

appraiseR

Engenharia de Avaliações no R

Droubi

15/10/2020

Introdução

appraiseR é um pacote **R** para facilitar a prática da Engenharia de Avaliações segundo a NBR 14.653-02 (NBR 14653-2: Avaliação de Bens Parte 2: Imóveis Urbanos 2011).

Conjuntos de dados disponíveis

Foram disponibilizados alguns conjuntos de dados visando possibilitar a confecção de testes dos algoritmos.

Uns destes conjuntos é um conjunto de 50 dados de apartamentos em Florianópolis. Este conjunto de dados é proveniente de Hochheim (2015).

```
data("centro_2015")
dados <- centro_2015@data
```

dados

```
## # A tibble: 53 x 7
##   valor area_total quartos suites garagens dist_b_mar padrao
##   <dbl>    <dbl>    <dbl> <dbl>    <dbl>    <dbl> <fct>
## 1 1060000      350         3     1        2      720 medio
## 2  510000     137.         3     1        1      665 medio
## 3  780000     165.         3     1        2      415 medio
## 4  550000     175.         3     1        1      320 medio
## 5  850000     123.         3     1        3      895 alto
## 6  300000      89.8         2     0        1      645 baixo
## 7  750000     174         2     1        2      860 alto
## 8  650000     123         3     1        1      745 alto
## 9  620000     121         3     1        1      745 alto
## 10 740000     109         3     1        1      300 medio
## # ... with 43 more rows
```

Seleção de modelos

A seleção de modelos pode ser feita através da função `bestfit`, fornecendo-se uma fórmula com as variáveis desejadas, sem qualquer transformação (o algoritmo se encarrega de testar as transformações).

```
best_fits <- bestfit(valor~area_total + quartos + suites + garagens +
                    dist_b_mar + padrao, data = dados)
```

O pacote gera um objeto da classe `bestfit`

```
class(best_fits)
```

```
## [1] "bestfit"
```

Uma vez feitas as combinações de transformações, elas podem ser vistas no console do **R**:

```
print(best_fits)
```

```
## Call:
## bestfit(formula = valor ~ area_total + quartos + suites + garagens +
##         dist_b_mar + padrao, data = dados)
##
## Best 10 fits:
##      id valor area_total quartos  suites garagens dist_b_mar  adj_R2
## 443   1 rsqrt      sqrt  rsqrt identity    sqrt    rsqrt 0.9480455
## 395   2 rsqrt  identity  rsqrt identity    sqrt    rsqrt 0.9477222
## 955   3 rsqrt      sqrt  rsqrt    sqrt    sqrt    rsqrt 0.9474578
## 907   4 rsqrt  identity  rsqrt    sqrt    sqrt    rsqrt 0.9472744
## 439   5 rsqrt      sqrt    log identity    sqrt    rsqrt 0.9471142
## 951   6 rsqrt      sqrt    log    sqrt    sqrt    rsqrt 0.9468425
## 391   7 rsqrt  identity    log identity    sqrt    rsqrt 0.9466028
## 903   8 rsqrt  identity    log    sqrt    sqrt    rsqrt 0.9465023
## 411   9 rsqrt      log  rsqrt identity    sqrt    rsqrt 0.9460101
## 407  10 rsqrt      log    log identity    sqrt    rsqrt 0.9455580
## ...
```

Através do comando `summary` é impresso no console um sumário do modelo mais ajustado:

```
summary(best_fits)
```

```
## Call:
## bestfit(formula = valor ~ area_total + quartos + suites + garagens +
##         dist_b_mar + padrao, data = dados)
##
## Best (Chosen) Transformations:
##      id valor area_total quartos  suites garagens dist_b_mar  adj_R2
## 443  1 rsqrt      sqrt  rsqrt identity      sqrt      rsqrt 0.9480455
##
## Best (Chosen) fit LM summary:
##
## Call:
## lm(formula = "rsqrt(valor) ~ sqrt(area_total) + rsqrt(quartos) + identity(suites) + sqrt(g
##     aragens) + rsqrt(dist_b_mar) + (padrao)",
##     data = dados, subset = NULL)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.144e-04 -5.344e-05  8.870e-07  4.272e-05  1.729e-04
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.780e-03  1.301e-04  13.681 < 2e-16 ***
## sqrt(area_total) -2.279e-05  5.295e-06  -4.304 9.81e-05 ***
## rsqrt(quartos)   6.561e-04  1.269e-04   5.169 6.14e-06 ***
## identity(suites) -4.240e-05  2.060e-05  -2.058  0.0459 *
## sqrt(garagens)  -2.711e-04  4.426e-05  -6.125 2.62e-07 ***
## rsqrt(dist_b_mar) -2.628e-03  5.099e-04  -5.154 6.45e-06 ***
## padraomedio    -2.214e-04  4.586e-05  -4.828 1.85e-05 ***
## padraoalto     -2.576e-04  4.605e-05  -5.595 1.52e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.428e-05 on 42 degrees of freedom
## (3 observations deleted due to missingness)
## Multiple R-squared:  0.9555, Adjusted R-squared:  0.948
## F-statistic: 128.7 on 7 and 42 DF, p-value: < 2.2e-16
##
## NBR-14.653-2 check:
## Minimum number of market data:
## [1] "n = 53 >= 42 --> Grau III"
## Max significance level allowed for each predictor:
## [1] "t máximo = 4.59 % < 10% --> Grau III"
## Max significance level allowed for F-test:
## [1] "p-valor F = 2.79e-24 % < 1% --> Grau III"
```

É possível explorar melhor a tabelas das transformações, que está armazenada sob o nome de tabs dentro do objeto.

```
names(best_fits)
```

```
## [1] "adj.R2"      "tab"          "call"          "combinations" "response"
## [6] "predictors"  "newdata"
```

Por exemplo, pode-se selecionar apenas os modelos cuja transformação da variável resposta foi a transformação logaritmica.

```
df <- best_fits$tab  
head(df[which(df$valor == "log"), ])
```

```
##      id valor area_total  quartos  suites garagens dist_b_mar  adj_R2  
## 335 32   log   identity    sqrt identity    sqrt    rsqrt 0.9425135  
## 327 34   log   identity      log identity    sqrt    rsqrt 0.9422108  
## 323 37   log   identity identity identity    sqrt    rsqrt 0.9420366  
## 331 42   log   identity  rsqrt identity    sqrt    rsqrt 0.9412708  
## 334 52   log   identity    sqrt identity    sqrt      log 0.9402942  
## 322 55   log   identity identity identity    sqrt      log 0.9399696
```

Pode-se, então, selecionar o sumário de um modelo diferente, através da escolha apropriada do `id` na tabela acima.

```
summary(best_fits, fit = 32)
```

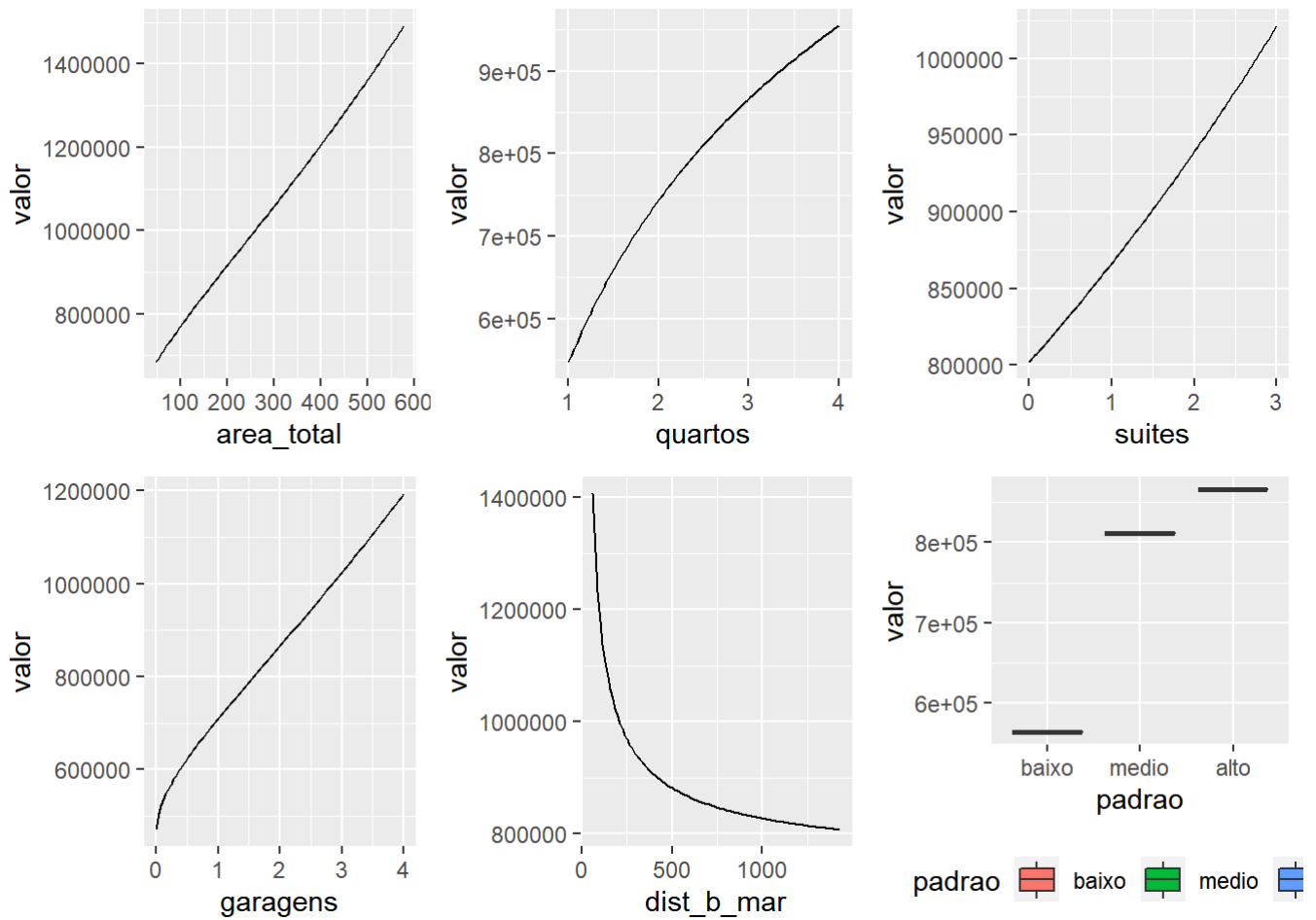
```
## Call:
## bestfit(formula = valor ~ area_total + quartos + suites + garagens +
##         dist_b_mar + padrao, data = dados)
##
## Best (Chosen) Transformations:
##      id valor area_total quartos  suites garagens dist_b_mar  adj_R2
## 335 32   log   identity    sqrt identity    sqrt    rsqrt 0.9425135
##
## Best (Chosen) fit LM summary:
##
## Call:
## lm(formula = "log(valor) ~ identity(area_total) + sqrt(quartos) + identity(suites) + sqrt
##    (garagens) + rsqrt(dist_b_mar) + (padrao)",
##      data = dados, subset = NULL)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.30047 -0.06983 -0.01359  0.07933  0.35987
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.144e+01  1.631e-01  70.112 < 2e-16 ***
## identity(area_total) 1.844e-03  2.819e-04   6.540 6.64e-08 ***
## sqrt(quartos)      4.468e-01  1.185e-01   3.770 0.000504 ***
## identity(suites)    1.055e-01  3.634e-02   2.903 0.005860 **
## sqrt(garagens)      4.282e-01  7.751e-02   5.525 1.91e-06 ***
## rsqrt(dist_b_mar)   4.212e+00  9.096e-01   4.631 3.49e-05 ***
## padraomedio        2.167e-01  8.199e-02   2.643 0.011491 *
## padraoalto         2.721e-01  8.198e-02   3.319 0.001872 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1492 on 42 degrees of freedom
## (3 observations deleted due to missingness)
## Multiple R-squared:  0.9507, Adjusted R-squared:  0.9425
## F-statistic: 115.8 on 7 and 42 DF,  p-value: < 2.2e-16
##
## NBR-14.653-2 check:
## Minimum number of market data:
## [1] "n = 53 >= 42 --> Grau III"
## Max significance level allowed for each predictor:
## [1] "t máximo = 1.15 % < 10% --> Grau III"
## Max significance level allowed for F-test:
## [1] "p-valor F = 2.31e-23 % < 1% --> Grau III"
```

Diagnósticos

Gráficos do modelo

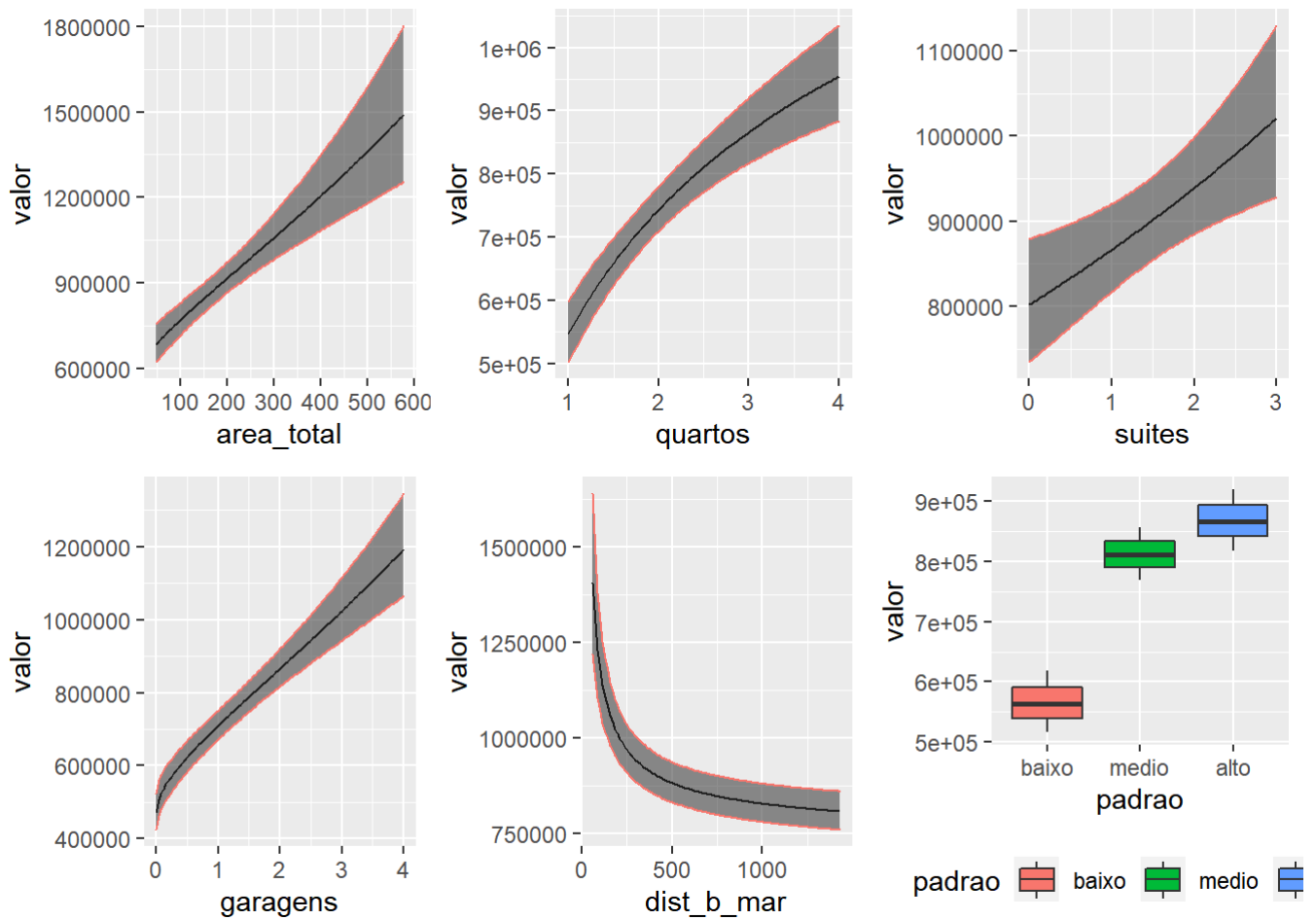
É possível elaborar gráficos do modelo através da função `plotmod`:

```
plotmod(best_fits)
```



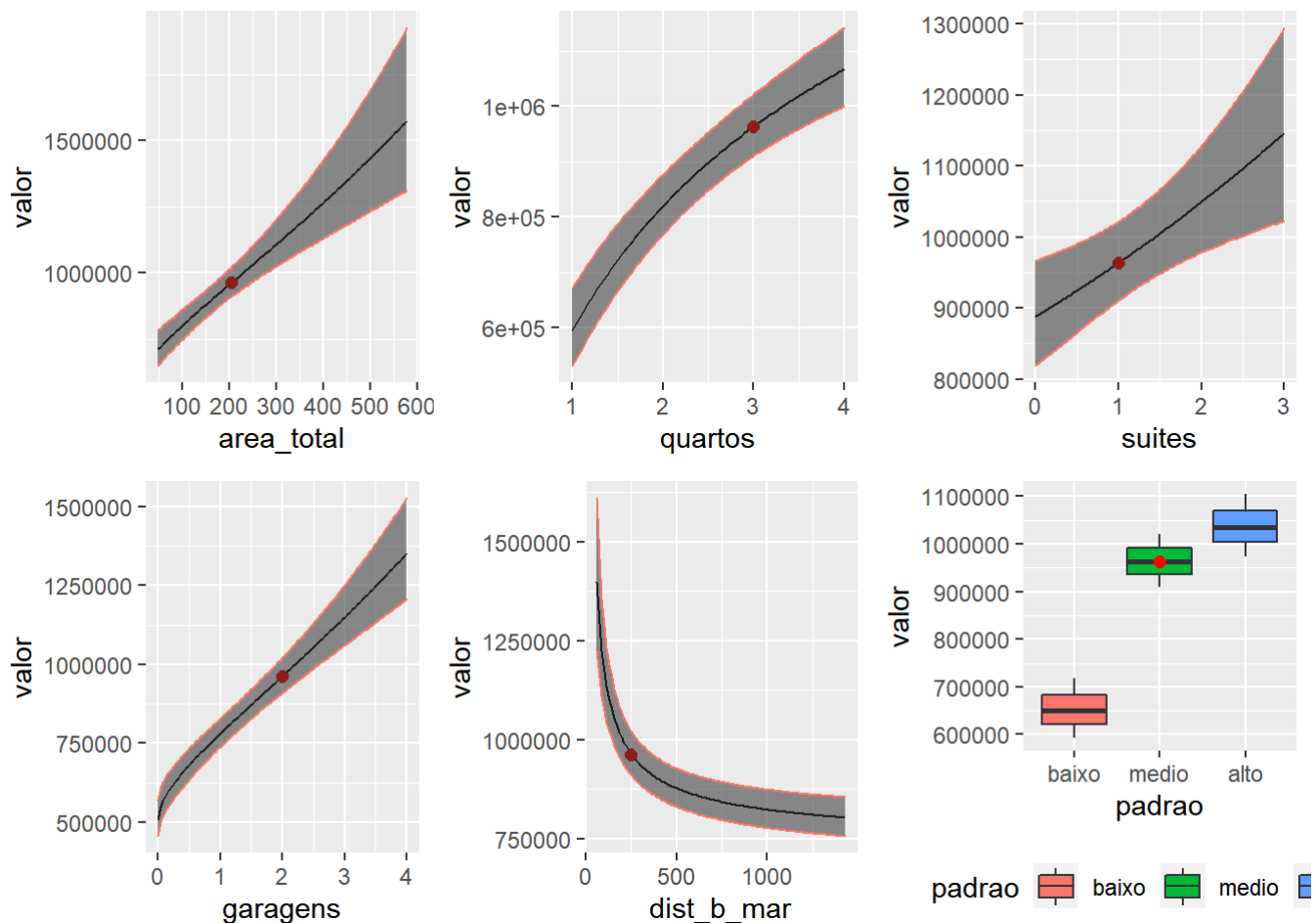
Para adicionar intervalos de confiança ou predição e escolher o nível adequado, faz-se:

```
plotmod(best_fits, interval = "confidence", level = 0.80)
```



É possível visualizar o comportamento do modelo no ponto de previsão de valores:

```
plotmod(best_fits, interval = "confidence", level = 0.80,
        local = list(area_total = 205, quartos = 3, suites = 1, garagens = 2,
                      dist_b_mar = 250, padrao = "medio"))
```

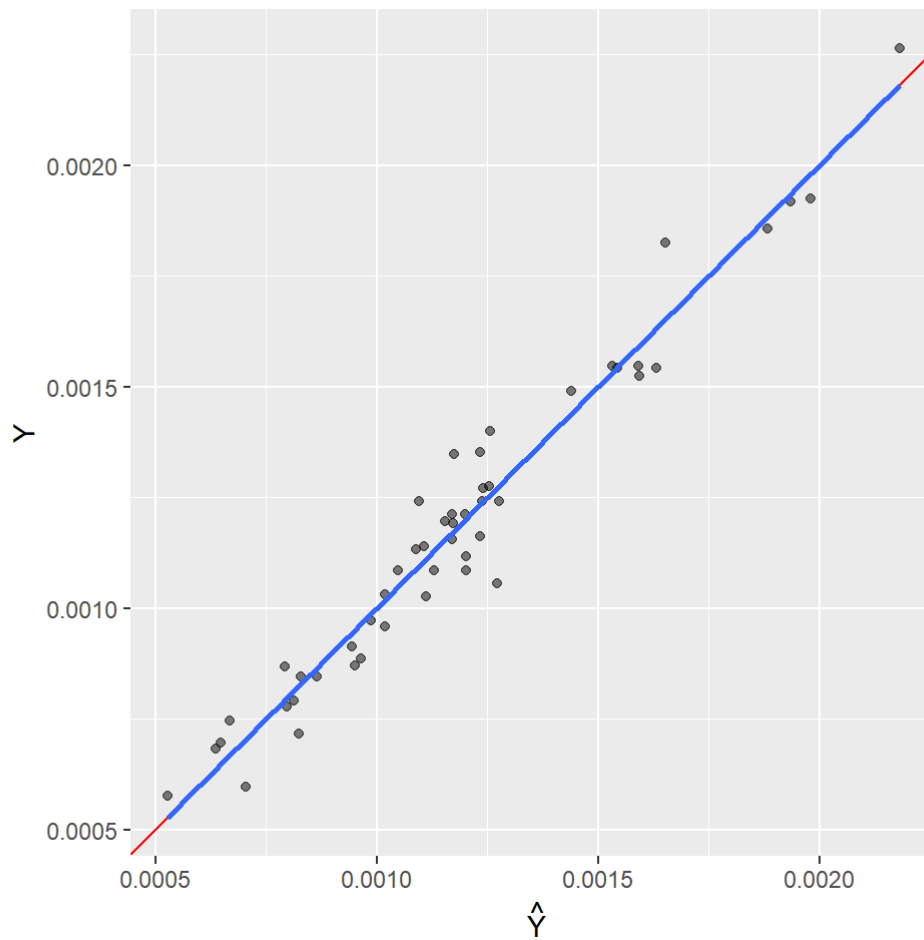


Poder de Predição

A função `power_plot` ajusta o gráfico do poder de predição do modelo. Por padrão, a função `power_plot` ajusta o poder de predição do modelo de maior $R^2_{ajustado}$.

```
power_plot(best_fits)
```

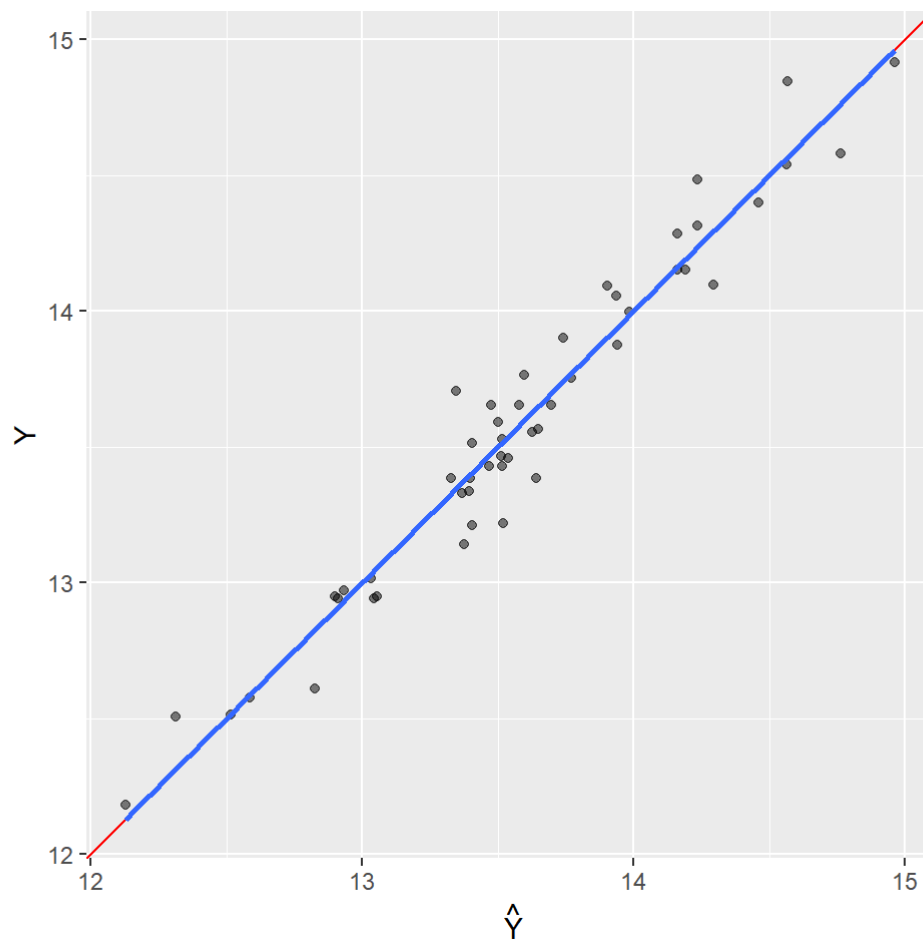
```
## `geom_smooth()` using formula 'y ~ x'
```

Porém, é possível selecionar outros modelos da tabela:

```
power_plot(best_fits, 32)
```

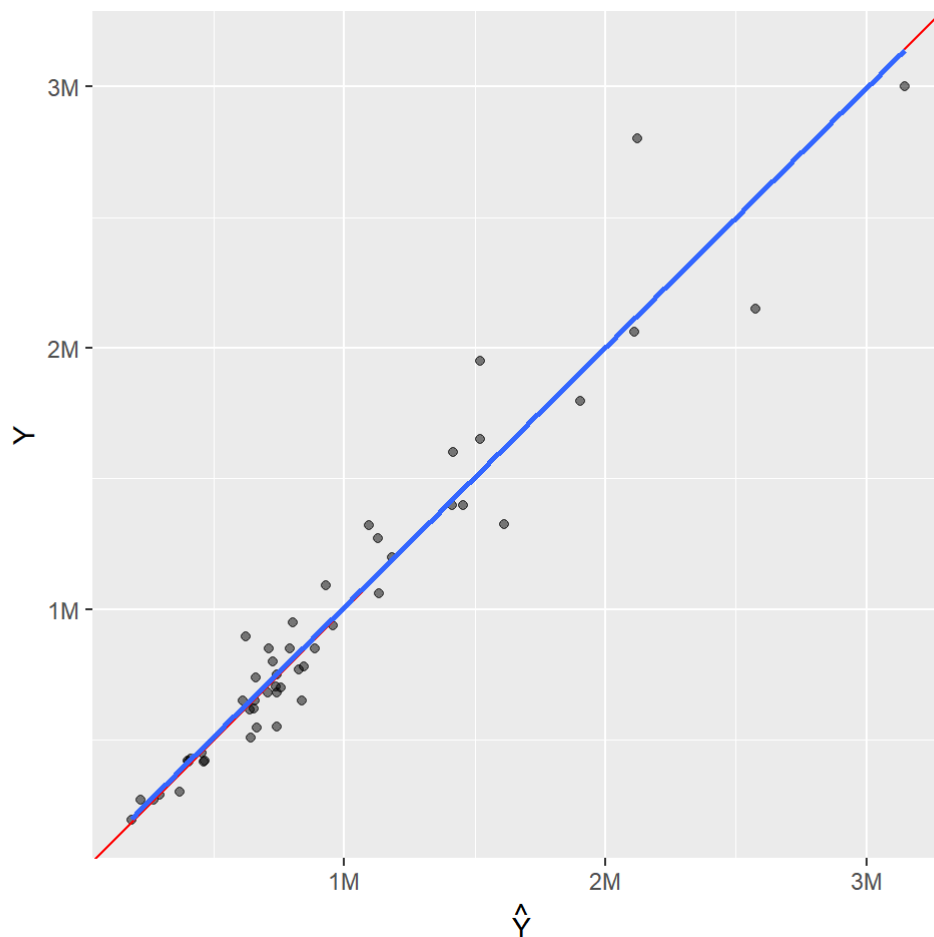
```
## `geom_smooth()` using formula 'y ~ x'
```



Assim como também é possível elaborar o gráfico na escala de valores original:

```
power_plot(best_fits, 32, func = "log")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Grau de precisão

As previsões podem ser feitas pela função `predict` aplicada ao método `bestfit`. Ao imprimí-las no console, serão mostrados os limites do intervalo escolhido, do campo de arbítrio, bem como a amplitude do intervalo e seu grau de precisão, segundo a NBR 14.653-02:

É possível fornecer parâmetros para a escolha das características dos imóveis a serem previstos através do argumento `newdata` (não mostrado). Caso não sejam fornecidos novos dados para ajuste, a função automaticamente irá procurar pelos avaliando no conjunto de dados (dados cuja variável resposta não tenha valores):

```
p <- predict(best_fits, interval = "confidence")
p
```

```
## Predictions:
##      fit      lwr      upr    AMP G.P.   C.A.I.   C.A.S.   L.I.   L.S.
## 1  650202.4  717559.8  591902.6  19.33  III  552672.1  747732.8  591902.6  717559.8
## 2  963580.5  1021476.7  910470.5  11.52  III  819043.4  1108117.6  910470.5  1021476.7
## 3  1035927.9  1105696.8  972560.0  12.85  III  880538.7  1191317.0  972560.0  1105696.8
```

```
p1 <- predict(best_fits, interval = "prediction")
p1
```

```
## Predictions:
```

```
##      fit      lwr      upr  AMP G.P.  C.A.I.  C.A.S.  L.I.  L.S.
## 1  650202.4  803989.5 536663.0 41.11   I 552672.1  747732.8 552672.1  747732.8
## 2  963580.5 1220578.2 779964.9 45.73   I 819043.4 1108117.6 819043.4 1108117.6
## 3 1035927.9 1326247.1 831463.4 47.76   I 880538.7 1191317.0 880538.7 1191317.0
```

Referências

Hochheim, Norberto. 2015. *Engenharia de Avaliações - Módulo Básico*. Florianópolis: IBAPE - SC.

NBR 14653-2: Avaliação de Bens Parte 2: Imóveis Urbanos. 2011. Rio de Janeiro: Associação Brasileira de Normas Técnicas.