Regression and Prediction

Howard Nguyen

2022-12-27

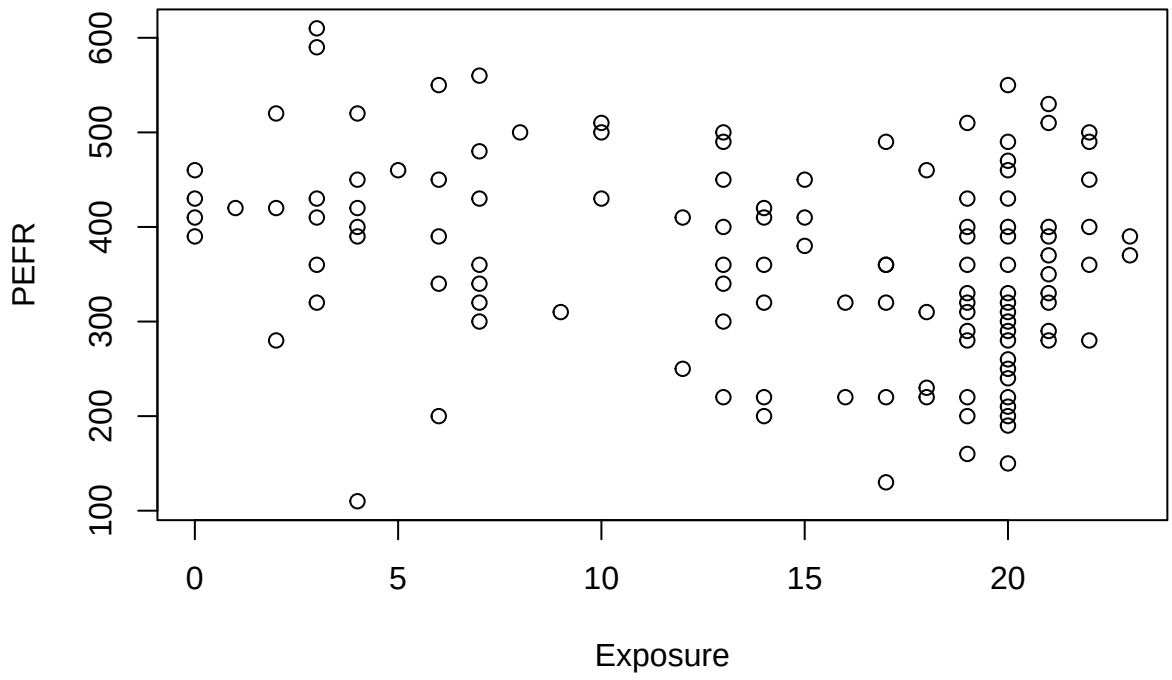
```
# Load R packages.
library(MASS)
library(dplyr)
library(tidyr)
library(ggplot2)
library(lubridate)
library(splines)
library(mgcv)
```

```
# Define paths to data sets.
lung <- read.csv('LungDisease.csv')
house <- read.csv(('house_sales.csv'), sep='\t')
```

Simple Linear Regression

The Regression Equation

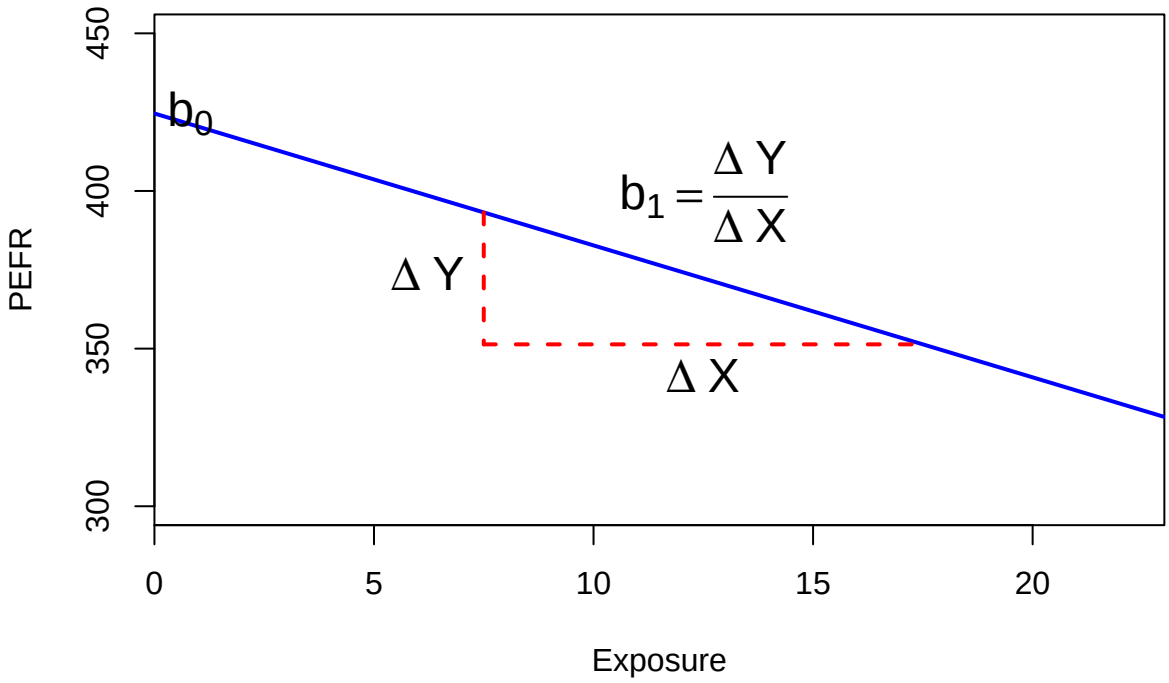
```
plot(lung$Exposure, lung$PEFR, xlab="Exposure", ylab="PEFR")
```




```
model <- lm(PEFR ~ Exposure, data=lung)
model
```

```
##
## Call:
## lm(formula = PEFR ~ Exposure, data = lung)
##
## Coefficients:
## (Intercept)     Exposure
##     424.583       -4.185
```

```
plot(lung$Exposure, lung$PEFR, xlab="Exposure", ylab="PEFR", ylim=c(300,450), type="n", xaxs="i")
abline(a=model$coefficients[1], b=model$coefficients[2], col="blue", lwd=2)
text(x=.3, y=model$coefficients[1], labels=expression("b"[0]),  adj=0, cex=1.5)
x <- c(7.5, 17.5)
y <- predict(model, newdata=data.frame(Exposure=x))
segments(x[1], y[2], x[2], y[2] , col="red", lwd=2, lty=2)
segments(x[1], y[1], x[1], y[2] , col="red", lwd=2, lty=2)
text(x[1], mean(y), labels=expression(Delta~Y), pos=2, cex=1.5)
text(mean(x), y[2], labels=expression(Delta~X), pos=1, cex=1.5)
text(mean(x), 400, labels=expression(b[1] == frac(Delta ~ Y, Delta ~ X)), cex=1.5)
```



Slope and intercept for the regression fit to the lung data

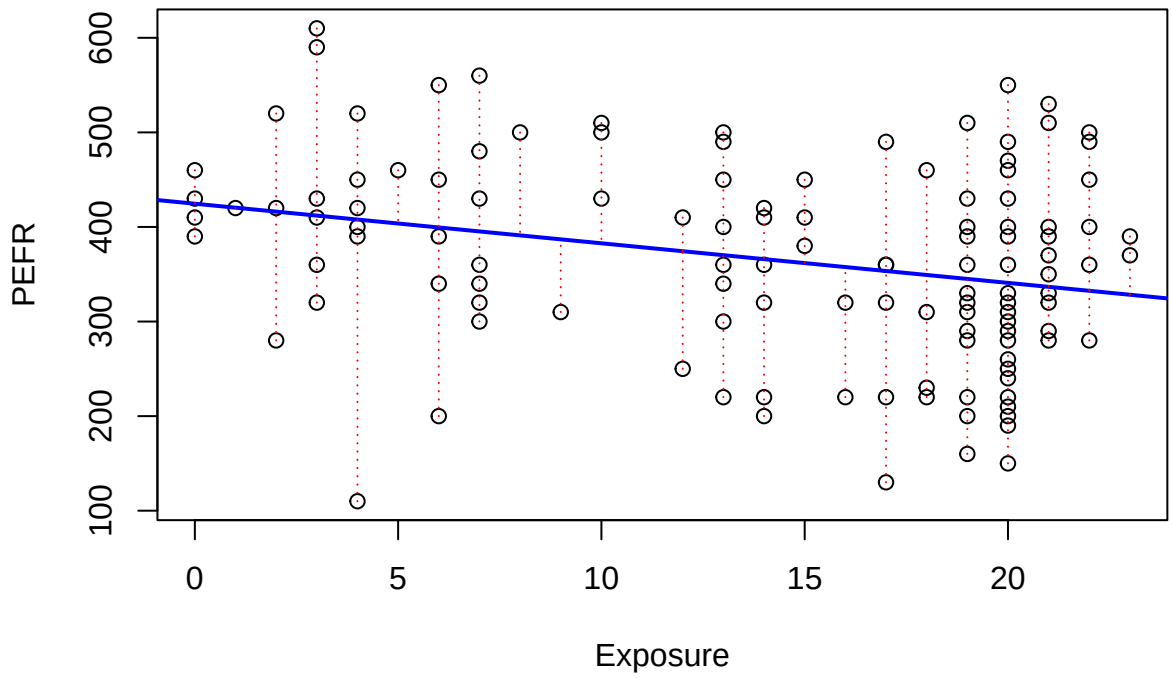
Fitted Values and Residuals

```
fitted <- predict(model)
resid <- residuals(model)

lung1 <- lung %>%
  mutate(Fitted=fitted,
         positive = PEFR>Fitted) %>%
  group_by(Exposure, positive) %>%
  summarize(PEFR_max = max(PEFR),
            PEFR_min = min(PEFR),
            Fitted = first(Fitted)) %>%
  ungroup() %>%
  mutate(PEFR = ifelse(positive, PEFR_max, PEFR_min)) %>%
  arrange(Exposure)
```

```
## `summarise()` has grouped output by 'Exposure'. You can override using the
## `.groups` argument.
```

```
plot(lung$Exposure, lung$PEFR, xlab="Exposure", ylab="PEFR")
abline(a=model$coefficients[1], b=model$coefficients[2], col="blue", lwd=2)
segments(lung1$Exposure, lung1$PEFR, lung1$Exposure, lung1$Fitted, col="red", lty=3)
```



Residuals from a regression line (to accommodate all the data, the y-axis scale differs from previous chart, hence the apparently different slope)

Multiple linear regression

Use the multiple linear regression in estimating the value of houses

```
print(head(house[, c('AdjSalePrice', 'SqFtTotLiving', 'SqFtLot', 'Bathrooms',
               'Bedrooms', 'BldgGrade')]))
```

```
##   AdjSalePrice SqFtTotLiving SqFtLot Bathrooms Bedrooms BldgGrade
## 1       300805          2400    9373      3.00        6         7
## 2      1076162          3764   20156      3.75        4        10
## 3       761805          2060   26036      1.75        4         8
## 4       442065          3200    8618      3.75        5         7
## 5       297065          1720    8620      1.75        4         7
## 6       411781           930    1012      1.50        2         8
```

```
house_lm <- lm(AdjSalePrice ~ SqFtTotLiving + SqFtLot + Bathrooms +
                 Bedrooms + BldgGrade,
               data=house, na.action=na.omit)
house_lm
```

```
##
## Call:
```

```
## lm(formula = AdjSalePrice ~ SqFtTotLiving + SqFtLot + Bathrooms +
##     Bedrooms + BldgGrade, data = house, na.action = na.omit)
##
## Coefficients:
##   (Intercept)  SqFtTotLiving        SqFtLot      Bathrooms       Bedrooms
##    -5.219e+05      2.288e+02     -6.047e-02     -1.944e+04     -4.777e+04
##     BldgGrade
##     1.061e+05
```

Assessing the Model

The most important performance metric from a data science perspective is root mean squared error, or RMSE. This measures the overall accuracy of the model and is a basis for comparing it to other models (including models fit using machine learning techniques). Similar to RMSE is the residual standard error, or RSE. The only difference is that the denominator is the degrees of freedom, as opposed to number of records. In practice, for linear regression, the difference between RMSE and RSE is very small, particularly for big data applications. The summary function in R computes RSE as well as other metrics for a regression model:

```
summary(house_lm)
```

```
##
## Call:
## lm(formula = AdjSalePrice ~ SqFtTotLiving + SqFtLot + Bathrooms +
##     Bedrooms + BldgGrade, data = house, na.action = na.omit)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1199479  -118908   -20977    87435  9473035
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -5.219e+05  1.565e+04 -33.342  < 2e-16 ***
## SqFtTotLiving  2.288e+02  3.899e+00  58.694  < 2e-16 ***
## SqFtLot       -6.047e-02  6.118e-02  -0.988    0.323
## Bathrooms     -1.944e+04  3.625e+03  -5.363 8.27e-08 ***
## Bedrooms      -4.777e+04  2.490e+03 -19.187  < 2e-16 ***
## BldgGrade      1.061e+05  2.396e+03  44.277  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 261300 on 22681 degrees of freedom
## Multiple R-squared:  0.5406, Adjusted R-squared:  0.5405
## F-statistic:  5338 on 5 and 22681 DF,  p-value: < 2.2e-16
```

Model Selection and Stepwise Regression

```
house_full <- lm(AdjSalePrice ~ SqFtTotLiving + SqFtLot + Bathrooms +
                   Bedrooms + BldgGrade + PropertyType + NbrLivingUnits +
                   SqFtFinBasement + YrBuilt + YrRenovated + NewConstruction,
                 data=house, na.action=na.omit)
```

```
## Code snippet 4.8
step_lm <- stepAIC(house_full, direction="both")
```

```
## Start:  AIC=563145.4
## AdjSalePrice ~ SqFtTotLiving + SqFtLot + Bathrooms + Bedrooms +
##     BldgGrade + PropertyType + NbrLivingUnits + SqFtFinBasement +
##     YrBuilt + YrRenovated + NewConstruction
##
##                    Df  Sum of Sq        RSS    AIC
## - NbrLivingUnits    1 6.4007e+09 1.3662e+15 563144
## - NewConstruction   1 1.0592e+10 1.3662e+15 563144
## - YrRenovated       1 2.5069e+10 1.3662e+15 563144
## - SqFtLot           1 1.0657e+11 1.3663e+15 563145
## <none>                           1.3662e+15 563145
## - SqFtFinBasement   1 1.4030e+11 1.3663e+15 563146
## - PropertyType      2 4.4207e+12 1.3706e+15 563215
## - Bathrooms         1 7.6325e+12 1.3738e+15 563270
## - Bedrooms          1 2.8212e+13 1.3944e+15 563607
## - YrBuilt           1 1.2906e+14 1.4952e+15 565191
## - SqFtTotLiving     1 1.3264e+14 1.4988e+15 565246
## - BldgGrade         1 1.9050e+14 1.5567e+15 566105
##
## Step:  AIC=563143.6
## AdjSalePrice ~ SqFtTotLiving + SqFtLot + Bathrooms + Bedrooms +
##     BldgGrade + PropertyType + SqFtFinBasement + YrBuilt + YrRenovated +
##     NewConstruction
##
##                    Df  Sum of Sq        RSS    AIC
## - NewConstruction   1 1.0801e+10 1.3662e+15 563142
## - YrRenovated       1 2.5628e+10 1.3662e+15 563142
## - SqFtLot           1 1.0731e+11 1.3663e+15 563143
## <none>                           1.3662e+15 563144
## - SqFtFinBasement   1 1.3828e+11 1.3663e+15 563144
## + NbrLivingUnits    1 6.4007e+09 1.3662e+15 563145
## - PropertyType      2 4.4301e+12 1.3706e+15 563213
## - Bathrooms         1 7.7500e+12 1.3739e+15 563270
## - Bedrooms          1 2.8273e+13 1.3944e+15 563606
## - YrBuilt           1 1.3013e+14 1.4963e+15 565206
## - SqFtTotLiving     1 1.3288e+14 1.4990e+15 565247
## - BldgGrade         1 1.9177e+14 1.5579e+15 566122
##
## Step:  AIC=563141.7
## AdjSalePrice ~ SqFtTotLiving + SqFtLot + Bathrooms + Bedrooms +
##     BldgGrade + PropertyType + SqFtFinBasement + YrBuilt + YrRenovated
##
##                    Df  Sum of Sq        RSS    AIC
## - YrRenovated       1 2.5893e+10 1.3662e+15 563140
## - SqFtLot           1 1.1494e+11 1.3663e+15 563142
## <none>                           1.3662e+15 563142
## - SqFtFinBasement   1 1.4534e+11 1.3663e+15 563142
## + NewConstruction   1 1.0801e+10 1.3662e+15 563144
## + NbrLivingUnits    1 6.6093e+09 1.3662e+15 563144
## - PropertyType      2 4.5301e+12 1.3707e+15 563213
```

```
## - Bathrooms        1 7.7487e+12 1.3739e+15 563268
## - Bedrooms         1 2.8269e+13 1.3945e+15 563604
## - SqFtTotLiving    1 1.3390e+14 1.5001e+15 565261
## - YrBuilt          1 1.3760e+14 1.5038e+15 565317
## - BldgGrade        1 1.9244e+14 1.5586e+15 566129
##
## Step:  AIC=563140.2
## AdjSalePrice ~ SqFtTotLiving + SqFtLot + Bathrooms + Bedrooms +
##     BldgGrade + PropertyType + SqFtFinBasement + YrBuilt
##
##                   Df  Sum of Sq        RSS    AIC
## - SqFtLot          1 1.1425e+11 1.3663e+15 563140
## <none>                          1.3662e+15 563140
## - SqFtFinBasement  1 1.4999e+11 1.3664e+15 563141
## + YrRenovated      1 2.5893e+10 1.3662e+15 563142
## + NewConstruction  1 1.1065e+10 1.3662e+15 563142
## + NbrLivingUnits   1 7.1825e+09 1.3662e+15 563142
## - PropertyType     2 4.5076e+12 1.3707e+15 563211
## - Bathrooms        1 7.7790e+12 1.3740e+15 563267
## - Bedrooms         1 2.8251e+13 1.3945e+15 563603
## - SqFtTotLiving    1 1.3388e+14 1.5001e+15 565259
## - YrBuilt          1 1.5091e+14 1.5171e+15 565515
## - BldgGrade        1 1.9244e+14 1.5587e+15 566128
##
## Step:  AIC=563140.1
## AdjSalePrice ~ SqFtTotLiving + Bathrooms + Bedrooms + BldgGrade +
##     PropertyType + SqFtFinBasement + YrBuilt
##
##                   Df  Sum of Sq        RSS    AIC
## <none>                          1.3663e+15 563140
## + SqFtLot          1 1.1425e+11 1.3662e+15 563140
## - SqFtFinBasement  1 1.4116e+11 1.3665e+15 563140
## + YrRenovated      1 2.5199e+10 1.3663e+15 563142
## + NewConstruction  1 1.8750e+10 1.3663e+15 563142
## + NbrLivingUnits   1 8.0521e+09 1.3663e+15 563142
## - PropertyType     2 4.4415e+12 1.3708e+15 563210
## - Bathrooms        1 7.7109e+12 1.3740e+15 563266
## - Bedrooms         1 2.8553e+13 1.3949e+15 563607
## - SqFtTotLiving    1 1.3748e+14 1.5038e+15 565313
## - YrBuilt          1 1.5080e+14 1.5171e+15 565513
## - BldgGrade        1 1.9234e+14 1.5587e+15 566126
```

```
step_lm
```

```
##
## Call:
## lm(formula = AdjSalePrice ~ SqFtTotLiving + Bathrooms + Bedrooms +
##     BldgGrade + PropertyType + SqFtFinBasement + YrBuilt, data = house,
##     na.action = na.omit)
##
## Coefficients:
##               (Intercept)              SqFtTotLiving
##                 6.179e+06                  1.993e+02
##                 Bathrooms                  Bedrooms
```

```
##                 4.240e+04                 -5.195e+04
##                 BldgGrade  PropertyTypeSingle Family
##                 1.372e+05                  2.291e+04
##     PropertyTypeTownhouse            SqFtFinBasement
##                 8.448e+04                  7.047e+00
##                   YrBuilt
##                -3.565e+03
```

```
lm(AdjSalePrice ~  Bedrooms, data=house)
```

```
##
## Call:
## lm(formula = AdjSalePrice ~ Bedrooms, data = house)
##
## Coefficients:
## (Intercept)     Bedrooms
##      117354       132991
```

Weight Regression

Weighted regression is used by statisticians for a variety of purposes; in particular, it is important for analysis of complex surveys. Data scientists may find weighted regression useful in two cases: • Inverse-variance weighting when different observations have been measured with different precision; the higher variance ones receiving lower weights. • Analysis of data where rows represent multiple cases; the weight variable encodes how many original observations each row represents. For example, with the housing data, older sales are less reliable than more recent sales. Using the DocumentDate to determine the year of the sale, we can compute a Weight as the number of years since 2005 (the beginning of the data):

```
### Weighted regression
house$Year = year(house$DocumentDate)
house$Weight = house$Year - 2005
```

We can compute a weighted regression with the lm function using the weight argument:

```
house_wt <- lm(AdjSalePrice ~ SqFtTotLiving + SqFtLot + Bathrooms +
                 Bedrooms + BldgGrade,
               data=house, weight=Weight, na.action=na.omit)
round(cbind(house_lm=house_lm$coefficients,
            house_wt=house_wt$coefficients), digits=3)
```

```
##                  house_lm    house_wt
## (Intercept)   -521871.368 -584189.329
## SqFtTotLiving     228.831     245.024
## SqFtLot            -0.060      -0.292
## Bathrooms       -19442.840  -26085.970
## Bedrooms        -47769.955  -53608.876
## BldgGrade       106106.963  115242.435
```

The coefficients in the weighted regression are slightly different from the original regression.

Key notes: • Multiple linear regression models the relationship between a response variable Y and multiple predictor variables X1, . . . , Xp. • The most important metrics to evaluate a model are root mean squared

error (RMSE) and R-squared (R2). • The standard error of the coefficients can be used to measure the reliability of a variable's contribution to a model. • Stepwise regression is a way to automatically determine which variables should be included in the model. • Weighted regression is used to give certain records more or less weight in fitting the equation.