Lab 2: Multiple Linear Regression

Sarah Wright

January 20, 2022

Introduction

Linear regression is a simple method for predicting a quantitative response variable Y on the basis of multiple predictors. That is, we assume that

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n.$$

Based on this model, given values $x_1, \ldots, x_n$ for the predictors, we predict the response variable to be

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n.$$

For the ith observation, the difference between the true response y_i and our predicted response $\hat{y}_i$ is $e_i = y_i - \hat{y}_i$, which we call the ith *residual*. The coefficients of a linear model are usually chosen using the *least squares criterion*. That is, if we have m observations on which to base our model, we choose the values of $\beta_0, \beta_1, \ldots, \beta_n$ to minimize the *Mean Squared Error* (MSE), which is defined as

$$(e_1^2 + e_2^2 + \ldots + e_m^2)/m.$$

There is a lot of statistical theory behind multiple linear regression which is explored in depth in other courses. The basis of this theory is the assumption that the true relationship between the predictors and the response variable is given by

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n + \epsilon$$

where ϵ is a mean-zero random error term. Based on these assumptions, there are a number of metrics for assessing which variables are important in our model, which we can learn from the `summary()` function in R. For the purposes of this course, we should know that the coefficients tell us how a one unit change in one of the predictors affects the response. We should also know that the p-values tell us how confident we can be that one of the coefficients is different from zero, where p-values close to 0 imply high confidence and p-values close to 1 imply low confidence.

Multiple Linear Regression Lab

In this lab, you will practice multiple linear regression by working with a simple weather data set. The data associated with this lab are in the file `MonthlyWeatherData.csv`, which contains 8+ years of monthly high temperature data (in degrees Fahrenheit) in Wooster, Ohio. The `Normal` column gives the normal average high for that month (based on the last decade). The `First7D` column contains the observed average high for only the first seven days of the month, and the `Observed` column contains the observed average high for the entire month.

1) Create a data frame in R called `WeatherData` consisting of the data from `MonthlyWeatherData.csv`. (Import the dataset and make any necessary changes to display it accurately. The resulting dataset should have 92 rows and 5 columns.)

When doing empirical modeling, we should set aside a portion of our data to be used at the end in assessing the performance of our final model(s). Thus, we usually have a *training* set of data we build our model on and a *test* or *hold out* set that we use to assess performance. With time-dependent data, we set aside the most-recent data (but would choose randomly if time was not a factor). (If you do not completely understand the idea of training and test sets, please reach out! This concept will serve as a basis for many (if not most) methods you'll be learning about this semester!)

2) For this lab, we will use only data up through 2013 as our training set. Create a data frame in R called `WeatherTrain` consisting of this data (the first 60 observations). Briefly explain why the `Normal` column is periodic, but the other columns are not. ???????????????????????

```r
WeatherTrain <- WeatherData %>% filter(Year <= 2013)
summary(WeatherTrain)
```

```
##        X            Month               Normal         First7D
##  Min.   : 1.00   Length:60          Min.   :32.60   Min.   :23.20
##  1st Qu.:15.75   Class :character   1st Qu.:47.52   1st Qu.:43.62
##  Median :30.50   Mode  :character   Median :61.80   Median :60.00
##  Mean   :30.50                      Mean   :60.58   Mean   :60.08
##  3rd Qu.:45.25                      3rd Qu.:76.28   3rd Qu.:77.20
##  Max.   :60.00                      Max.   :83.30   Max.   :92.30
##     Observed          Year
##  Min.   :27.00   Min.   :2009
##  1st Qu.:43.60   1st Qu.:2010
##  Median :61.60   Median :2011
##  Mean   :60.73   Mean   :2011
##  3rd Qu.:78.10   3rd Qu.:2012
##  Max.   :88.20   Max.   :2013
```

```r
WeatherTrain
```

```
##     X Month Normal First7D Observed Year
## 1   1   Jan   32.6    35.1     27.0 2009
## 2   2   Feb   35.7    32.9     39.1 2009
## 3   3   Mar   49.9    44.3     53.2 2009
## 4   4   Apr   61.5    54.9     61.2 2009
## 5   5   May   73.5    66.6     72.1 2009
## 6   6  June   79.8    73.0     78.6 2009
## 7   7  July   83.3    74.7     78.1 2009
## 8   8   Aug   82.0    78.6     80.1 2009
## 9   9   Sep   75.1    77.2     73.3 2009
## 10 10   Oct   62.1    61.3     58.9 2009
## 11 11   Nov   51.0    52.4     54.5 2009
## 12 12   Dec   40.4    40.6     36.7 2009
## 13 13   Jan   32.6    23.2     30.3 2010
## 14 14   Feb   35.7    32.2     31.8 2010
## 15 15   Mar   49.9    38.4     51.8 2010
## 16 16   Apr   61.5    78.5     66.7 2010
```

```
## 17 17   May   73.5    74.1      73.6 2010
## 18 18  June   79.8    77.0      80.2 2010
## 19 19  July   83.3    85.1      84.9 2010
## 20 20   Aug   82.0    83.6      84.2 2010
## 21 21   Sep   75.1    81.3      75.3 2010
## 22 22   Oct   62.1    59.5      63.8 2010
## 23 23   Nov   51.0    47.6      53.0 2010
## 24 24   Dec   40.4    29.7      30.2 2010
## 25 25   Jan   32.6    34.9      29.2 2011
## 26 26   Feb   35.7    33.0      37.6 2011
## 27 27   Mar   49.9    40.8      46.1 2011
## 28 28   Apr   61.5    52.4      60.7 2011
## 29 29   May   73.5    59.8      71.6 2011
## 30 30  June   79.8    81.5      79.3 2011
## 31 31  July   83.3    86.6      87.3 2011
## 32 32   Aug   82.0    86.4      81.5 2011
## 33 33   Sep   75.1    79.1      73.0 2011
## 34 34   Oct   62.1    65.4      61.6 2011
## 35 35   Nov   51.0    60.2      56.7 2011
## 36 36   Dec   40.4    45.9      43.6 2011
## 37 37   Jan   32.6    40.6      39.9 2012
## 38 38   Feb   35.7    44.1      42.4 2012
## 39 39   Mar   49.9    50.8      62.6 2012
## 40 40   Apr   61.5    59.0      60.2 2012
## 41 41   May   73.5    80.2      78.1 2012
## 42 42  June   79.8    71.4      81.6 2012
## 43 43  July   83.3    92.3      88.2 2012
## 44 44   Aug   82.0    88.4      83.5 2012
## 45 45   Sep   75.1    84.7      73.4 2012
## 46 46   Oct   62.1    65.4      61.5 2012
## 47 47   Nov   51.0    42.2      48.6 2012
## 48 48   Dec   40.4    52.6      43.6 2012
## 49 49   Jan   32.6    32.4      37.7 2013
## 50 50   Feb   35.7    27.7      34.7 2013
## 51 51   Mar   49.9    35.0      41.9 2013
## 52 52   Apr   61.5    51.4      61.6 2013
## 53 53   May   73.5    74.5      73.9 2013
## 54 54  June   79.8    71.6      78.5 2013
## 55 55  July   83.3    81.1      81.5 2013
## 56 56   Aug   82.0    76.8      80.2 2013
## 57 57   Sep   75.1    77.2      74.4 2013
## 58 58   Oct   62.1    76.2      63.8 2013
## 59 59   Nov   51.0    56.1      46.4 2013
## 60 60   Dec   40.4    45.3      38.8 2013
```

```r
unique(WeatherData$Normal)
```

```
##  [1] 32.6 35.7 49.9 61.5 73.5 79.8 83.3 82.0 75.1 62.1 51.0 40.4
```

```r
#unique(WeatherData$First7D)#compare to the values of Normal
#unique(WeatherData$Observed)#compare to the values of Normal
```

Normal is periodic because the varible can only be one of speific valuse:

```
    32.6 35.7 49.9 61.5 73.5 79.8 83.3 82.0 75.1 62.1 51.0 40.4
  The other varibales can be anything.
```

In predictive modeling, we often begin with a simple baseline model, to which we compare other models. Any more complicated model must outperform the baseline model to be considered useful. In this case, there are two obvious baseline models we might use. The first would be to predict the final average high for the entire month from just the `Normal` value. The other would be to predict the final average high for the entire month from just the first seven day average (`First7D`).

3) Which of the two models suggested do you think would be a better baseline model? Why?

```
I think the model that uses the Normal value would be better becasue it has more information. Usir
```

4) Compute the mean squared error (MSE) associated with the two baseline models (just use the data in `WeatherTrain`). Save these as `mse_normal` and `mse_first`. Which seems like the better baseline model?

```
mse_normal <- mean(summary(lm(Observed~Normal, data = WeatherTrain))$residuals^2)
mse_first <- mean(summary(lm(Observed~First7D, data = WeatherTrain))$residuals^2)
mse_normal
```

```
## [1] 14.16408
```

```
mse_first
```

```
## [1] 33.71628
```

```
Mse_normal seems like the better base line model
```

The above result does not mean the information on the first seven days is predictively useless. Rather, we need to pair that short-term data together with our prior expectations (in this case, the normal temperatures) to get a better prediction of each month's final average temperature.

5) Build a two-input linear model for the final average high temperature, by using the syntax `lm(Y ~ X1 + X2, data)`. Save your model as `lmfit1`. What are the coefficients associated with each factor? Note that the other term is the intercept. (Remember, you can use `summary()` to call the coefficients of your model.) ???????????????

```
lmfit1 <-summary(lm(Observed ~ First7D + Normal, data = WeatherData))
lmfit1
```

```
##
## Call:
## lm(formula = Observed ~ First7D + Normal, data = WeatherData)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.5200 -1.7977 -0.1185  1.7792 12.1723
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)  0.37933    1.24675   0.304    0.762
## First7D      0.28952    0.04887   5.925   5.8e-08 ***
## Normal       0.70823    0.05253  13.482   < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.366 on 89 degrees of freedom
## Multiple R-squared:  0.9668, Adjusted R-squared:  0.9661
## F-statistic:  1298 on 2 and 89 DF,  p-value: < 2.2e-16
```

6) One coefficient is substantially larger than the other. Given that each column has numbers that are approximately the same in magnitude, what does this tell us about the relative importance of the two factors? Does this agree with your analysis of the potential baseline models? (HINT: Compare with your answers from Question 4.)

   Normal has a greater than the coeifficent of First7D and a smaller p-value,it tells us that Normal is a better predictor of final tempeture highs compared to the first 7 days of the month This agress with the analysis of potential baseline models.

The phenomenon observed here is common in nearly all kinds of empirical modeling situations. As the sample size increases, the averages generally move toward expected levels. This phenomenon is called *regression to the mean*.

7) If you apply the `names()` function to the linear model you built, you will see that R stores a lot of information about the model. Access the `residuals` of the model and use them to find the MSE of your model (on the training data). Save this value as `mse1`. How much better is this model than the "normal" baseline model (by % reduction in MSE)?

```r
names(lmfit1)
```

```
##  [1] "call"          "terms"         "residuals"     "coefficients"
##  [5] "aliased"       "sigma"         "df"            "r.squared"
##  [9] "adj.r.squared" "fstatistic"    "cov.unscaled"
```

```r
mse1 <- mean(lmfit1$residuals^2)
mse_normal- mse1
```

```
## [1] 3.200441
```

```
This model is much better than the the normal baseline model by 3.200441% reduction in MSE.
```

We can also build models using *categorical* predictors instead of just numerical predictors.

8) Build a linear model called `lmfit2` to predict `Observed` using only the `First7D` and `Month` columns. And then use `summary()` function to look at the regression coefficients.

```r
lmfit2 <- summary(lm(Observed ~ First7D + Month, data = WeatherData))
lmfit2
```

```
##
## Call:
## lm(formula = Observed ~ First7D + Month, data = WeatherData)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.1588 -1.5507 -0.0528  1.8048  8.9661
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  36.66199    3.48744  10.513  < 2e-16 ***
## First7D       0.42754    0.05664   7.548 6.58e-11 ***
## MonthAug     10.11655    2.12786   4.754 8.80e-06 ***
## MonthDec    -14.40934    1.87234  -7.696 3.40e-11 ***
## MonthFeb    -16.01121    2.06554  -7.752 2.65e-11 ***
## MonthJan    -18.16175    2.14795  -8.455 1.13e-12 ***
## MonthJuly    11.37154    2.11445   5.378 7.45e-07 ***
## MonthJune    10.73422    1.88904   5.682 2.14e-07 ***
## MonthMar     -3.84020    1.90714  -2.014 0.047458 *
## MonthMay      6.92997    1.73793   3.987 0.000148 ***
## MonthNov     -8.32633    1.67806  -4.962 3.93e-06 ***
## MonthOct     -2.45170    1.70916  -1.434 0.155392
## MonthSep      3.27520    2.11686   1.547 0.125813
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.213 on 79 degrees of freedom
## Multiple R-squared:  0.9732, Adjusted R-squared:  0.9691
## F-statistic: 239.1 on 12 and 79 DF,  p-value: < 2.2e-16
```

9) Notice that there is a coefficient listed for eleven months. When you build a linear model with a categorical variable, R will introduce a *baseline* which serves as a category of comparison for the other categories. The baseline is the one month that is not listed in the summary output. Which month serves as a baseline?

```
April is the baseline.
```

10) Compute the MSE of this new model and save it as `mse2`. Is this better or worse than the MSE for `lmfit1`? Why do you think this is?

```r
mse2 <- mean(lmfit2$residuals^2)
mse2
```

```
## [1] 8.862719
```

```
This model is worse by about 5%. This could be because there are more coeiffiecnts and some of th
```

Of course, the best model is not necessarily the one that fits our training data the best. Such a model may *overfit* the data, and we actually want a model that gives us the best predictions when applied to our test set.

11) Create a data frame in R called `WeatherTest` consisting of the most recent 32 observations of `WeatherData`. This will be your test set. (Make sure there is no overlap with `WeatherTrain`!)

```
WeatherTest <- WeatherData %>% arrange(desc(Year))
WeatherTest <- slice(WeatherTest, 1:32)
```

12) Compute the MSE of the normal base model and the two linear models applied to the *test* set. To get predictions from the linear models, use the syntax `predict(model, newdata)` where `newdata` is the data you want predictions for. Which of the three models performs best? Why do you think this is in terms of how complicated the models are?

```
mse_normal_Test <- mean(summary(lm(Observed~Normal, data = WeatherTest))$residuals^2)
mse1_Test <- mean(lmfit1$residuals^2)
mse2_Test <- mean(lmfit2$residuals^2)

mse_normal_Test
```

```
## [1] 17.31283
```

```
mse1_Test
```

```
## [1] 10.96364
```

```
mse2_Test
```

```
## [1] 8.862719
```

```
predict(lm(Observed ~ First7D + Normal, data = WeatherData), WeatherTest) #lmfit1 model
```

```
##        1        2        3        4        5        6        7        8
## 33.77460 39.87864 48.05366 59.10644 71.02156 80.05784 82.79722 83.96107
##        9       10       11       12       13       14       15       16
## 33.36928 35.42003 45.41902 61.07518 74.75637 78.11806 82.39189 81.93443
##       17       18       19       20       21       22       23       24
## 78.90055 63.38200 56.56290 42.25192 31.66111 34.23300 45.88226 60.69880
##       25       26       27       28       29       30       31       32
## 70.64519 79.56566 82.44980 81.55805 77.42400 62.57134 51.46734 40.92013
```

```
predict(lm(Observed ~ First7D + Normal, data = WeatherData), WeatherTest, interval = "confidence") #lmf
```

```
##         fit      lwr      upr
## 1  33.77460 32.42679 35.12242
## 2  39.87864 38.06696 41.69032
## 3  48.05366 47.01046 49.09685
## 4  59.10644 58.02892 60.18396
## 5  71.02156 69.82336 72.21977
## 6  80.05784 79.03237 81.08332
## 7  82.79722 81.65981 83.93463
## 8  83.96107 82.68910 85.23303
## 9  33.36928 32.04963 34.68892
```

```
## 10 35.42003 34.20956 36.63051
## 11 45.41902 43.68192 47.15613
## 12 61.07518 60.35890 61.79146
## 13 74.75637 73.80259 75.71016
## 14 78.11806 76.95189 79.28423
## 15 82.39189 81.22721 83.55658
## 16 81.93443 80.84809 83.02076
## 17 78.90055 77.34520 80.45591
## 18 63.38200 62.56997 64.19403
## 19 56.56290 54.56229 58.56350
## 20 42.25192 41.04547 43.45837
## 21 31.66111 30.30802 33.01419
## 22 34.23300 32.91571 35.55030
## 23 45.88226 44.28064 47.48387
## 24 60.69880 59.94413 61.45348
## 25 70.64519 69.35571 71.93466
## 26 79.56566 78.54111 80.59021
## 27 82.44980 81.28994 83.60965
## 28 81.55805 80.46236 82.65374
## 29 77.42400 76.23856 78.60944
## 30 62.57134 61.85788 63.28481
## 31 51.46734 50.66275 52.27194
## 32 40.92013 39.85387 41.98638
```

```r
predict(lm(Observed ~ First7D + Month, data = WeatherData), WeatherTest) #lmfit2 model
```

```
##        1        2        3        4        5        6        7        8
## 33.72056 41.64286 51.03488 59.06494 71.03985 81.59919 82.62128 84.44457
##        9       10       11       12       13       14       15       16
## 33.12201 35.05878 47.14429 61.97219 76.55508 78.73469 82.02273 81.45181
##       17       18       19       20       21       22       23       24
## 77.34669 62.29948 57.96399 41.83385 30.59954 33.30588 47.82835 61.41639
##       25       26       27       28       29       30       31       32
## 70.48405 80.87238 82.10824 80.89601 75.16625 61.10238 50.43934 39.86718
```

```r
predict(lm(Observed ~ First7D + Month, data = WeatherData), WeatherTest, interval = "confidence") #lmfi
```

```
##         fit      lwr      upr
## 1  33.72056 31.44134 35.99979
## 2  41.64286 38.89759 44.38812
## 3  51.03488 48.75600 53.31375
## 4  59.06494 56.71104 61.41884
## 5  71.03985 68.68868 73.39102
## 6  81.59919 79.28842 83.90996
## 7  82.62128 80.35337 84.88920
## 8  84.44457 82.10753 86.78161
## 9  33.12201 30.85737 35.38665
## 10 35.05878 32.79087 37.32670
## 11 47.14429 44.76546 49.52312
## 12 61.97219 59.70861 64.23577
## 13 76.55508 74.15388 78.95628
## 14 78.73469 76.45687 81.01251
## 15 82.02273 79.73694 84.30853
```

```
## 16 81.45181 79.18238 83.72124
## 17 77.34669 74.83204 79.86135
## 18 62.29948 59.88188 64.71708
## 19 57.96399 55.00859 60.91939
## 20 41.83385 39.39659 44.27112
## 21 30.59954 28.27646 32.92262
## 22 33.30588 30.95587 35.65590
## 23 47.82835 45.49931 50.15739
## 24 61.41639 59.15528 63.67751
## 25 70.48405 68.08849 72.87961
## 26 80.87238 78.59350 83.15125
## 27 82.10824 79.82566 84.39082
## 28 80.89601 78.60917 83.18285
## 29 75.16625 72.74638 77.58613
## 30 61.10238 58.67157 63.53319
## 31 50.43934 48.00583 52.87284
## 32 39.86718 37.44160 42.29276
```

```r
predict(lm(Observed~Normal, data = WeatherTest), WeatherTest) #normal model
```

```
##        1        2        3        4        5        6        7        8
## 32.34360 35.45760 49.72173 61.37412 73.42831 79.75676 83.27257 81.96670
##        9       10       11       12       13       14       15       16
## 32.34360 35.45760 49.72173 61.37412 73.42831 79.75676 83.27257 81.96670
##       17       18       19       20       21       22       23       24
## 75.03554 61.97683 50.82670 40.17883 32.34360 35.45760 49.72173 61.37412
##       25       26       27       28       29       30       31       32
## 73.42831 79.75676 83.27257 81.96670 75.03554 61.97683 50.82670 40.17883
```

```r
predict(lm(Observed~Normal, data = WeatherTest), WeatherTest, interval = "confidence") #normal model
```

```
##         fit      lwr      upr
## 1  32.34360 29.45861 35.22859
## 2  35.45760 32.79260 38.12260
## 3  49.72173 47.90210 51.54136
## 4  61.37412 59.82209 62.92615
## 5  73.42831 71.54357 75.31306
## 6  79.75676 77.52113 81.99240
## 7  83.27257 80.81232 85.73282
## 8  81.96670 79.59181 84.34159
## 9  32.34360 29.45861 35.22859
## 10 35.45760 32.79260 38.12260
## 11 49.72173 47.90210 51.54136
## 12 61.37412 59.82209 62.92615
## 13 73.42831 71.54357 75.31306
## 14 79.75676 77.52113 81.99240
## 15 83.27257 80.81232 85.73282
## 16 81.96670 79.59181 84.34159
## 17 75.03554 73.06975 77.00132
## 18 61.97683 60.42254 63.53112
## 19 50.82670 49.05447 52.59893
## 20 40.17883 37.82937 42.52829
## 21 32.34360 29.45861 35.22859
```

```
## 22 35.45760 32.79260 38.12260
## 23 49.72173 47.90210 51.54136
## 24 61.37412 59.82209 62.92615
## 25 73.42831 71.54357 75.31306
## 26 79.75676 77.52113 81.99240
## 27 83.27257 80.81232 85.73282
## 28 81.96670 79.59181 84.34159
## 29 75.03554 73.06975 77.00132
## 30 61.97683 60.42254 63.53112
## 31 50.82670 49.05447 52.59893
## 32 40.17883 37.82937 42.52829
```

```
lmfit1 or the first linear model is the best performs the best compared to the other two models. Despi
```