Correlation, Causation and Linear Regression

Thileepan Paulraj

21 December 2018

Correlation and Causation

Two variables are said to be correlated when they vary together. An unit increase in one variable should result in an unit increase or decrease in another variable for the variables to be correlated. Variables could also be linearly and non-linearly (parabolic correlation, exponential correlation) correlated.

Let's take the cars data set from R and see how it's mileage and weight variables are correlated. Let's use pearson's correlation for that.

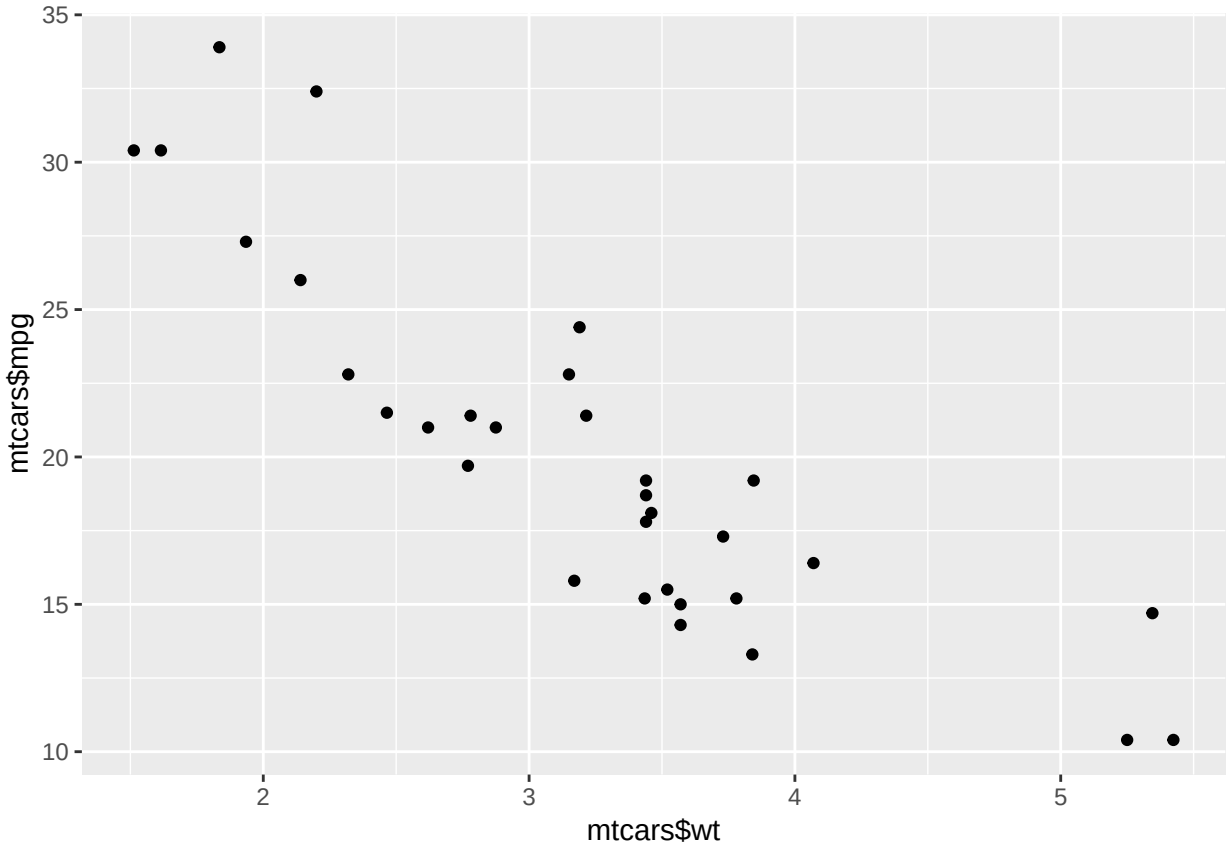
```
cor.test(mtcars$mpg, mtcars$wt)
```

```
##
##  Pearson's product-moment correlation
##
## data:  mtcars$mpg and mtcars$wt
## t = -9.559, df = 30, p-value = 1.294e-10
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.9338264 -0.7440872
## sample estimates:
##        cor
## -0.8676594
```

Correlation coefficients are usually between -1 and +1. Here our correlation coefficient is high, which indicates high correlation between the variables. the negative sign indicates a negative correlation which means as the weight of the car increases it's mileage decreases.

Let's visualize the correlation also.

```
ggplot(data =mtcars, aes(x = mtcars$wt, y = mtcars$mpg)) + geom_point()
```



From the plot we can clearly see that as the weight of the car increases it's mileage decreases.

Causation

When two variables vary together it doesn't mean that one variable is the cause of the variation in the other variable. In some cases, however one the change in one variable might be the cause for the change in the other variable. An example of this could be the rise in temperature during summer and the number of hours of usage of air conditioners.

Regression

Correlation measures the strength of association between two variables but regression fits a linear relationship between a set of predictor variables and the output. Using this linear relationship we can predict future values of the output variable if we have our predictor variables.

Linear Regression Example

Let's take a real life data set and apply linear regression to it to predict future results. I will be using the loans data set to predict the **interest rate** given to each customer.

```
train_data  = read.csv('loan_data_train.csv', stringsAsFactors = F)
test_data = read.csv('loan_data_test.csv', stringsAsFactors = F)
```

```
glimpse(train_data)
```

```
## Observations: 2,200
## Variables: 15
## $ ID                              <int> 79542, 75473, 67265, 80167, 172...
## $ Amount.Requested                <chr> "25000", "19750", "2100", "2800...
## $ Amount.Funded.By.Investors      <chr> "25000", "19750", "2100", "2800...
## $ Interest.Rate                   <chr> "18.49%", "17.27%", "14.33%", "...
## $ Loan.Length                     <chr> "60 months", "60 months", "36 m...
## $ Loan.Purpose                    <chr> "debt_consolidation", "debt_con...
## $ Debt.To.Income.Ratio            <chr> "27.56%", "13.39%", "3.50%", "1...
## $ State                           <chr> "VA", "NY", "LA", "NV", "OH", "...
## $ Home.Ownership                  <chr> "MORTGAGE", "MORTGAGE", "OWN", ...
## $ Monthly.Income                  <dbl> 8606.56, 6737.50, 1000.00, 7083...
## $ FICO.Range                      <chr> "720-724", "710-714", "690-694"...
## $ Open.CREDIT.Lines               <chr> "11", "14", "13", "12", "6", "2...
## $ Revolving.CREDIT.Balance        <chr> "15210", "19070", "893", "38194...
## $ Inquiries.in.the.Last.6.Months <int> 3, 3, 1, 1, 2, 2, 0, 1, 0, 1, 0...
## $ Employment.Length               <chr> "5 years", "4 years", "< 1 year...
```

```
dim(train_data)
```

```
## [1] 2200   15
```

```
glimpse(test_data)
```

```
## Observations: 300
## Variables: 14
## $ ID                              <int> 20093, 62445, 65248, 81822, 579...
## $ Amount.Requested                <int> 5000, 18000, 7200, 7200, 22000,...
## $ Amount.Funded.By.Investors      <chr> "5000", "18000", "7200", "7200"...
## $ Loan.Length                     <chr> "60 months", "60 months", "60 m...
## $ Loan.Purpose                    <chr> "moving", "debt_consolidation",...
## $ Debt.To.Income.Ratio            <chr> "12.59%", "4.93%", "25.16%", "1...
## $ State                           <chr> "NY", "CA", "LA", "NY", "MI", "...
## $ Home.Ownership                  <chr> "RENT", "RENT", "MORTGAGE", "MO...
## $ Monthly.Income                  <dbl> 4416.67, 5258.50, 3750.00, 3416...
## $ FICO.Range                      <chr> "690-694", "710-714", "750-754"...
## $ Open.CREDIT.Lines               <chr> "13", "6", "13", "14", "9", "."...
## $ Revolving.CREDIT.Balance        <int> 7686, 11596, 7283, 4838, 20181,...
## $ Inquiries.in.the.Last.6.Months <int> 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0...
## $ Employment.Length               <chr> "< 1 year", "10+ years", "6 yea...
```

```
dim(test_data)
```

```
## [1] 300  14
```

```
dim(train_data)
```

```
## [1] 2200   15
```

```
dim(test_data)
```

```
## [1] 300  14
```

As we can see from the dimensions of the train and test data we can see the test data has one variable less, this is the variable **Interest.Rate** which we have to predict.

Data Pre Processing

Before we start any kind of machine learning modeling, the data needs to be pre-processed. Since we have two different data sets (train and test) we can combine them together and do the pre processing.

In order to combine them together both the data sets needs to be of the same dimension. So let's add a variable called **Interest.rate** with NA values in the test data set also.

```
test_data$Interest.Rate = NA
```

Now, both the data sets are of the same dimension and we could combine them easily, but we need some indication to differentiate our training set from test set. Let's add a variable called **data** to both the data sets which will be equal to **train** for training data set and **test** for test data set.

```
train_data$data = 'train'
test_data$data = 'test'
```

Now, let's combine the data sets

```
all_data = rbind(train_data, test_data)
glimpse(all_data)
```

```
## Observations: 2,500
## Variables: 16
## $ ID                              <int> 79542, 75473, 67265, 80167, 172...
## $ Amount.Requested                <chr> "25000", "19750", "2100", "2800...
## $ Amount.Funded.By.Investors      <chr> "25000", "19750", "2100", "2800...
## $ Interest.Rate                   <chr> "18.49%", "17.27%", "14.33%", "...
## $ Loan.Length                     <chr> "60 months", "60 months", "36 m...
## $ Loan.Purpose                    <chr> "debt_consolidation", "debt_con...
## $ Debt.To.Income.Ratio            <chr> "27.56%", "13.39%", "3.50%", "1...
## $ State                           <chr> "VA", "NY", "LA", "NV", "OH", "...
## $ Home.Ownership                  <chr> "MORTGAGE", "MORTGAGE", "OWN", ...
## $ Monthly.Income                  <dbl> 8606.56, 6737.50, 1000.00, 7083...
## $ FICO.Range                      <chr> "720-724", "710-714", "690-694"...
## $ Open.CREDIT.Lines               <chr> "11", "14", "13", "12", "6", "2...
## $ Revolving.CREDIT.Balance        <chr> "15210", "19070", "893", "38194...
## $ Inquiries.in.the.Last.6.Months <int> 3, 3, 1, 1, 2, 2, 0, 1, 0, 1, 0...
## $ Employment.Length               <chr> "5 years", "4 years", "< 1 year...
## $ data                            <chr> "train", "train", "train", "tra...
```

Converting characters to numeric variables

From the table above, we can see that some numeric variables are represented as character variables in the data set. Let's try to convert them back to numeric first

```
all_data = all_data %>%
  mutate(Interest.Rate = as.numeric(gsub("%", "", Interest.Rate)),
  Amount.Requested = as.numeric(Amount.Requested),
  Amount.Funded.By.Investors = as.numeric(Amount.Funded.By.Investors),
  Loan.Length = as.numeric(gsub("months", "", Loan.Length)),
  Debt.To.Income.Ratio = as.numeric(gsub("%", "", Debt.To.Income.Ratio)),
  Revolving.CREDIT.Balance = as.numeric(Revolving.CREDIT.Balance),
  Open.CREDIT.Lines = as.numeric(Open.CREDIT.Lines))
```

```
## Warning in evalq(as.numeric(Amount.Requested), <environment>): NAs
```

```
## introduced by coercion

## Warning in evalq(as.numeric(Amount.Funded.By.Investors), <environment>):
## NAs introduced by coercion

## Warning in evalq(as.numeric(gsub("months", "", Loan.Length)),
## <environment>): NAs introduced by coercion

## Warning in evalq(as.numeric(Revolving.CREDIT.Balance), <environment>): NAs
## introduced by coercion

## Warning in evalq(as.numeric(Open.CREDIT.Lines), <environment>): NAs
## introduced by coercion
```

```
glimpse(all_data)
```

```
## Observations: 2,500
## Variables: 16
## $ ID                              <int> 79542, 75473, 67265, 80167, 172...
## $ Amount.Requested                <dbl> 25000, 19750, 2100, 28000, 2425...
## $ Amount.Funded.By.Investors      <dbl> 25000.00, 19750.00, 2100.00, 28...
## $ Interest.Rate                   <dbl> 18.49, 17.27, 14.33, 16.29, 12....
## $ Loan.Length                     <dbl> 60, 60, 36, 36, 60, 36, 60, 36,...
## $ Loan.Purpose                    <chr> "debt_consolidation", "debt_con...
## $ Debt.To.Income.Ratio            <dbl> 27.56, 13.39, 3.50, 19.62, 23.7...
## $ State                           <chr> "VA", "NY", "LA", "NV", "OH", "...
## $ Home.Ownership                  <chr> "MORTGAGE", "MORTGAGE", "OWN", ...
## $ Monthly.Income                  <dbl> 8606.56, 6737.50, 1000.00, 7083...
## $ FICO.Range                      <chr> "720-724", "710-714", "690-694"...
## $ Open.CREDIT.Lines               <dbl> 11, 14, 13, 12, 6, 2, 5, 11, 24...
## $ Revolving.CREDIT.Balance        <dbl> 15210, 19070, 893, 38194, 31061...
## $ Inquiries.in.the.Last.6.Months <int> 3, 3, 1, 1, 2, 2, 0, 1, 0, 1, 0...
## $ Employment.Length               <chr> "5 years", "4 years", "< 1 year...
## $ data                            <chr> "train", "train", "train", "tra...
```

Converting a range variable (FICO.Range) to a numeric value

```
all_data = all_data %>%
  mutate( f1 = as.numeric(substr(FICO.Range, 1,3)),
          f2 = as.numeric(substr(FICO.Range, 5,7)),
          fico = (f1+f2)/2
        ) %>%
  select(-FICO.Range, -f1, -f2)

glimpse(all_data)
```

```
## Observations: 2,500
## Variables: 16
## $ ID                              <int> 79542, 75473, 67265, 80167, 172...
## $ Amount.Requested                <dbl> 25000, 19750, 2100, 28000, 2425...
## $ Amount.Funded.By.Investors      <dbl> 25000.00, 19750.00, 2100.00, 28...
## $ Interest.Rate                   <dbl> 18.49, 17.27, 14.33, 16.29, 12....
## $ Loan.Length                     <dbl> 60, 60, 36, 36, 60, 36, 60, 36,...
## $ Loan.Purpose                    <chr> "debt_consolidation", "debt_con...
## $ Debt.To.Income.Ratio            <dbl> 27.56, 13.39, 3.50, 19.62, 23.7...
## $ State                           <chr> "VA", "NY", "LA", "NV", "OH", "...
```

```
## $ Home.Ownership                 <chr> "MORTGAGE", "MORTGAGE", "OWN", ...
## $ Monthly.Income                 <dbl> 8606.56, 6737.50, 1000.00, 7083...
## $ Open.CREDIT.Lines              <dbl> 11, 14, 13, 12, 6, 2, 5, 11, 24...
## $ Revolving.CREDIT.Balance       <dbl> 15210, 19070, 893, 38194, 31061...
## $ Inquiries.in.the.Last.6.Months <int> 3, 3, 1, 1, 2, 2, 0, 1, 0, 1, 0...
## $ Employment.Length              <chr> "5 years", "4 years", "< 1 year...
## $ data                           <chr> "train", "train", "train", "tra...
## $ fico                           <dbl> 722, 712, 692, 712, 732, 787, 6...
```

Converting Employment length variable into a numeric variable.

```
all_data = all_data %>%
  mutate( el = ifelse(substr(Employment.Length, 1,2) == 10, 10, Employment.Length),
          el = ifelse(substr(Employment.Length, 1,1) == "<", 0, el),
          el = gsub("years", "", el),
          el = gsub("years", "", el),
          el = as.numeric(el)
          ) %>%
  select(-Employment.Length)
```

```
## Warning in evalq(as.numeric(el), <environment>): NAs introduced by coercion
```

```
glimpse(all_data)
```

```
## Observations: 2,500
## Variables: 16
## $ ID                             <int> 79542, 75473, 67265, 80167, 172...
## $ Amount.Requested               <dbl> 25000, 19750, 2100, 28000, 2425...
## $ Amount.Funded.By.Investors     <dbl> 25000.00, 19750.00, 2100.00, 28...
## $ Interest.Rate                  <dbl> 18.49, 17.27, 14.33, 16.29, 12....
## $ Loan.Length                    <dbl> 60, 60, 36, 36, 60, 36, 60, 36,...
## $ Loan.Purpose                   <chr> "debt_consolidation", "debt_con...
## $ Debt.To.Income.Ratio           <dbl> 27.56, 13.39, 3.50, 19.62, 23.7...
## $ State                          <chr> "VA", "NY", "LA", "NV", "OH", "...
## $ Home.Ownership                 <chr> "MORTGAGE", "MORTGAGE", "OWN", ...
## $ Monthly.Income                 <dbl> 8606.56, 6737.50, 1000.00, 7083...
## $ Open.CREDIT.Lines              <dbl> 11, 14, 13, 12, 6, 2, 5, 11, 24...
## $ Revolving.CREDIT.Balance       <dbl> 15210, 19070, 893, 38194, 31061...
## $ Inquiries.in.the.Last.6.Months <int> 3, 3, 1, 1, 2, 2, 0, 1, 0, 1, 0...
## $ data                           <chr> "train", "train", "train", "tra...
## $ fico                           <dbl> 722, 712, 692, 712, 732, 787, 6...
## $ el                             <dbl> 5, 4, 0, 10, 10, NA, 2, 0, NA, ...
```