Regression Analysis with R

Stephen Bernardo

2023-11-18

Load Libraries

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(magrittr)
```

```
##
## Attaching package: 'magrittr'
##
## The following object is masked from 'package:purrr':
##
##     set_names
##
## The following object is masked from 'package:tidyr':
##
##     extract
```

```
library(lubridate)
```

Reading Data

```
df <- read_csv("day.csv")
```

```
## Rows: 731 Columns: 16
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl  (15): instant, season, yr, mnth, holiday, weekday, workingday, weathers...
## date  (1): dteday
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Preparing Data

Extract month from dteday column

```
df$month_name <- month(df$dteday, label = TRUE)
```

Turning month_name to character data type

```
df$month_name <- as.character(df$month_name)
```

Running regression models

Model 1

Linear Regression model

```
model1 <- lm(cnt ~ month_name, data = df)

summary(model1)
```

```
##
## Call:
## lm(formula = cnt ~ month_name, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5177.2 -1095.2  -249.3  1290.0  4669.7
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     4484.9      196.7  22.799  < 2e-16 ***
## month_nameAug   1179.5      275.9   4.275 2.17e-05 ***
## month_nameDec  -1081.1      275.9  -3.918 9.79e-05 ***
## month_nameFeb  -1829.6      281.8  -6.492 1.58e-10 ***
## month_nameJan  -2308.6      275.9  -8.366 3.09e-16 ***
## month_nameJul   1078.8      275.9   3.909 0.000101 ***
## month_nameJun   1287.5      278.2   4.628 4.38e-06 ***
## month_nameMar   -792.6      275.9  -2.873 0.004192 **
## month_nameMay    864.9      275.9   3.134 0.001793 **
## month_nameNov   -237.7      278.2  -0.854 0.393113
## month_nameOct    714.3      275.9   2.589 0.009829 **
## month_nameSep   1281.6      278.2   4.607 4.83e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1524 on 719 degrees of freedom
## Multiple R-squared:  0.3906, Adjusted R-squared:  0.3813
## F-statistic:  41.9 on 11 and 719 DF,  p-value: < 2.2e-16
```

The Adjusted R-squared for this is 0.3813. This means taht month_name explains the cnt by the said amount. The reference month used is August because the data is set to charcter type.

d) With either a code chunk or regular text, use the coefficient estimates from Model1 to report the predicted cnt for the months of January and June. 10 points (5 points for each correct prediction)

Data frame for prediction

```
new_data <- data.frame(month_name = c("Jan", "Jun"))
```

Predicting counts for January and June

```
predicted_counts <- predict(model1, newdata = new_data)
```

Results

```
result <- data.frame(month_name = new_data$month_name, predicted_counts)
print(result)
```

```
##   month_name predicted_counts
## 1        Jan         2176.339
## 2        Jun         5772.367
```

Model 2

Multiple Linear Regression Model

```
model2 <- lm(cnt ~ temp + month_name, data = df)
```

Summary of Model 2

```
summary(model2)
```

```
##
## Call:
## lm(formula = cnt ~ temp + month_name, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4896.6 -1080.0  -228.4  1245.2  3372.9
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1554.39     390.76   3.978 7.66e-05 ***
## temp           6235.14     729.40   8.548  < 2e-16 ***
## month_nameAug  -308.08     315.42  -0.977   0.3290
## month_nameDec  -170.96     283.80  -0.602   0.5471
## month_nameFeb  -764.81     296.15  -2.582   0.0100 *
## month_nameJan  -852.31     313.41  -2.719   0.0067 **
## month_nameJul  -701.18     335.50  -2.090   0.0370 *
## month_nameJun   -47.47     307.78  -0.154   0.8775
## month_nameMar  -297.20     269.38  -1.103   0.2703
## month_nameMay    86.73     278.37   0.312   0.7555
## month_nameNov   390.66     275.22   1.419   0.1562
## month_nameOct   620.72     263.30   2.357   0.0187 *
## month_nameSep   368.25     285.93   1.288   0.1982
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1453 on 718 degrees of freedom
## Multiple R-squared:  0.4469, Adjusted R-squared:  0.4377
## F-statistic: 48.35 on 12 and 718 DF,  p-value: < 2.2e-16
```

The coefficient for month_nameJan is in Model1 is -2308.6, whereas it increased to -852.31 in model 2. One possible reason is model fit, which leads to adjusting of existing variables when an additional coefficient is added.

Predicted count for January when the temperature is .25

Create a data frame for prediction

```
new_data2 <- data.frame(temp = 0.25, month_name = "Jan")
```

Predict counts for January and June

```
predicted_counts2 <- predict(model2, newdata = new_data2)
```

Display the results

```
result2 <- data.frame(month_name = new_data2$month_name, temp = new_data2$temp, predicted_counts2)
print(result2)
```

```
##   month_name temp predicted_counts2
## 1        Jan 0.25          2260.863
```