Correlation and regression using the Lahman database for baseball

Michael Lopez, Skidmore College

Overview

The Lahman package is a gold mine for statisticians interested in studying baseball. In today's lab, we are going to explore parts of the Lahman package, while also linking to the statistical concepts in a bivariate analysis, including correlation, regression, and R-squared.

First, let's install the package, and then we load the libraries that we'll need for this lab. As always, once a package is downloaded, you do not need to run the `install.packages()` code again.

```r
#install.packages("Lahman")
library(Lahman)
library(mosaic)
```

There is so much to the Lahman database, and in this course, we'll only touch the tip of the iceberg. First, scroll to the data frame list on the last page of the tutorial. Under **Data sets**, there is a list of roughly 25 data sets that come with the Lahman package.

You can also get a list of the data frames by entering the following command.

```r
LahmanData
```

1. Which data frames in the Lahman package has the largest number of observations? Which have the fewest?

We're going to start by using the `Teams` data.

```r
data(Teams)
head(Teams)
tail(Teams)
```

The `Teams` data contains team-level information for every year of baseball season, from 1871 - 2014. That's a lot of years! To make things a bit easier, we are going to focus on the modern era of baseball, which is generally considered to be the period from 1970 onwards.

2. Recall from Lab 1
   2.1 Write a command to generate the names of the `Teams` data.
   2.2 Write a command that gives you the dimensions of the `Teams` data.
   2.3 Identify the observation in row 500 and column 10.

To reduce a larger data frame into a subset of rows that contain only information meeting a certain criterion, we are going to take advantage of the `filter()` command. *Note*: The `filter()` command comes from the `dplyr` package, which is one of `R`'s best packages for data manipulation. It loads automatically with `mosaic`, and you can read more by going to the tutorial here.

```r
Teams.1 <- filter(Teams, yearID >= 1970)
head(Teams.1)
```

We store the newer data set as `Teams.1`, and you can tell `Teams.1` is a smaller data set because it begins in 1970, and not 1871.

Next, let's create a few missing team-level variables that we are going to need for the lab. In `dplyr`, we create a series of new variables using the `mutate()` command.

```r
Teams.1 <- mutate(Teams.1, X1B = H - X2B - X3B - HR,
                  TB = X1B + 2*X2B + 3*X3B + 4*HR,
                  RC = (H + BB)*TB/(AB + BB),
                  RC.new = (H + BB - CS)*(TB + (0.55*SB))/(AB+BB) )
```

This creates four new team-level variables, `X1B` (number of singles), `TB` (total bases), `RC` (runs created), and `RC.new` (a newer runs created formula). The first formula for runs created is the basic one, discussed previously in lecture, while the second one uses a tweak for stolen bases.

Notice that in the above code, the data name (`Teams.1`) remained unchanged. Alternatively, we could have created a new name (say, `Teams.2`) if we had wanted.

Bivariate analysis

Correlation and scatter plots

Using `Teams.1`, let's quantify the strength, shape, and direction of the association between runs created and runs.

```r
xyplot(R ~ RC, data = Teams.1, main = "Runs ~ Runs Created, 1970 - 2015")
cor(R ~ RC, data = Teams.1)
cor(R ~ RC, data = Teams.1)^2
```

```r
xyplot(R ~ RC.new, data = Teams.1, main = "Runs ~ Runs Created, 1970 - 2015")
cor(R ~ RC.new, data = Teams.1)
cor(R ~ RC.new, data = Teams.1)^2
```

3. Which variable - `RC` or `RC.new` - shows a stronger link with runs? What does this entail about the updated stolen bases formula?

Let's identify some additional ways of quantifying the association between two variables.

As an example, let's look at the relationship between a team's home runs and its runs.

```r
cor(R ~ HR, data = Teams.1)
cor(R ~ HR, data = Teams.1)^2
```

4. Interpret the correlation coefficient and the R-squared values between home runs and runs.

Simple linear regression.

From introductory statistics, you'll remember the simple linear regression (SLR) equation. Recall, here's the estimated SLR fit:

$$\hat{y_i} = \hat{\beta}_0 + \hat{\beta}_1 * x_i$$

In the model above, $\hat{y_i}$ represents our predicted value of an outcome variable given x_i, while $\hat{\beta}_0$ and $\hat{\beta}_1$ are the estimated intercepts and slopes, respectively.

In our example, we can plug in our variable names as follows:

$$\hat{R_i} = \hat{\beta}_0 + \hat{\beta}_1 * HR_i$$

5. Make a scatter plot of `R` as a function of `HR`. Using the function, make educated guesses for $\hat{\beta}_0$ and $\hat{\beta}_1$.

Fortunately, we don't have to make educated guesses. Let's look at the fit a model of runs as a function of home runs using the R command `lm()`.

```
fit.1 <- lm(R ~ HR, data = Teams.1)
summary(fit.1)
```

There's a ton of stuff in the code above, most of which is useful.

First, the `summary()` code gives the regression output, as well as other model characteristics.

6. Write the estimated regression line. Does the actual estimated regression line resemble your guesses from question 5)?
7. Interpret both the intercept and the slope (for `HR`) in the estimated regression equation. Is the intercept a useful term here?
8. Estimate the number of runs that a team with 250 home runs would score.

There is also code to draw an estimated regression line, assuming that you've already stored a linear model.

```
plotModel(fit.1)
```

In the code above, `R` knows that `fit.1` contains a linear model of runs as a function of home runs. Hopefully this reflects the line of best fit that you would've drawn by hand.

Let's return to the output of the regression equation.

```
summary(fit.1)
```

9. Is the link between home runs and runs significant? How can you tell?

Related note on significance testing Let's turn out attention to another variable with a lesser link to runs, `SB`. The following code is used to produce identical output.

```
fit.2 <- lm(R ~ SB, data = Teams.1)
summary(fit.2)
cor.test(Teams.1$R, Teams.1$SB)
```

Can you figure out the link between the two outputs from above?

In the first, the relative significance of the slope parameter (as judged by a $p - value = 0.008619$) is identical to the relative significance of the test for the significance of the correlation coefficient (also a $p - value$ of 0.008619).

10. Use a similar code to determine if there is there a significant linear association between team runs and the number of times that team was caught stealing?

Analyzing several variables simultaneously

We close the lab by looking at how one would analyze a subset of variables to judge which are most strongly associated with team success.

First, we use the `select()` command to reduce our data set from several dozen columns to only the ones we are interested in studying.

```
Teams.3 <- select(Teams.1, R, RC.new, RC, RA, HR, SO, attendance)
```

Next, we calculate the pairwise correlation coefficient between each variable.

```
cor.matrix <- cor(Teams.3)
cor.matrix
round(cor.matrix, 3)
```

11. Of the variables listed, which boast the strongest and weakest correlations with runs scored? Sidenote: What does the `round()` command do?

This is a good time to point out that, as is usually the case with observational data like this, strong links between two variables do not entail that one variable causes the other. While hitting home-runs will likely cause teams to score more runs, it is not neccessarily the case that higher attendence causes teams to score more runs.

12. Think of one reason why attendence and runs are significantly correlated besides saying that attendence causes teams to score more runs.

Plotting correlations

There are lots of fun plots to make in R. Here are a few.

```
install.packages('corrplot')
library(corrplot)
corrplot(cor.matrix, method="number")
corrplot(cor.matrix, method="circle", type = "lower")
```

Better team metrics: on base, slugging, OPS

Perhaps other functions of traditional baseball statistics are worth looking at. Here, we calculate on base percentage (OBP), slugging percentage (SLG), and OPS, which is the sum of OBP and SLG.

```
Teams.1 <- mutate(Teams.1, OBP = (H + BB)/(AB + BB),
                  SLG = (X1B + 2*X2B + 3*X3B + 4*HR)/AB,
                  OPS = OBP + SLG)
cor.matrix <- cor(select(Teams.1, R, RC, OBP, SLG, OPS))
round(cor.matrix, 2)
corrplot(cor.matrix, method="number")
```

Repeatability

In addition to wanting a metric to correlate with success and to reflect individual talent, it is worth looking at how well a metric can predict a future, unknown performance.

This requires some careful coding. Using the `dplyr` package, we create a new data frame, `Teams.2`, which arranges the team-level data by franchise and year, and then calculates the number of runs scored in the following year as the variable `next.R`.

```
Teams.2 <- Teams.1 %>%
  arrange(franchID, yearID) %>%
  group_by(franchID) %>%
  mutate(next.R = lead(RC))
head(Teams.2)
```

13. Verify that the last column of `Teams.2` contains the future runs scored for each team in each row.

To close, we look at which team-level variables most strongly correlate with future runs scored.

```
cor.matrix <- cor(select(ungroup(Teams.2), next.R, R, RC, OBP, SLG,
        OPS, HR, X1B), use="pairwise.complete.obs")
round(cor.matrix, 2)
corrplot(cor.matrix, method="number")
```

14. Which variables appear to be the best at predicting a team's runs scored in the following year? Which appears to be the worst?
15. Related: Compare the correlation of number of singles (`X1B`) to `next.R` and `R`, as well as number of home runs (`HR`) to `next.R` and `R`. Think carefully about how this would impact your recommendations of how to build a team. Which measures should be looked at most closely? Which measures appear to be mostly noise?
16. For fun: Go to the Shiny app here. Take a few guesses, and then click `View scatterplot of correlation between your guesses and actual correlation`. How good are you at guessing the correlation coefficient?