HW1 Number1

Aaron Coates

4/17/2019

```r
setwd("~/Documents/GitHub/MMSS_311_2")
sick <- read.csv("/Users/aaroncoates/Downloads/sick_data.csv")

library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(ggplot2)
library(glmnet)
```

```
## Loading required package: Matrix

## Loading required package: foreach

## Loaded glmnet 2.0-16
```

```r
library(broom)
library(aod)
```

First, I turn positive results to "1" and negative results to "0".

```r
sick$result.d <- sapply(sick$result, function(x){
  if(x == "Positive"){
    1
  } else {
    0
  }})
```

This is the OLS regression.

```r
regOLS <- glm(result.d ~ temp + bp, data=sick) %>% broom::tidy()
regOLS
```

```
## # A tibble: 3 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept) -5.21     0.514        -10.1 4.61e-23
## 2 temp         0.0628   0.00506       12.4 4.92e-33
## 3 bp          -0.00829  0.000470     -17.6 1.03e-60
```

```r
for (i in 1:1000) {sick$predictionOLS[i] <-
  regOLS[1, 2] + regOLS[2, 2]*sick$temp[i] + regOLS[3,2]*sick$bp[i]}
```

Above, I generated predicted values using the OLS regression. I summarize the results below.

```
(sum(with(sick, result.d==1 & predictionOLS>=.5)) +
   sum(with(sick, result.d==0 & predictionOLS<=.5)))/1000
```

```
## [1] 0.964
```

So, the model is accurate 96.4% of the time. Now, I will rearrange variables to find the line of best fit.

```
intOLS <- (regOLS[1, 2] -.5)/(-regOLS[3, 2])
slopeyOLS <- regOLS[2, 2]/(-regOLS[3,2])
intOLS
```

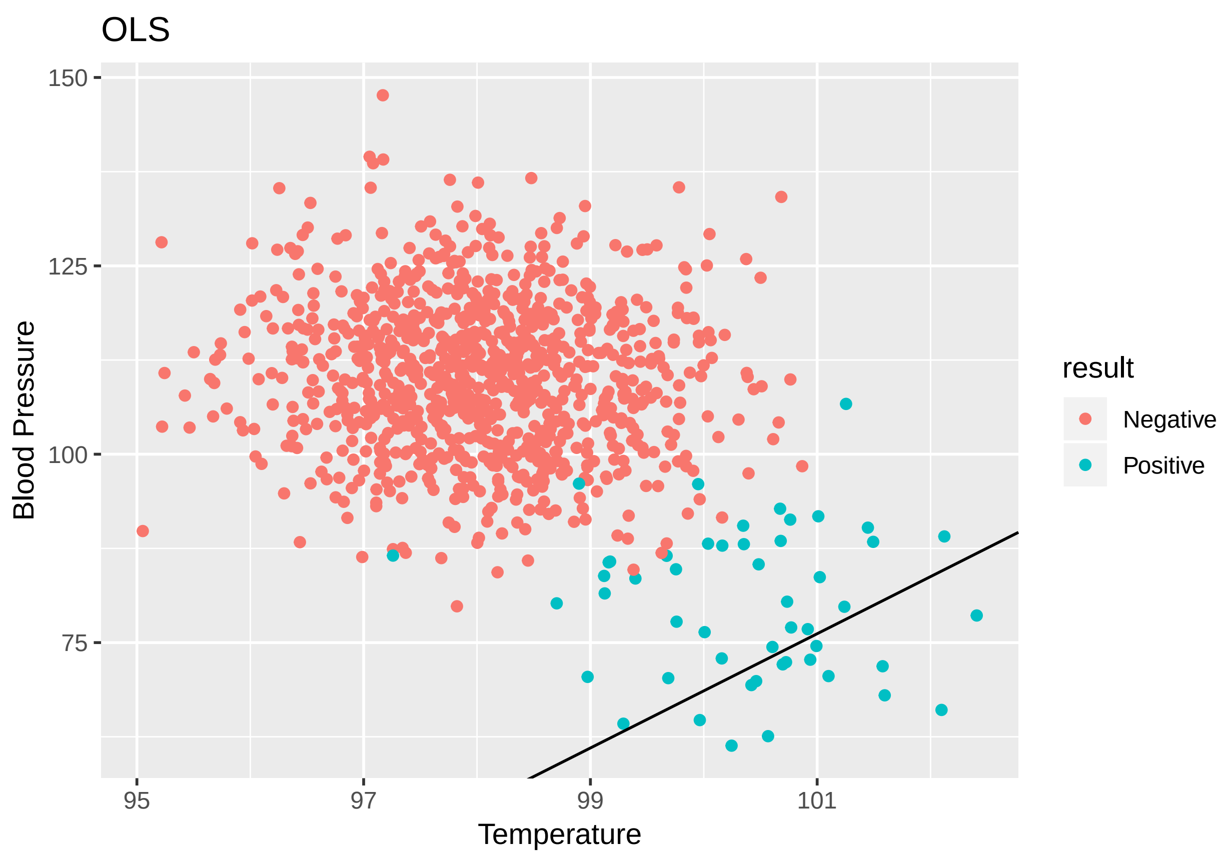
```
##    estimate
## 1 -689.4861
```

```
slopeyOLS
```

```
##   estimate
## 1 7.580789
```

I plot the results below.

```
plotOLS <- ggplot(data = sick, aes(x = temp, y = bp, col=result)) +
  geom_point() + geom_abline(intercept = intOLS[1, 1],
  slope = slopeyOLS[1, 1]) +
  labs(title ="OLS", x = "Temperature", y = "Blood Pressure")
```



Now, I will perform the logistic regression.

```
sick$result <- factor(sick$result)
logit <- glm(result ~ temp + bp, data = sick,
```

```r
             family = "binomial") %>% broom::tidy()
logit
```

```
## # A tibble: 3 x 5
##   term        estimate std.error statistic      p.value
##   <chr>          <dbl>     <dbl>     <dbl>        <dbl>
## 1 (Intercept) -199.      46.8        -4.26 0.0000206
## 2 temp           2.31     0.492       4.70 0.00000260
## 3 bp            -0.350    0.0638     -5.48 0.0000000414
```

Now, I will generate predicted values and then summarize the results.

```r
for (i in 1:1000) {sick$predictionLOGIT[i] <-
  logit[1, 2] + logit[2, 2]*sick$temp[i] + logit[3,2]*sick$bp[i]}
```

```r
(sum(with(sick, result.d==1 & predictionLOGIT>=.5)) +
   sum(with(sick, result.d==0 & predictionLOGIT<=.5))) / 1000
```

```
## [1] 0.992
```

So, the model is 99.2% accurate. Now, I will find the line of best fit by rearranging coefficients.

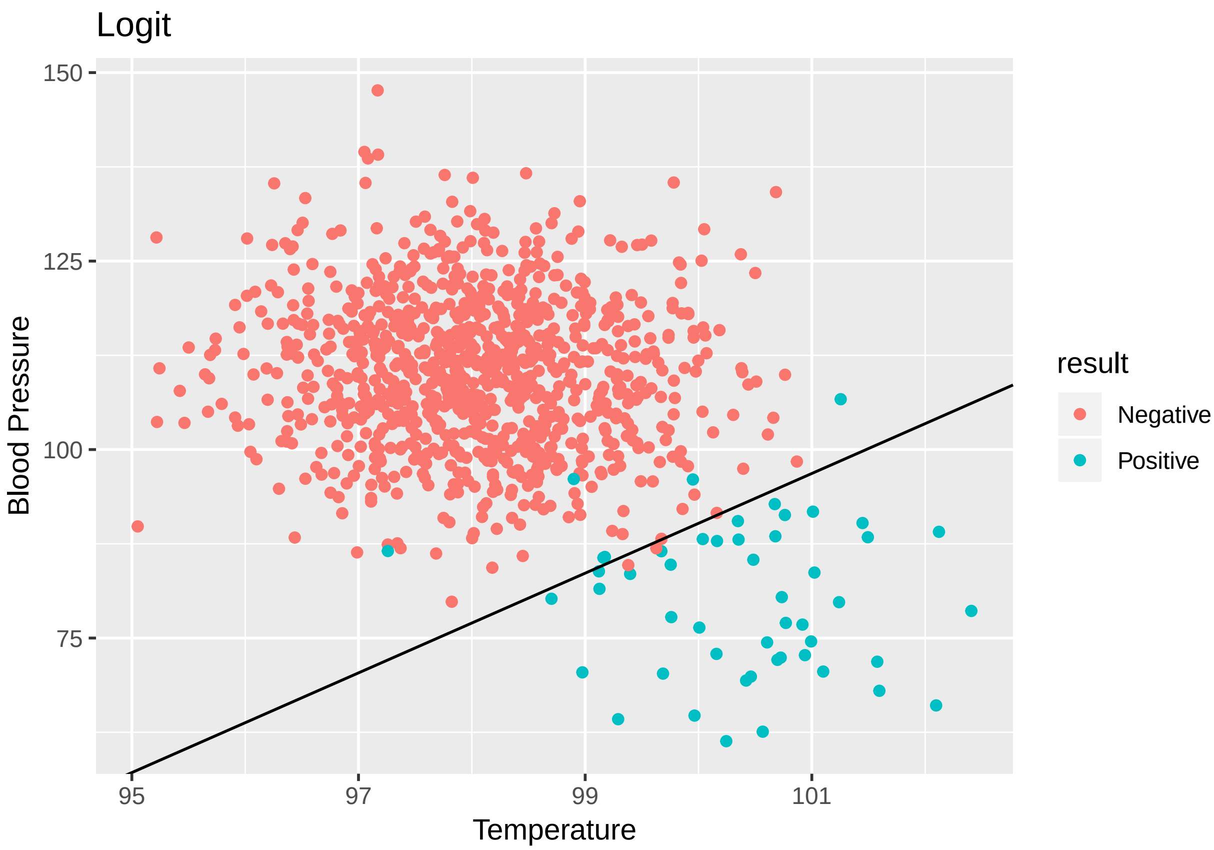
```r
intLOG <- (logit[1, 2] -.5)/(-logit[3, 2])
slopeyLOG <- logit[2, 2]/(-logit[3,2])
intLOG
```

```
##    estimate
## 1 -571.0098
```

```r
slopeyLOG
```

```
##   estimate
## 1 6.612235
```

```r
plotLOG <- ggplot(data = sick, aes(x = temp, y = bp, col=result)) +
  geom_point() + geom_abline(intercept = intLOG[1, 1],
  slope = slopeyLOG[1, 1]) + labs(title ="Logit",
  x = "Temperature", y = "Blood Pressure")
```


Logit
150
125
100
75
Blood Pressure
95
97
99
101
Temperature
result
Negative
Positive