631216252 0.4427839
## 4 0.89854026 0.21398499 0.67206922 1.00000000 0.390044553 0.5696622
## 5 0.11632369 -0.00855345 0.63121625 0.39004455 1.000000000 0.2980703
## 6 0.37283952 0.454742256 0.44278386 0.56966218 0.298070321 1.0000000
## 7 0.07024871 0.258392387 -0.05318379 0.06077006 -0.021435278 0.1112784
## 8 0.84309862 0.413286721 0.33927360 0.81697679 -0.038662520 0.6159163
## 9 0.26001262 0.346671044 0.26296923 0.36227311 -0.019958008 0.5724481
## 10 0.94591843 0.361181944 0.49820787 0.91273546 0.117281707 0.5686133
## 11 0.45300157 0.195680762 0.75107249 0.72435041 0.695949692 0.5249648
## # ... with 5 more variables: nulo <dbl>, pacote <dbl>, remocao <dbl>,
## #   sadt <dbl>, taxa <dbl>

#### Apenas a primeira tabela (^) será demonstrada -----
caiN.corP <- caiN.precor %>%
  cor(., method = 'pearson');#caiN.corP

caiS.corP <- caiS.precor %>%
  cor(., method = 'pearson');#caiS.corP

#### Representação por simbolos -----
Ccorgrid <- symnum(caiC.corP);Ccorgrid

##           c d e h mt md n p r s t
## consulta      1
## domiciliar    1
## emergencia    . 1
## honorario     + , 1
## material      , . 1
## medicamento . . . . 1
## nulo          1
## pacote        + . . + , 1
## remocao       . . . . 1
## sadt          * . . * . + . 1
## taxa          . , , , . . . 1
## attr("legend")
## [1] 0 ' ' 0.3 '.' 0.6 ', ' 0.8 '+' 0.9 '*' 0.95 'B' 1

#### Apenas a primeira tabela (^) será demonstrada -----
Ncorgrid <- symnum(caiN.corP);#Ncorgrid
Scorgrid <- symnum(caiS.corP);#Scorgrid

#### Tratamento para apresentação gráfica -----
caiC.corP[lower.tri(caiC.corP)] <- NA
caiN.corP[lower.tri(caiN.corP)] <- NA
caiS.corP[lower.tri(caiS.corP)] <- NA
```

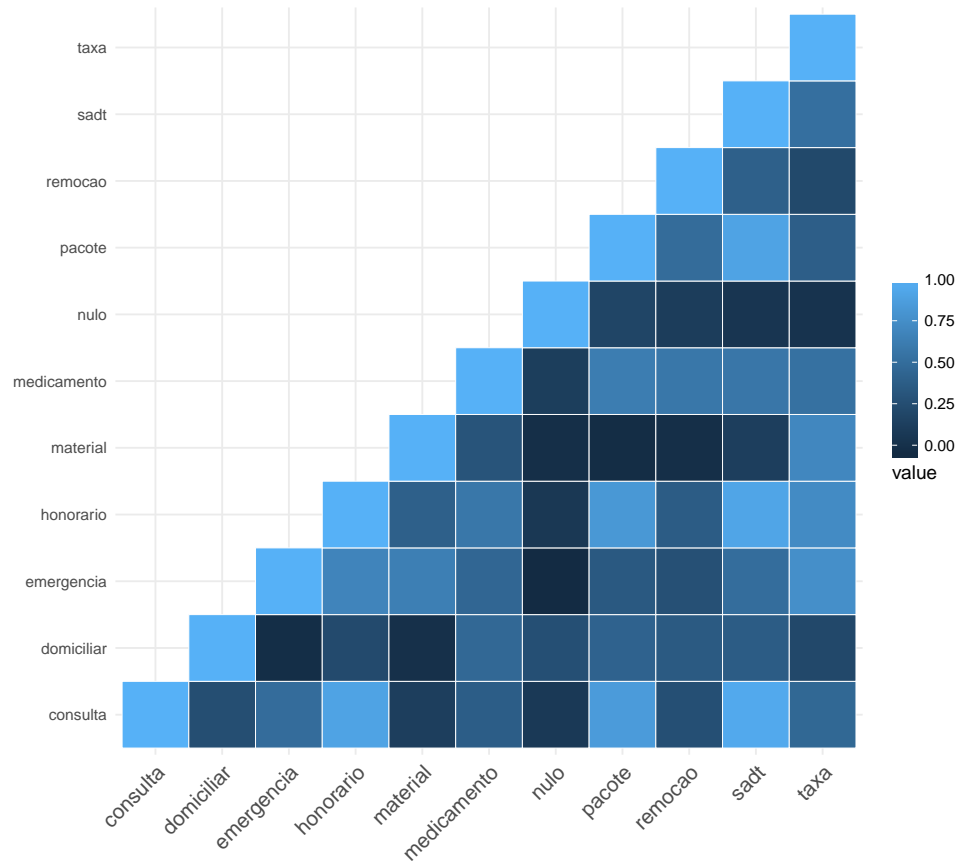
Gráficos para Comunicação

Retorno para Conteúdo

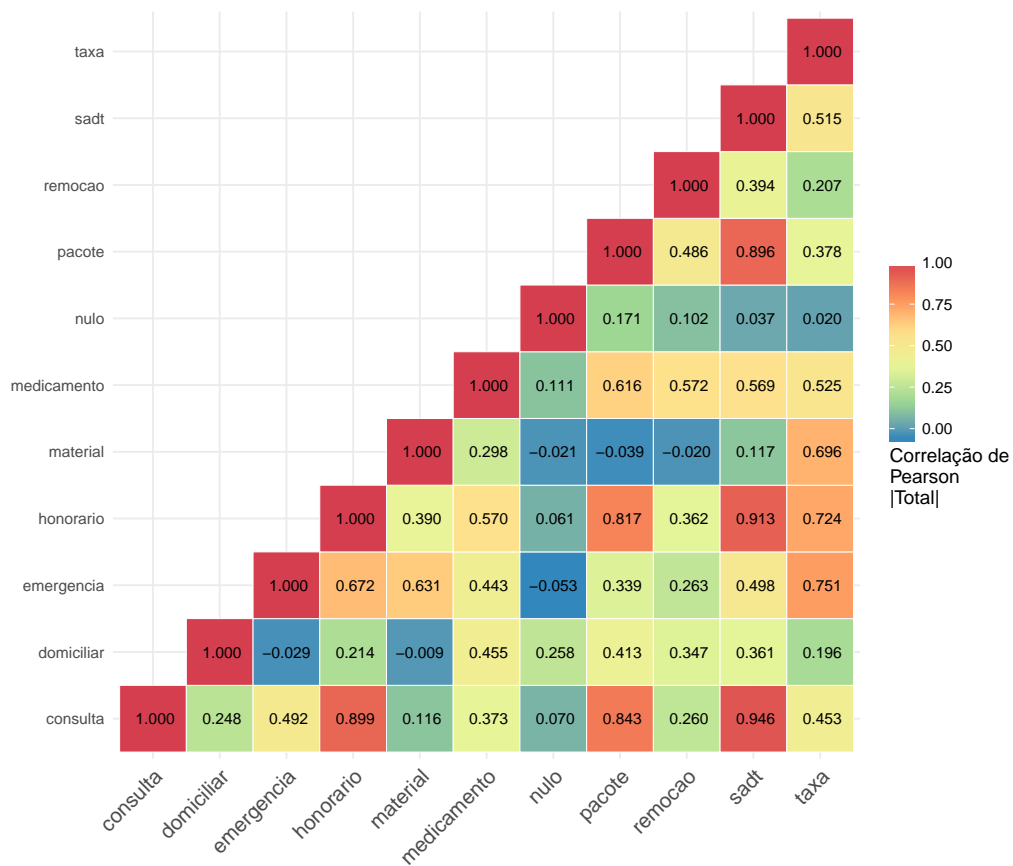
Embora suficientes para avaliação estatística de profissionais, os instrumentos apresentados até então não são adequados para demonstração de resultados e apreciação de públicos em geral (outras capacidades sobre visualização serão tratadas ao final deste estudo, no tópico anterior à conclusão).

Com objetivo de evoluir da análise para comunicação, os dados de correlação foram submetidos a manipulação por um dos pacotes especialistas do “R” para apresentações gráficas (GGPlot2), foram então gerados os mapas de correlação com estética mais adequada:

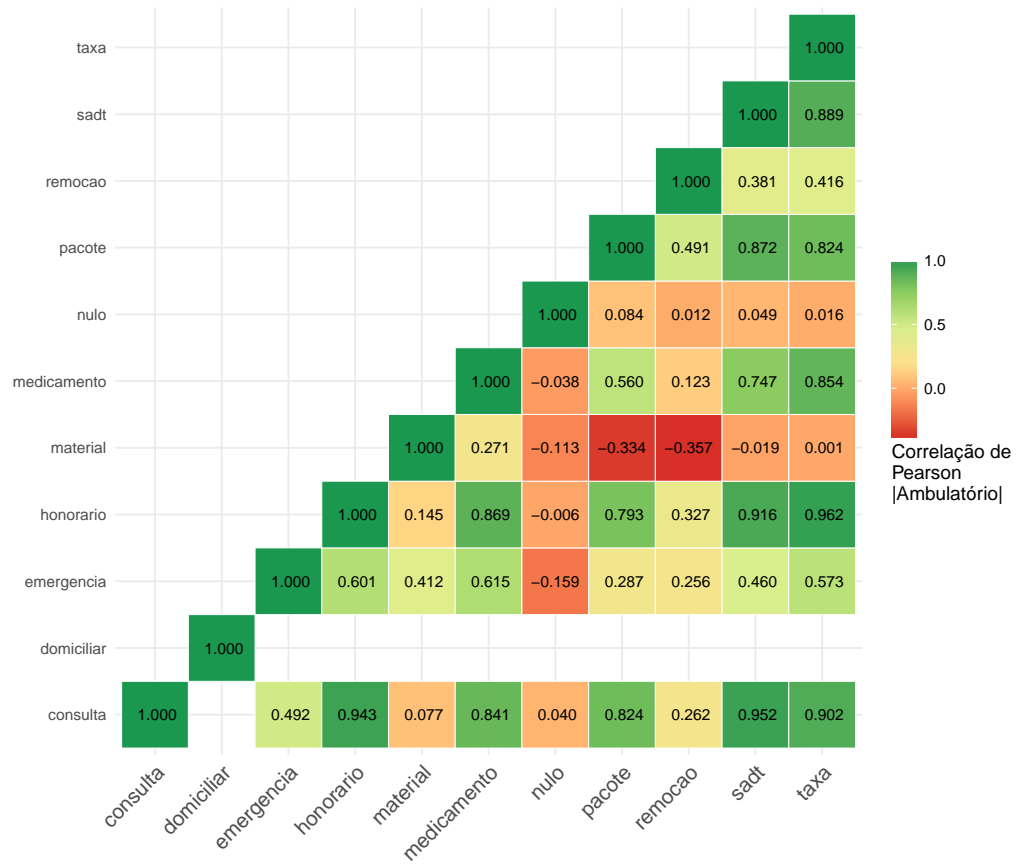
```
#### Gráfico gerado a partir do comando abaixo (exemplo) ----
GGcorC <- ggplot(data = melt(caiC.corP, na.rm = TRUE), aes(Var2, Var1, fill = value)) +
  geom_tile(color = "white") +
  theme_minimal()+
  theme(axis.text.x = element_text(angle = 45,
                                    vjust = 1,
                                    size = 12,
                                    hjust = 1),
        axis.title.x = element_blank(),
        axis.title.y = element_blank())+
  guides(fill = guide_colorbar(barwidth = 1,
                               barheight = 7,
                               title.position = "bottom")) +
  coord_fixed();GGcorC
```

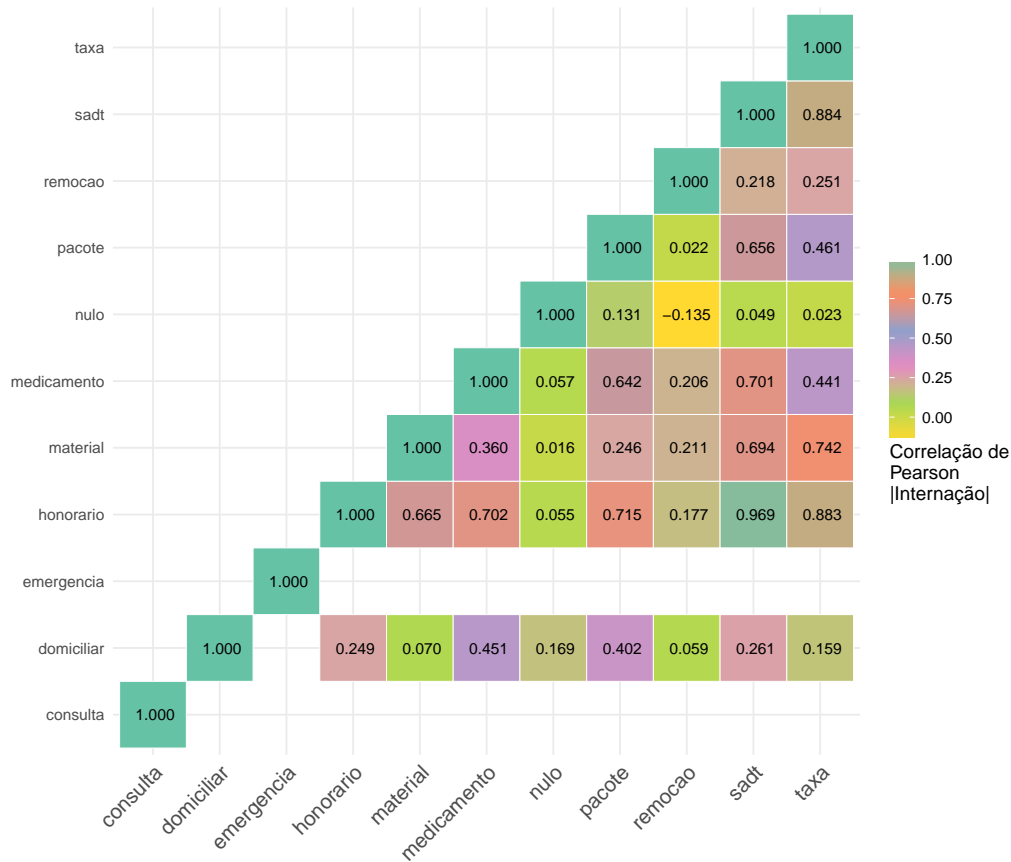


```
#### Camada adicionada ao gráfico original -----
GGcorC +
  geom_text(aes(Var2, Var1, label = format(value, digits = 1, nsmall= 3)),
    color = "black",
    size = 3) +
  scale_fill_distiller(palette = "Spectral",
    #trans = "reverse",
    #space = "Lab",
    name="Correlação de\nPearson\n|Total|")
```

Demais gráficos (Correlação de componentes com Internação = N,S), comandos do “R” para geração foram ocultados:





Direcionamento no Estudo

Retorno para Conteúdo

Após verificadas as influências, se fez necessária uma seleção para detalhamento investigativo, além de um entendimento sobre o sentido da correlação. Deste modo foi realizada uma consulta pelo BI de Contas para verificação de quais componentes correlacionados se mostravam relevantes, por valor e percentual no faturamento acumulado entre 2014 e 2017:

```
#### Tabela criada manualmente para simplificação de nomes --
(depara_componente <- read_excel('depara_componente.xlsx'))
```

```
## # A tibble: 11 x 2
##   `Componente assistencial` componente
##   <chr> <chr>
## 1 Material material
## 2 SADT sadt
## 3 Honorários honorario
## 4 <NA> nulo
## 5 Diárias e Taxas taxa
## 6 Medicamento medicamento
## 7 Consulta Eletiva consulta
## 8 Valor Referencial pacote
## 9 Home Care domiciliar
```

```

## 10      Urgência e Emergência  emergencia
## 11      Remoção                remocao

#### Importações (Geradas pelo Power BI com um click) -----
`dataset` = read.csv('C:/Users/nelson.junior/Documents/R/REditorWrapper_e74c054e-473b-47b7-b164-7b93f69
`dataset2` = read.csv('C:/Users/nelson.junior/Documents/R/REditorWrapper_6b9a785a-59bf-48ad-abdd-044716

#### Criação de resumos, por indicador de internação -----
resumo_valor_N <- dataset %>%
  select(internacao = "Indicador de Internação",
         componente = "Componente assistencial",
         valor = "Valor Total",
         percentual = "Valor Total.1",
         quantidade = "Quantidade Aprovada") %>%
  mutate(percentual = as.numeric(format(percentual*100, digits = 2)))

resumo_valor_S <- dataset2 %>%
  select(internacao = "Indicador de Internação",
         componente = "Componente assistencial",
         valor = "Valor Total",
         percentual = "Valor Total.1",
         quantidade = "Quantidade Aprovada") %>%
  mutate(percentual = as.numeric(format(percentual*100, digits = 2)))

#### Renomeação/substituição/exclusão de colunas (estudo) ---
depara_componente <- depara_componente %>%
  select(componente = "Componente assistencial", comp = "componente")

resumo_valor_S <- merge(resumo_valor_S, depara_componente, by = "componente") %>%
  mutate(componente = comp) %>%
  select(-comp)

resumo_valor_N <- merge(resumo_valor_N, depara_componente, by = "componente") %>%
  mutate(componente = comp) %>%
  select(-comp)

#### Unificação de Tabelas-----
resumo_valor <- rbind(resumo_valor_N, resumo_valor_S) %>%
  arrange(internacao, desc(percentual)) %>%
  select(-valor); resumo_valor

```

##	componente	internacao	percentual	quantidade
## 1	sadt	Não	33.9	41438502
## 2	medicamento	Não	23.4	15855368
## 3	emergencia	Não	11.7	1742006
## 4	honorario	Não	11.1	4268442
## 5	consulta	Não	9.8	3496017
## 6	pacote	Não	9.0	1366023
## 7	material	Não	1.2	4468319
## 8	pacote	Sim	33.7	603225
## 9	taxa	Sim	19.0	3697349
## 10	medicamento	Sim	18.1	97291398
## 11	material	Sim	13.5	34129894
## 12	domiciliar	Sim	7.0	383263
## 13	honorario	Sim	6.3	1700725

```
## 14          sadt          Sim          2.4      5274850
```

Foram então filtradas as maiores Correlações, e agregado o valor como uma nova coluna, para correlações com internação, e sem internação. Atenção especial foi dada ao relacionamento de pacotes para internação = “S”:

```
#### Transformação de tabela de correlação e filtro -----
topcorS <- melt(caiS.corP, na.rm = TRUE) %>%
  filter((value >= 0.85 | ((Var1 == "pacote" | Var2 == "pacote")
                        & value >= 0.70)) & value != 1) %>%
  arrange(desc(value))

topcorN <- melt(caiN.corP, na.rm = TRUE) %>%
  filter(value >= 0.85 & value != 1) %>%
  arrange(desc(value))

#### Lookup de Tabelas -----
topcorS_W <- add_column(topcorS, Var1_prct = resumo_valor_S[match(topcorS$Var1, resumo_valor_S$componente)])
topcorS_W <- add_column(topcorS_W, Var2_prct = resumo_valor_S[match(topcorS$Var2, resumo_valor_S$componente)])

topcorN_W <- add_column(topcorN, Var1_prct = resumo_valor_N[match(topcorN$Var1, resumo_valor_N$componente)])
topcorN_W <- add_column(topcorN_W, Var2_prct = resumo_valor_N[match(topcorN$Var2, resumo_valor_N$componente)])

#### Reordenação de colunas -----
names(topcorS_W)
```

```
## [1] "Var1"      "Var2"      "value"      "Var1_prct" "Var2_prct"

(topcorS_W <- select(topcorS_W, "Var1", "Var1_prct", "Var2", "Var2_prct", Corr = "value"))
```

```
##      Var1 Var1_prct  Var2 Var2_prct      Corr
## 1 honorario      6.3  sadt      2.4 0.9691678
## 2      sadt      2.4  taxa     19.0 0.8835740
## 3 honorario      6.3  taxa     19.0 0.8833631
## 4 honorario      6.3 pacote    33.7 0.7151972
```

```
(topcorN_W <- select(topcorN_W, "Var1", "Var1_prct", "Var2", "Var2_prct", Corr = "value"))
```

```
##      Var1 Var1_prct      Var2 Var2_prct      Corr
## 1 honorario     11.1      taxa      NA 0.9620606
## 2  consulta      9.8      sadt     33.9 0.9520582
## 3  consulta      9.8 honorario     11.1 0.9431398
## 4 honorario     11.1      sadt     33.9 0.9156441
## 5  consulta      9.8      taxa      NA 0.9017094
## 6      sadt     33.9      taxa      NA 0.8885665
## 7  pacote      9.0      sadt     33.9 0.8724481
## 8 honorario     11.1 medicamento    23.4 0.8688183
## 9 medicamento    23.4      taxa      NA 0.8536851
```

Devido à relevância de Taxas e Pacotes em internação, foram averiguadas as correlações cruzadas (presumidamente altas) a partir de consultas e emergência, que são considerados como ambulatoriais.

Para fins de estudo, foram criadas tabelas separadas e individualizadas para a análise, porém uma abordagem similar às tabelas de correlação geral poderia ter sido utilizada, com menos processos intermediários na ferramenta, e diminuição de passos:

```
#### Preparação: Transformação de tabela agrupada em plana --
cai_df <- cai %>%
  as.data.frame(.) %>%
```

```

as.tibble()

#### Cor. Emergencia, Consulta > Taxas, Pacotes em Internação
cai_ECn_PTs <- cai_df %>%
  filter(internacao == "s") %>%
  arrange(amq) %>%
  select(taxa) %>%
  as.tibble()

cai_temp <- cai_df %>%
  filter(internacao == "n") %>%
  arrange(amq) %>%
  select(emergencia) %>%
  as.tibble()

cai_ECn_PTs <- cai_ECn_PTs %>% add_column(emergencia = cai_temp$emergencia)

cai_temp <- cai_df %>%
  filter(internacao == "n") %>%
  arrange(amq) %>%
  select(consulta) %>%
  as.tibble()

cai_ECn_PTs <- cai_ECn_PTs %>% add_column(consulta = cai_temp$consulta)

cai_temp <- cai_df %>%
  filter(internacao == "s") %>%
  arrange(amq) %>%
  select(pacote) %>%
  as.tibble()

cai_ECn_PTs <- cai_ECn_PTs %>% add_column(pacote = cai_temp$pacote)
cai_ECn_PTs <- cai_ECn_PTs[1:77,]

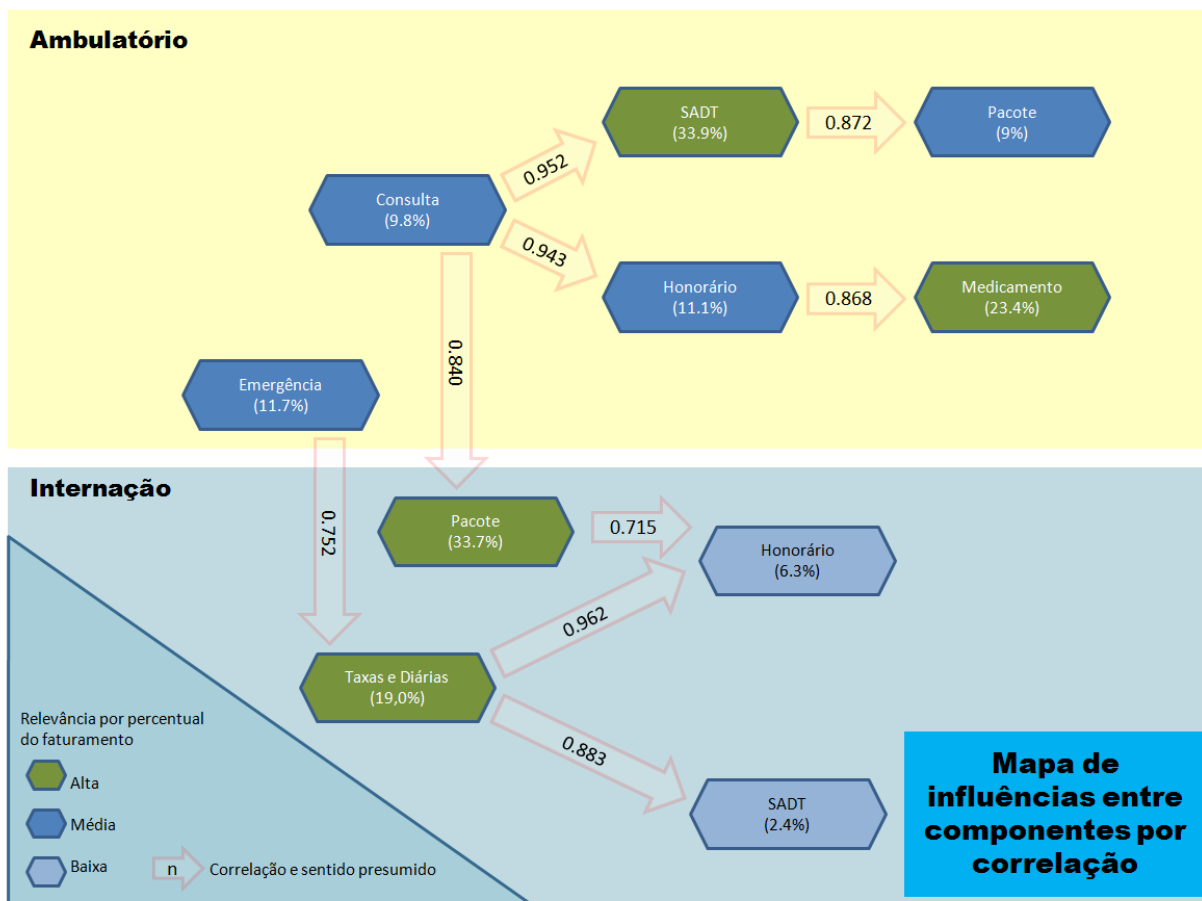
cai_ECn_PTs.corP <- cai_ECn_PTs %>%
  cor(., method = 'pearson');cai_ECn_PTs.corP

```

```

##           taxa emergencia consulta  pacote
## taxa      1.0000000  0.7520463 0.4258082 0.3996761
## emergencia 0.7520463  1.0000000 0.5137216 0.3966162
## consulta   0.4258082  0.5137216 1.0000000 0.8409378
## pacote     0.3996761  0.3966162 0.8409378 1.0000000

```



* Mapeamento sujeito à mudanças

Análise Detalhada sobre Consulta e Exames

Retorno para Conteúdo

Preparação

Retorno para Conteúdo

Com base nas descobertas apontadas pela correlação, e considerando o sentido de influência conhecido de consultas sobre exames, foi então realizada uma investigação de cunho estatístico inferencial, com técnicas mais relevantes para determinação de um modelo preditivo, onde a variável independente é a quantidade de consultas, e a resposta é a quantidade de exames. Tais análises deverão ser realizadas sobre outros conteúdos e elementos, com diversos e adequados modelos de regressão, e com maior detalhamento para efetividade e precisão, como por exemplo, a quebra de consultas e exames de uma determinada especialidade médica.

Preparação: Consulta sobre exames em Internação -----

```
cai_SCs <- cai_df %>%
  filter(internacao == "s") %>%
  arrange(amq) %>%
```

```

select(sadt) %>%
as.tibble()

cai_temp <- cai_df %>%
  filter(internacao == "n") %>%
  arrange(amq) %>%
  select(consulta) %>%
  as.tibble()

cai_SCs <- cai_SCs %>% add_column(consulta = cai_temp$consulta)

#### Preparação: Consulta sobre exames sem Internação -----

cai_SCn <- cai_df %>%
  filter(internacao == "n") %>%
  arrange(amq) %>%
  select(sadt) %>%
  as.tibble()

cai_SCn <- cai_SCn %>% add_column(consulta = cai_temp$consulta)

#### Preparação: Consulta sobre exames totais -----

#rm(cai_SCc)

cai_SCc <- cai_df %>%
  select(amq,sadt) %>%
  group_by(amq) %>%
  summarise(sadt=sum(sadt)) %>%
  as.data.frame(.) %>%
  select(sadt) %>%
  as.tibble()

## Inclusão de coluna "z", análise de variação por desvios padrão, outliers acima de 3
zSc <- scale(cai_SCc$sadt, center = TRUE, scale = TRUE)
cai_SCc <- cai_SCc %>% add_column(consulta = cai_temp$consulta, z = zSc[1:dim(cai_SCc)[1]])

```

Seleção de Escopo pelo Indicador de Internação

Retorno para Conteúdo

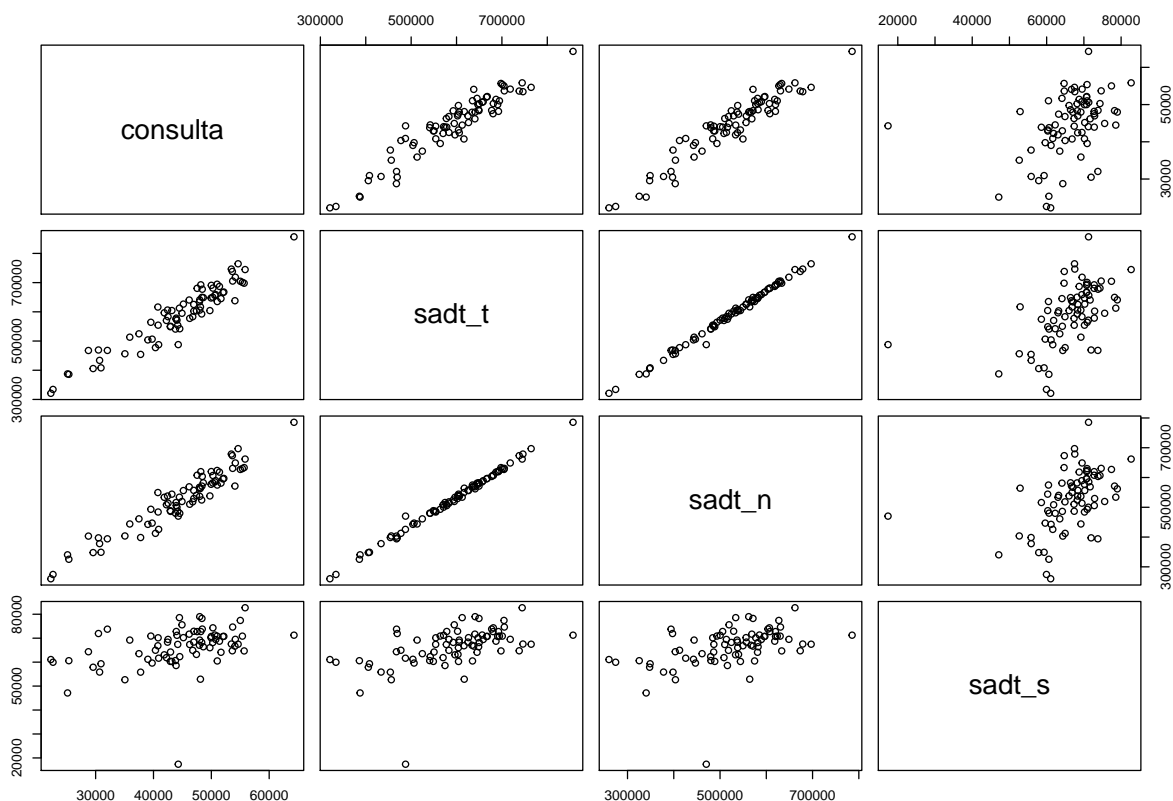
Também foi criada uma tabela para avaliação de relevância de SADT total, com e sem internação:

```

cai_g <- cai_SCc %>%
  select(consulta, sadt_t = sadt) %>%
  add_column(sadt_n = cai_SCn$sadt) %>%
  add_column(sadt_s = cai_SCs$sadt)

plot(cai_g)

```

A matriz de gráficos de dispersão indicou uma forte semelhança de comportamento entre **Consulta Eletiva sobre SADT total**, e **Consulta Eletiva sobre SADT sem internação (ambulatorial)**. Também pode ser verificado que **Consulta Eletiva sobre SADT com internação (hospitalar)** apresenta uma relação mais difusa, pois exames restritos ao ambiente hospitalar geralmente são disparados por outros componentes, fato que poderia ser comprovado com uma análise de correlação específica, caso a resposta gráfica não fosse suficiente. Com base nos resultados, o estudo foi então direcionado para modelo entre **Consulta e SADT total**.

Regressão Linear

Retorno para Conteúdo

Modelo Inicial

Retorno para Conteúdo

Após determinação das variáveis de estudo, a criação do modelo é bastante simplificada pela ferramenta, através da função `lm` (Linear Model), onde “`sadt ~ consulta`” representa uma equação de previsão de quantidades de `sadt` por meio da informação de quantidades de `consulta`:

```
#### Avaliar regressão -----
lm(sadt ~ consulta, data = cai_SCc) %>%
summary()
```

```
##
## Call:
## lm(formula = sadt ~ consulta, data = cai_SCc)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -100255  -22857   -1071    22639    69417
##
## Coefficients:
##              Estimate Std. Error t value      Pr(>|t|)
## (Intercept) 72791.6402  20768.4007   3.505      0.000769 ***
## consulta     11.6330     0.4576  25.420 < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 33740 on 76 degrees of freedom
## Multiple R-squared:  0.8948, Adjusted R-squared:  0.8934
## F-statistic: 646.2 on 1 and 76 DF,  p-value: < 0.00000000000000022

#### Avaliar se intervalo de confiança de 99% inclui 0 -----
confint(lm(sadt ~ consulta, data = cai_SCc), level=0.99)

##              0.5 %          99.5 %
## (Intercept) 17919.89902 127663.38137
## consulta     10.42386     12.84206
```

Formula:

$y = \beta_0 + \beta_1 x$, onde β_0 = intercepto e β_1 = inclinação

$qt\text{sadt} = \beta_0 + (\beta_1 *_{qt} \text{consulta})$

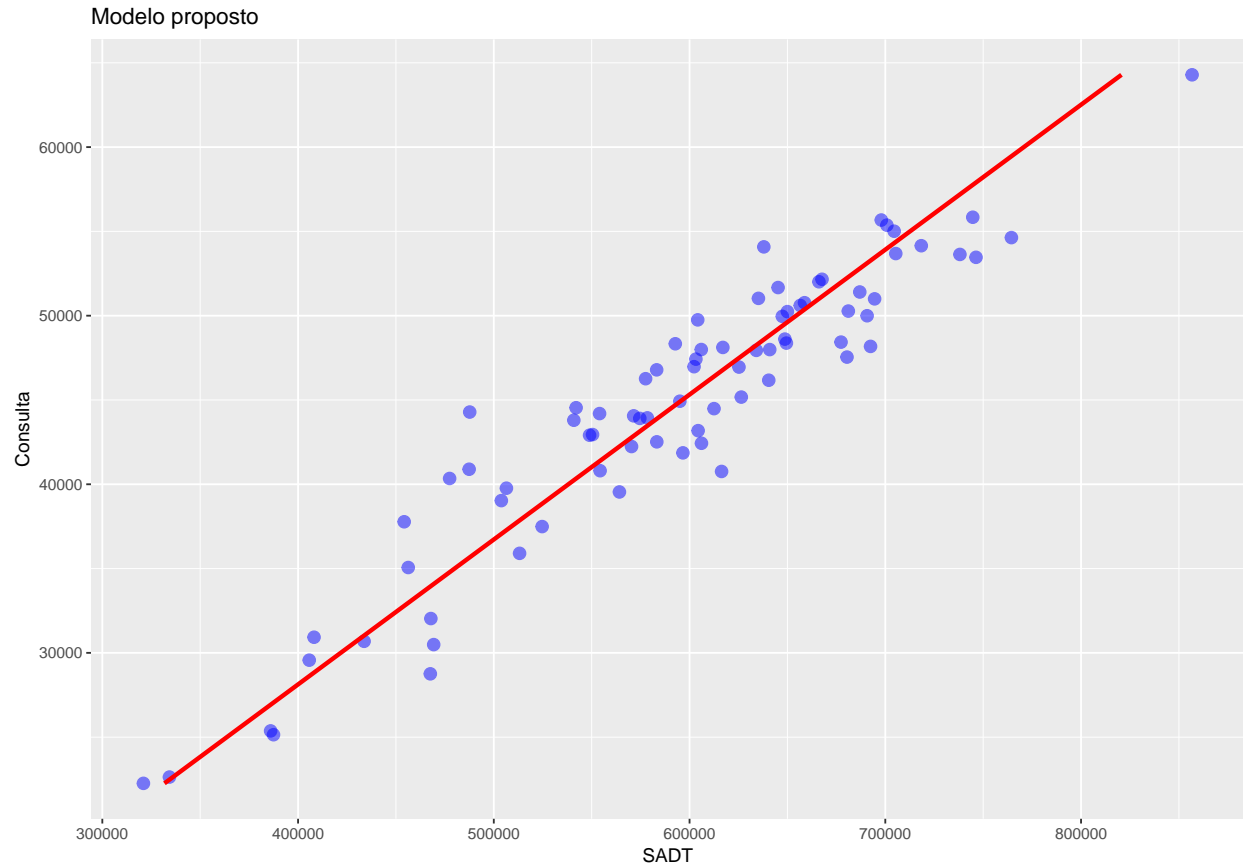
$qt\text{sadt} = 72791.6402 + (11.6330 *_{qt} \text{consulta})$

Apresentando valores de p inferiores a **0,01** para ambos os coeficientes da equação e também para a estatística de variação F , a regressão simples pode ser considerada de boa qualidade e suficiente, comprovando as tendências previamente apontadas na correlação. Caso necessário, modelos de outras naturezas poderiam ser aplicados na ferramenta (lineares múltiplos, quadráticos e não lineares, como polinomiais e exponenciais).

O valor do coeficiente de determinação r^2 de **0.8948** significa que a quantidade de consultas explica **89,48%** da variação de quantidade de exames, e portanto o valor do modelo como instrumento preditivo aparenta ser relevante, inicialmente.

O resultado estatístico pode ser visualizado no gráfico abaixo:

```
#### Gerar objeto com modelo -----
lm.cai_SCc <- lm(sadt ~ consulta, data = cai_SCc)
#### Incluir conluna com valores previstos -----
cai_SCc <- cai_SCc %>% add_predictions(lm.cai_SCc) #modelr
#### Gráfico do modelo -----
ggplot(cai_SCc, aes(y = consulta)) +
  geom_point(aes(x = sadt), color = "blue", alpha = 0.5, size = 3) +
  geom_line(aes(x = pred), color = "red", size = 1.2) +
  labs(title = 'Modelo proposto',
       x = 'SADT', y = 'Consulta')
```



Aspectos e Avaliação da Regressão

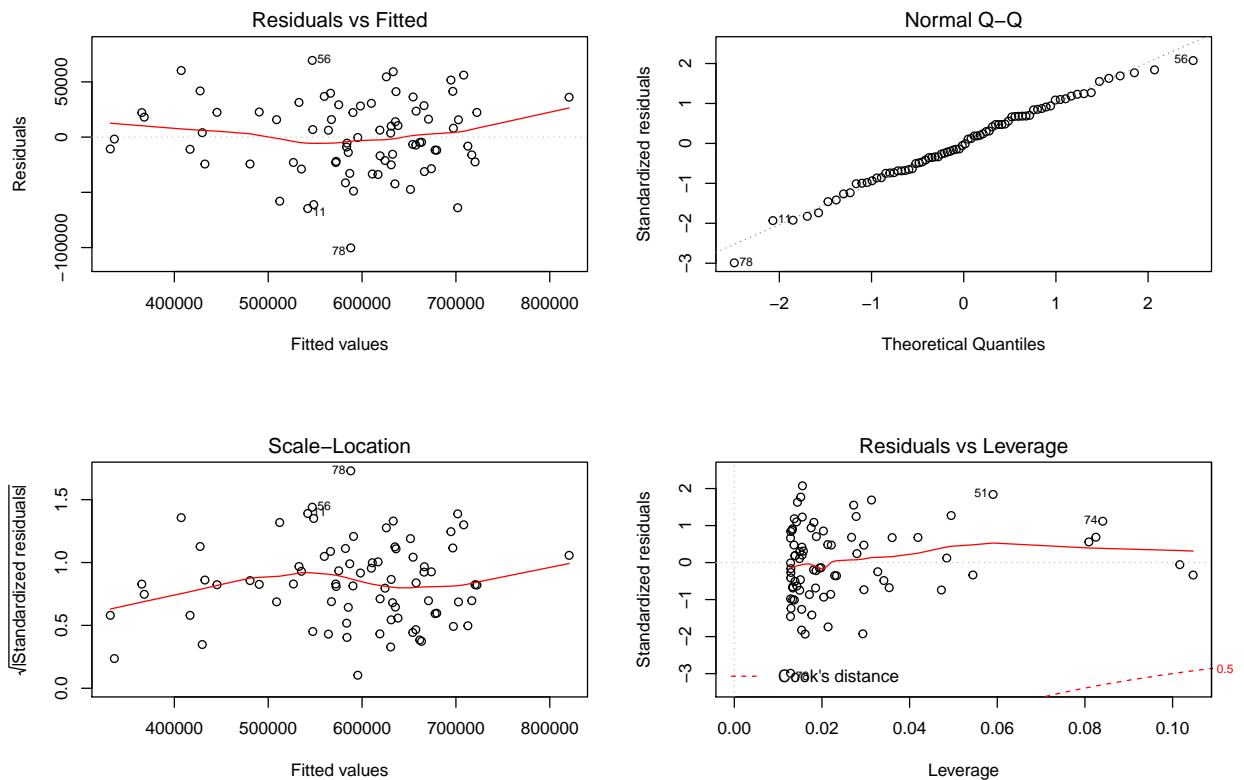
Retorno para Conteúdo

Embora as evidências apontem para um modelo bem construído, existe ainda a necessidade de verificar todas as quatro premissas da regressão: Linearidade (que pode ser verificada acima); Independência de variáveis (de suma importância no uso de séries temporais, devido à possibilidade de Autocorrelação: a ocorrência de dependência entre observações adjacentes no tempo), normalidade e Igualdade de variâncias para valores previstos (de SADT) ao longo de todos os valores de previsão (consulta).

A função padrão de plotagem de gráfico aplicada ao objeto do modelo na ferramenta, é excelente para verificação rápida de Linearidade (gráfico 1), normalidade (gráfico 2) e Igualdade (gráfico 3), além de oferecer também um teste de alavancagem (gráfico 4), que verifica se alguma observação extrema tem influência forte sobre todo o modelo:

```
## Gráficos
## (1) (2)
## (3) (4)
par(mfrow = c(2,2), oma = c(0, 0, 1.1, 0))
#### Validando modelo pelos gráficos padrão -----
plot(lm.cai_ScC)
```

lm(sadt ~ consulta)



```
#### Retornando à configuração de plotagem com Gráfico Único -
par(mfrow = c(1,1), oma = c(0, 0, 0, 0))
```

Considerações sobre Séries Temporais

Retorno para Conteúdo

No aspecto de independência das variáveis, se faz necessária a utilização de técnicas como os testes de **Durbin-Watson** ou **Breusch-Godfrey**, para verificar autocorrelação. Uma vez identificada a técnica estatística desejada, uma consulta em sites de busca geralmente é suficiente para apontar a função e pacote correspondentes, estes podem ser adicionados de maneira simples por meio do ambiente RStudio, tornam-se disponíveis para uso nos scripts.

```
#### Definição do repositório para download -----
## options(repos = 'http://vps.fmvz.usp.br/CRAN/')

#### Instalação do novo pacote -----
## install.packages("lmtest")

#### Ativação do pacote com o teste estatístico -----
library("lmtest", lib.loc="~/R/win-library/3.4")
#### Testes de autocorrelação -----
dwtest(lm.cai_ScC)
```

```
##
## Durbin-Watson test
##
## data:  lm.cai_SCc
## DW = 0.86926, p-value = 0.00000001839
## alternative hypothesis: true autocorrelation is greater than 0
```

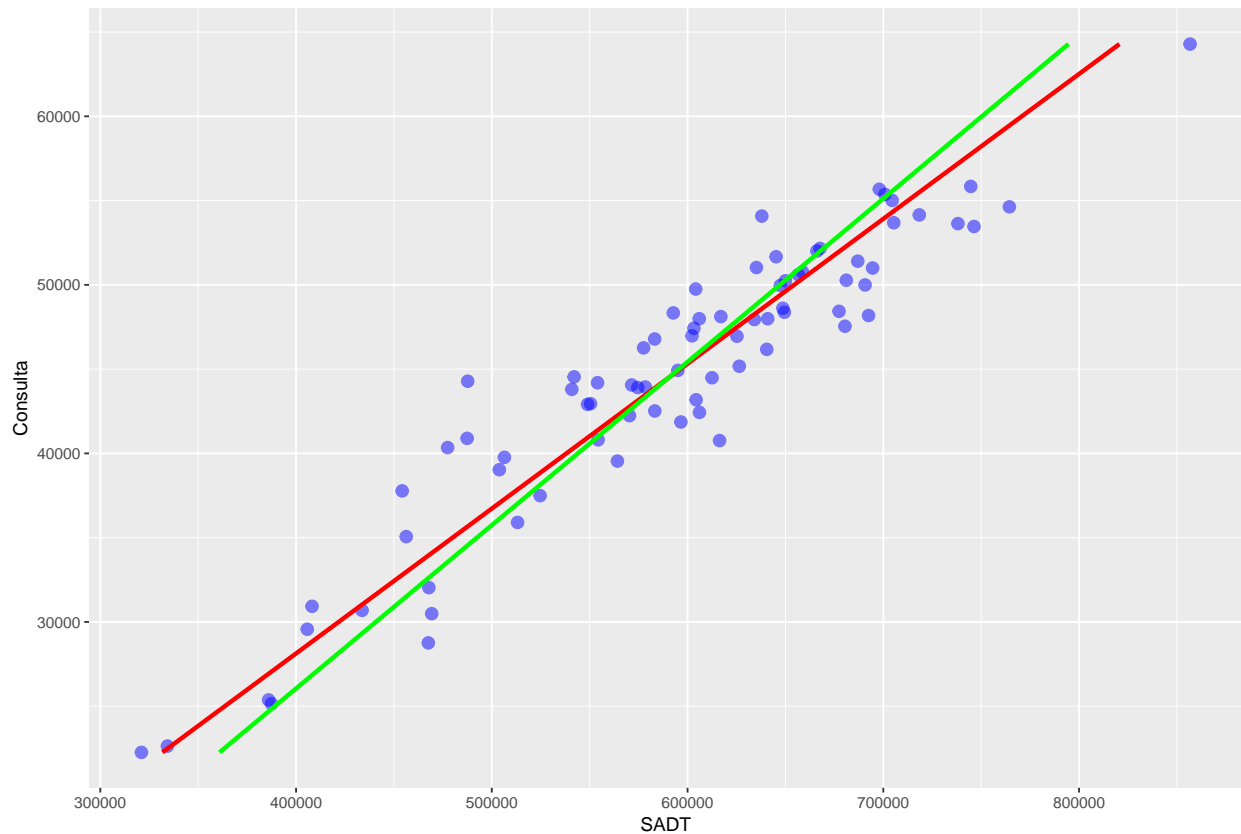
```
bgtest(lm.cai_SCc, order = 2)
```

```
##
## Breusch-Godfrey test for serial correlation of order up to 2
##
## data:  lm.cai_SCc
## LM test = 31.39, df = 2, p-value = 0.0000001527
```

Ambos os testes apontam valor p tendendo a zero, o que indica a rejeição da hipótese nula (que assume a Autocorrelação = 0), e portanto uma transformação é necessária, para enquadramento do princípio de independência de variáveis. Novamente, uma vez identificada a técnica estatística (neste caso, transformação de **Cochrane Orcutt**), uma pesquisa é feita, a função e pacote são identificados, importados na ferramenta, e utilizados após ativação. Em seguida encontra-se o gráfico com ambos os modelos, original com autocorrelação (em vermelho) e transformado, sem ruídos (ρ) de autocorrelação (em verde):

```
#### Ativação do pacote com transformação para Autocorrelação
library("orcutt", lib.loc="~/R/win-library/3.4")
#### Transformação dos modelo para eliminação do ruído -----
lmT.SCc <- cochrane.orcutt(lm.cai_SCc)
#### Inclusão de coluna com valores previstos ajustados -----
cai_SCc <- cai_SCc %>% add_column(predT = as.vector(fitted(lmT.SCc))) #modelr
#### Novo gráfico, com ambos modelos -----
ggplot(cai_SCc, aes(y = consulta)) +
  geom_point(aes(x = sadt), color = "blue", alpha = 0.5, size = 3) +
  geom_line(aes(x = pred), color = "red", size = 1.2) +
  geom_line(aes(x = predT), color = "green", size = 1.2) +
  labs(title = 'Modelo original e transformado, procedimento Cochrane Orcutt, *em verde',
        x = 'SADT', y = 'Consulta')
```

Modelo original e transformado, procedimento Cochrane Orcutt, *em verde



Além da análise gráfica, uma revisão do sumário do novo modelo pode ser realizado com uma função específica do novo pacote:

```
summary.orcutt(lmT.SCc)
```

```
## Call:
## lm(formula = sadt ~ consulta, data = cai_SCc)
##
##               Estimate Std. Error t value      Pr(>|t|)
## (Intercept) 131162.26639 16252.75561   8.070 0.0000000000008775 ***
## consulta    10.31995    0.28613  36.068 < 0.0000000000000022 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 25356.08 on 75 degrees of freedom
## Multiple R-squared:  0.9455 , Adjusted R-squared:  0.9448
## F-statistic: 1300.9 on 1 and 75 DF, p-value: < 3.917e-49
##
## Durbin-Watson statistic
## (original): 0.86926 , p-value: 1.839e-08
## (transformed): 2.06436 , p-value: 6.253e-01
```

Formula:

$y = \beta_0 + \beta_1 x$, onde β_0 = intercepto e β_1 = inclinação

$qt\text{sadt} = \beta_0 + (\beta_1 *_{qt} \text{consulta})$

$qt\text{sadt} = 131162.26639 + (10.31995 *_{qt} \text{consulta})$

Primeiramente, é importante avaliar o resultado do teste de **Durbin Watson** para o modelo transformado, com valor próximo à **2**, e $p = \mathbf{0.6253}$, o que significa que não há evidências suficientes para rejeitar h_0 (*hipótese nula*), onde a autocorrelação é igual a *zero*. O modelo ajustado revela as mesmas boas características, com valor p para o teste t dos coeficientes aproximadamente zero, assim como p para a estatística de variação (F). O valor de r^2 dos novos resultados mostram que a variável quantidade de consultas explica **94,55%** da variação de SADT, demonstrando a validade do novo modelo para predição.

Avaliação Específica para Objetos do Pacote “Orcutt”

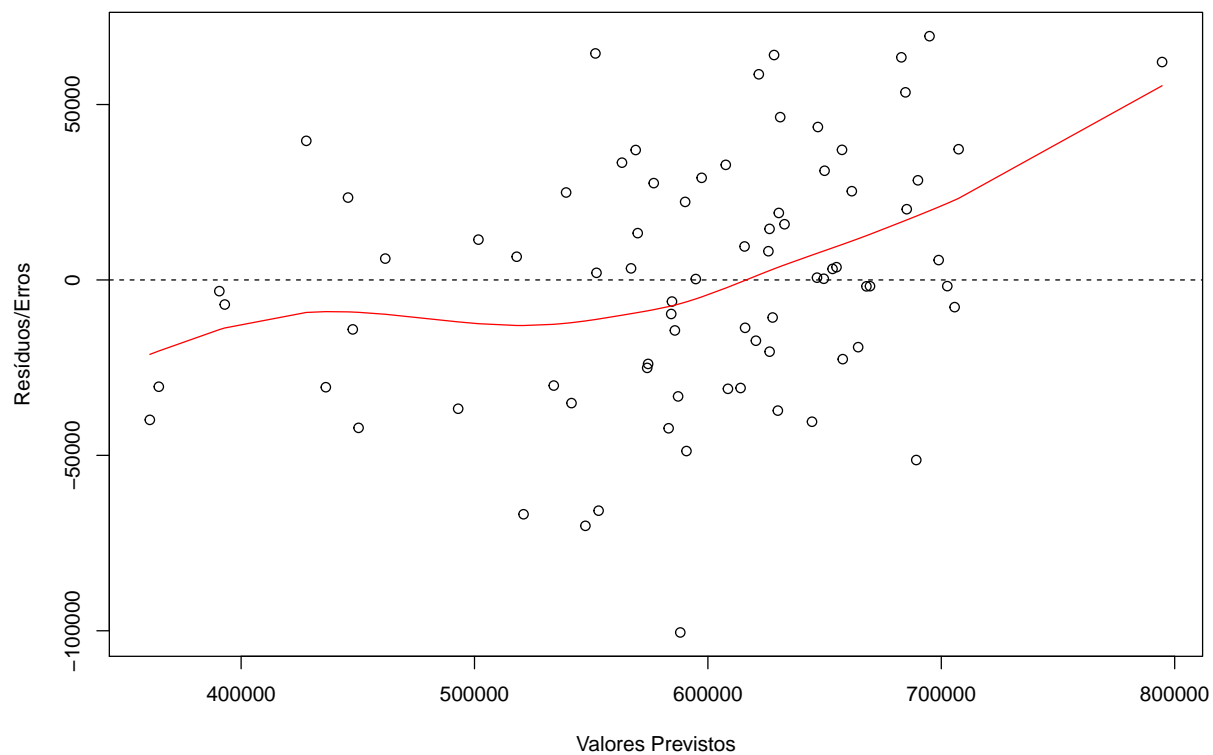
Retorno para Conteúdo

Devido à particularidade do pacote utilizado para a transformação (com objetos de modelo linear proprietários), uma análise dos aspectos da regressão por funções padrão da ferramenta é limitada. Se faz necessária programação específica para diagnóstico de linearidade, normalidade e igualdade de variâncias:

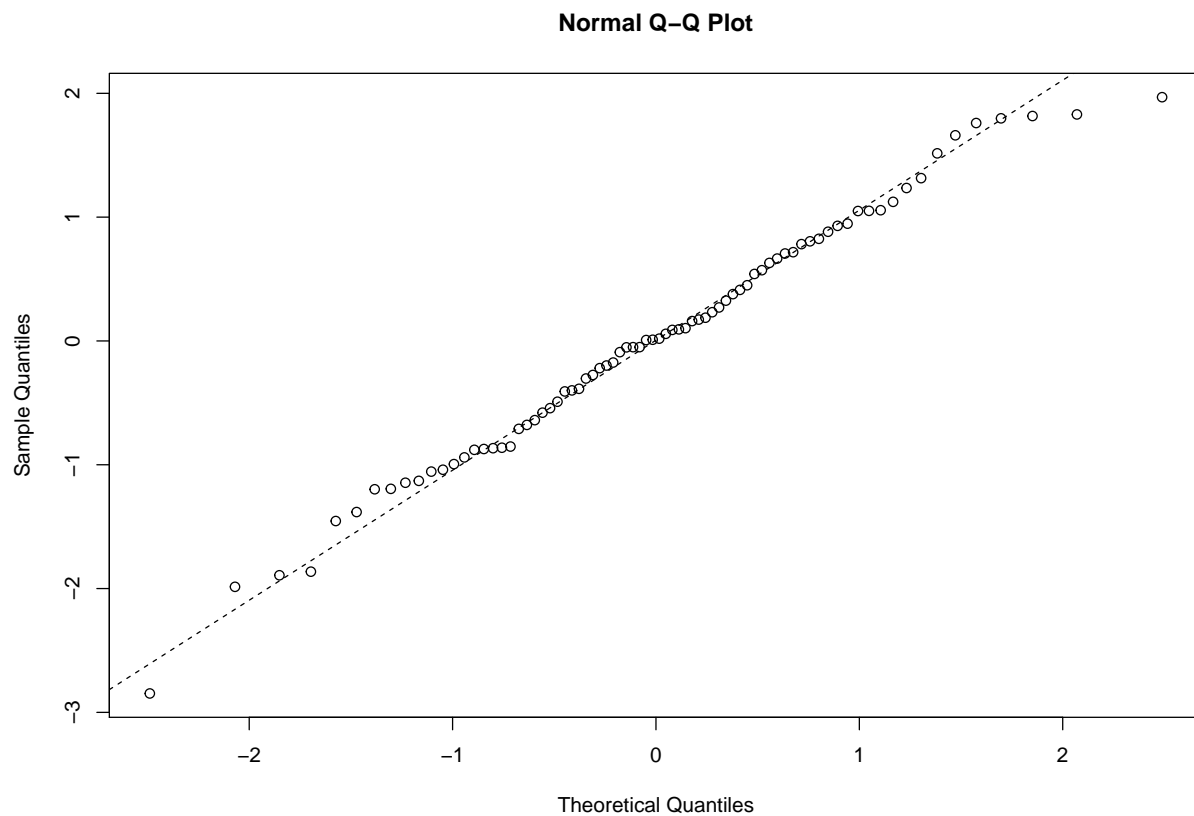
Padronização de resíduos: $Std.e = (\frac{e_i}{\sigma_e})_{1 \leq i \leq n}$, onde $\sigma = \sqrt{\frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n-1}}$

```
#### Valores previstos -----
fit_r <- fitted(lmT.SCc)
#### Gráfico padrão (1) Linearidade -----
plot(fit_r, resid(lmT.SCc),
     ylab = "Resíduos/Erros",
     xlab = "Valores Previstos",
     main = "Linearidade e Variância")
#### Linha de referência -----
smt = smooth.spline(fit_r, resid(lmT.SCc), spar=1)
abline(h = 0, lty = 2)
lines(smt, col='red', lwd=1)
```

Linearidade e Variância

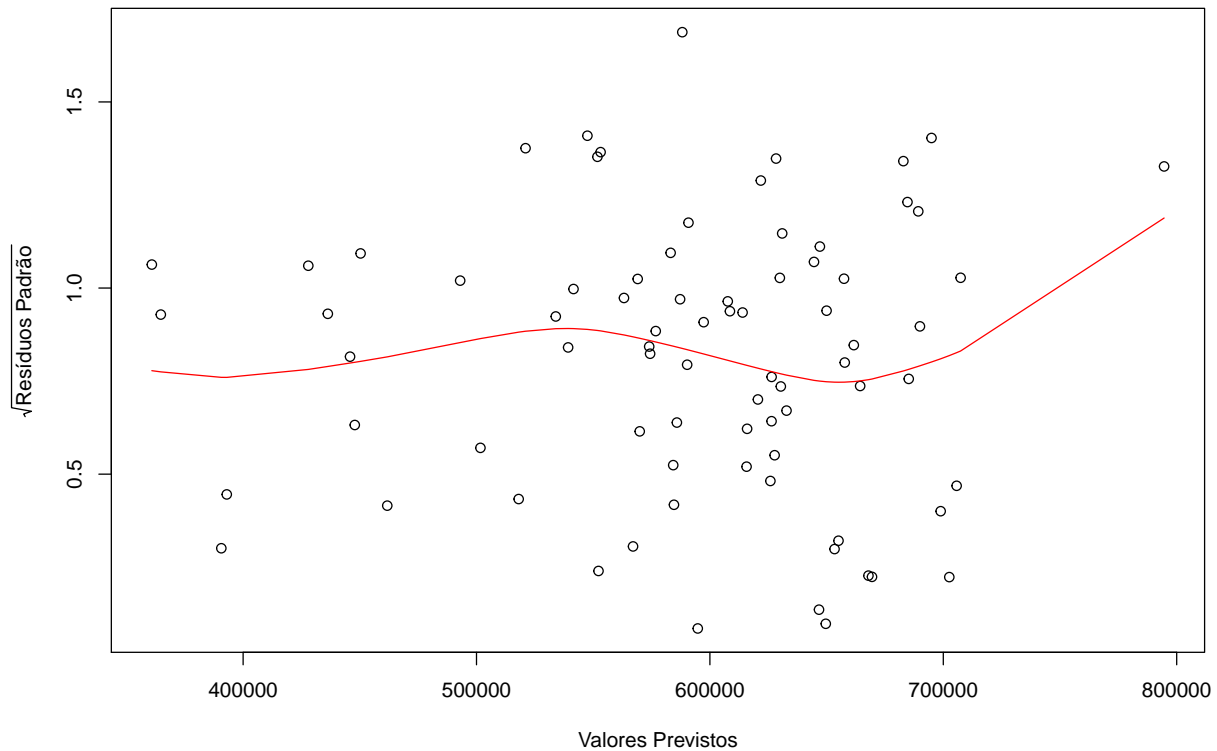


```
#### Resíduos padronizados -----  
std_r <- resid(lmT.SCc)/sd(resid(lmT.SCc))  
# ou std_r <- scale(resid(lmT.SCc), center = FALSE, scale = TRUE)  
  
#### Gráfico padrão (2) Normalidade -----  
qqnorm(std_r)  
#### Linha de referência -----  
qqline(std_r, lty = 2)
```

```
#### Raiz de resíduos padronizados -----
std_r <- sqrt(abs(std_r))
#### Gráfico padrão (3) Igualdade -----
plot(fit_r, std_r,
     ylab = expression(sqrt("Resíduos Padrão")),
     xlab = "Valores Previstos",
     main = "Escala e Localização de Resíduos")
#### Linha de referência -----
smt = smooth.spline(fit_r, std_r, spar=1)
lines(smt, col='red', lwd=1)
```

Escala e Localização de Resíduos



Alternativa de Transformação

Retorno para Conteúdo

Uma alternativa de eliminação dos ruídos da autocorrelação é o procedimento de **Hildreth-Lu**, porém esta abordagem exige aplicação da transformação para todas as novas predições, pois existe uma alteração da escala dos dados:

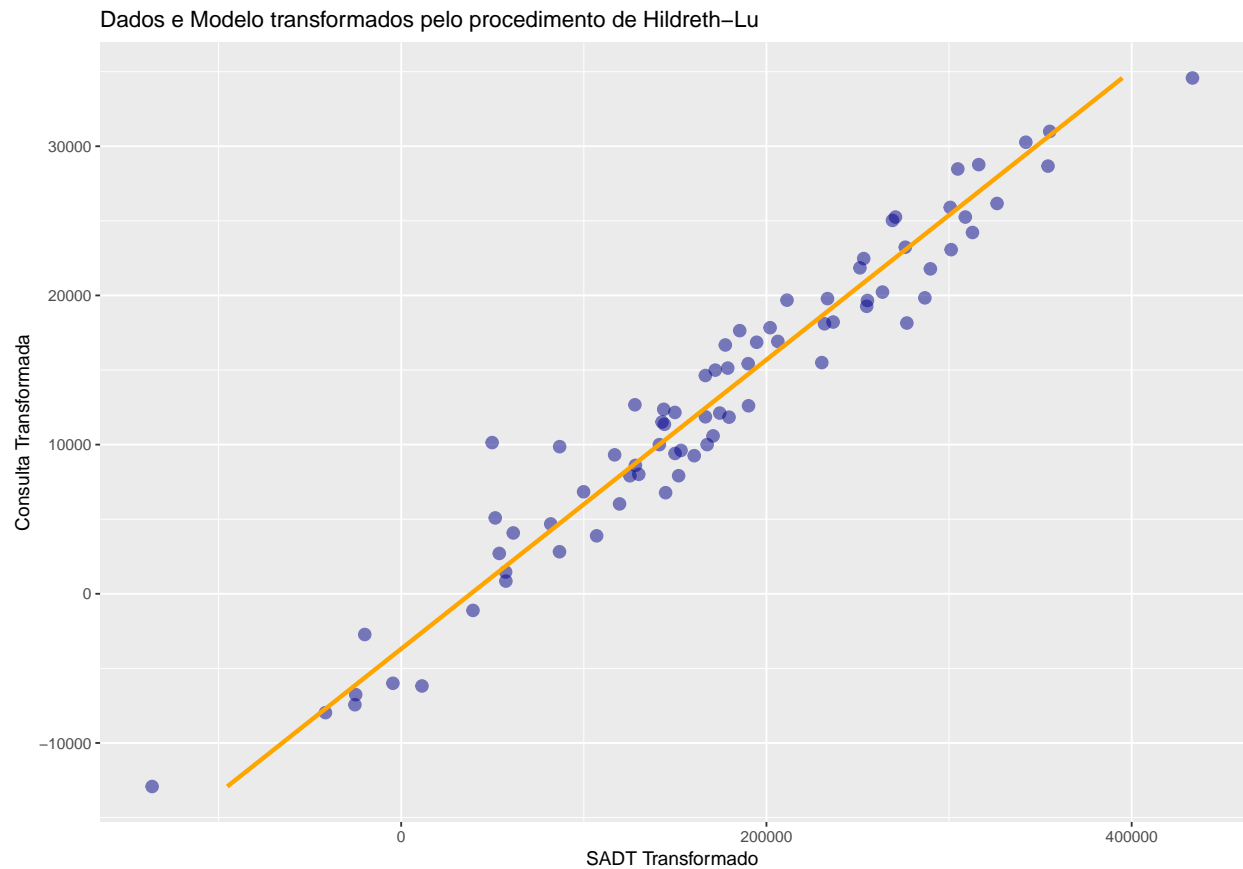
```
#### Ativação do pacote com a segunda forma de transformação
library("HoRM", lib.loc="~/R/win-library/3.4")
#### Valor de rho a partir da primeira transformação -----
lmT.SCc["rho"]

## $rho
## [1] 0.7097007

#### Regressão com o novo procedimento -----
lmT_hil.SCc <- hildreth.lu(x = cai_SCc$consulta, y = cai_SCc$sadt, rho = 0.7097007)
## rho representa o ruído gerado pela autocorrelação

#### Extração de dados a partir do objeto do modelo -----
cai_SCc_Hil <- as.tibble(lmT_hil.SCc[["model"]]) %>%
  select(sadt = y, consulta = x)
#### Inclusão de valores previstos __-----
cai_SCc_Hil <- add_predictions(cai_SCc_Hil, lmT_hil.SCc)
#### Novo gráfico -----
ggplot(cai_SCc_Hil, aes(y = consulta)) +
```

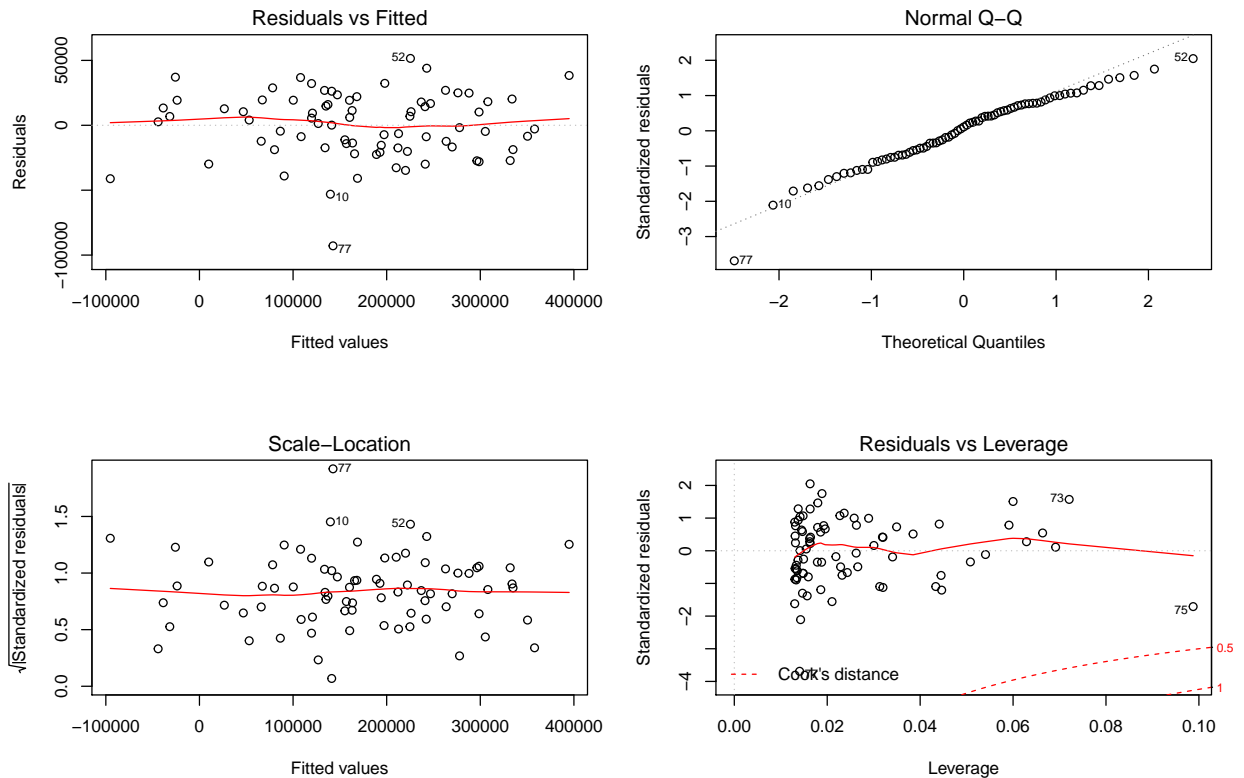
```
geom_point(aes(x = sadt), color = "dark blue", alpha = 0.5, size = 3) +
geom_line(aes(x = pred), color = "orange", size = 1.2) +
labs(title = 'Dados e Modelo transformados pelo procedimento de Hildreth-Lu',
      x = 'SADT Transformado', y = 'Consulta Transformada')
```



A vantagem do pacote “*HoRM*” sobre o pacote “*Orcutt*” é a geração de objeto de modelo linear padrão, o que facilita os diagnósticos básicos do “R”.

```
## Gráficos
## (1) (2)
## (3) (4)
par(mfrow = c(2,2), oma = c(0, 0, 1.1, 0))
#### Validando modelo pelos gráficos padrão -----
plot(lmT_hil.SCc)
```

lm(y ~ x)



```
#### Retornando à configuração de plotagem com Gráfico Único -
par(mfrow = c(1,1), oma = c(0, 0, 0, 0))
#### Sumário do novo modelo -----
summary(lmT_hil.SCc)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -92880 -17315   2678  19256  51528
##
## Coefficients:
##              Estimate Std. Error t value      Pr(>|t|)
## (Intercept) 38076.3141  4718.1636    8.07 0.000000000000878 ***
## x           10.3200    0.2861   36.07 < 0.000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 25360 on 75 degrees of freedom
## Multiple R-squared:  0.9455, Adjusted R-squared:  0.9448
## F-statistic: 1301 on 1 and 75 DF, p-value: < 0.0000000000000022
```

```
#### Avaliar se intervalo de confiança de 99% inclui 0 -----
confint(lmT_hil.SCc, level=0.99)
```

```
##              0.5 %      99.5 %
## (Intercept) 25606.28772 50546.34046
## x              9.56372   11.07618
```

O sumário do terceiro modelo também confirma a rejeição das hipóteses nulas (coeficientes iguais a zero), pelos testes t e F , e apresentando o mesma explicação de **94,55%** da variação de SADT por consulta, pois o mesmo valor de ruído foi aplicado à segunda transformação a partir do cálculo realizado no procedimento da primeira transformação (Orcutt).

Outros Experimentos

Retorno para Conteúdo

Outras técnicas e visualizações foram aplicadas dentro do estudo, e estão representadas a seguir, com respectivos comentários:

```
#### Intervalos para os valores previstos, IC 95% -----
as.tibble(predict.lm(lmT_hil.SCc, level=0.95, interval = 'confidence'))
```

```
## # A tibble: 77 x 3
##       fit      lwr      upr
##       <dbl>    <dbl>    <dbl>
## 1 288008.491 279419.7240 296597.26
## 2 100247.759  93240.2181 107255.30
## 3  90552.777  83226.6490  97878.91
## 4  86409.543  78939.6960  93879.39
## 5 334963.583 324307.3404 345619.82
## 6 163091.914 157311.6399 168872.19
## 7   9911.758  -759.6457  20583.16
## 8 263568.162 255927.4867 271208.84
## 9 160510.360 154715.3907 166305.33
## 10 139869.994 133836.4317 145903.56
## # ... with 67 more rows
```

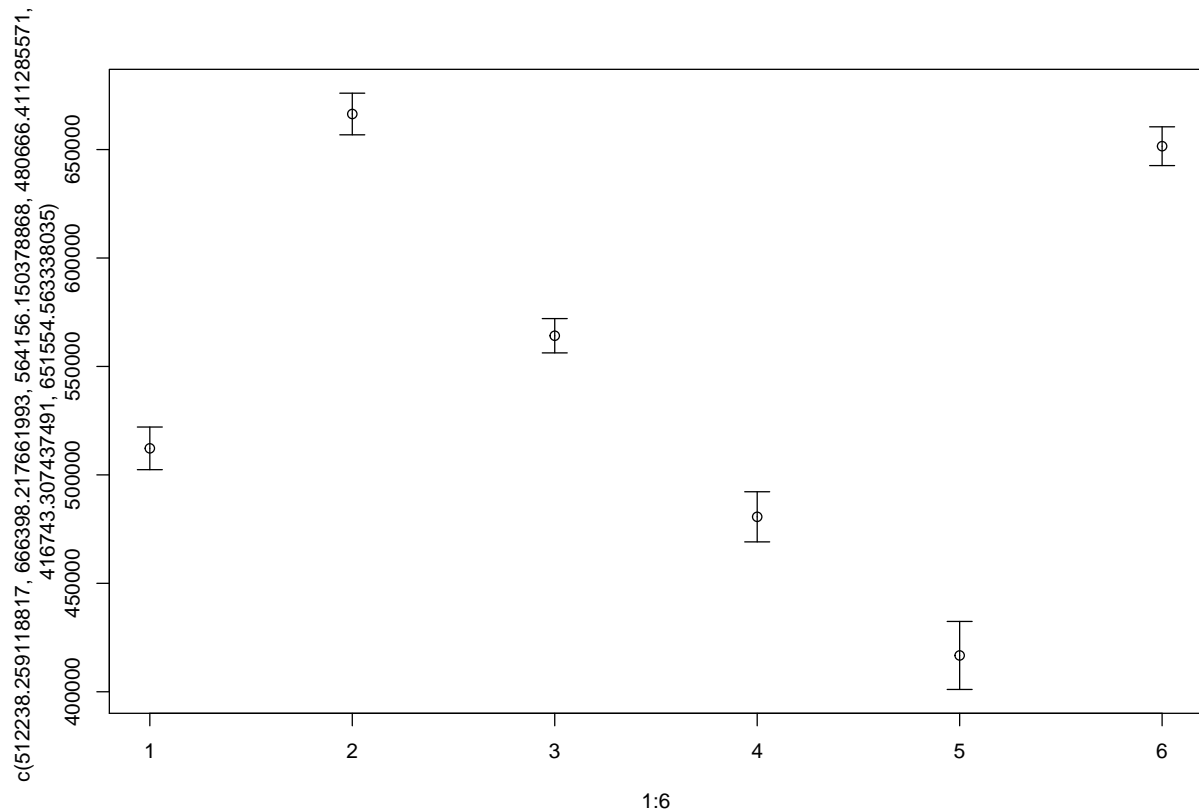
```
#### Preparação/Gráfico de intervalos do modelo original ----
```

```
#rm(SCc.CI)
```

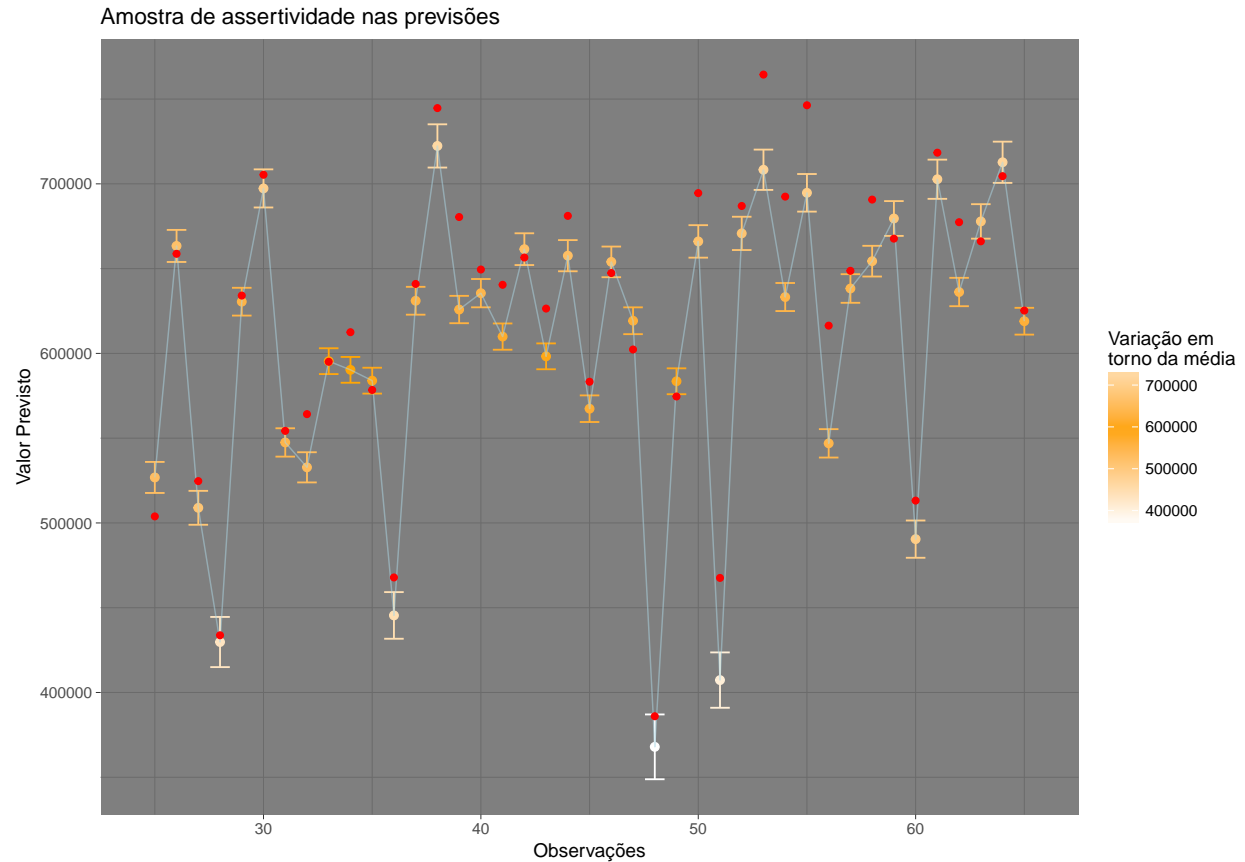
```
SCc.CI <- as.tibble(predict.lm(lm.cai_SCc, level=0.95, interval = 'confidence'))
SCc.CI <- add_column(SCc.CI, rn = as.numeric(row.names(SCc.CI)))
cai_SCc <- add_column(cai_SCc, rn = as.numeric(row.names(SCc.CI)))
```

```
#### Intervalos pelo pacote plotrix -----
```

```
plotCI(SCc.CI$fit[1:6], ui = SCc.CI$upr[1:6], li = SCc.CI$lwr[1:6], ylab = NULL, xlab = NULL)
```



```
#### Intervalos com apresentação melhorada pelo GGplot2 -----
ggplot(SCc.CI[25:65,], aes(x = rn, y = fit, col = fit)) +
  geom_point(size = 2) +
  geom_errorbar(aes(ymax = upr, ymin = lwr)) +
  geom_line(color = "light blue", size = 0.4, alpha = 0.5) +
  geom_point(data = cai_SCc[25:65,], aes(y = sadt), col = "red", size = 1.5) +
  theme_dark() +
  scale_colour_gradient2(name="Variação em\ntorno da média",
    low = "white",
    mid = "orange",
    midpoint = 591713,
    high = "white") +
  labs(title = 'Amostra de assertividade nas previsões',
    x = 'Observações', y = 'Valor Previsto')
```



```
#### Teste de distribuição T para média de SADT, IC 95% -----
t.test(cai_ScC$sadt, conf.level = 0.95)
```

```
##
## One Sample t-test
##
## data: cai_ScC$sadt
## t = 50.574, df = 77, p-value < 0.00000000000000022
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## 568415.6 615010.4
## sample estimates:
## mean of x
## 591713
```

```
#### Teste de distribuição T para média do modelo, IC 95% ---
t.test(fitted(lmT.SCc), conf.level = 0.95) #IC para modelo
```

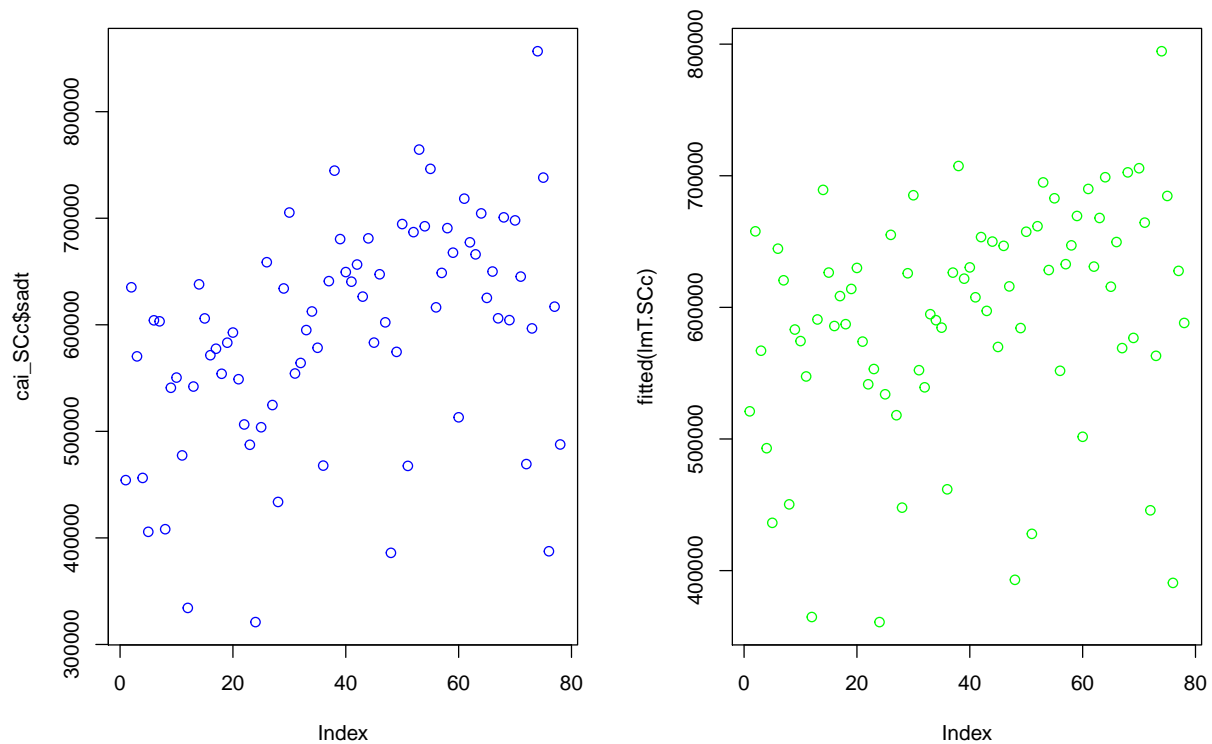
```
##
## One Sample t-test
##
## data: fitted(lmT.SCc)
## t = 60.248, df = 77, p-value < 0.00000000000000022
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## 571963.2 611063.3
## sample estimates:
```

```
## mean of x
## 591513.2

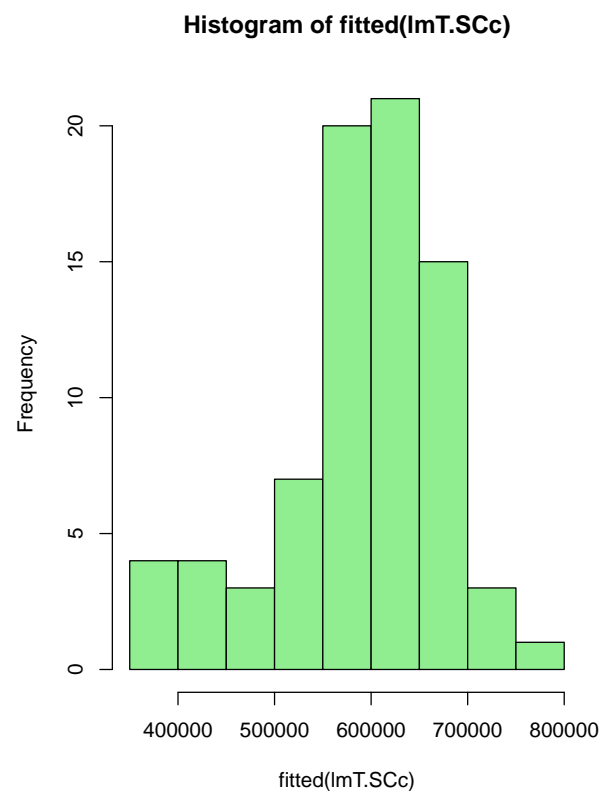
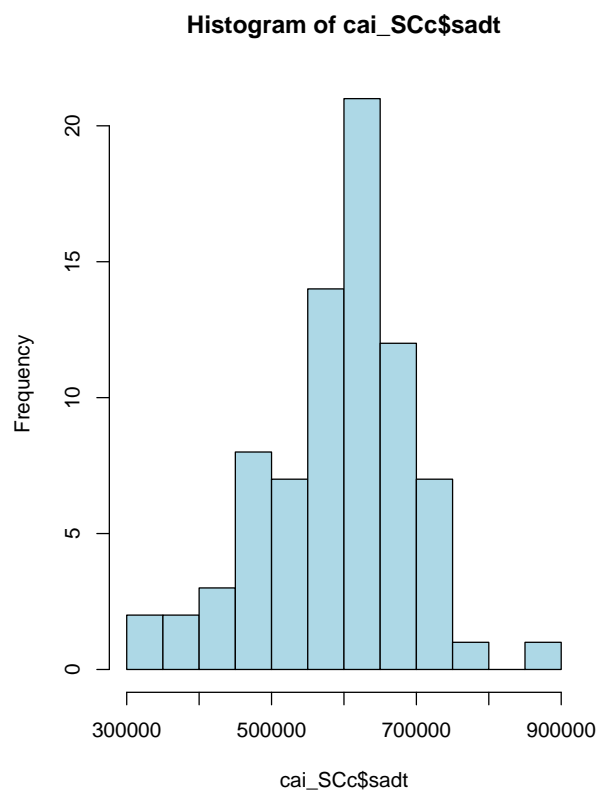
#### Intervalo para coeficientes do modelo original, IC 95% -
confint(lm.cai_SCc, level=0.95)

##                2.5 %          97.5 %
## (Intercept) 31427.7853 114155.49509
## consulta    10.7215    12.54441

#### Gráfico padrão de SADT e valores previstos -----
par(mfrow = c(1,2), oma = c(0, 0, 1.1, 0))
plot(cai_SCc$sadt, col = 'blue')
plot(fitted(lmT.SCc), col = 'green')
```

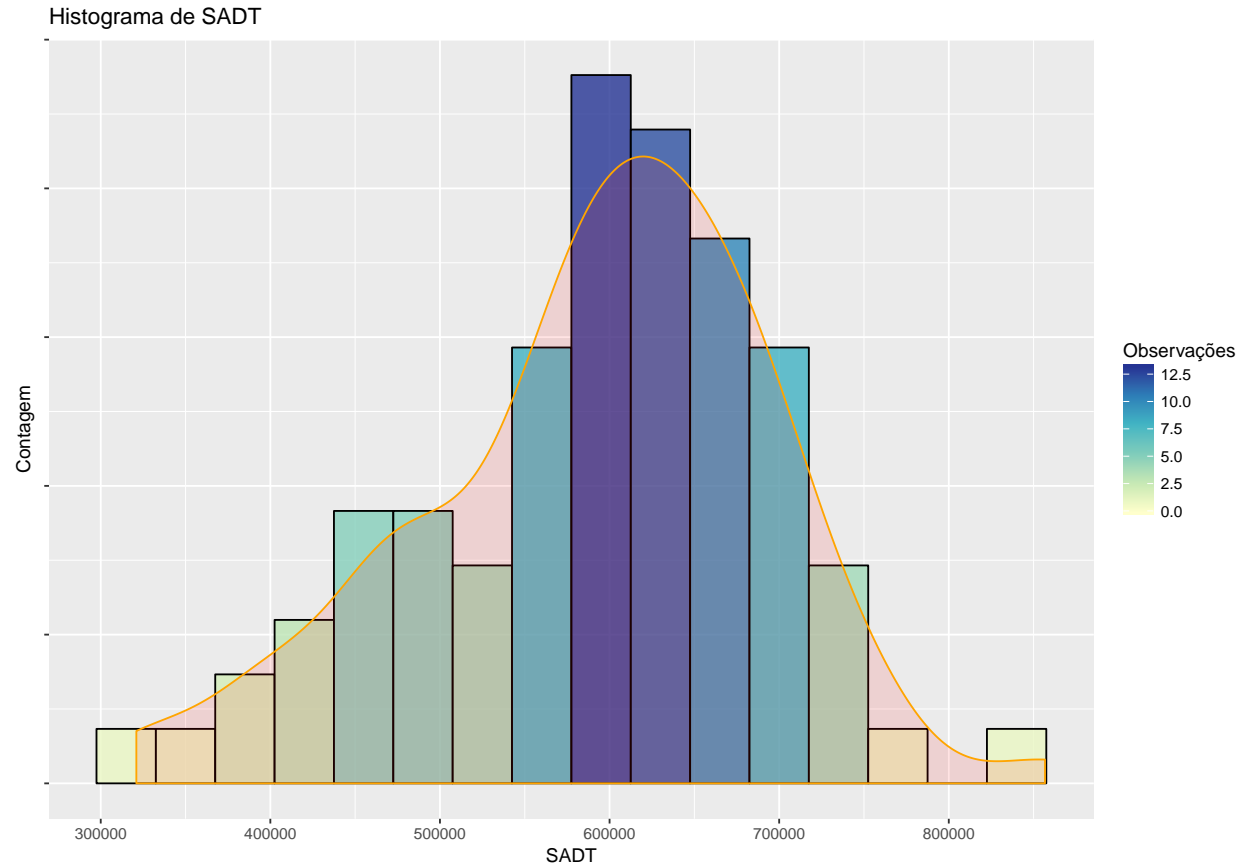


```
#### Histograma padrão para SADT e modelo -----
hist(cai_SCc$sadt, col = 'light blue')
hist(fitted(lmT.SCc), col = 'light green')
```

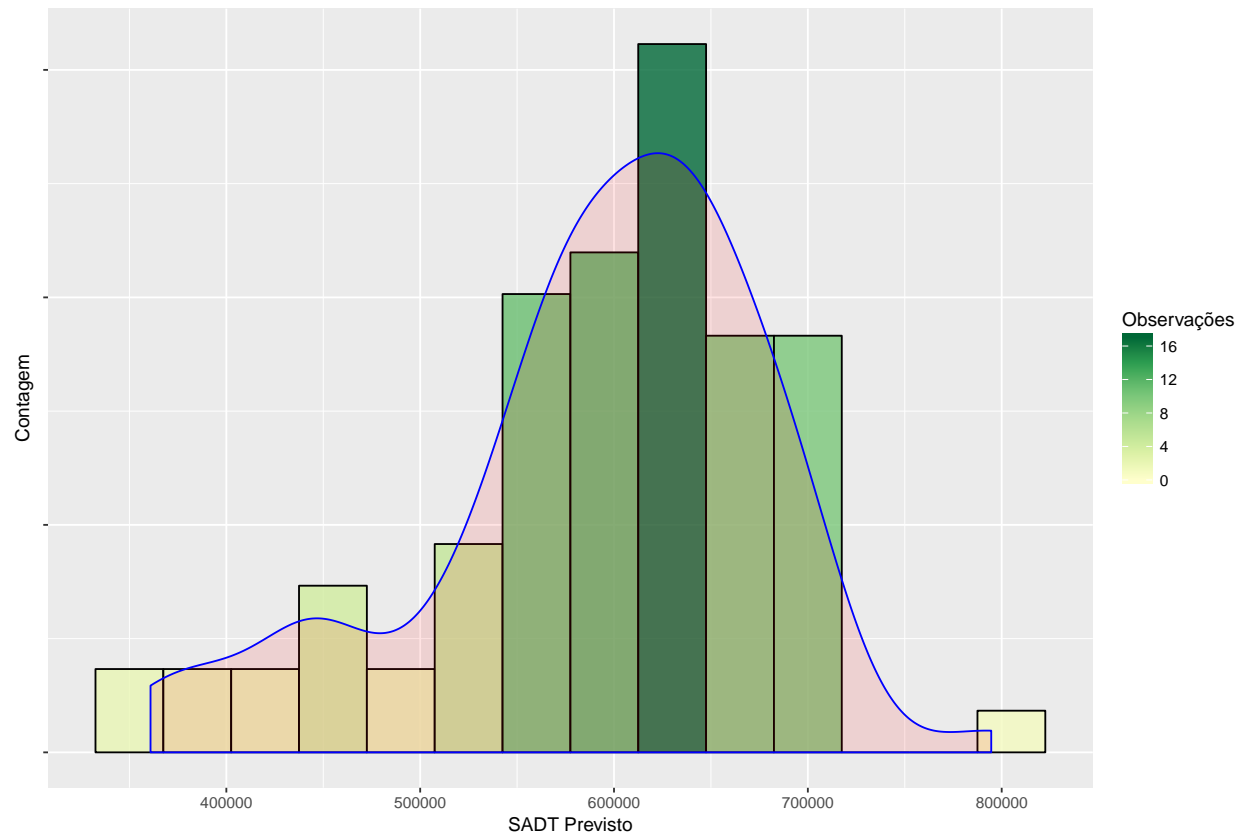
```
par(mfrow = c(1,1), oma = c(0, 0, 0, 0))

#### Histograma para SADT e modelo pelo GGPlot2 -----
ggplot(cai_SCc,aes(sadt)) +
  stat_bin(aes(y =..density..,
              fill = ..count..),
          col = "black",
          binwidth = 35000,
          alpha = 0.8) +
  geom_density(fill = "red",
              color = "orange",
              alpha = 0.11) +
  scale_x_continuous(breaks = seq(200000, 800000, by = 100000)) +
  scale_y_continuous(labels = NULL) +
  labs(title = 'Histograma de SADT', x = 'SADT', y = 'Contagem') +
  scale_fill_distiller(name = 'Observações',
                      palette = 'YlGnBu',
                      direction = 1)
```

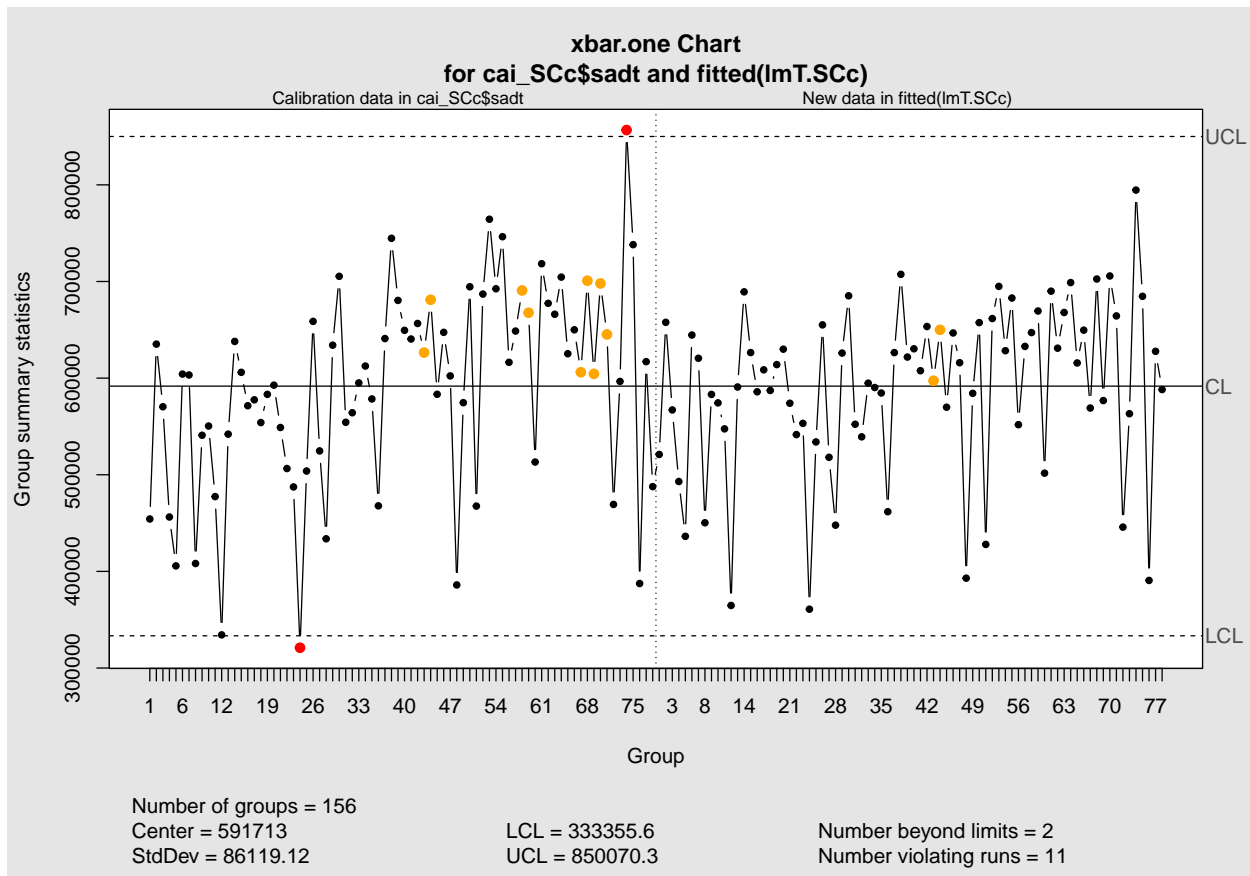


```
ggplot(cai_ScC,aes(predT)) +
  stat_bin(aes(y =..density..,
              fill = ..count..),
          col = "black",
          binwidth = 35000,
          alpha = 0.8) +
  geom_density(fill = "red",
              color = "blue",
              alpha = 0.11) +
  scale_x_continuous(breaks = seq(200000, 800000, by = 100000)) +
  scale_y_continuous(labels = NULL) +
  labs(title = 'Histograma de SADT Previsto pelo modelo', x = 'SADT Previsto', y = 'Contagem') +
  scale_fill_distiller(name = 'Observações',
                      palette = 'YlGn',
                      direction = 1)
```

Histograma de SADT Previsto pelo modelo

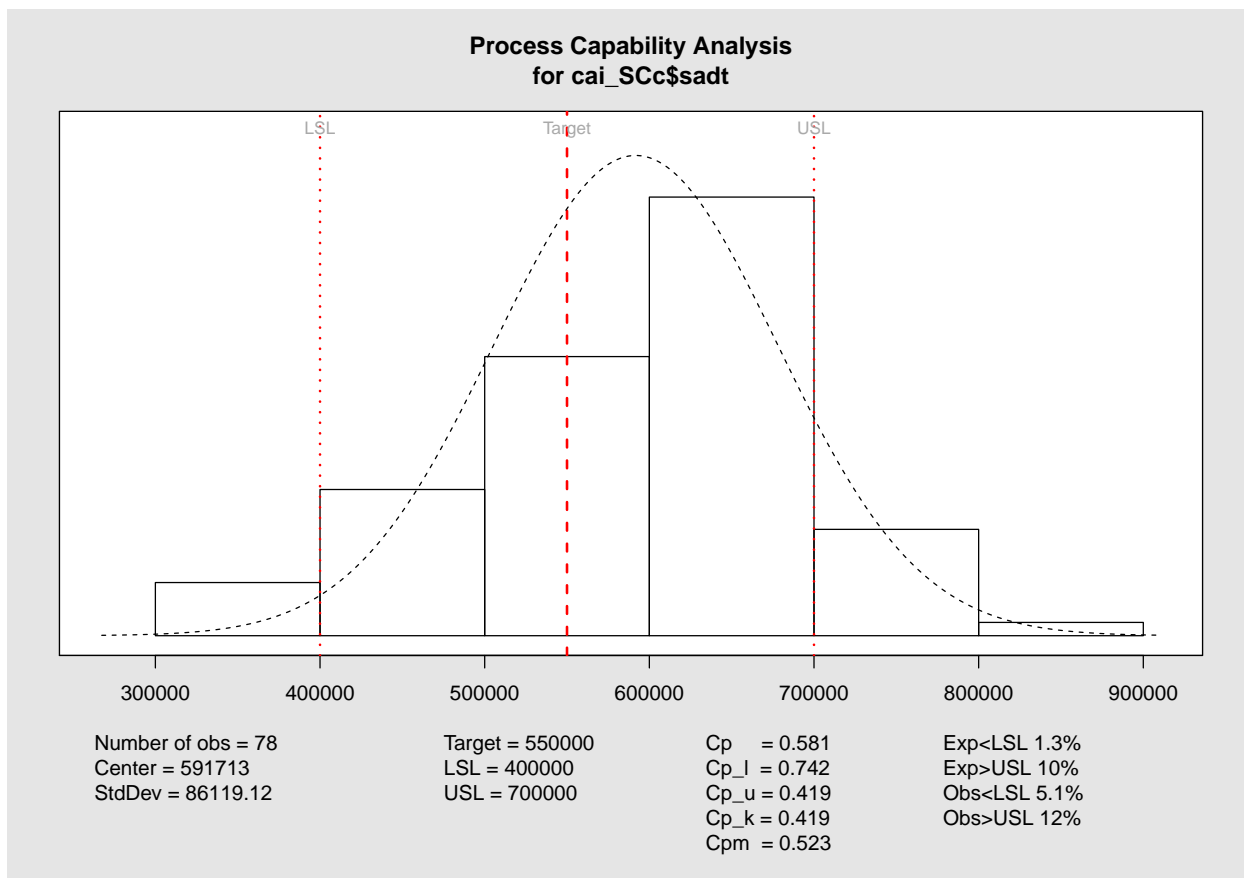


```
#### Gráfico de Controle Estatístico de Processo -----
qcc.cai_SCc <- qcc(cai_SCc$sadt, type = 'xbar.one', newdata = fitted(lmT.SCc));qcc.cai_SCc
```



```
## List of 15
## $ call      : language qcc(data = cai_SCc$sadt, type = "xbar.one", newdata = fitted(lmT.SCc))
## $ type      : chr "xbar.one"
## $ data.name  : chr "cai_SCc$sadt"
## $ data       : num [1:78, 1] 454221 635192 570367 456290 405717 ...
## .. attr(*, "dimnames")=List of 2
## $ statistics : Named num [1:78] 454221 635192 570367 456290 405717 ...
## .. attr(*, "names")= chr [1:78] "1" "2" "3" "4" ...
## $ sizes      : int [1:78] 1 1 1 1 1 1 1 1 1 1 ...
## $ center     : num 591713
## $ std.dev    : num 86119
## $ newstats   : Named num [1:78] 521009 657769 567067 493000 436292 ...
## .. attr(*, "names")= chr [1:78] "1" "2" "3" "4" ...
## $ newdata    : num [1:78, 1] 521009 657769 567067 493000 436292 ...
## .. attr(*, "dimnames")=List of 2
## $ newsizes   : Named int [1:78] 1 1 1 1 1 1 1 1 1 1 ...
## .. attr(*, "names")= chr [1:78] "1" "2" "3" "4" ...
## $ newdata.name: chr "fitted(lmT.SCc)"
## $ nsigmas    : num 3
## $ limits     : num [1, 1:2] 333356 850070
## .. attr(*, "dimnames")=List of 2
## $ violations  :List of 2
## - attr(*, "class")= chr "qcc"

#### Histograma e densidade com alvos para capacidade -----
process.capability (qcc.cai_SCc, spec.limits=c(400000,700000))
```



```
##
## Process Capability Analysis
##
## Call:
## process.capability(object = qcc.cai_SCc, spec.limits = c(400000, 700000))
##
## Number of obs = 78          Target = 5.5e+05
##      Center = 5.917e+05      LSL = 4e+05
##      StdDev = 8.612e+04      USL = 7e+05
##
## Capability indices:
##
##      Value      2.5%    97.5%
## Cp      0.5806  0.4890  0.6720
## Cp_l    0.7420  0.6257  0.8584
## Cp_u    0.4191  0.3358  0.5024
## Cp_k    0.4191  0.3199  0.5184
## Cpm     0.5225  0.4332  0.6117
##
## Exp<LSL 1.3%  Obs<LSL 5.1%
## Exp>USL 10%  Obs>USL 12%
```

Visualizações de Dados

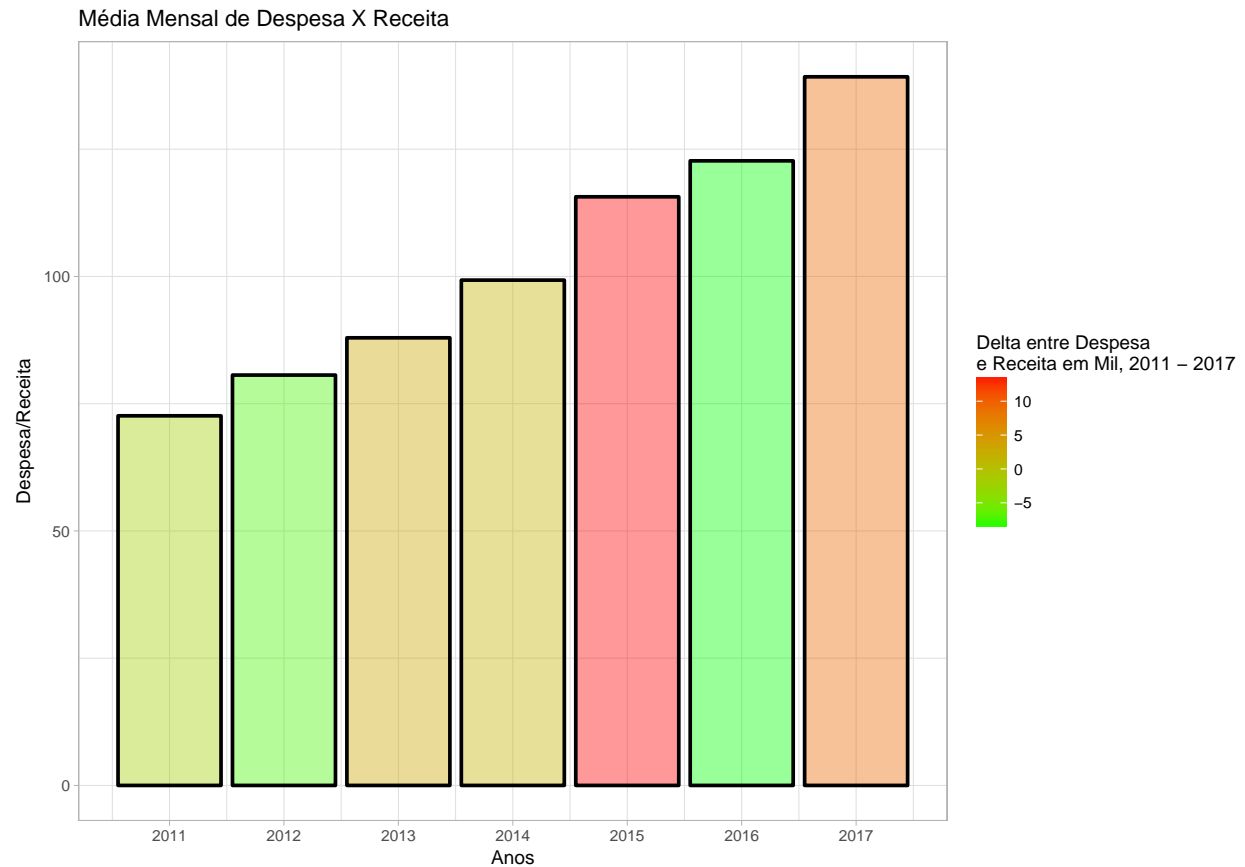
Retorno para Conteúdo

Além de suas capacidades estatísticas, o potencial de utilização da ferramenta apenas para demonstração de resultados é muito elevado, pois quase todos os aspectos da área de desenho podem ser modificados, e o conceito de gráfico em camadas permite a acumulação e troca de estruturas que não são possíveis no MS Excel:

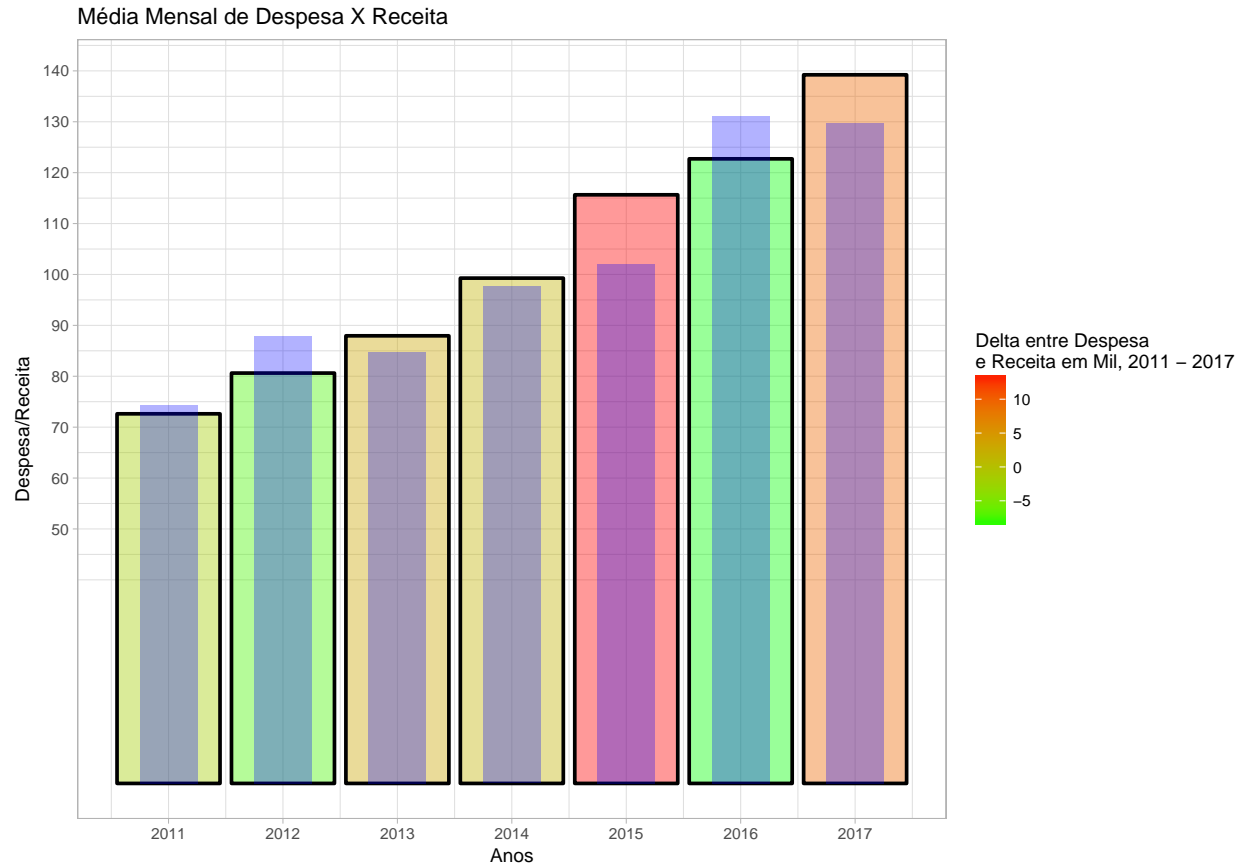
```
#### Importação de dados e inclusão de coluna -----
rec_des <- read_excel('Gráfico e Previsão 2017 2.xlsx')
rec_des <- rec_des %>% add_column(Diff = rec_des$Despesa - rec_des$Receita);as.tibble(rec_des)

## # A tibble: 7 x 4
##   Ano   Despesa   Receita   Diff
##   <dbl>   <dbl>   <dbl>   <dbl>
## 1  2011  72621707  74403028 -1781321
## 2  2012  80623725  87995035 -7371310
## 3  2013  87945889  84768559  3177330
## 4  2014  99281839  97643015  1638824
## 5  2015 115650596 102083161 13567435
## 6  2016 122717187 131155095 -8437907
## 7  2017 139227700 129831998  9395702

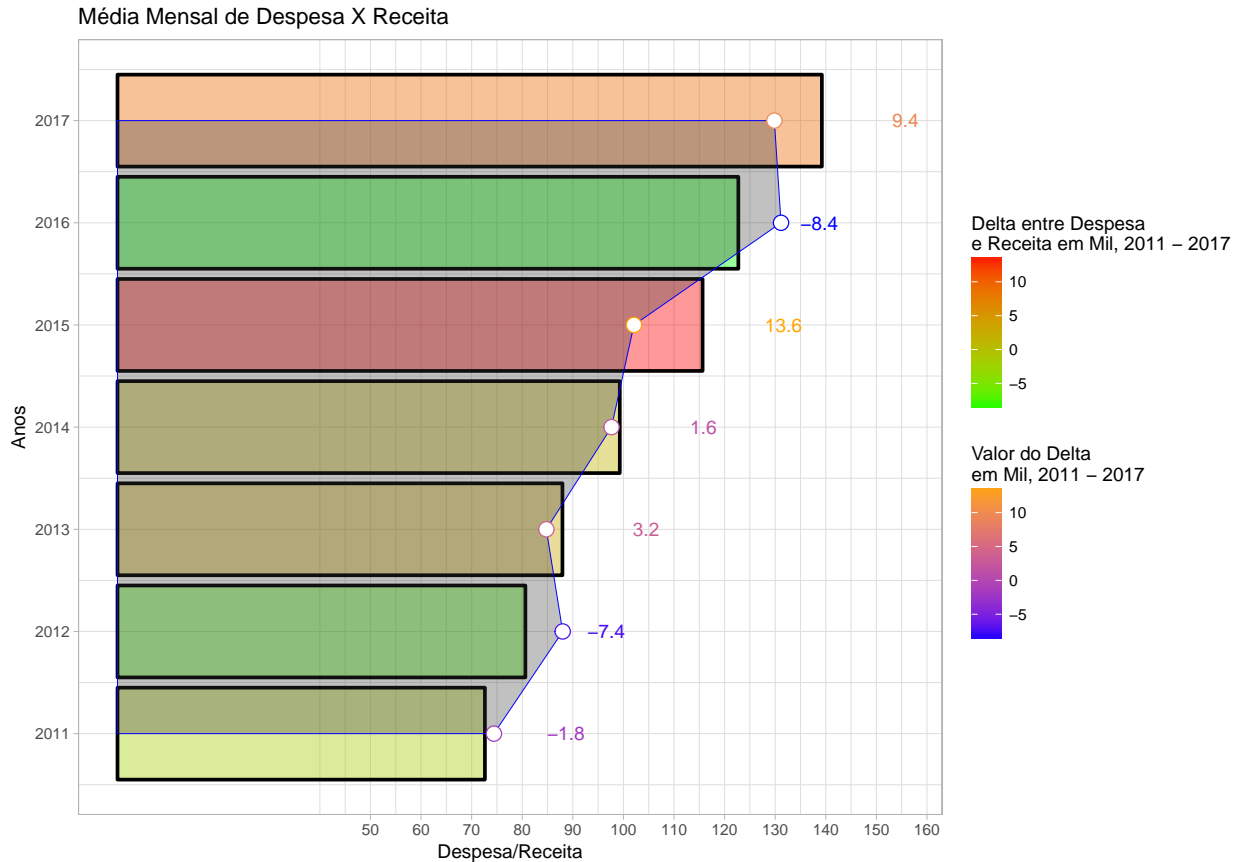
#### Gráfico base -----
GGrec_des <- ggplot(data = rec_des, aes(x = Ano)) + # Dados básicos e eixos comuns
  # Barras de despesa
  geom_col(aes(y = Despesa/1000000, fill = Diff/1000000),
    width = 0.9,
    alpha=0.4,
    #stat = "sum",
    col = "black",
    size = 1) +
  # Efeito gradiente para qualquer preenchimento usado na estética do gráfico
  scale_fill_gradient(name="Delta entre Despesa\ne Receita em Mil, 2011 - 2017",
    low = "green",
    high = "red",
    space = "Lab",
    guide = "colourbar") +
  # Controle da quebra da escala no eixo X
  scale_x_continuous(breaks = seq(2011, 2017, by = 1)) +
  # Mudança de orientação do texto da legenda do eixo x
  theme(axis.text.x = element_text(angle = 45,
    vjust = 1,
    size = 9,
    hjust = 1)) +
  # Tema geral da área do gráfico
  theme_light() +
  # Textos de eixos e Título
  labs(title = 'Média Mensal de Despesa X Receita', x = 'Anos', y = 'Despesa/Receita');GGrec_des
```



```
#### Gráfico com adição de camada com coluna de receita -----
GGrec_des +
  # Controle da quebra da escala no eixo y
  scale_y_continuous(breaks = seq(50, 180, by = 10)) +
  # Barras de receita
  geom_bar(aes(y = Receita/1000000),
            width = 0.5, alpha=0.3,
            fill = "blue",
            stat = "sum",
            size = 1)
```



```
#### Gráfico com adição de camada com área de receita -----
GGrec_des +
  # Área de receita
  geom_area(aes(y = Receita/1000000),
    col = "blue",
    size = 0.2,
    alpha = 0.3) +
  # Pontos para marcação da receita sobre a área
  geom_point(aes(y = Receita/1000000, col = Diff/1000000),
    size=3.8, shape=21, fill="white") +
  # Texto com valores
  geom_text(aes(y = (Despesa/1000000) + 16,
    label = format((Diff/1000000), digits = 2),
    col = Diff/1000000),
    size = 4) +
  # Efeito gradiente para qualquer cor usada na estética do gráfico
  scale_colour_gradient(name="Valor do Delta\nem Mil, 2011 - 2017",
    low = "blue",
    high = "orange",
    space = "Lab",
    guide = "colourbar") +
  # Controle da quebra da escala no eixo y
  scale_y_continuous(breaks = seq(50, 180, by = 10)) +
  # troca de eixos somente para apresentação
  coord_flip()
```

Conclusão

Retorno para Conteúdo

A partir do estudo realizado foram confirmadas as seguintes premissas:

- A plataforma de código aberto “R” para computação estatística constitui uma ferramenta profissional, flexível e com alto poder de aplicação, desde de funções inerentes até suas estruturas secundárias para manipulação de dados e apresentação gráfica. Empresas como Google, Pfizer, Microsoft, Uber, Facebook, IBM, Ford, Novartis, Roche, New York Times (esta última para para visualização de dados) já fazem uso em grande escala, contando com equipes dedicadas e especialistas, incluindo desenvolvimento personalizado da ferramenta.
- Os resultados encontrados no piloto estatístico de Consultas X Exames confirmam o forte relacionamento entre estes componentes, e seu modelo linear poderá trazer uma vantagem estratégica no relacionamento com prestadores, principalmente no controle de resultados do projeto de “Banco de Consultas”.

Estas respostas apontam para grande valia de futuros trabalhos realizados em conjunto com equipes técnicas do Planserv, e podem alavancar decisões em diversas disciplinas relacionadas a informação, com grandes contribuições para suporte a decisão, identificação de padrões, diagnóstico de distorções e estratégia em geral.