Modelos Lineales en R: Parte 2

Datos de Casas en Ames, Iowa US

Estos datos contienen 2930 casas en la ciudad de Ames en Iowa en Estados Unidos. Los datos originales han sido modificados para facilitar uso y hacen parte del paquete modeldata. *modeldata* hace parte de tidyverse

El paquete que usaremos para ajustar modelos el dia de hoy se llama parnisp

Importemos las librerias necesarias

```
library(tidymodels)
```

```
## -- Attaching packages -------------------------------------- tidymodels 0.2.0 --

## v broom        0.7.12     v recipes      0.2.0
## v dials        0.1.0      v rsample      0.1.1
## v dplyr        1.0.8      v tibble       3.1.6
## v ggplot2      3.3.5      v tidyr        1.2.0
## v infer        1.0.0      v tune         0.2.0
## v modeldata    0.1.1      v workflows    0.2.6
## v parsnip      0.2.1      v workflowsets 0.2.1
## v purrr        0.3.4      v yardstick    0.0.9

## -- Conflicts ----------------------------------------- tidymodels_conflicts() --
## x purrr::discard() masks scales::discard()
## x dplyr::filter()  masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## x recipes::step()  masks stats::step()
## * Dig deeper into tidy modeling with R at https://www.tmwr.org
```

Resolvamos los conflictos entre los distintos paquetes

```
tidymodels_prefer(quiet=FALSE)
```

```
## [conflicted] Will prefer dplyr::filter over any other package
## [conflicted] Will prefer dplyr::select over any other package
## [conflicted] Will prefer dplyr::slice over any other package
## [conflicted] Will prefer dplyr::rename over any other package
## [conflicted] Will prefer dials::neighbors over any other package
## [conflicted] Will prefer parsnip::fit over any other package
## [conflicted] Will prefer parsnip::bart over any other package
## [conflicted] Will prefer parsnip::pls over any other package
## [conflicted] Will prefer purrr::map over any other package
## [conflicted] Will prefer recipes::step over any other package
## [conflicted] Will prefer themis::step_downsample over any other package
## [conflicted] Will prefer themis::step_upsample over any other package
## [conflicted] Will prefer tune::tune over any other package
## [conflicted] Will prefer yardstick::precision over any other package
## [conflicted] Will prefer yardstick::recall over any other package
## [conflicted] Will prefer yardstick::spec over any other package
## -- Conflicts ------------------------------------------ tidymodels_prefer() --
```

Carguemos los datos**

```
data(ames)

head(ames)
```

```
## # A tibble: 6 x 74
##   MS_SubClass            MS_Zoning Lot_Frontage Lot_Area Street Alley Lot_Shape
##   <fct>                  <fct>            <dbl>    <int> <fct>  <fct> <fct>
## 1 One_Story_1946_and_New~ Resident~          141    31770 Pave   No_A~ Slightly~
## 2 One_Story_1946_and_New~ Resident~           80    11622 Pave   No_A~ Regular
## 3 One_Story_1946_and_New~ Resident~           81    14267 Pave   No_A~ Slightly~
## 4 One_Story_1946_and_New~ Resident~           93    11160 Pave   No_A~ Regular
## 5 Two_Story_1946_and_New~ Resident~           74    13830 Pave   No_A~ Slightly~
## 6 Two_Story_1946_and_New~ Resident~           78     9978 Pave   No_A~ Slightly~
## # ... with 67 more variables: Land_Contour <fct>, Utilities <fct>,
## #   Lot_Config <fct>, Land_Slope <fct>, Neighborhood <fct>, Condition_1 <fct>,
## #   Condition_2 <fct>, Bldg_Type <fct>, House_Style <fct>, Overall_Cond <fct>,
## #   Year_Built <int>, Year_Remod_Add <int>, Roof_Style <fct>, Roof_Matl <fct>,
## #   Exterior_1st <fct>, Exterior_2nd <fct>, Mas_Vnr_Type <fct>,
## #   Mas_Vnr_Area <dbl>, Exter_Cond <fct>, Foundation <fct>, Bsmt_Cond <fct>,
## #   Bsmt_Exposure <fct>, BsmtFin_Type_1 <fct>, BsmtFin_SF_1 <dbl>, ...
```

Revisemos las dimensiones del dataframe

```
dim(ames)
```

```
## [1] 2930   74
```

Revisemos la lista de variables

```
names(ames)
```

```
##  [1] "MS_SubClass"        "MS_Zoning"          "Lot_Frontage"
##  [4] "Lot_Area"           "Street"             "Alley"
##  [7] "Lot_Shape"          "Land_Contour"       "Utilities"
## [10] "Lot_Config"         "Land_Slope"         "Neighborhood"
## [13] "Condition_1"        "Condition_2"        "Bldg_Type"
## [16] "House_Style"        "Overall_Cond"       "Year_Built"
## [19] "Year_Remod_Add"     "Roof_Style"         "Roof_Matl"
## [22] "Exterior_1st"       "Exterior_2nd"       "Mas_Vnr_Type"
## [25] "Mas_Vnr_Area"       "Exter_Cond"         "Foundation"
## [28] "Bsmt_Cond"          "Bsmt_Exposure"      "BsmtFin_Type_1"
## [31] "BsmtFin_SF_1"       "BsmtFin_Type_2"     "BsmtFin_SF_2"
## [34] "Bsmt_Unf_SF"        "Total_Bsmt_SF"      "Heating"
## [37] "Heating_QC"         "Central_Air"        "Electrical"
## [40] "First_Flr_SF"       "Second_Flr_SF"      "Gr_Liv_Area"
## [43] "Bsmt_Full_Bath"     "Bsmt_Half_Bath"     "Full_Bath"
## [46] "Half_Bath"          "Bedroom_AbvGr"      "Kitchen_AbvGr"
## [49] "TotRms_AbvGrd"      "Functional"         "Fireplaces"
## [52] "Garage_Type"        "Garage_Finish"      "Garage_Cars"
## [55] "Garage_Area"        "Garage_Cond"        "Paved_Drive"
## [58] "Wood_Deck_SF"       "Open_Porch_SF"      "Enclosed_Porch"
## [61] "Three_season_porch" "Screen_Porch"       "Pool_Area"
## [64] "Pool_QC"            "Fence"              "Misc_Feature"
## [67] "Misc_Val"           "Mo_Sold"            "Year_Sold"
## [70] "Sale_Type"          "Sale_Condition"     "Sale_Price"
```

```
## [73] "Longitude"          "Latitude"
```

Procesemos los datos

```
ames <- ames %>%
  mutate(Sale_Price_log10 = log10(Sale_Price))
```

Particion Estratificada

```
set.seed(123)
ames_split <- initial_split(ames, prop = 0.80, strata = Sale_Price_log10)
ames_train <- training(ames_split)
ames_test  <- testing(ames_split)
```

Verifiquemos los cuartiles de cada uno de los conjuntos

```
quantile(ames_train$Sale_Price_log10)
```

```
##       0%      25%      50%      75%     100%
## 4.106837 5.112270 5.204120 5.329398 5.872156
```

```
quantile(ames_test$Sale_Price_log10)
```

```
##       0%      25%      50%      75%     100%
## 4.544068 5.112270 5.204798 5.329091 5.877947
```

Creemos el modelo

Especifiquemos el *engine* con [set_engine](https://---

title: 'Modelos Lineales en R: Parte 2' output: pdf_document: default html_document: default —

Datos de Casas en Ames, Iowa US

Estos datos contienen 2930 casas en la ciudad de Ames en Iowa en Estados Unidos. Los datos originales han sido modificados para facilitar uso y hacen parte del paquete modeldata. *modeldata* hace parte de tidyverse

Importemos las librerias necesarias

```
library(tidymodels)
```

Resolvamos los conflictos entre los distintos paquetes

```
tidymodels_prefer(quiet=FALSE)
```

```
## [conflicted] Removing existing preference
## [conflicted] Will prefer dplyr::filter over any other package
## [conflicted] Removing existing preference
## [conflicted] Will prefer dplyr::select over any other package
## [conflicted] Removing existing preference
## [conflicted] Will prefer dplyr::slice over any other package
## [conflicted] Removing existing preference
## [conflicted] Will prefer dplyr::rename over any other package
## [conflicted] Removing existing preference
## [conflicted] Will prefer dials::neighbors over any other package
## [conflicted] Removing existing preference
```

```
## [conflicted] Will prefer parsnip::fit over any other package
## [conflicted] Removing existing preference
## [conflicted] Will prefer parsnip::bart over any other package
## [conflicted] Removing existing preference
## [conflicted] Will prefer parsnip::pls over any other package
## [conflicted] Removing existing preference
## [conflicted] Will prefer purrr::map over any other package
## [conflicted] Removing existing preference
## [conflicted] Will prefer recipes::step over any other package
## [conflicted] Removing existing preference
## [conflicted] Will prefer themis::step_downsample over any other package
## [conflicted] Removing existing preference
## [conflicted] Will prefer themis::step_upsample over any other package
## [conflicted] Removing existing preference
## [conflicted] Will prefer tune::tune over any other package
## [conflicted] Removing existing preference
## [conflicted] Will prefer yardstick::precision over any other package
## [conflicted] Removing existing preference
## [conflicted] Will prefer yardstick::recall over any other package
## [conflicted] Removing existing preference
## [conflicted] Will prefer yardstick::spec over any other package
## -- Conflicts --------------------------------------------- tidymodels_prefer() --
```

Carguemos los datos**

```
data(ames)
head(ames)
```

```
## # A tibble: 6 x 74
##   MS_SubClass            MS_Zoning Lot_Frontage Lot_Area Street Alley Lot_Shape
##   <fct>                  <fct>            <dbl>    <int> <fct>  <fct> <fct>
## 1 One_Story_1946_and_New~ Resident~         141    31770 Pave   No_A~ Slightly~
## 2 One_Story_1946_and_New~ Resident~          80    11622 Pave   No_A~ Regular
## 3 One_Story_1946_and_New~ Resident~          81    14267 Pave   No_A~ Slightly~
## 4 One_Story_1946_and_New~ Resident~          93    11160 Pave   No_A~ Regular
## 5 Two_Story_1946_and_New~ Resident~          74    13830 Pave   No_A~ Slightly~
## 6 Two_Story_1946_and_New~ Resident~          78     9978 Pave   No_A~ Slightly~
## # ... with 67 more variables: Land_Contour <fct>, Utilities <fct>,
## #   Lot_Config <fct>, Land_Slope <fct>, Neighborhood <fct>, Condition_1 <fct>,
## #   Condition_2 <fct>, Bldg_Type <fct>, House_Style <fct>, Overall_Cond <fct>,
## #   Year_Built <int>, Year_Remod_Add <int>, Roof_Style <fct>, Roof_Matl <fct>,
## #   Exterior_1st <fct>, Exterior_2nd <fct>, Mas_Vnr_Type <fct>,
## #   Mas_Vnr_Area <dbl>, Exter_Cond <fct>, Foundation <fct>, Bsmt_Cond <fct>,
## #   Bsmt_Exposure <fct>, BsmtFin_Type_1 <fct>, BsmtFin_SF_1 <dbl>, ...
```

Revisemos las dimensiones del dataframe

```
dim(ames)
```

```
## [1] 2930   74
```

Revisemos la lista de variables

```
names(ames)
```

```
##  [1] "MS_SubClass"        "MS_Zoning"          "Lot_Frontage"
##  [4] "Lot_Area"           "Street"             "Alley"
```

```
##  [7] "Lot_Shape"          "Land_Contour"       "Utilities"
## [10] "Lot_Config"         "Land_Slope"         "Neighborhood"
## [13] "Condition_1"        "Condition_2"        "Bldg_Type"
## [16] "House_Style"        "Overall_Cond"       "Year_Built"
## [19] "Year_Remod_Add"     "Roof_Style"         "Roof_Matl"
## [22] "Exterior_1st"       "Exterior_2nd"       "Mas_Vnr_Type"
## [25] "Mas_Vnr_Area"       "Exter_Cond"         "Foundation"
## [28] "Bsmt_Cond"          "Bsmt_Exposure"      "BsmtFin_Type_1"
## [31] "BsmtFin_SF_1"       "BsmtFin_Type_2"     "BsmtFin_SF_2"
## [34] "Bsmt_Unf_SF"        "Total_Bsmt_SF"      "Heating"
## [37] "Heating_QC"         "Central_Air"        "Electrical"
## [40] "First_Flr_SF"       "Second_Flr_SF"      "Gr_Liv_Area"
## [43] "Bsmt_Full_Bath"     "Bsmt_Half_Bath"     "Full_Bath"
## [46] "Half_Bath"          "Bedroom_AbvGr"      "Kitchen_AbvGr"
## [49] "TotRms_AbvGrd"      "Functional"         "Fireplaces"
## [52] "Garage_Type"        "Garage_Finish"      "Garage_Cars"
## [55] "Garage_Area"        "Garage_Cond"        "Paved_Drive"
## [58] "Wood_Deck_SF"       "Open_Porch_SF"      "Enclosed_Porch"
## [61] "Three_season_porch" "Screen_Porch"       "Pool_Area"
## [64] "Pool_QC"            "Fence"              "Misc_Feature"
## [67] "Misc_Val"           "Mo_Sold"            "Year_Sold"
## [70] "Sale_Type"          "Sale_Condition"     "Sale_Price"
## [73] "Longitude"          "Latitude"
```

Procesemos los datos

```
ames <- ames %>%
  mutate(Sale_Price_log10 = log10(Sale_Price))
```

Particion Estratificada

```
set.seed(123)
ames_split <- initial_split(ames, prop = 0.80, strata = Sale_Price_log10)
ames_train <- training(ames_split)
ames_test  <- testing(ames_split)
```

Verifiquemos los cuartiles de cada uno de los conjuntos

```
quantile(ames_train$Sale_Price_log10)
```

```
##       0%      25%      50%      75%     100%
## 4.106837 5.112270 5.204120 5.329398 5.872156
```

```
quantile(ames_test$Sale_Price_log10)
```

```
##       0%      25%      50%      75%     100%
## 4.544068 5.112270 5.204798 5.329091 5.877947
```

Creemos el modelo

Especifiquemos el *engine* con set_engine

Primero miremos cuales *engines* hay par modelos lineales

```
show_engines("linear_reg")
```

```
## # A tibble: 7 x 2
##   engine mode
##   <chr>  <chr>
## 1 lm     regression
## 2 glm    regression
## 3 glmnet regression
## 4 stan   regression
## 5 spark  regression
## 6 keras  regression
## 7 brulee regression
```

Ajustemos el modelo/reference/set_engine.html)

linear_reg

```
lm_model <- linear_reg() %>%
  set_engine("lm")
```

Ajustemos el modelo

fit

```
lm_form_fit <- lm_model %>%
  fit(Sale_Price_log10 ~ Longitude + Latitude, data = ames_train)
```

```
lm_form_fit
```

```
## parsnip model object
##
##
## Call:
## stats::lm(formula = Sale_Price_log10 ~ Longitude + Latitude,
##     data = data)
##
## Coefficients:
## (Intercept)    Longitude     Latitude
##    -300.251       -2.013        2.782
```

```
lm_xy_fit <- lm_model %>%
  fit_xy(x = ames_train %>% select(Longitude, Latitude),
         y = ames_train %>% pull(Sale_Price_log10)
         )
```

```
lm_xy_fit
```

```
## parsnip model object
##
##
## Call:
## stats::lm(formula = ..y ~ ., data = data)
##
## Coefficients:
## (Intercept)    Longitude     Latitude
```

```
##    -300.251       -2.013        2.782
```

Ejercicio: Escoger dos variables diferentes

1. Entrenar el modelo
2. Comparar los coeficientes

```
lm_form_fit <- lm_model %>%
  fit(Sale_Price_log10 ~ Year_Built + Year_Sold, data = ames_train)
lm_form_fit
```

```
## parsnip model object
##
##
## Call:
## stats::lm(formula = Sale_Price_log10 ~ Year_Built + Year_Sold,
##     data = data)
##
## Coefficients:
## (Intercept)   Year_Built    Year_Sold
##    4.585758     0.003629    -0.003248
```

Acceder a la informacion del modelo que entrenamos extract_fit_engine

```
model_res <- lm_form_fit %>%
  extract_fit_engine() %>%
  summary()

model_res
```

```
##
## Call:
## stats::lm(formula = Sale_Price_log10 ~ Year_Built + Year_Sold,
##     data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.02941 -0.08195 -0.01221  0.07561  0.58443
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.586e+00  4.375e+00   1.048    0.295
## Year_Built   3.629e-03  9.659e-05  37.571   <2e-16 ***
## Year_Sold   -3.248e-03  2.176e-03  -1.493    0.136
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1393 on 2339 degrees of freedom
## Multiple R-squared:  0.377,  Adjusted R-squared:  0.3764
## F-statistic: 707.6 on 2 and 2339 DF,  p-value: < 2.2e-16
```

Comparemos ahora los resultados del modelo usando como predictores latitud y longitud

```
lm_form_fit <- lm_model %>%
  fit(Sale_Price_log10 ~ Longitude + Latitude, data = ames_train)
```

```r
model_res <- lm_form_fit %>%
  extract_fit_engine() %>%
  summary()

model_res
```

```
##
## Call:
## stats::lm(formula = Sale_Price_log10 ~ Longitude + Latitude,
##     data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.02781 -0.09482 -0.01501  0.09799  0.57143
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -300.2509    14.5815  -20.59   <2e-16 ***
## Longitude     -2.0134     0.1297  -15.53   <2e-16 ***
## Latitude       2.7817     0.1817   15.31   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1614 on 2339 degrees of freedom
## Multiple R-squared:  0.1639, Adjusted R-squared:  0.1632
## F-statistic: 229.3 on 2 and 2339 DF,  p-value: < 2.2e-16
```

predict

```r
ames_test_small <- ames_test %>%
  slice(1:5)

ames_test_small
```

```
## # A tibble: 5 x 75
##   MS_SubClass            MS_Zoning Lot_Frontage Lot_Area Street Alley Lot_Shape
##   <fct>                  <fct>            <dbl>    <int> <fct>  <fct> <fct>
## 1 One_Story_1946_and_New~ Resident~          80    11622 Pave   No_A~ Regular
## 2 One_Story_1946_and_New~ Resident~          81    14267 Pave   No_A~ Slightly~
## 3 Two_Story_1946_and_New~ Resident~          74    13830 Pave   No_A~ Slightly~
## 4 One_Story_1946_and_New~ Resident~          70    10500 Pave   No_A~ Regular
## 5 One_Story_1946_and_New~ Resident~          83    10159 Pave   No_A~ Slightly~
## # ... with 68 more variables: Land_Contour <fct>, Utilities <fct>,
## #   Lot_Config <fct>, Land_Slope <fct>, Neighborhood <fct>, Condition_1 <fct>,
## #   Condition_2 <fct>, Bldg_Type <fct>, House_Style <fct>, Overall_Cond <fct>,
## #   Year_Built <int>, Year_Remod_Add <int>, Roof_Style <fct>, Roof_Matl <fct>,
## #   Exterior_1st <fct>, Exterior_2nd <fct>, Mas_Vnr_Type <fct>,
## #   Mas_Vnr_Area <dbl>, Exter_Cond <fct>, Foundation <fct>, Bsmt_Cond <fct>,
## #   Bsmt_Exposure <fct>, BsmtFin_Type_1 <fct>, BsmtFin_SF_1 <dbl>, ...
```

```r
sales_price_small <- predict(lm_form_fit, new_data = ames_test_small)

sales_price_small
```

```
## # A tibble: 5 x 1
```

```
##   .pred
##   <dbl>
## 1  5.22
## 2  5.22
## 3  5.28
## 4  5.24
## 5  5.31
```

Como podemos comparar las predicciones con los datos reales

```
ames_test_small %>%
  select(Sale_Price_log10) %>%
  bind_cols(predict(lm_form_fit, ames_test_small))
```

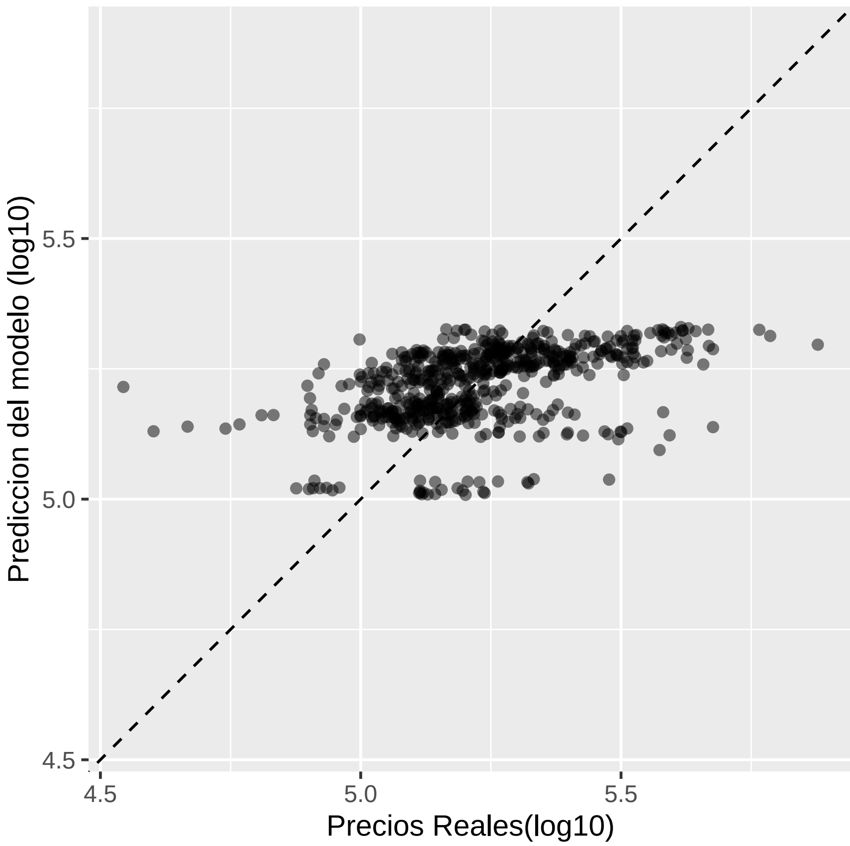
```
## # A tibble: 5 x 2
##   Sale_Price_log10 .pred
##              <dbl> <dbl>
## 1             5.02  5.22
## 2             5.24  5.22
## 3             5.28  5.28
## 4             5.06  5.24
## 5             5.60  5.31
```

Intro a ggplot

```
ames_test_pred <- ames_test %>%
  select(Sale_Price_log10) %>%
  bind_cols(predict(lm_form_fit, ames_test))
head(ames_test_pred)
```

```
## # A tibble: 6 x 2
##   Sale_Price_log10 .pred
##              <dbl> <dbl>
## 1             5.02  5.22
## 2             5.24  5.22
## 3             5.28  5.28
## 4             5.06  5.24
## 5             5.60  5.31
## 6             5.33  5.31
```

```
ggplot(ames_test_pred, aes(x= Sale_Price_log10, y=.pred)) +
  geom_abline(lty=2) +
  geom_point(alpha = 0.5) +
  labs(y = "Prediccion del modelo (log10)", x = "Precios Reales(log10)") +
  coord_obs_pred()
```



Estimacion del rendimiento del modelo

```
ames_metrics <-  metric_set(rmse, mae)

ames_metrics(ames_test_pred,
             truth = Sale_Price_log10,
             estimate = .pred)
```

```
## # A tibble: 2 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard       0.160
## 2 mae     standard       0.123
```