Regression for a Categorical Outcome

Machine Learning - Section 3.1.3

Marc Omar Haddad

Published: 22 February, 2020

Updated: 23 February, 2020

The regression approach discussed in previous lessons can also be applied to Categorical Data.

```
data("heights")
y = heights$height
set.seed(2, sample.kind = "Rounding")

test_index = createDataPartition(y, times = 1, p = 0.5, list = FALSE)
train_set = heights %>% slice(-test_index)
test_set = heights %>% slice(test_index)
```

We will define the outcome as $Y = 1$ for `female`, and $Y = 0$ for `male`, and with feature $X =$`height`. With this definition we are interested in the **Conditional Probability of being `female` when given `height`**; represented mathematically as:

$$\Pr(Y = 1 \mid X = x)$$

So, we ask ourselves: *What is the **conditional probability of being female** if you are **66 inches** tall?*

We can calculate the probability by simply rounding `height` entries that are near 66 inches to 66, and then calculating the proportion of females.

```
train_set %>%
  filter(height == round(66)) %>%
  summarize(`Conditional Prob. of being Female` = mean(sex == "Female"))
```

Conditional Prob. of being Female
0.2142857

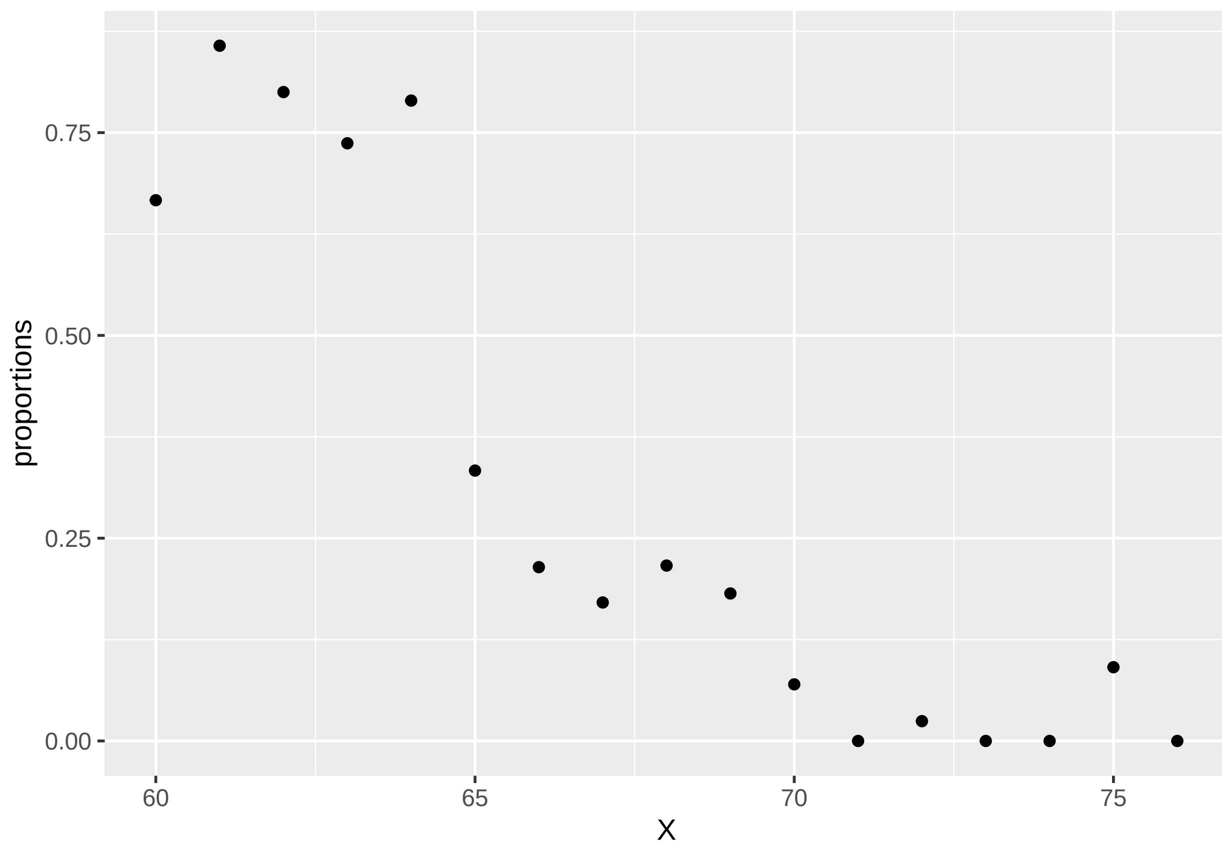
The Conditional Probability of being `female` given a `height` of 66 inches is 21.4%.

Now we will repeat the same exercise for multiple values of X.

```
X = seq(60, 76, 1)

proportions = map_dbl(X, function(x) {
  train_set %>%
    filter(height == round(x)) %>%
    summarize(proportion = mean(sex == "Female")) %>%
    .$proportion
})

qplot(X, proportions)
```



Since the results of the above plot appear to be linear, we can try regression.

Reminder: When using regression we assume that **Conditional Probability** can be *expressed* as a **linear function**:

$$p(x) = \Pr(Y = 1 \mid X = x) = \beta_0 + \beta_1 x$$

We plug in our values into the `lm()` function to yield an estimate of β_0 and β_1.

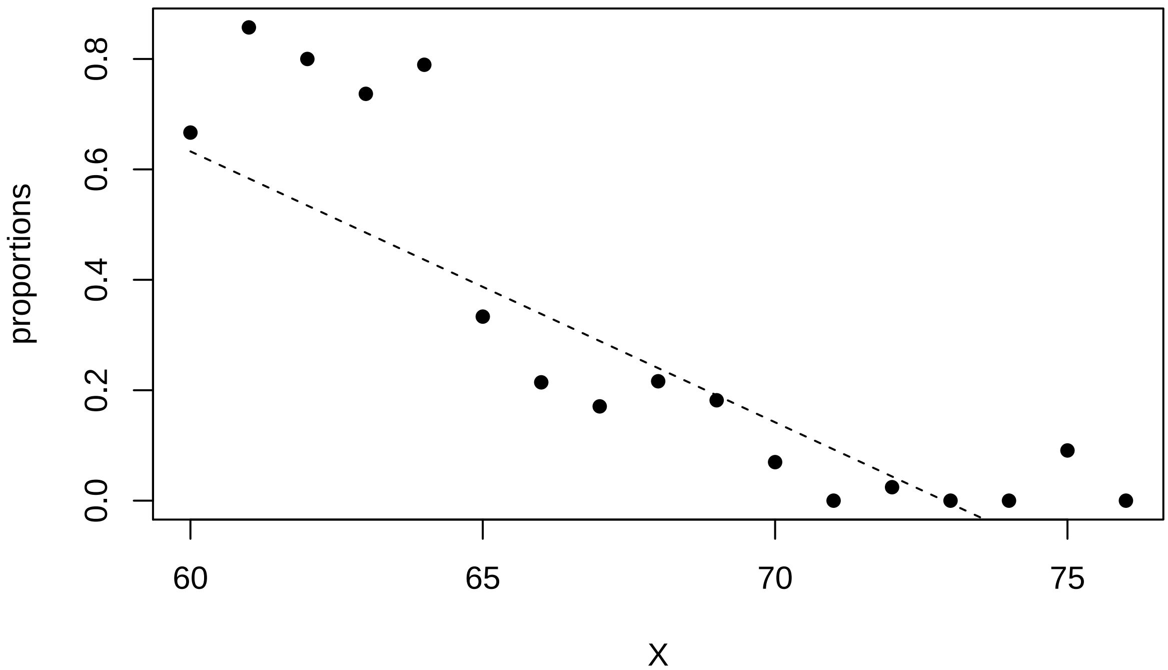
```
lm_fit = mutate(train_set, y = as.numeric(sex == "Female")) %>%
  lm(y ~ height, data = .)
lm_fit$coefficients
```

```
## (Intercept)      height
##  3.57681946 -0.04907022
```

Now that we have our estimates, we can extract an actual prediction. The estimate of our Conditional Probability is:

$$\hat{p}(x) = \hat{\beta}_0 + \hat{\beta}_1 x$$

We can visualize our **Conditional Expectations / Conditional Probabilities** by plugging the values of `lm_fit$coefficients` into the above function and plotting the resulting line.



Using this formula, we form a prediction by defining a **Decision Rule**. In this case our Decision Rule is:

Predict `female` if $\hat{p}(x) > 0.5$.

We can then use the `confusionMatrix()` function to assess our model.

```
p_hat = predict(lm_fit, test_set)
y_hat = ifelse(p_hat > 0.5, "Female", "Male") %>% factor()
confusionMatrix(y_hat, test_set$sex)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Female Male
##     Female     20   15
##     Male       98  393
##
##                Accuracy : 0.7852
##                  95% CI : (0.7476, 0.8195)
##     No Information Rate : 0.7757
```

```
##     P-Value [Acc > NIR] : 0.3218
##
##                   Kappa : 0.177
##
##  Mcnemar's Test P-Value : 1.22e-14
##
##             Sensitivity : 0.16949
##             Specificity : 0.96324
##          Pos Pred Value : 0.57143
##          Neg Pred Value : 0.80041
##              Prevalence : 0.22433
##          Detection Rate : 0.03802
##    Detection Prevalence : 0.06654
##       Balanced Accuracy : 0.56636
##
##        'Positive' Class : Female
##
```